

Semantic Foundations of Dataspaces

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Johannes Theissen-Lipp (geb. Lipp), M. Sc.

aus

Groß-Gerau, Deutschland

Berichter: Prof. Dr. Stefan Decker
Prof. Dr. Christoph Quix
Prof. Edward Curry

Tag der mündlichen Prüfung: 25. September 2024

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Abstract

Digital transformation is rapidly reshaping industries, organizations, and societies around the world, driven by the exponential growth of data from the Internet and the Web. This data, encompassing both open and closed resources, drives innovation and provides competitive advantages. However, there is a growing demand for data sovereignty, where individuals and organizations control their data throughout its lifecycle. Initiatives such as dataspace, funded with €4-6 billion in Europe, and the Internet of Production focus on data autonomy and integration, but interoperability and common understanding remain critical challenges. The fields of Semantic Web technologies and Findable, Accessible, Interoperable, and Re-usable (FAIR) data provide frameworks for addressing these challenges. These semantic technologies enable seamless data exchange and meaningful interpretation to maximize the potential of data, yet they are not sufficiently utilized in dataspace.

This thesis explores the semantic foundations of dataspace, aiming to integrate semantic technologies to improve data interoperability and facilitate common understanding among stakeholders. It also advances semantic technologies to enhance their impact. This work proposes new concepts, principles, best practices, and solutions to improve data management in dataspace. The key contributions of this thesis include:

- **A Tailored Data Lifecycle Model for Dataspace:** This model addresses the unique challenges and requirements of dataspace, ensuring a streamlined and structured approach to managing data and metadata throughout their lifecycle.
- **Principles of Information Models:** These models define a common core in dataspace, providing a structured representation of data, services, participants, and interactions, and serving as a foundation for data sovereignty and interoperability mechanisms.
- **Extended Data Access Principles:** These principles cover various types of heterogeneity, improving data accessibility by managing resource constraints, handling dynamic data changes, and providing rich metadata to increase the quantity and quality of information in dataspace.
- **Methods for Improving Common Domain Understanding:** This includes techniques for identification, best practices for vocabulary development, involvement of non-experts, effective recommendation systems, evolution of semantic information over time, and improved validation techniques to enhance data interoperability and (re)use in dataspace.

By combining these contributions, this thesis confidently integrates semantic technologies into dataspace. This integration enhances data and service interoperability and facilitates common understanding within these environments, promoting proper data management and reuse throughout the data lifecycle. Our work provides valuable insights into the effective management of information within the digital ecosystem of dataspace, contributing to the advancement of knowledge in today's data-driven digital transformation.

Kurzfassung

Die digitale Transformation verändert weltweit Branchen, Organisationen und Gesellschaften durch das exponentielle Wachstum von Daten. Diese Daten, ob offen oder vertraulich, treiben Innovationen und schaffen Wettbewerbsvorteile. Es wächst jedoch das Bedürfnis nach Datensouveränität, bei der Personen und Organisationen die Kontrolle über ihre Daten während des gesamten Lebenszyklus behalten. Initiativen wie Datenräume, die in Europa mit 4 bis 6 Milliarden Euro gefördert werden, und das Internet of Production konzentrieren sich auf Datenautonomie und -integration, doch Interoperabilität und gemeinsames Verständnis bleiben Herausforderungen. Semantic Web-Technologien und Findable, Accessible, Interoperable, and Re-usable (FAIR)-Daten bieten Grundlagen für die Bewältigung dieser Herausforderungen, sind jedoch in Datenräumen noch nicht ausreichend genutzt.

Diese Dissertation erforscht die semantischen Grundlagen von Datenräumen und integriert semantische Technologien zur Verbesserung der Dateninteroperabilität und des gemeinsamen Verständnisses. Zudem werden semantische Technologien weiterentwickelt, um ihre Wirkung zu verstärken. Diese Arbeit entwickelt neue Konzepte, Prinzipien, Best Practices und Lösungen zur Verbesserung des Datenmanagements in Datenräumen. Die Kernbeiträge dieser Arbeit sind:

- **Ein angepasstes Datenlebenszyklusmodell:** Ein strukturiertes Modell zur Verwaltung von Daten und Metadaten in Datenräumen, welches die Herausforderungen und Anforderungen von Datenräumen berücksichtigt.
- **Prinzipien von Informationsmodellen:** Ein gemeinsamer Kern für die strukturierte Darstellung von Daten, Diensten, Teilnehmern und Interaktionen als Grundlage für Datensouveränität und Interoperabilität.
- **Erweiterte Datenzugangsprinzipien:** Für verschiedene Arten von Heterogenität eine Verbesserung der Datenzugänglichkeit von Daten durch Ressourcenmanagement, dynamische Datenänderungen und umfangreiche Metadaten. Dies erhöht die Quantität und Qualität von Informationen in Datenräumen.
- **Methoden zur Verbesserung des gemeinsamen Verständnisses:** Techniken zur Identifizierung, Entwicklung von Vokabularen, Einbeziehung von Fachfremden, Empfehlungssysteme, Informationsaktualisierung über Zeit und Validierung zur Verbesserung der Dateninteroperabilität und -(nach)nutzung.

Durch die Kombination dieser Beiträge integriert diese Arbeit semantische Technologien in Datenräume. Dies verbessert die Interoperabilität von Daten und Diensten, erleichtert ein gemeinsames Verständnis und fördert das Management und die Wiederverwendung von Daten während des gesamten Lebenszyklus. Diese Arbeit liefert Erkenntnisse über das effektive Management von Informationen im digitalen Ökosystem der Datenräume und trägt zum Fortschritt der datengetriebenen digitalen Transformation bei.

Acknowledgments

This dissertation is the result of my longest and hardest journey so far. Over the years, I have done research at three employers: the Chair of Information Management in Mechanical Engineering (IMA) at RWTH Aachen University, the Fraunhofer Institute for Applied Information Technology (FIT), and the Chair of Databases and Information Systems (DBIS) at RWTH Aachen University. Although a Ph.D. thesis is an individual accomplishment, research is a team effort driven by open exchange, discussion, collaboration, support, and fun. I would like to express my deepest gratitude to everyone who has supported me along the way. I would like to pay special tribute to a few people.

First of all, I would like to thank my dissertation committee. I especially thank Stefan Decker for the opportunity to complete my Ph.D. under his supervision and in the context of various projects on dataspace and the Cluster of Excellence “Internet of Production”. Stefan Decker created an environment at DBIS and Fraunhofer FIT where I could apply my skills to a variety of challenging tasks and conduct cutting-edge research with all the resources I needed. His expertise, interest, research fellows, and trust pushed me forward. I would also like to thank Christoph Quix for our collaboration on dataspace and data management systems, his great responsibility, encouragement in hard times, scientific acumen, valuable feedback on my research, and review of my thesis. Thanks to Edward Curry for his outstanding work on data spaces, semantic technologies, and data infrastructures in general. Our discussions and workshop series have been fruitful for my research. Finally, I would like to thank Gerhard Lakemeyer for chairing my dissertation committee.

Major parts of this work have been funded by the German Research Foundation (DFG) under the project number 442146713 (NFDI4Ing – National Research Data Infrastructure for the Engineering Sciences) and within the framework of the German Excellence Strategy – EXC-2023 Internet of Production – 390621612, by the Modernity Fund (mFUND) of the Federal Ministry of Transport and Digital Infrastructure (BMVI) under the grant 19F2083B (Mobility Data Space), by the Federal Ministry of Education and Research (BMBF) under the grant numbers FAIRDS05 (FAIR Data Spaces) and 01|S17031 (InDaSpacePlus), by the Federal Government Commissioner for Culture and the Media (BKM) under the grant number 2522DIG012, and by the Gaia-X Federation Services.

In the following, I would like to express my gratitude to many important people (in no particular order and disregarding titles) who guided and supported me throughout my studies.

I am grateful for the environment, project work, solved problems, discussions, research and fun with my former and current colleagues. For my time at IMA I thank Lucian Poth, André Pomp, Andreas Kirmse, Vadim Kraus, Alexia Fenollar Solvay, Christian Kohlschein, Alexander Paulus, Jannik Peters, Vladimir Samsonov, Christian Schwier, Nicolai Mathar, Max Haberstroh and Tobias Meisen. Special thanks to André Pomp for convincing me to pursue a Ph.D. and to Tobias Meisen for always pushing me to aim high. For my time at FIT, I thank Iraklis Dimitriadis, Christoph Lange-Bever, Haydar Akyürek, Matthias Böckmann, Paul Moosmann, Lina Molinas Comet, Philipp Hochmann, and Christoph Quix. Special thanks to Christoph Lange-Bever for creating so many opportunities and pushing me to

aim high. For my time at DBIS, I thank Lars Gleim, Sascha Welten, Alexander Neumann, Md. Rezaul Karim, Kim Fidomski, István Koren, Sandra Geisler, Felix Schwinger, Michal Slupczynski, Lasse Nitz, Till Döhmen, and Soo-Yon Kim. Special thanks to Lars Gleim for encouraging me to join DBIS and for the deep scientific discussions, and to Md. Rezaul Karim, Sascha Welten, and Sandra Geisler for inspiring me to keep going.

I am also grateful for the countless hours of administrative (and sometimes moral) support without which my productive work would not have been possible. For IMA, my thanks go to Julia Theissen, Sarah Bujarski, Claudia Capellmann, Ralf Schmitz, Nicolai Mathar, and Margit Werden. For FIT, my thanks go to Bettina Ruland, Leany Maaßen, Rinora Isufi, and Sandra Gattys-Schweer. For DBIS, my thanks to Leany Maaßen, Reinhard Linde, Tatianna Liberzon, and Romina Mennicken. Special thanks to Leany Maaßen for her continued support on so many levels.

I further want to thank all of my co-authors, thesis co-supervisors, and collaborators for everything we have accomplished together. Special thanks to (in alphabetical order): Benedikt Arnold, Sebastian Bader, Pascal Bibow, Michael Cochez, Lina Molinas Comet, Edward Curry, Manuela Dalibor, Stefan Decker, Iraklis Dimitriadis, Holger Drees, Marc Fuhrmans, Sandra Geisler, Lars Gleim, Benedikt Heinrichs, Malte Heithoff, Sayed Hoseini, Dorothea Iglezakis, Matthias Jarke, Max Kocher, István Koren, Achim Kramer, Moritz Kröger, Christoph Lange-Bever, Giacomo Lanza, Tobias Meisen, Judith Michael, Jonas Odenthal, Alexander Paulus, Jan Pennekamp, André Pomp, Sebastian Pretzsch, Christoph Quix, Michael Rath, Maximilian Rudack, Bernhard Rumpe, Vladimir Samsonov, Patrick Sapel, Katrin Schilling, Mauritius Schmitz, Niklas Schäfer, Alexia Fenollar Solvay, Sebastian Ulrich, and Andreas Wortmann. I would also like to thank the students without whom this work would not have been possible: Philipp Noodt, Erik Pinders, Siyabend Sakik, Enis Zejnilovic, Max Kocher, Maximilian Hermans, Philipp Hochmann, Hussain Ali, Paul Moosmann, Olumayowa Valerie Aborowa, and Lina Molinas Comet.

A dissertation is a long journey. Sadly, people have passed away during this time. I would have loved to tell them proudly of my accomplishment and how important they were along the way. Sarah Bujarski, you were an inspiring colleague and a great friend. Matthias Jarke, you were a great pioneer and a passionate researcher. But the most heartbreaking thing is that I can no longer talk to my beloved father about my achievements or anything else in life. Fritz, I love you and miss you so much.

Finally, I would like to thank my family and friends. This thesis is the result of hard work with the worst work-life balance I could have ever imagined. Many people have compensated a lot in my personal life and supported me in many ways. Special thanks go to my parents, Fritz and Vero, who laid the foundation for my interests early on and have been incredibly supportive on all levels ever since. I also thank Robert, Volker, Annika, Jana, Luke, Pal, Ferdi, Fabi, Marcel, Patrick, Matthias, and Karly for everything you have done and for encouraging me to keep going. Finally, but most of all, I would like to thank my wife Julia and my daughter Clara for what they had to go through for this dissertation, for their incredible support and for carrying me, especially on the last mile.

Aachen, May 2024
Johannes Theissen-Lipp



*“An expert is a man who has made all the mistakes
that can be made, in a very narrow field.”*

– Niels Bohr

Declaration of Authorship

I, Johannes Theissen-Lipp

declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I do solemnly swear that:

1. This work was done wholly or mainly while in candidature for the doctoral degree at this faculty and university;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others or myself, this is always clearly attributed;
4. Where I have quoted from the work of others or myself, the source is always given. This thesis is entirely my own work, with the exception of such quotations;
5. I have acknowledged all major sources of assistance;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published before. A detailed list can be found on the following page.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, I used OpenAI's generative AI (ChatGPT v3.5 & v4) and DeepL to improve the writing, make suggestions, and for rephrasing. After using these tools/services, I carefully reviewed and edited the content as needed. I take full responsibility for the content of this dissertation.

Published Papers

Parts of this dissertation are based on the following previously published peer-reviewed papers. All contributors to these published papers are listed as co-authors. A detailed attribution of the contributions can be found in each of the bibliographic entries below.

List of Core Publications

- [Arn+24] Benedikt T. Arnold, Johannes Theissen-Lipp, Diego Collarana, Christoph Lange, Sandra Geisler, Edward Curry, and Stefan Decker. “Towards Enabling FAIR Dataspaces Using Large Language Models.” In: *The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, May 26–27, 2024, Hersonissos, Greece*. Contributions: BA and SD identified the research gap. BA, JTL, and DC designed and conceived the work. BA performed, analyzed, and interpreted the work. BA drafted the paper. BA, JTL, DC, CL, SG, and EC reviewed and revised the paper. SD and CL provided guidance and funding. CEUR-WS. June 2024. URL: <https://ceur-ws.org/Vol-3705/short09.pdf>.
- [Bad+20] Sebastian Bader, Jaroslav Pullmann, Christian Mader, Sebastian Tramp, Christoph Quix, Andreas Müller, Haydar Akyürek, Matthias Böckmann, Benedikt Imbusch, Johannes Theissen-Lipp, Sandra Geisler, and Christoph Lange. “The International Data Spaces Information Model - An Ontology for Sovereign Exchange of Digital Content.” In: *Proceedings of the 19th International Semantic Web Conference*. Contributions: SB identified the research gap. SB, JP, CM, ST, and CL designed, conceived, and interpreted the work. JP, CM, ST, CL, JL, HA, MB, BI, and SB performed, analyzed, and evaluated the work. CQ, AM, and SG provided feedback and reviewed the paper. CL and SB provided funding and guidance. Cham: Springer International Publishing, 2020, pp. 176–192. DOI: [10.1007/978-3-030-62466-8_12](https://doi.org/10.1007/978-3-030-62466-8_12).
- [Bec+21] Fabian Becker, Pascal Bibow, Manuela Dalibor, Aymen Gannouni, Viviane Hahn, Christian Hopmann, Matthias Jarke, István Koren, Moritz Kröger, Johannes Lipp, Judith Maibaum, Judith Michael, Bernhard Rumpe, Patrick Sapel, Niklas Schäfer, Georg J. Schmitz, Günther Schuh, and Andreas Wortmann. “A Conceptual Model for Digital Shadows in Industry and Its Application.” In: *Conceptual Modeling*. Ed. by Aditya Ghose, Jennifer Horkoff, Vítor E. Silva Souza, Jeffrey Parsons, and Joerg Evermann. Vol. 13011. Lecture Notes in Computer Science. Contributions: JMi and MD identified the research gap and designed and conceived the work. MD, JMi, PB, FB, MK, JL, PS, NS, JMa, AG, VH, and GS performed, analyzed, and interpreted the work. IK and AW reviewed the paper and provided feedback. AW, GS, BR, CH, and MJ provided guidance and funding. Cham: Springer International Publishing, 2021, pp. 271–281. ISBN: 978-3-030-89021-6. DOI: [10.1007/978-3-030-89022-3_22](https://doi.org/10.1007/978-3-030-89022-3_22).
- [Dre+21] Holger Drees, Dennis Oliver Kubitz, Johannes Lipp, Sebastian Pretzsch, and Christoph Schlueter Langdon. “Mobility Data Space—First Implementation and Business Opportunities.” In: *29th ITS World Congress*. Contributions: HD and

- SP designed and conceived the basic work. JL, DK, and SP performed, analyzed, and interpreted the work. HD, DK, JL, SP, and CL drafted and revised the paper. Hamburg, Germany: ERTICO – ITS Europe, Oct. 2021, pp. 1–12.
- [LGD20] Johannes Lipp, Lars Gleim, and Stefan Decker. “Towards Reusability in the Semantic Web: Decoupling Naming, Validation, and Reasoning.” In: *Proceedings of the 11th Workshop on Ontology Design and Patterns(WOP2020) co-located with 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, November 01-06, 2020*. Contributions: JL and LG designed, conceived, performed, and interpreted the work. SD provided feedback and reviewed the paper. SD provided funding and guidance. CEUR Workshop Proceedings. 2020. DOI: [10.24406/publica-fhg-409120](https://doi.org/10.24406/publica-fhg-409120). URL: <http://ontologydesignpatterns.org/wiki/WOP:2020>.
- [Lip+20] Johannes Lipp, Maximilian Rudack, Uwe Vroomen, and Andreas Bührig-Polaczek. “When to Collect What? Optimizing Data Load via Process-driven Data Collection.” In: *Proceedings of the 22nd International Conference on Enterprise Information Systems*. Contributions: JL and MR designed, conceived, and interpreted the work. JL performed, analyzed, and interpreted the work. JL and MR drafted and revised the paper. UV and ABP provided funding and guidance. SCITEPRESS - Science and Technology Publications, 2020, pp. 220–225. ISBN: 978-989-758-423-7. DOI: [10.5220/0009439502200225](https://doi.org/10.5220/0009439502200225).
- [Lip+21a] Johannes Lipp, Lars Gleim, Michael Cochez, Iraklis Dimitriadis, Hussain Ali, Daniel Hoppe Alvarez, Christoph Lange, and Stefan Decker. “Towards Easy Vocabulary Drafts with Neologism 2.0.” In: *The Semantic Web: ESWC 2021 Satellite Events*. Ed. by Ruben Verborgh, Anastasia Dimou, Aidan Hogan, Claudia d’Amato, Ilaria Tiddi, Arne Bröring, Simon Mayer, Femke Ongenaes, Riccardo Tommasini, and Mehwish Alam. Vol. 12739. Lecture Notes in Computer Science. Contributions: JL, ID, LG, and MC designed, conceived, and interpreted the work. HA, JL, DHA, and ID performed, analyzed, and evaluated the work. JL, ID, LG, and MC drafted and revised the paper. CL and SD reviewed the paper and provided guidance and funding. Cham: Springer International Publishing, 2021, pp. 21–26. ISBN: 978-3-030-80417-6. DOI: [10.1007/978-3-030-80418-3_4](https://doi.org/10.1007/978-3-030-80418-3_4).
- [Lip+21b] Johannes Lipp, Michael Rath, Maximilian Rudack, Uwe Vroomen, and Andreas Bührig-Polaczek. “Flexible OPC UA Data Load Optimizations on the Edge of Production.” In: *Enterprise Information Systems*. Ed. by Joaquim Filipe, Michał Śmiałek, Alexander Brodsky, and Slimane Hammoudi. Vol. 417. Lecture Notes in Business Information Processing. Contributions: JL and MRu designed, conceived, and interpreted the work. JL and MRa performed, analyzed, and interpreted the work. JL, MRa, and MRu drafted and revised the paper. UV and ABP provided funding and guidance. Cham: Springer International Publishing, 2021, pp. 43–61. ISBN: 978-3-030-75417-4. DOI: [10.1007/978-3-030-75418-1_3](https://doi.org/10.1007/978-3-030-75418-1_3).
- [Lip+21c] Johannes Lipp, Siyabend Sakik, Moritz Kröger, and Stefan Decker. “LISSU: Integrating Semantic Web Concepts into SOA Frameworks.” In: *Proceedings of the 23rd International Conference on Enterprise Information Systems*. Won an award for the **Best Student Paper**. Contributions: JL and MK designed,

conceived, and interpreted the work. SS, JL, and MK performed, analyzed, and evaluated the work. JL, MK, and SS drafted and revised the paper. SD provided funding, feedback, and guidance. SCITEPRESS - Science and Technology Publications, 2021, pp. 855–865. ISBN: 978-989-758-509-8. DOI: [10.5220/0010481408550865](https://doi.org/10.5220/0010481408550865).

- [LS20] Johannes Lipp and Katrin Schilling. “The Semantic Web in the Internet of Production: A Strategic Approach with Use-Case Examples.” In: *2020 IARIA 14th International Conference on Advances in Semantic Processing*. Contributions: JL and KS designed, conceived, performed, and interpreted the work. JL and KS drafted and revised the paper. IARIA, 2020, pp. 68–72. ISBN: 978-1-61208-813-6. URL: http://www.thinkmind.org/index.php?view=article&articleid=semapro_2020_2_80_30040.
- [Mic+23] Judith Michael, István Koren, Iraklis Dimitriadis, Judith Fulterer, Aymen Gannouni, Malte Heithoff, Annkristin Hermann, Katharina Hornberg, Moritz Kröger, Patrick Sapel, Niklas Schäfer, Johannes Theissen-Lipp, Stefan Decker, Christian Hopmann, Matthias Jarke, Bernhard Rumpe, Robert H. Schmitt, and Günther Schuh. “A Digital Shadow Reference Model for Worldwide Production Labs.” In: *Internet of Production*. Ed. by Christian Brecher, Günther Schuh, Wil van der Aalst, Matthias Jarke, Frank T. Piller, and Melanie Padberg. Interdisciplinary Excellence Accelerator Series. Cham: Springer International Publishing, 2023, pp. 1–29. ISBN: 978-3-030-98062-7. DOI: [10.1007/978-3-030-98062-7_3-3](https://doi.org/10.1007/978-3-030-98062-7_3-3).
- [Moo+24] Paul Moosmann, Johannes Theissen-Lipp, Christina Gillmann, and Christoph Lange. “Enhanced RDF Validation Techniques for Dataspaces.” In: *Proceedings of the 20th International Conference on Semantic Systems, 17–19 September 2024*. Status: Accepted (currently being published). Contributions: JTL identified the research gap. PM and JTL designed and conceived the work. PM performed, analyzed, and evaluated the work. JTL and PM drafted the paper. PM, JTL, CG, and CL reviewed and revised the paper. CL provided funding. Amsterdam, the Netherlands: SEMANTiCS 2024 EU, Sept. 2024.
- [Pau+18] Alexander Paulus, André Pomp, Lucian Poth, Johannes Lipp, and Tobias Meisen. “Gathering and Combining Semantic Concepts from Multiple Knowledge Bases.” In: *Proceedings of the 20th International Conference on Enterprise Information Systems*. Won an award for the **Best Paper**. Contributions: APo and APa designed, conceived, and interpreted the work. JL and LP performed, analyzed, and evaluated the work. APo, APa, JL, and LP drafted and revised the paper. TM provided funding and guidance. SCITEPRESS - Science and Technology Publications, 2018, pp. 69–80. ISBN: 978-989-758-298-1. DOI: [10.5220/0006800700690080](https://doi.org/10.5220/0006800700690080).
- [Pau+19] Alexander Paulus, André Pomp, Lucian Poth, Johannes Lipp, and Tobias Meisen. “Recommending Semantic Concepts for Improving the Process of Semantic Modeling.” In: *Enterprise Information Systems*. Ed. by Slimane Hammoudi, Michał Śmiałek, Olivier Camp, and Joaquim Filipe. Vol. 363. Lecture Notes in Business Information Processing. Contributions: APo and APa designed, conceived, and interpreted the work. JL and LP performed, analyzed, and evaluated the work. APo, APa, JL, and LP drafted and revised the paper.

- TM provided funding and guidance. Cham: Springer International Publishing, 2019, pp. 350–369. ISBN: 978-3-030-26168-9. DOI: [10.1007/978-3-030-26169-6_17](https://doi.org/10.1007/978-3-030-26169-6_17).
- [PLM19a] Andre Pomp, Johannes Lipp, and Tobias Meisen. “You are Missing a Concept! Enhancing Ontology-Based Data Access with Evolving Ontologies.” In: *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. Contributions: JL and AP designed, conceived, and interpreted the work. JL performed and evaluated the work. JL and AP drafted and revised the paper. TM provided funding and guidance. IEEE, 2019, pp. 98–105. ISBN: 978-1-5386-6783-5. DOI: [10.1109/ICOSC.2019.8665620](https://doi.org/10.1109/ICOSC.2019.8665620).
- [PLM19b] André Pomp, Johannes Lipp, and Tobias Meisen. “Enabling the Continuous Evolution of Ontologies for Ontology-Based Data Management.” In: *International Journal of Robotic Computing* (Oct. 2019). Contributions: JL and AP designed, conceived, and interpreted the work. JL performed, analyzed, and evaluated the work. JL and AP drafted and revised the paper. TM provided funding and guidance., pp. 19–43. DOI: [10.35708/TAI1868-126244](https://doi.org/10.35708/TAI1868-126244).
- [The+22a] Johannes Theissen-Lipp, Max Kocher, Stefan Josef Decker, Patrick Sapel, Mauritius Schmitz, and Christian Hopmann. “Collecting Data and Metadata by Transforming between Differently Expressive Query Languages.” In: *Proceedings of the 16th International Conference on Semantic Computing*. Contributions: JTL and MK identified the research gap and designed and conceived the work. PS and MS provided technical infrastructure and injection molding expertise. JTL, MK, and MS performed, analyzed, and interpreted the work. JTL, MK, and PS drafted and revised the paper. SD and MS reviewed the paper and provided feedback. SD and CH provided guidance and funding. 2022. DOI: [10.1109/ICSC52841.2022.00044](https://doi.org/10.1109/ICSC52841.2022.00044). URL: <https://publications.rwth-aachen.de/record/842364>.
- [The+22b] Johannes Theissen-Lipp, Max Kocher, Michael Rath, Sebastian Ulrich, Maximilian Rudack, and Stefan Decker. “Integrating an XPath-Enhanced OPC UA Data Collection Into Industrial Communication.” In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. Contributions: JTL and MK identified the research gap and designed and conceived the work. MRu provided technical infrastructure and high pressure die casting expertise. JTL, MK, SU, and MRa performed, analyzed, and interpreted the work. JTL, MK, MRa, SU, and MRu drafted and revised the paper. SD provided feedback, guidance, and funding. IEEE, 2022, pp. 1–8. ISBN: 978-1-6654-9996-5. DOI: [10.1109/ETFA52439.2022.9921562](https://doi.org/10.1109/ETFA52439.2022.9921562).
- [The+22c] Johannes Theissen-Lipp, Moritz Kröger, Benedikt Heinrichs, and Stefan Decker. “LISSU: Continuous Monitoring of SOA Communication with Constraint-Based Validation.” In: *SN Computer Science* 3.4 (2022). Was honored with the **Young Researcher Award 2022** by the [Cluster of Excellence](#). Contributions: JL and MK designed, conceived, and interpreted the work. BH, JL, and MK performed, analyzed, and evaluated the work. JL, MK, and BH drafted and revised the paper. SD provided funding, feedback, and guidance. DOI: [10.1007/s42979-022-01170-5](https://doi.org/10.1007/s42979-022-01170-5).

- [The+23] Johannes Theissen-Lipp, Max Kocher, Christoph Lange, Stefan Decker, Alexander Paulus, André Pomp, and Edward Curry. “Semantics in Dataspaces: Origin and Future Directions.” In: *Companion Proceedings of the ACM Web Conference 2023*. WWW ’23 Companion. Contributions: JTL and SD identified the research gap and designed and conceived the work. JTL and MK performed, analyzed, and interpreted the work. JTL, MK, APo, and APa drafted and revised the paper. SD and EC provided feedback and guidance. SD provided funding. Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1504–1507. ISBN: 9781450394192. DOI: [10.1145/3543873.3587689](https://doi.org/10.1145/3543873.3587689).
- [The+24] Johannes Theissen-Lipp, Niklas Schäfer, Max Kocher, Philipp Hochmann, Michael Riesener, and Stefan Decker. “Extending Semantic RML Mappings with Additional Source Formats.” In: *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*. Contributions: JTL identified the research gap. JTL and NS designed and conceived the work. PH, JTL, and NS performed, analyzed, and interpreted the work. JTL, MK, and NS drafted and revised the paper. SD and MR provided guidance and funding. INSTICC. SciTePress, 2024, pp. 201–208.

List of Secondary Publications

- [Arn+22] Susanne Arndt, Benjamin Farnbacher, Marc Fuhrmans, Stephan Hachinger, Johanna Hickmann, Nils Hoppe, Martin Thomas Horsch, Dorothea Iglezakis, Ashish Karmacharya, Giacomo Lanza, Sophia Leimer, Johannes Munke, Džulia Terzijska, Johannes Theissen-Lipp, Cord Wiljes, and Jürgen Windeck. *Meta-data4Ing: An ontology for describing the generation of research data within a scientific activity*. Tech. rep. Contributions: SA, MF, DI, and GL identified the research gap and designed and conceived the work. All authors performed, analyzed, and interpreted the work. All authors drafted and revised the paper. NFDI, 2022. DOI: [10.5281/zenodo.5957104](https://doi.org/10.5281/zenodo.5957104).
- [Des+24] Rohit A. Deshmukh, Diego Collarana, Joshua Gelhaar, Johannes Theissen-Lipp, Christoph Lange, Benedikt T. Arnold, Edward Curry, and Stefan Decker. “Challenges and Opportunities for Enabling the Next Generation of Cross-Domain Dataspaces.” In: *The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, May 26–27, 2024, Hersonissos, Greece*. Contributions: RD, CL, JTL, and SD identified the research gap. RD, DC, JG, JTL, and BA designed and conceived the work. RD drafted the paper. RD, DC, JG, JTL, CL, BA, EC, and SD reviewed and revised the paper. SD and CL provided guidance and funding. CEUR-WS. June 2024. URL: <https://ceur-ws.org/Vol-3705/short12.pdf>.
- [Hau+24] Marco Hauff, Lina M. Comet, Paul Moosmann, Christoph Lange, Ioannis Chrysakis, and Johannes Theissen-Lipp. “FAIRness in Dataspaces: The Role of Semantics for Data Management.” In: *The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, May 26–27, 2024, Hersonissos, Greece*. Contributions: JTL, IC, and MH identified the research gap. JTL, MH, and IC designed and conceived the work. MH, LC, PM, JTL, and IC performed, analyzed, and interpreted the work. MH, LC, PM, IC, and JTL drafted and revised the

- paper. CL reviewed the paper and provided funding. CEUR-WS. June 2024. URL: <https://ceur-ws.org/Vol-3705/paper01.pdf>.
- [HTQ24] Sayed Hoseini, Johannes Theissen-Lipp, and Christoph Quix. “A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes.” In: *Journal of Web Semantics* 81 (2024). Contributions: SH designed, conceived, performed, and interpreted the work. SH drafted and revised the paper. JTL and CQ reviewed the paper and provided feedback. CQ provided guidance and funding., p. 100819. ISSN: 1570-8268. DOI: [10.1016/j.websem.2024.100819](https://doi.org/10.1016/j.websem.2024.100819). URL: <https://www.sciencedirect.com/science/article/pii/S1570826824000052>.
- [LGD21] Johannes Lipp, Lars Christoph Gleim, and Stefan Decker. “Chapter 20. Towards Easy Reusability in the Semantic Web.” In: *Advances in Pattern-Based Ontology Engineering*. Ed. by Eva Blomqvist. Vol. 51. Studies on the Semantic Web. Contributions: JL and LG designed, conceived, performed, and interpreted the work. SD provided feedback and reviewed the paper. SD provided funding and guidance. IOS Press, 2021, pp. 331–336. DOI: [10.3233/SSW210023](https://doi.org/10.3233/SSW210023). URL: <https://publications.rwth-aachen.de/record/818920>.
- [Ode+19] Jonas Odenthal, Georg Jacobs, Achim Kramer, and Johannes Lipp. “Future-Site - Test Center for the Development of Tomorrow’s Construction Site.” In: *ATZheavy duty worldwide* 12.4 (2019). Contributions: JL and JO designed, conceived, performed, analyzed, and interpreted the work. JO and JL drafted and revised the paper. AK contributed feedback and reviewed the paper. GJ and AK provided funding and guidance., pp. 58–61. ISSN: 2524-8782. DOI: [10.1007/s41321-019-0057-z](https://doi.org/10.1007/s41321-019-0057-z).
- [Ott+18] Thomas Otte, Nora Metzner, Johannes Lipp, Max Schulze Schwienhorst, Alexia Fenollar Solvay, and Tobias Meisen. “User-centered Integration of Automated Air Mobility into Urban Transportation Networks.” In: *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. Contributions: TO, JL, and NM designed, conceived, performed, analyzed, and interpreted the work. TO, JL, and NM drafted and revised the paper. MS and AFS contributed feedback and reviewed the paper. TM and AFS provided funding and guidance. IEEE, 2018, pp. 1–10. ISBN: 978-1-5386-4112-5. DOI: [10.1109/DASC.2018.8569820](https://doi.org/10.1109/DASC.2018.8569820).
- [Pau+18] Alexander Paulus, Andre Pomp, Lucian Poth, Johannes Lipp, and Tobias Meisen. “Towards Combining Semantic Concepts from Multiple Knowledge Bases.” In: *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. Contributions: APo and APa designed, conceived, and interpreted the work. JL and LP performed, analyzed, and evaluated the work. APo, APa, JL, and LP drafted and revised the paper. TM provided funding and guidance. IEEE, 2018, pp. 312–313. ISBN: 978-1-5386-4408-9. DOI: [10.1109/ICSC.2018.00062](https://doi.org/10.1109/ICSC.2018.00062).
- [Sam+19] Vladimir Samsonov, Johannes Lipp, Philipp Noodt, Alexia Fenollar Solvay, and Tobias Meisen. “More Machine Learning for Less: Comparing Data Generation Strategies in Mechanical Engineering and Manufacturing.” In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. Contributions: JL and VS designed, conceived, and interpreted the work. PN, JL, and VS performed and analyzed the work. JL, VS, and PN drafted and revised the paper. AFS re-

viewed the paper. TM and AF provided funding and guidance. IEEE, 2019, pp. 799–807. ISBN: 978-1-7281-2485-8. DOI: [10.1109/SSCI44817.2019.9002663](https://doi.org/10.1109/SSCI44817.2019.9002663).

- [Sch+22] Sven Schiller, Markus Landwehr, Georg Vinogradov, Iraklis Dimitriadis, Haydar Akyürek, Johannes Lipp, Philipp Ganser, and Thomas Bergs. “Towards Ontology-based Lifecycle Management in Blisk Manufacturing.” In: *Procedia CIRP* 112 (2022). Contributions: SS, JL, ML, and ID designed, conceived, and interpreted the work. JL, HA, ID, GV, and SS performed, analyzed, and evaluated the work. SS, HA, and JL drafted and revised the paper. PG and TB reviewed the paper., pp. 280–285. ISSN: 22128271. DOI: [10.1016/j.procir.2022.09.085](https://doi.org/10.1016/j.procir.2022.09.085). URL: <https://www.sciencedirect.com/science/article/pii/S2212827122012483>.

- [TDC23] Johannes Theissen-Lipp, Stefan Decker, and Edward Curry. “The First International Workshop on Semantics in Dataspaces.” In: *Companion Proceedings of the ACM Web Conference 2023*. WWW ’23 Companion. Contributions: JTL and SD identified the research gap and designed, conceived, and interpreted the work. JTL, SD, and EC planned and organized the workshop. JTL drafted the paper. SD and EC reviewed the paper. SD provided funding. Austin, TX, USA: Association for Computing Machinery, 2023, p. 1439. ISBN: 9781450394192. DOI: [10.1145/3543873.3589750](https://doi.org/10.1145/3543873.3589750).

Supervised Theses

Parts of this dissertation are based on the following bachelor and master theses that I supervised. For the theses where I was the first advisor, I conceived the project idea, goals, and basic architecture and evaluation design, while my co-advisors provided feedback and domain expertise. As a secondary advisor, I provided feedback and expertise in the same way. The student developed the detailed thesis concept, implementation and evaluation, and the thesis report. All advisors provided feedback and reviewed the solutions and manuscripts.

List of Supervised Theses

- [Abo22] Olumayowa Valerie Aborowa. “Enriching Model-Driven Generation of Enterprise Information Systems with Ontologies.” Advisors: Johannes Theissen-Lipp and Malte Heithoff. Master’s thesis. RWTH Aachen University, 2022.
- [Ali22] Hussain Ali. “Improving Node Embedding via Additional Similarity Metrics Based on Ontology Class Proximity.” Advisor: Johannes Theissen-Lipp. Master’s thesis. RWTH Aachen University, 2022.
- [Com22] Lina Teresa Molinas Comet. “Establishment of a Guideline for the Intuitive Creation of Semantic Models in the Internet of Production (IoP).” Advisors: Iraklis Dimitriadis, Patrick Sapel, and Johannes Theissen-Lipp. Master’s thesis. RWTH Aachen University, 2022.
- [Her22] Maximilian Hermans. “Expert Interview-based Recommendations for Semantic Uplifting Tools in the Internet of Production.” Advisors: Johannes Theissen-Lipp and Niklas Schäfer. Bachelor’s thesis. RWTH Aachen University, 2022.
- [Hoc22] Philipp Hochmann. “Erweiterung von semantischen RML Mappings am Beispiel von heterogenen Quelldaten.” Advisors: Johannes Theissen-Lipp and Niklas Schäfer. Master’s thesis. RWTH Aachen University, 2022.
- [Koc21] Max Kocher. “Verbesserter Datenzugriff in OPC UA durch Transformation von XQuery-Ausdrücken.” Advisors: Johannes Lipp and Patrick Sapel. Bachelor’s thesis. RWTH Aachen University, 2021.
- [Moo22] Paul Moosmann. “Enhancing SHACL Validation Through Constraint Templates and Inference.” Advisor: Johannes Theissen-Lipp. Master’s thesis. RWTH Aachen University, 2022.
- [Noo19] Philipp Noodt. “Energy Control: Predicting a Machine Tool’s Energy Consumption Using Machine Learning.” Advisors: Johannes Lipp and Vladimir Samsonov. Master’s thesis. RWTH Aachen University, 2019.
- [Pin19] Erik Pinders. “A Tailored Communication Infrastructure for the Construction Site of Tomorrow.” Advisors: Johannes Lipp and Jan Pennekamp. Bachelor’s thesis. RWTH Aachen University, 2019.
- [Sak20] Siyabend Sakik. “Integrating Semantic Web Concepts into an RPC Framework.” Advisors: Johannes Lipp and Moritz Kröger. Bachelor’s thesis. RWTH Aachen University, 2020.
- [Zej21] Enis Zejnilovic. “A Batch Recommender for Fast Ontology Prototyping.” Advisors: Johannes Lipp and Lars Gleim. Master’s thesis. RWTH Aachen University, 2021.

Preface

This thesis is intended to be read as a digital-first document. As such, it makes extensive use of hyperlinks throughout the text and employs color schemes and figures optimized for on-screen reading.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Research Questions	4
1.3	Thesis Outline & Contributions	5
2	Background and Research Context	6
2.1	Background on Dataspaces	6
2.1.1	Dataspace Initiatives	6
2.1.2	Foundations of Dataspaces	10
2.2	Background on Semantics	11
2.2.1	Semantics Web Technologies	11
2.2.2	Evolving Semantic Web Standards	21
2.3	Application Domains of Dataspaces	24
3	Developing a Data Lifecycle Model for Dataspaces	30
3.1	Methodology and Requirements	30
3.2	Analyzing the State of the Art	32
3.2.1	General Data Lifecycle Models	32
3.2.2	Data Lifecycle Models in Dataspace Initiatives	45
3.3	Towards a Data Lifecycle Model for Dataspaces	48
4	Information Models for Dataspaces	51
4.1	Definition and Principles of Information Models	52
4.2	Example 1: The IDS Information Model	53
4.3	Example 2: The Digital Shadow Reference Model	58
4.4	Discussion and Conclusion	61
5	Principles of Heterogeneous Data Access for Dataspaces	63
5.1	Requirements and Approach	64
5.2	Enabling Data Accessibility from Heterogeneous Systems	65
5.2.1	Related Work and Background	67
5.2.2	Approach	67
5.2.3	Results	68
5.2.4	Evaluation	72
5.3	Adaptive Data Load Management	79
5.3.1	Motivating Domain Example	79
5.3.2	Background and Related Work	81
5.3.3	Approach	84
5.3.4	Results	85
5.3.5	Evaluation	92

5.4	Managing Data and Metadata Changes	95
5.4.1	Motivating Domain Example	96
5.4.2	Related Work and Background	98
5.4.3	Approach	101
5.4.4	Results	104
5.4.5	Evaluation	108
5.5	Generating Rich Metadata	115
5.5.1	Related Work and Background	115
5.5.2	Approach	118
5.5.3	Results	118
5.5.4	Assessment of Two Example Extensions	123
5.6	Discussion and Conclusion	126
6	Creating a Common Domain Understanding in Dataspaces	128
6.1	Requirements and Approach	129
6.2	Identification Techniques	131
6.2.1	Background and Motivation	131
6.2.2	Approach	133
6.2.3	Discussion	135
6.3	Best Practices for Vocabulary Creation	136
6.3.1	Related Work	137
6.3.2	Approach	141
6.3.3	Results	142
6.3.4	Case Study: Experiences from the Internet of Production	156
6.4	Vocabulary Creation for Non-experts	158
6.4.1	Related Work	159
6.4.2	Approach	161
6.4.3	Results	161
6.5	Recommendations for Vocabulary Use	163
6.5.1	Use Case Requirements	164
6.5.2	Related Work	165
6.5.3	Approach	171
6.5.4	Results	173
6.5.5	Evaluation	184
6.6	Evolving Semantic Annotations	192
6.6.1	Related Work	193
6.6.2	Approach	200
6.6.3	Results	200
6.6.4	Evaluation	203
6.7	Enhanced Validation Techniques	209
6.7.1	Related Work	209
6.7.2	Approach	211
6.7.3	Results	214
6.7.4	Evaluation	217
6.8	Discussion and Conclusion	220
7	Conclusion and Outlook	222
7.1	Key Contributions	223

7.2 Discussion and Limitations	224
7.3 Directions for Future Work	225
Bibliography	228
List of Figures	275
List of Tables	281
Listings	283
Appendices	284
A List of Acronyms	285
B Curriculum Vitae	289

1 Introduction

Digital transformation is changing the way we work and live at high speed. It has become a pervasive force that is reshaping industries, organizations, and societies around the world. The evolution of the digital landscape (see [figure 1.1](#)) has witnessed the emergence of the Internet, followed by The Web, where data is growing at an exponential rate. Within this vast expanse, we encounter a spectrum of data that includes a fraction of open data resources alongside a substantial reservoir of closed data that individuals and organizations seek to manage and control. Such data has emerged as a driver of innovation, enabling organizations to gain insights, make decisions, and create competitive advantages. But simply aggregating and sharing data is no longer enough; individuals are demanding control over their personal and proprietary data throughout the entire data lifecycle.

More recently, the concept of *dataspaces* has emerged, promising *data sovereignty*: Individuals and organizations can gain greater autonomy and control over their data assets. Dataspaces aim to provide a unified exchange of data from various sources and meet a multitude of requirements, including trust, interoperability, and data sovereignty. A second initiative, the *Internet of Production*, focuses on integrating products, processes, and data within the digital ecosystem of production. These initiatives have predominantly concentrated on infrastructure and integration aspects for data and participants, leaving room for other crucial dimensions of data management.

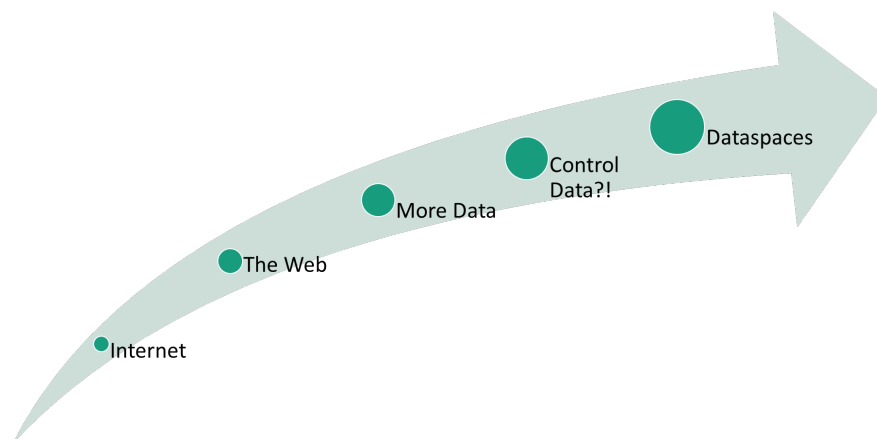


Figure 1.1: In today’s era of ever-increasing data, dataspaces promise control over data.

For successful data management, two critical factors must be addressed: Interoperability and establishing common understanding. Interoperability ensures seamless data exchange and integration across diverse systems, while common understanding fosters effective collaboration and decision-making. Achieving these objectives across the entire data lifecycle poses a significant challenge. Fortunately, the research fields of *Semantic Web*, Findable, Accessible, Interoperable, and Re-usable ([FAIR](#)) data, and semantics in general offer promising solutions to address the issues of interoperability and common understanding. The Semantic Web and the general field of semantics provide a framework for structuring and linking

1 Introduction

data in machine-readable formats, enabling automated processing and meaningful interpretation. Additionally, the [FAIR](#) data principles advocate for making data findable, accessible, interoperable, and reusable to maximize its potential.

This thesis researches on the semantic foundations of dataspaces. It aims to investigate the data lifecycle within dataspaces, with a focus on enhancing data management practices. We seek to integrate semantic technologies into dataspaces to improve data interoperability and facilitate a common understanding across different stakeholders. Additionally, the thesis aims to evolve semantic technologies to have a better impact, for example in the context of the Internet of Production. By doing so, it aims to achieve proper data management along the data lifecycle in dataspaces, leveraging improved semantic solutions.

In conclusion, this thesis sets out to explore the role of semantic technologies and solution in general in dataspaces, taking into account the challenges and opportunities presented therein. It aims to provide valuable insights into the effective management of information within the digital ecosystem, contributing to the advancement of knowledge in the field of data-driven digital transformation.

Chapter Outline

This chapter is structured as follows: We first motivate challenges on global scale and subsequently derive research questions for this thesis to address effective data management within dataspaces. We finally present the outline of this thesis, including an overview of our contributions.

1.1 Motivation

The exponential growth and increasing diversity of data on a global scale has led to a complex and interconnected digital landscape. In this information-rich environment, there is a need for a comprehensive understanding of the semantic foundations underlying the management and exchange of data. While traditional database systems have addressed some aspects of data management, they often fall short in providing a holistic view of the global digital landscape as a socio-technical system.

A fundamental challenge of the global digital landscape is the need for interoperability, common understanding, and data sovereignty. As the on-demand integration of globally distributed organizations, data, and services promises to unlock value and create advantage, methods for enabling interoperability are needed. The sheer magnitude of data, combined with the interconnectedness of disparate information structures and infrastructures, necessitates the establishment of common languages that can accurately and reliably describe resources to enable their use and reuse. Data sovereignty, including solutions to control data and services even after they have been shared, is the third pressing need that arises when balancing the benefits and drawbacks of sharing information.

Dataspaces provide a framework that goes beyond traditional approaches such as databases and aims to encompass the socio-technical aspects of the information landscape. In recent years, several dataspace efforts (especially in Europe) have been launched to manage and exchange data in specific domains or communities, such as mobility, manufacturing, and research. Data sovereignty is one of the core principles of dataspaces, distinguishing them from open data portals and enabling exclusive self-determination of economic data goods. Over

the last decade, different definitions of dataspace have emerged [Cur20], but there is still no standardized guidance on how to implement them. In 2020, the European Commission underlined the importance of dataspace by announcing in the European Data Strategy an investment of €4-6 billion in dataspace and cloud infrastructures [Eur20a].

The research community’s point of view on dataspace started with initial proposals and then further developed over about twenty years. Franklin and Halevy in [FHM05; HFM06] envisioned dataspace as one of the first ways to deal with the high heterogeneity of data sources, as it allows data access to be abstracted using shared concepts, in contrast to relational databases. Dataspace do not provide an integration layer to bridge heterogeneity, but keep every data source as is, without the need to transfer data to a centralized storage location. Skipping the integration or homogenization part of the data exchange reduces the upfront workload and focuses on a best-effort approach to data integration “as needed” [Cur20]. Several other definitions of dataspace emerged over the years [Cur20], often based on the original idea from Franklin and others. They often focus on a specific characteristic or a specific approach, but do not provide a general definition of what a *dataspace* is. Otto et al. in [OHW22] describe dataspace as a business collaboration format driven by the desire to achieve shared goals. More and more sophisticated and specialized approaches were established up to today.

In parallel, the field of semantics (esp. of data) received attention in scientific communities, while its potential is not yet fully utilized in actual implementations of dataspace [Aue22]. The Semantic Web was proposed by Tim Berners-Lee in 2001 [BHL01] as an extension of the World Wide Web that aims to enable machines to understand and process information on the web. A summary of two decades of the Semantic Web was published by Hogan in [Hog20] in 2020: The foundation of the Semantic Web lies in the use of standardized technologies such as Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL Protocol and RDF Query Language (SPARQL) to represent and query data in a structured and machine-readable format. The goal of the Semantic Web is to facilitate automated knowledge sharing, intelligent search, and the integration of data from various sources to enable more meaningful and context-aware interactions on the web. Semantic Web solutions in general do not intend to replace other solutions like relational databases or machine learning, but aim to cooperate closely with them. Among many other concepts and solutions, this field proposes an improved implementation of identifiers.

The importance of the semantic foundations of dataspace is reflected in high-level descriptions, but is neither sufficiently incorporated in practice nor sufficiently investigated in research. Research gaps include (i) a clear understanding of dataspace initiatives and the term *dataspace* itself, (ii) information models providing common understanding in dataspace, (iii) access to heterogeneous data within these, (iv) and methodologies for proper information use and reuse. In summary, there is indeed a need to leverage and enhance the semantic foundations of dataspace to facilitate effective data management and exchange in the context of the global digital landscape.

1.2 Research Questions

In pursuit of effective management of data within dataspace and related initiatives such as the Internet of Production, we consider four core research questions to address the potential of semantics throughout the data lifecycle:

RQ 1. *What are the key stages and challenges in the data lifecycle within dataspace, and how can a comprehensive data lifecycle model address these challenges?*

This research question aims to investigate the stages and challenges in the data lifecycle within dataspace, with the objective of developing a comprehensive data lifecycle model to address these challenges. It seeks to identify the key stages involved in managing data within dataspace and to understand the obstacles that arise during its lifecycle, such as data integration issues and data reusability concerns.

RQ 2. *How can both human and machine understanding be achieved in dataspace and what are useful information models to agree on general concepts within dataspace?*

This research question explores the influence of common information models and shared vocabularies on the design, implementation, and usage of dataspace. It considers both human and machine understanding, and addresses conceptual representations and common languages for shared interpretation. In addition, this research question aims to develop effective information models for practical application and usefulness within dataspace.

RQ 3. *How can data access from heterogeneous sources, such as manufacturing systems, be effectively facilitated within dataspace to support seamless integration and utilization of diverse data?*

This research question focuses on the challenge of accessing data from heterogeneous sources, specifically within the context of manufacturing systems, and aims to explore methods for enabling effective data integration and utilization within dataspace. This exploration includes an examination of strategies and techniques for facilitating data access from diverse sources to address the challenges associated with integrating heterogeneous data into dataspace. The goal is to enable seamless integration and utilization of data from manufacturing systems and similar sources, contributing to improved data interoperability for following data lifecycle phases.

RQ 4. *How can reuse of data and knowledge within dataspace be achieved, and how can recommendations, best practices, and validation mechanisms enhance data and metadata quality to support this reuse?*

This research question aims to explore strategies and approaches that enable the reuse of data and knowledge within dataspace. A particular emphasis lies on the role of recommendations, best practices, and validation mechanisms in improving the quality of data and metadata. Recommendations and best practices provide guidance on data integration, analysis, and utilization, while validation mechanisms ensure the reliability and accuracy of the data and metadata. By investigating these aspects, the research aims to identify methods for enhancing metadata quality and promoting valuable reuse of data and knowledge within dataspace.

1.3 Thesis Outline & Contributions

To answer the research questions formulated above, this thesis is structured as follows:

- [Chapter 1](#) introduces the context of this work, motivates our research, and derives research questions for the remainder of this thesis.
- [Chapter 2](#) explores the theoretical and practical foundations, including dataspace, background on semantics, and application domains of this thesis.
- [Chapter 3](#) addresses [RQ 1](#) by identifying and evaluating data lifecycle models ([DLCMs](#)) and finally proposing a [DLCM](#) for dataspace that serves as a foundation for the following chapters.
- [Chapter 4](#) addresses [RQ 2](#) by defining the principles of information models and presenting two information models that define common core aspects of dataspace.
- [Chapter 5](#) addresses [RQ 3](#) by advancing the principles of data access for dataspace, covering various types of heterogeneity. It presents methods for enabling data access from heterogeneous systems, adaptively managing data loads, managing changes in data and metadata, and generating rich metadata.
- [Chapter 6](#) addresses [RQ 4](#) by contributing to a common domain understanding in dataspace. It includes methods for identification techniques, for establishing best practices, for involving non-experts in vocabulary creation, for creating recommendations for vocabulary (re)use, for evolving semantic annotations, and for improving validation techniques.
- [Chapter 7](#) concludes our work, lists our contributions, discusses its limitations, and motivates future research directions.

2 Background and Research Context

This chapter lays the foundation for this thesis by presenting its background and research context.

Chapter Outline

This chapter is structured as follows: [Section 2.1](#) first presents the fundamental principles of dataspace, including concrete initiatives in practice as well as the research perspective. [Section 2.2](#) then elaborates on the background of semantics, covering semantic technologies as well as the Linked and [FAIR](#) data principles. [Section 2.3](#) finally presents the application domains of this work.

2.1 Background on Dataspace

This section presents the background of dataspace, since the contributions of this thesis are mainly designed for and evaluated in dataspace. Our presentation is twofold: First, we identify, describe, and analyze real-world dataspace initiatives driven by a variety of stakeholders, including industry. Note that the landscape of dataspace initiatives is an evolving field, and we are working with the current snapshot of 2023 in this work. Then, we present the theoretical foundations of dataspace from an academic perspective. Note that substantial parts of both of the following sections were published while this thesis was being completed [[The+23](#)].

2.1.1 Dataspace Initiatives

This section presents a complete list of dataspace initiatives based on defined criteria and is intended to give an overview of the broad dataspace landscape. As the landscape of dataspace initiatives continues to evolve, this thesis provides a snapshot of the situation in 2023. The criteria for the following survey on dataspace initiatives are: National (German) or international (European) scope, because it is consistent with the predominant influence of European values in the evolving landscape of dataspace, particularly emphasizing key aspects such as data sovereignty. An industrial or research focus, because it allows distinguishing between initiatives tailored to industrial applications and those geared towards research and development, providing insights into the different purposes driving dataspace initiatives. A sufficiently high technology readiness level (e.g., 3 or higher), because it ensures that the surveyed initiatives have moved beyond the conceptual or experimental stage, indicating a level of maturity and feasibility that holds promise for practical implementation and adoption. Convincing community, impact, and reach, because it ensures that the selected initiatives have garnered significant support, recognition, and engagement, underscoring their potential

to make a meaningful impact in the dataspace domain. Table 2.1 summarizes the following dataspace initiatives, their scope, and their goals.

In 2014 the Fraunhofer Society in Germany launched the Industrial Data Space [Ott+16] initiative with the objective of creating a reference architecture enabling sovereign and secure data exchange between companies without them losing control over their data. In 2016 the Industrial Data Spaces Association was founded as a legal entity, which was renamed to the International Data Spaces Association (IDSA), and the initiative to International Data Spaces (IDS) in 2018 [OJ19]. The IDSA published the version 3 of the IDS *reference architecture model* in 2019 [Ott+19] and the most recent version 4.2.0 in 2023 [Int23], and defines five essential services for enabling sovereign data exchange¹. These services are concerned with the certification of participants and components, the management of metadata, participants and issued certificates, and the enforcement of security rules. The IDS Information Model [Bad+20b] is an RDF Schema (RDFS)/OWL ontology, which defines fundamental concepts for describing actors, their interactions, the resources exchanged by them, and data usage restrictions. This model introduces semantics into the IDS and works in conjunction with the FAIR principles, which are implemented in all major IDS components². It comes with application profiles, which for example reuse contents from Data Catalog Vocabularies (DCATs), to guide users during application. For more details, see section 4.2. The IDSA Data Space Radar [Mer23] lists various dataspace projects based on the IDS reference architecture, which for example includes the *Mobility Data Space* [Dre+21].

The IDS references architecture is one of the building blocks for the international initiative Gaia-X, which aims “to develop an open software layer of control, governance, and the implementation of a common set of policies and rules to be applied to any existing cloud/ edge technology stack to obtain transparency, sovereignty and interoperability across data and services” [Bon21; Pla19]. Gaia-X states to be based on European values, as those formulated by the joint declaration of the EU member states³. Gaia-X proposes machine-readable *self descriptions* [SL22] that follow Web standards such as the Linked Data principles [Wor09] and use the *World Wide Web Consortium (W3C) Verifiable Credentials Data Model* [Spo+19; GAI22]. These self descriptions add semantics to artifacts like participants or services, and come with application profile that implement mandatory fields and best practices. Gaia-X defines a *dataspace participant* as “a legal or natural person, who is identified, onboarded, and has a Gaia-X self description. A participant [...] may be a provider, a consumer, or a federator.” [Pol22; GAI22]. Ten Gaia-X lighthouse projects⁴ tackle various domains and aim for a coherent dataspace. One example in our 2023 observations is Catena-X⁵, which aims to create a secure and standardized data-based ecosystem for companies ranging from original equipment manufacturers and tier suppliers to small and medium enterprises across the automotive value chain.

¹<https://internationaldataspaces.org/adopt/essential-services/>

²<https://internationaldataspaces.org/ids-and-the-fair-data-principles/>

³“The next generation EU cloud offering [...] should aim for the highest standards in terms of data protection, cybersecurity, data portability/reversibility, interoperability, transparency, openness, energy efficiency, performance and reliability. Completely interoperable, open, multi-vendor cloud platforms and services, based on European, international or open source standards, will enable users to migrate effectively to the cloud, reaping its full economic benefits and availing of a high degree of choice in the market” [EU 20].

⁴<https://gaia-x.eu/who-we-are/lighthouse-projects/>

⁵<https://catena-x.net/en/>

2 Background and Research Context

The German National Research Data Infrastructure (NFDI) [HWS21], (funded with around €900 million) consists of 26 consortia⁶ representing various fields in science, engineering, and humanities. It aims to develop solutions and services for research data management and presents the FAIR principles [Wil+16] as a key element. The NFDI was recommended in 2016 [Rat16] and ties into the European Open Science Cloud (EOSC) [EOS22], which is a co-created eco-system funded by EU Member States and Associated Countries built on recommendations from 2016 [CRI16]. Within their consortia, the NFDI provides a variety of resources including ontologies, tools, and application profile guiding proper ontology usage. The EOSC aims to provide a range of services and resources for Open Science to the research community and intends to build on semantic technologies such as persistent identifiers (PIDs), Linked Data, and the FAIR principles [EOS22]. The EOSC portal⁷ offers training material, tools, research outputs, and metadata schemas in the form of profiles. The FAIR Data Spaces project envisions to merge NFDI and Gaia-X components into dataspace building blocks for science and industry in compliance with the FAIR principles⁸.

The European Strategy for Data [Eur20a] proposed the creation of several common European dataspace aiming at various domains (e.g., health, manufacturing, finance, green deal, etc.), based on the experience with the EOSC [Eur22]. The NFDI, EOSC, and the common European dataspace share the commitment to the FAIR data principles (see section 2.2.2). In section 3.2.2 we clarify the relationship between FAIR data and semantics in dataspace.

The Data Spaces Support Centre [Fra22] aims to contribute to the creation of common dataspace that collectively create a data-sovereign, interoperable and trusted environment for data sharing. It is funded by the European Commission as part of the Digital Europe program. The recently released *Data Spaces Blueprint v1.0* [Dat24] document, which claims to be a consistent and comprehensive set of guidelines to support the development of dataspace, includes a definition of the term *dataspace*: “A *dataspace* is a distributed system defined by a governance framework. It facilitates secure and trustworthy data transactions between participants, emphasizing trust and data sovereignty. A *dataspace* supports one or more use cases.” [Dat24].

The SOcial LIinked Data (SOLID) project [Ver22], initiated by Tim Berners-Lee at MIT, focuses on data sovereignty of individuals by providing *data pods*, in which individuals (and organizations) store their data and can make them available to selected data consumers. The data is made available according to Linked Data and Semantic Web standards. As such, SOLID has similar goals to Gaia-X and the EOSC, and has also been referenced by the latter [EOS22]. Unlike the other national and supranational initiatives initiated or funded, SOLID is a grassroots initiative.

Table 2.1 summarizes the dataspace initiatives, their scope, and how they define their goals. While dataspace were initially viewed as a solution to manage heterogeneity, more recent dataspace approaches shifted the focus to the actual exchange of data while keeping data lineage and sovereignty, as proposed by the IDSA in their IDS reference architecture model.

⁶<https://www.nfdi.de/consortia/?lang=en>

⁷<https://eosc-portal.eu/eosc-providers-hub/what-are-eosc-profiles>

⁸<https://www.nfdi.de/fair-data-spaces/?lang=en>

Table 2.1: Overview of dataspace initiatives, their scope, and goals.

Name	Scope	Goal
IDS [OJ19; Ott+19; Bad+20b; Int23]	Global	“...create the future of the global, digital economy with IDS , a secure, sovereign system of data sharing in which all participants can realize the full value of their data.”
Gaia-X [Pla19; GAI22; SL22; Bon21]	Europe	“... create an open, transparent , and secure federated digital ecosystem , where data and services respond to common rules and can be securely built, collated, and shared.”
NFDI [HWS21; Rat16]	National	“... valuable data from science and research are systematically accessed, networked and made usable in a sustainable and qualitative manner for the entire German science system.”
EOSC [EOS22; CRI16; Eur20a; Eur22]	European	“...provide European researchers, innovators, companies and citizens with a federated and open multi-disciplinary environment where they can publish, find and reuse data, tools and services for research, innovation and educational purposes.”
SOLID [Ver22; W3C19]	Global	“...a specification that lets people store their data securely in decentralized data stores called Pods. Pods are like secure personal web servers for your data..”

As another example, the OpenDEI project⁹ identified the following technical and governance design principles for dataspace, which are reflected in current efforts such as the [IDS](#) and Gaia-X:

- Data sovereignty: exclusive self-determination of economic data goods
- Decentralized soft infrastructure: inter-operable implementations based on unified set of agreements (functional, technical, operational, legal, economic)
- Data level playing field: new entrants face no insurmountable barriers to entry
- Public-private governance: sound governance by representation of all stakeholders

The role of semantics as a design principle for dataspace is highlighted by the requirement for data to follow the [FAIR](#) principles (e.g., for the [EOSC](#), [NFDI](#), and the common European dataspace), and for the use of semantic descriptions (e.g., self descriptions in Gaia-X [GAI22]), and ontologies (in all dataspace).

Table 2.1 summarizes this section. While initiatives such as the [NFDI](#) act on national scope in Germany, others such as Gaia-X or the [EOSC](#) are on European scope. Only the [IDS](#) explicitly addresses the entire globe already, but the other initiatives will eventually yield technologies and solutions for a global scale. We did not specify the scope for [SOLID](#), because it is a specification and not a concrete dataspace implementation. It could however be interpreted as combination of available apps¹⁰ and its scope could range from local to global. All presented dataspace initiatives mention the use of semantics and semantic technologies

⁹<https://www.opendei.eu/>

¹⁰<https://solidproject.org/apps>

in general, but their actual implementation is still limited in depth or breadth. The ongoing developments are generally progressing in a manner that allows for a seamless integration of political and scientific strategies (see [EU 20; Eur20a; Eur22]). This thesis addresses these limitations and research gaps.

2.1.2 Foundations of Dataspaces

This section aims to provide insights on the definition of the term *dataspace* and design principles for dataspaces from a research perspective.

The scientific study of dataspaces is a central area of research that brings together principles from computer science, data management, information retrieval, and domain-specific expertise. Understanding the scientific foundations of dataspaces is essential to the development of effective techniques, methodologies, and tools for harnessing the potential of these complex information ecosystems.

Historically, traditional data management techniques have focused on centralized databases with well-defined schemas, suitable for structured data originating from controlled environments. However, today's data landscape includes a variety of structured, semi-structured, and unstructured data sources, often generated in real time from sources such as sensors and mobile devices. The inherent heterogeneity and dynamics of these sources require new approaches to data integration and analysis. Dataspaces provide a solution by enabling the flexible correlation of disparate data sources, allowing for the spontaneous discovery and exploration of relationships, patterns, and trends that may not be apparent in isolation.

The term *dataspace* was coined nearly two decades ago [FHM05] and has evolved since then. Scientific definitions range from

1. an abstraction for *data management in an identifiable scope* [HFM06] over
2. *a multi-sided data platform* connecting participants in an ecosystem [OJ19] to
3. interlinking data towards *loosely connected (global) information* [HB11a].

The research community's perspective on dataspaces began with initial proposals and evolved over a span of approximately two decades. Initially envisioned as a primary facilitator for dataspaces by the original authors [FHM05; HFM06], it was seen as a solution to address the considerable heterogeneity in data sources. This approach abstracts data access using shared concepts, distinguishing itself from relational databases.

Dataspaces do not incorporate an integration layer to bridge heterogeneity; instead, each data source remains unaltered, eliminating the need to transfer data to a centralized storage location. Franklin, Halevy, and Maier [FHM05] proposed some lightweight integration, but suggested performing most transformations on demand via a *pay-as-you-go* principle. This omission of the integration or homogenization aspect in data exchange reduces the initial workload and concentrates on an *as needed* best-effort strategy for data integration [Cur20].

Over the years, various definitions of dataspaces emerged [Cur20], frequently rooted in the original concept proposed by Franklin and others. These interpretations often emphasize specific characteristics or approaches without providing a comprehensive definition of dataspace. Otto et al. characterize dataspaces as a collaborative business format driven by the aspiration to attain shared objectives [OHW22].

As time progressed, more sophisticated and specialized approaches to dataspace have been developed, expanding the landscape of possibilities up to the present day. Many of such implementations and scientific notions follow different interpretations of the term *dataspace*, but agree on some usage of semantic technologies.

2.2 Background on Semantics

This section elaborates on the background of semantics. It includes Semantic Web technologies as well as the Linked and FAIR data principles, and a discussion on these.

2.2.1 Semantics Web Technologies

This chapter and this thesis in general follow the nomenclature of [Hit21] and refer to the term *Semantic Web* as field of research instead of a concrete artifact. While *The Semantic Web* may or may not come into existence some day, the research field *Semantic Web* exists since more than two decades by now. The term *Semantic Web Technologies* was defined by Hitzler as “the set of methods and tools arising out of the field” [Hit21]. This work uses the terms (the) *Semantic Web*, *Semantic Web Technologies*, and *semantics* to refer to the research field and such methods and tools, respectively. Hitzler mentioned three main publication channels for further reference, namely the Semantic Web journal¹¹, the Journal of Web Semantics¹², and the proceedings of the annual International Semantic Web Conference¹³.

There are three perspectives of semantics [Hit21], which all pursue efficient methods for data sharing, discovery, integration, and reuse:

1. Enhance the World Wide Web with machine-understandable information by enriching data with expressive metadata.
2. Do not limit application to the World Wide Web, but apply semantic technologies for information integration and management to establish efficient methods and tools for data sharing, discovery, integration, and reuse.
3. Follow a practical approach and investigate foundations and applications of ontologies, linked data, and Knowledge Graphs (KGs) with the W3C standards and recommendations.

The following sections cover the main parts of the Semantic Web field, but also include adjacent fields that provide concepts and solutions or possible uses in dataspace, such as the FAIR principles, Linked Data, and knowledge graphs. We call these additional solutions *evolving Semantic Web standards* (see section 2.2.2) because they add concepts to the original Semantic Web to better address emerging needs in the digital landscape. The major part of this overview is structured after the general idea of a Semantic Web stack [Ber00] proposed by Tim Berners-Lee in 2000 and an updated version of it [Ide17], which is depicted in figure 2.1 and includes additional fields such as data lifting or validation. The Semantic Web stack, also called *Semantic Web Layer Cake*, illustrates the architecture of the Semantic Web as visionary artifact. We traverse from the fundamentals displayed on the bottom to more

¹¹<https://www.semantic-web-journal.net/>

¹²<https://www.journals.elsevier.com/journal-of-web-semantics>

¹³<https://sws.semanticweb.org/content/international-semantic-web-conference-iswc>

2 Background and Research Context

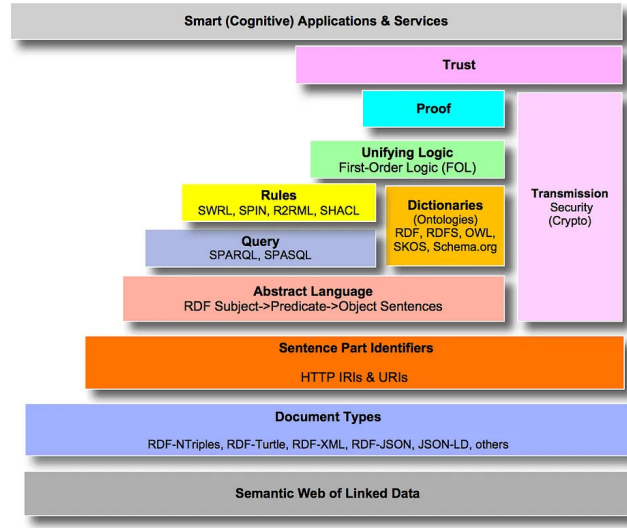


Figure 2.1: A tweaked Semantic Web stack proposed by [Ide17], based on the original proposal of Tim Berners-Lee from 2000 [Ber00].

sophisticated approaches depicted on the upper area of figure 2.1. If not stated otherwise, the definitions for individual concepts are from two books [Hit+08; DFH11] or the online documentation of available W3C standards [Wor09].

We structure the background on semantics based on the research questions (see section 1.2) and chapters of this thesis. The subsections below are organized in the following order. Chapter 4 fosters human and machine understanding via common information models, which is represented by the first subsection below. Chapter 5 enables data access from heterogeneous sources, and background for it includes document types and formats (including serialization), query methods and languages, and rules for validation techniques, and application profiles, and data mapping. Chapter 6 enables a common domain understanding to improve reuse of data and knowledge in dataspace. Background for this chapter includes rules and inference, identifiers, domain vocabularies and ontologies, semantic annotation, and tool support used in dataspace initiatives (see section 2.1.1).

Domain-agnostic Information Models

In the context of this thesis, we use domain-agnostic information models as abstract representations that capture the essential structure, relationships, and semantics of data elements, independent of any specific domain. By *domain-agnostic*, we mean that these models are designed to be applicable across domains without being tailored to any particular field. These models provide a universal framework for organizing and understanding data, enabling interoperability and integration across various contexts. The key characteristic of domain-agnostic information models is their focus on fundamental data modeling concepts, such as entities, attributes, relationships, and constraints, while intentionally omitting domain-specific details.

Domain-agnostic information models serve as a foundational layer that can be applied across different domains, reducing the need for extensive customization when communicating in dataspace. By abstracting away domain-specific complexities, these models facilitate data sharing, consistency, and efficient communication among stakeholders from different fields.

It is important to note the distinctions between information models and domain vocabularies, which includes the scope and granularity, semantic depth, reusability scope, and abstraction level. Information models provide a high-level view of dataspace organization, focusing on the “what” and “how” of data representation, whereas domain vocabularies offer specific terminology and semantics within a particular domain. Additionally, information models cater to various domains, providing a general context, while domain vocabularies are tailored to specific fields, offering specialized languages.

This common core, represented by information models, can be complemented with domain-specific extensions, such as domain vocabularies, as needed, ensuring comprehensive semantic representation and integration across relevant scopes. Examples of information models include the *IDS Information Model* and the *Reference Model for Worldwide Production Labs* from the Internet of Production project, detailed in [sections 4.2](#) and [4.3](#) respectively.

Document Types and Formats

The Extensible Markup Language ([XML](#)) is a markup language was published as [W3C](#) recommendation in 1998 as file format for sharing arbitrary data [[Bra+98](#)]. Its design goals support simplicity, generality, and usability across the Internet and it was intended to be used “*to build new formats at low cost (due to widely available tools to manipulate content in those new formats)*”¹⁴. [XML](#) is a descendant of the Standard Generalized Markup Language and describes a class of [XML](#) documents in a format that is both machine-readable and human-readable.

The XML Path Language ([XPath](#)) [[RDS17a](#)] and XML Query Language ([XQuery](#)) [[RDS17b](#)] are standardized languages to select and query nodes in XML. Nodes represent the individual components of an XML documents tree structure and have predecessors and successors. There are twelve axes in [XPath](#) (e.g., child, parent, preceding, or ancestor) that can be combined in path expressions to query nodes in a hierarchic way. The latest version 3.1 supports not only [XML](#) but also JavaScript Object Notation ([JSON](#)). [JSON](#) is a metaformat that is useful for expressing tree-based data structures¹⁵. The main advantage of [JSON](#) is that it concisely expresses the data structures commonly used in programming languages, making it very easy to create and consume. Schema languages such as the [XML](#) Schema Definition Language [[Gao+12](#); [Pet+12](#)] and RELAX NG¹⁶ enable [XML](#)-based languages to be validated. The [W3C](#) recommendation *Extensible Stylesheet Language Transformation* [[Kay07](#)] uses the underlying stylesheet language for transforming [XML](#)-like input, e.g., documents or databases, to [XML](#)-like output such as [XML](#) or HyperText Markup Language ([HTML](#)) documents.

Initially, most content on the Semantic Web was [XML](#)-based. Languages for defining vocabularies (cf. [section 2.2.1](#)) initially used only [XML](#), until Decker et al. [[Dec+00](#)] discussed the need for another representation and inference layer on top of existing layers (such as [XML](#)). Their proposal of a general method for encoding arbitrary ontology representation languages [RDF](#) and [RDFS](#) started an evolution over the years that led to other formats such as [JSON](#) and JSON-LD [[Spo+20](#)], the N-Triples syntax [[CS14](#)], or the Terse [RDF](#) Triple Language *Turtle* [[Bec+14](#)]. The increased popularity of these additional formats is due to their simplicity, interoperability, terseness, or human-readability.

¹⁴<https://www.w3.org/standards/webarch/>

¹⁵<https://www.w3.org/standards/webarch/metaformats>

¹⁶<https://relaxng.org/>

2 Background and Research Context

A [W3C](#) website states “*Protocols are the vehicle for exchanging our ideas*”¹⁷. This section covers the main languages and protocols in the scope of semantic technologies and dataspace. It is necessarily very selective, because these are two rich fields with diverse applications. Hypertext Transfer Protocol ([HTTP](#)) is the core protocol of the Web, which was standardized by the *Internet Engineering Task Force* and the [W3C](#). It is “*an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems*” [[NFB96](#)] and its current version is [HTTP/3](#) [[Bis22](#)].

Query: SPARQL

Query in the context of the Semantic Web means “*technologies and protocols that can programmatically retrieve information*”¹⁸. Just as any data sources need specific query languages, the [RDF](#)-based Semantic Web technologies need their own.

[SPARQL](#) is the most prominent candidate and provides both a query language for [RDF](#) and accompanying protocols. It originated from 2002’s SquishQL [[MSR02](#)] and was standardized by [W3C](#) in 2008 and the current revision is [SPARQL 1.1](#) [[Ara+13](#)] from 2013. [SPARQL](#) facilitates querying and manipulating [RDF](#) graph content in an [RDF](#) store or on the Web, e.g., through a [SPARQL](#) endpoint accessible via [HTTP](#). The [SPARQL 1.1](#) recommendation consists of a query language, query result formats such as [JSON](#) or [XML](#), an update language, service descriptions, entailment, and many more. Queries range from simple graph pattern matching to complex queries, with [SPARQL SELECT](#) as the most common query. Complex queries are similar to Structured Query Language ([SQL](#)) for relational databases and include features such as union, count, optional query parts, filters, value aggregation, nested queries, boolean yes/no queries, and construct queries. Technically, [SPARQL](#) queries are based on triple patterns, and a [SPARQL](#) engine returns the resources for all triples that match given patterns.

While a more comprehensive overview is given by a survey from 2005 [[Bai+05](#)], other query languages in the context of the Semantic Web include for example [[Bai+09](#)]:

- for [XML](#): [XPath](#) [[RDS17a](#)], [XSLT](#) [[Kay07](#)], [XQuery](#) [[RDS17b](#)], [XML-QL](#) [[Deu+99](#)], and [Xcerpt](#) [[SB04](#); [LS06](#)],
- for [RDF](#): [SPARQL](#) [[Ara+13](#)], [RQL](#) [[Kar+03](#)], [SeRQL](#) [[BK03](#)], [Algae](#) [[Pru04](#)], [TRIPLE](#) [[Dec+04](#)], and [Xcerpt](#) [[SB04](#); [LS06](#)], and
- for [OWL](#): the [RDF](#) query languages such as [SPARQL](#) and [OWL-QL](#) [[FHH04](#)].

Rules

This section covers rules in the context of the Semantic Web, which can be broadly separated into three areas. The first area, namely rules and inference, refers to the process of deriving logical consequences of assumed premises. That is, to derive new knowledge by combining existing facts and rules. Second, validation refers to the act of confirming something as correct. This includes verifying instances against constraints and reporting the result. Last, the area of data uplifting and mapping covers transforming or linking source data to semantically-enriched data such as [RDF](#). During this process, one achieves the planned result by applying a set of mapping rules or links to data in scope.

¹⁷Citation from <https://www.w3.org/standards/webarch/>

¹⁸Citation from <https://www.w3.org/standards/semanticweb/query>

Rules and Inference Although the two terms *inference* and *reasoning* are distinguished in logic, we in this thesis refer to them as synonyms that describe the act of deducing logical consequences from asserted axioms and thus creating new knowledge facts. Since semantic technologies usually model data as a set of relationships between resources, inference in this context means that automatic procedures generate new relationships based on data and additional information (rules)¹⁹. It is a implementation choice whether to treat such new information in the form of relationships implicitly or explicitly add it to the set of data. While classification schemes with classes and subclasses are widely used (e.g., in ontologies), rules can define a more general mechanism for generating new relationships.

There are many rule languages, but we in this overview cover three approaches for rules and inference suggested by the W3C. The Rule Interchange Format [KB13] is based on the observation that there is no one-fits-all rule language and thus focuses on exchanging rules among rules systems. It is a W3C recommendation and enables experts specifying declarative rules and production rules. The W3C standard Semantic Web Rule Language [Hor+04] includes a syntax for Horn-like rules and can be used to express rules and logic. Rules specified in this language consist of an antecedent (body) and a consequent (head), which must hold whenever the respective body holds. Finally, the SPARQL Inferencing Notation [KHI11] is a SPARQL-based rule and constraint language that is represented in RDF triples.

Validation and Application Profiles In the context of the Semantic Web, validation and application profiles play an important role in ensuring the quality and interoperability of semantic data. They are often part of quality assurance and control measures, which are defined as “*mechanisms for preventing errors from entering a data set that are used a priori to ensure high data quality before collection and to monitor and maintain data quality during and after the data collection process*” [MJ12]. Validation ensures that data conforms to a particular set of syntax and semantic rules, while application profiles provide a standard way to describe the structure and constraints of data for a particular application domain. Application profiles are typically defined using RDF vocabularies and ontologies, and their validation can be performed using tools such as RDF validators. By adhering to these standards, semantic data can be easily shared and reused across different applications and domains, leading to improved data integration and interoperability.

A basic option for validating JSON data is using JSON Schema [JSO22] to define constraints for specific key-value pairs. Although widely used, it has two major limitations that are crucial for the scope of this work. JSON Schema primarily focuses on data structure over semantics, limiting its ability to express complex semantic constraints such as comparing values of different fields, such as “*field A must be greater than field B*” directly in the schema. JSON Schema lacks native support for globally unique identifiers or PIDs, complicating the enforcement of constraints pertaining to global uniqueness within the schema without direct support for referencing external resources.

The two most common languages for validation including semantics are Shapes Constraint Language (SHACL) and Shape Expressions (ShEx), both of which support the widely used serialization formats JSON Linked Data (JSON-LD) and Turtle. SHACL is a language for validating RDF graphs against a set of conditions [KK17], which are provided as so-called *shapes* in the form of an RDF graph. These RDF graphs are called *shapes graph* in SHACL, while those being validated are named *data graphs*. SHACL shapes graphs

¹⁹See <https://www.w3.org/standards/semanticweb/inference>

can be seen as a description of conditions or as a description of data graphs that satisfy these conditions. These shapes graphs are not only used for validation but also for code generation, data visualization and integration, or the generation of user interfaces on the Web. [SHACL](#) provides multiple methods for defining shapes and targets as well as non-validating characteristics such as grouping or ordering, which can be used by user interface tools. The core constraint components include for example datatypes, cardinality, value ranges, string constraints and regular expressions, and logical operations. [SHACL](#) supports [SPARQL](#)-based constraints too, which add a lot of flexibility via [SPARQL](#)'s *SELECT* and *ASK* queries. Validators for [SHACL](#) consume shapes graphs and data graphs, and usually return a validation report in the form of an [RDF](#) graph, which states whether the data conforms to the given constraints. If not, it additionally produces a list of violations with comprehensive descriptions. [SHACL](#) started as [W3C](#) working draft in 2015 and became a [W3C](#) recommendation in 2017.

The [ShEx](#) language [[PLS14](#)] also describes [RDF](#) graph structures and can be used to validate [RDF](#) documents, assure expected patterns at interfaces, and generate forms or code for user interfaces. The terms *schema* and *model* are [ShEx](#)' counterpart of [SHACL](#)'s shapes graph and data graph, respectively. Shapes in [ShEx](#) can be defined with its [ShExC](#) syntax, which is similar to Turtle and [SPARQL](#) but is strongly inspired by regular expression languages, or using [RDF](#) serializations such as [JSON-LD](#) or Turtle. As [SHACL](#), [ShEx](#) supports a wide range of constraints, including cardinality, value range, datatype restrictions, and more. Both the validation functionality and the report of these two languages are analogous. The development around [ShEx](#) [[PLS14](#)] started in 2001²⁰ and the recent version 2.1 was published by the *W3C ShEx Community Group* in 2019²¹.

Application profiles play an important role for interoperability and contain “*set of metadata elements, policies, and guidelines defined for a particular application*”²². An application profile might use and combine elements from one or more vocabularies or contexts, and is not complete without documentation. Probably the most prominent example is the [DCAT](#) Application Profile for data portals in Europe²³, which is based on [DCAT](#) and provides interoperability with other applications by reusing established vocabularies. The many extensions of this [DCAT](#)-AP include for example [GeoDCAT-AP](#)²⁴ for describing geospatial data and services, a national extension²⁵, and [napDCAT-AP](#) [[Scr+22](#)] for transportation metadata at national access points.

Data Mapping and Data Uplifting Various source systems contain valuable metadata, but are heterogeneous in terms of format (comma-separated values ([CSV](#)), [JSON](#), [XML](#), etc.) and schema. *Data mapping* and *data uplifting* are two concepts that are used to help connect and integrate data from such different sources.

Bernstein and Melnik in [[BM07](#)] referred to data mappings as task in the generic approach of model management. We refer to *data mapping* in general as the process of transforming data from one representation to another, typically in order to enable interoperability between

²⁰<https://www.w3.org/2001/sw/wiki/ShEx>

²¹<http://shex.io/shex-semantic-20191008/>

²²<https://www.dublincore.org/specifications/dublin-core/usageguide/2001-04-12/glossary/>

²³<https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe>

²⁴<https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/geodcat-application-profile-data-portals-europe>

²⁵<https://www.dcat-ap.de/>

different systems or applications. For example, if we have data stored in one system in a particular format, and we want to use this data in another system that expects a different format, we might need to map the data from one format to another. This can involve defining mappings between the different data elements and structures, and may also involve transforming the data itself (e.g., converting data types or normalizing values). Data mapping as standalone approach usually must be accompanied by methods that ensure common understanding of the mapped data, especially when data from different source systems is merged.

More specifically, data uplifting refers to the process of converting data from a lower level of abstraction to a higher level of abstraction, typically using ontologies or other semantic models. In the context of the Semantic Web, we also refer to the term data uplifting when (semi-)structured data is transformed to [RDF](#) and enriched semantically, which allows explicit and uniform machine-readability²⁶. This process adds information to the source data by adding references to vocabularies. This information can, for example, include classification details (e.g., by describing taxonomic structures), links between entries of the data, or arbitrary metadata for parts of the data or the data as whole. This can help to make the data more easily understandable and reusable, and can also help to support reasoning and inference over the data.

An example of a mapping language is the RDF Mapping Language ([RML](#)) [[Dim+14](#)], which is based on the [W3C](#) recommendation *R2RML* [[DSC12](#)]. It is influenced by related solutions such as *RML-star* [[Igl+22](#)] and is used to convert heterogeneous data formats such as [CSV/SQL/XML](#) into [RDF](#) triples [[Dim+13](#); [Dim+20](#)].

Both data mapping and data uplifting are important techniques for integrating data from different sources. Although one of the core paradigms of dataspace is to “*keep data at source*”, it is mapped for use and reuse applications (cf. [section 3.2](#)). By using semantic models and mappings, we can create more interoperable and understandable data for dataspace so that they can support a wide range of applications and use cases.

Identifiers

An identifier is a tag that gives a unique name to an entity: a place, a person, or a thing. A [PID](#) is a permanent reference to a digital resource. Unlike Uniform Resource Locators ([URLs](#)), which can break, a persistent identifier will reliably point to a digital entity.

*“We share things by their names. [...] Uniform Resource Identifier ([URI](#)) is the way to name things on the Web and manipulate them.”*²⁷ Tim Berners-Lee motivated in a 2007 presentation²⁸ to use [URIs](#) for identifying things and highlighted persistence as key criteria for choosing [URIs](#). He elaborated on such [URI](#) persistence with the approach *Cool URIs don’t change*²⁹, which however is often not correctly carried out in practice. The following four instructions were included in the presentation and build a foundation for Linked Data (see [section 2.2.2](#)):

1. Use [URIs](#) as names for things.

²⁶Uniformity requires shared understanding of used vocabularies, which however is way easier to accomplish for vocabularies than for arbitrary source data.

²⁷<https://www.w3.org/standards/webarch/>

²⁸dig.csail.mit.edu/2007/Talks/0108-swuri-tbl

²⁹<https://www.w3.org/Provider/Style/URI>

2 Background and Research Context

2. Use [HTTP URIs](#) so that people can look up those names.
3. When someone looks up a [URI](#), provide useful information.
4. Include links to other [URIs](#), so that they can discover more things.

There is a guide on *Best Practice Recipes for Publishing RDF Vocabularies* [BP08a] and a service for permanent [URLs](#) (PURL in short) on the Web³⁰.

Over the years, several systems have been proposed for [PIDs](#), including object-focused (e.g., for documents or researchers) and object-agnostic solutions. Examples include *Uniform Resource Name* namespaces [Dai+02], ISBNs³¹, ISSN³², digital object identifiers ([DOIs](#))³³, internal identifiers (e.g., catalog numbers at the Publication Office of the European Union³⁴), the European legislation identifier³⁵, arXiv.org identifiers, PubMed IDs, the Web of Science accession number, crossref.org, DataCite, and orcid.org IDs.

Note that reliability and persistence are not provided by the concept of identifiers alone, but are purely a matter of service [KB22]. Insufficient maintenance of identifiers and links can result in identifiers no longer being dereferencable, and thus things no longer being findable or accessible [Käf+13]. This phenomenon is also referred to as *link rot* (see [section 2.2.2](#)).

Vocabularies and Ontologies

Vocabularies and similar artifacts are used to represent knowledge. The following definitions follow [Lip17] and cover the terms *concept*, *dictionary*, *taxonomy*, and *ontology*. *Concepts* or *classes* are an abstraction or generalization of ideas or objects. They extract similarities of their generalized ideas and express a meaning. A *dictionary* is a simple approach for storing knowledge and only contains concepts but no connections between these. It holds pairs of labels and descriptions, while a label often has more than one meaning³⁶. *Taxonomies* additionally add a hierarchical order in form of a tree to organize concepts. Extending these with non-hierarchical relationships leads to *ontologies* that contain meaningful expressions within specific domain areas and also may also contain axioms and restrictions. Gruber [Gru+93] in 1993 defined an ontology more generic as “a formal, explicit specification of a shared conceptualization”. Moreover, [MJ12] interprets an ontology as “a formal representation or classification of concepts and their relationships within a domain of interest”.

[RDF](#) is a language designed for expressing arbitrary information about arbitrary things, which became a [W3C](#) standard in 2004 and was completed as revision [RDF 1.1](#) [Sch+14] in 2014. It was extended with the data modeling vocabulary [RDFS](#) [BGM14], which can be seen as a semantic extension to [RDF](#). In short, [RDF](#) is a syntax for expressing directed, labeled and typed graphs [Hit21]. [XSPARQL](#) [Bis+12] enables mappings between [RDF](#) and [XML](#) by combining [XQuery](#) and [SPARQL](#) (cf. [section 2.2.1](#)). The [OWL](#) also became a [W3C](#) standard in 2004, and its revision [OWL 2](#) [Bao+12] was established in 2012. It is based on description logic and provides classes, properties, individuals, and data values in the form of [RDF](#) documents.

³⁰<http://purl.org/>

³¹International Standard Book Number <https://www.iso.org/standard/65483.html>

³²International Standard Serial Number <https://www.iso.org/standard/73846.html>

³³<https://www.doi.org/hb.html>

³⁴<https://op.europa.eu/en/web/webguide/identifiers>

³⁵<https://eur-lex.europa.eu/eli-register/about.html>

³⁶For example, the term *bank* has more than a dozen meanings: <https://www.dictionary.com/browse/bank>.

In general, RDF and OWL are use subject-predicate-object triples to describe things such as data. We define such *metadata* as “*documentation describing all aspects of the data (e.g. who, why, what, when and where) that would allow one to understand the physical format, content and context of the data, as well as possibly how to acquire, use and cite the data*” [MJ12].

Semantic Annotation

Semantic annotation refers to the process of adding metadata or semantic information to data such as web pages, documents, and multimedia resources to make the content more meaningful. In the context of the Semantic Web, semantic annotation is used to represent the content of a resource in a machine-understandable way using ontologies and controlled vocabularies. More precisely, Semantic annotation aims at the problem of finding a suitable semantic concept for a given label of a data attribute (e.g., the label of a column in a table) [GKL12; Ram+15].

The *semantic modeling* domain goes one step further by generating additional meaningful semantic relationships and more abstract concepts (e.g., combining the concepts of name, age, and hair color into the abstract concept of a person) [Tah+15; Tah+16]. The adjacent research area of *semantic tagging* aims to generate or suggest accurate tags for unstructured and binary data using various solutions such as tag ontologies, search engines like Google, or external knowledge bases such as WikiData, WordNet, DBpedia, OpenCyc or Wikipedia. Other useful sources include BabelNet, which is a fully automatic multilingual semantic network but lacks the possibility to be extended by third-parties, and WolframAlpha, which covers specific knowledge not found in other knowledge bases. Such annotations can put data into a semantic context and create domain-specific meanings to make the content easily understandable by human and machines [Pau+19]. Semantic annotation and modeling can be done manually or automatically using natural language processing and machine learning techniques. For example, Goel et al. in [GKL12] train a machine learning model that learns the representation of data to propose semantic concepts. In contrast, [Sye+10] and [Wan+12] use external knowledge bases to propose concepts based on textual data labels.

The resulting annotations, models, and tags can be used for various purposes such as search, data integration, knowledge discovery, and reasoning. Overall, semantic annotation and adjacent disciplines play a vital role for Semantic Web technologies and dataspace by enabling both machines and humans to better understand the content of resources and making it possible to create more intelligent and effective applications.

Semantic Web Tools Used in Dataspace

Table 2.2 presents an overview of Semantic Web tool support, which are used by the dataspace initiatives presented above. Note that, like the overview of dataspace initiatives above, this is a snapshot of the current landscape in 2023, which may not be complete or accurate by the time this thesis is read. The table for each tool presents its name, description, and use case. In addition, we evaluate the use of each tool in the dataspace initiatives IDS and Gaia-X, where we were able to gain deep technical insight. Protégé is used in both initiatives by both Semantic Web experts and domain experts for ontology engineering. The RDF4J framework is used in the IDS *CodeGen* tool³⁷.

³⁷<https://github.com/International-Data-Spaces-Association/Java-Representation-of-IDS-Information-Model>

Table 2.2: Semantic Web tools used in dataspace initiatives such as [IDS](#) and Gaia-X.

Tool Name	Description	Use Case
Protégé [Noy+01] and WebPro- tégé [Tud+13]	A free, open-source ontology editor and knowledge management system	Creating and editing ontologies
TopBraid Composer [Top20]	A comprehensive development environment for the Semantic Web	Developing and managing semantic web applications
OWLGrEd [Bär+10]	A graphical ontology editor for creating and editing OWL ontologies	Design ontology structures and relationships
RDF4J [Ecl16]	A Java-based framework for working with RDF data	Storing, querying, and manipulating RDF data
Virtuoso [Ope98]	A scalable RDF triplestore and SPARQL endpoint	Storing and querying large RDF datasets
Apache Jena [The11a]	A Java framework for building semantic web and Linked Data applications	Building and deploying Semantic Web applications
GraphDB [Ont15]	A high-performance RDF database and SPARQL endpoint	Storing and querying RDF data
PoolParty [Sem09]	A platform for managing enterprise KGs	Building and maintaining KGs
GATE [GAT95]	A natural language processing framework with support for RDF	Extracting and analyzing RDF data from unstructured text
WIDOCO [Gar17]	A step by step generator of HTML templates with the documentation of ontologies	Generating documentation for ontologies
WebVOWL [Loh+16 ; WLA18]	A web-based visualization tool for ontologies	Visualizing ontologies and their relationships
OWL API [HB11b]	A Java application programming interface (API) for working with OWL ontologies	Building tools that manipulate OWL ontologies
OntoHuman [Opa22]	A tool for creating human-readable documentation of ontologies	Improving documentation for ontologies
ReSpec [ReS12]	A script that makes it easier to write technical documents	Write W3C specifications and similar documents
LinkML [Mox+21]	A general purpose modeling language for Linked Data and other formalisms	Creating and managing data models
SHACL Play! [Fra20]	An online tool box for validating RDF data against SHACL constraints	Validating RDF data or visualizing SHACL

Both initiatives apply the Apache Jena framework for backend applications (e.g., a [SPARQL](#) demo in the [IDS](#)³⁸). The [IDS](#) and Gaia-X use WIDOCO and WebVOWL to generate documentation such as the releases of the [IDS information model](#)³⁹. The [OWL API](#) is thereby used implicitly, because WebVOWL builds on it. Sub-projects of the [IDS](#) use ReSpec to format and publish technical specifications. Gaia-X experiments with LinkML to simplify data model definitions for Semantic Web experts. SHACL Play! is used in Gaia-X to automatically generate visualizations in the form of Unified Modeling Language ([UML](#)) diagrams for their application profiles modeled in [SHACL](#) (c.f. a script in the *Service Characteristics* repository⁴⁰).

2.2.2 Evolving Semantic Web Standards

This section covers solutions and standards that were not part of the original Semantic Web research, but have been introduced more recently to address emerging needs. We first present and discuss Linked Data, and then present the [FAIR](#) data principles and the concept of knowledge graphs. Note that the terms *FAIR data principles*, *FAIR data*, and *FAIR principles* might be used interchangeably in both literature and this thesis. Both the [FAIR](#) principles and Linked Data are strong concepts for improving data use and reuse on multiple layers. They not only provide tools and concepts for proper semantic modeling and interlinking of data and metadata, but also give guidance on how to implement helpful actions in practice. This enables valuable information being stored for understanding by both human and machine. They go, in theory, hand in hand with Semantic Web technologies and with requirements posted by modern dataspace.

Linked Data

Linked Data is a method of publishing and interlinking structured data on the Web, making it machine-readable and interconnected [[BHB09](#)]. It follows a set of principles that include using unique and persistent [URIs](#) to identify resources, providing explicit links between resources, and using standard Web protocols such as [HTTP](#) for retrieval. By adhering to these principles, Linked Data enables the creation of a decentralized and distributed web of interconnected data sources. This linkage enables “*large scale integration of, and reasoning on, data on the Web*”⁴¹. Tim Berners-Lee proposed in this note on Linked Data⁴² four rules, which are often referred to as *Linked Data principles*:

1. Use [URIs](#) as names for things.
2. Use [HTTP URIs](#) so that people can look up those names.
3. When someone looks up a [URI](#), provide useful information, using the standards ([RDF*](#), [SPARQL](#)).
4. Include links to other [URIs](#) so that they can discover more things.

³⁸<https://github.com/International-Data-Spaces-Association/InformationModel/tree/develop/examples/queries-using-sparql>

³⁹w3id.org/idsa/core

⁴⁰<https://gitlab.com/gaia-x/technical-committee/service-characteristics/-/tree/develop/toolchain/visualization>

⁴¹<https://www.w3.org/standards/semanticweb/data>

⁴²<https://www.w3.org/DesignIssues/LinkedData.html>

Table 2.3: 5-star deployment scheme for (Linked) Open Data from 5stardata.info.

Stars / Level	Description
★	Available on the web (whatever format) but with an open licence, to be Open Data
★★	Available as machine-readable structured data (e.g. excel instead of image scan of a table)
★★★	as (★★) plus non-proprietary format (e.g. CSV instead of excel)
★★★★	All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
★★★★★	All the above, plus: Link your data to other people’s data to provide context

This note also proposes a star scheme to emphasize widespread adoption of Linked Data and specifically *Linked Open Data*, which were also published on a dedicated website⁴³. The star rating, which ranges from one to five stars, can be applied to datasets to represent their level of compliance with Linked Data principles. An overview of this 5-star scheme is shown in [table 2.3](#).

Prominent Linked Data efforts include DBPedia [[Leh+15](#)], SKOS [[MB09](#)], the Schema.org vocabulary [[GBM15](#)], and Wikidata [[VK14](#)]. The concept of Linked Data however imposes crucial challenges to real-world applications. They suggest using [URIs](#) as names for things [[HB11a](#)]. A study about Linked Data [[Käf+13](#)] found that each year about 10% of Linked Data [URIs](#) are no longer dereferencable. While the [W3C](#) suggests that “cool [URIs](#) don’t change”⁴⁴, this assumption is neither valid for the [HTML](#) Web nor for the Web of Data. For dataspace a loss of 10% per year of identifiers results in a large data deficit. Such drawbacks are caused by several reasons including:

- **Data Quality and Consistency:** Linked Data relies on data being accurately and consistently structured. However, in practice, data from different sources might vary in terms of quality, format, and semantics. Poor data quality can lead to confusion, misinterpretation, and hinder the creation of meaningful relationships.
- **High Initial Investment:** Transforming existing data into Linked Data requires a significant initial investment of time, effort, and resources. Organizations need to restructure their data, assign appropriate [URIs](#), and define relationships. This can be a barrier, especially for organizations with large and complex datasets.
- **Semantic Interoperability:** Linked Data aims to facilitate semantic interoperability by using standardized vocabularies and ontologies. However, achieving consensus on these standards across different communities and domains can be challenging. Differences in interpretation and use of terms can lead to misunderstandings and limit the effectiveness of linking data.
- **User Adoption and Accessibility:** For Linked Data to be successful, users need to understand and interact with it. If the user experience is not intuitive, or if users are not aware of how to navigate and leverage linked resources, the potential benefits might remain unrealized.

⁴³<https://5stardata.info/>

⁴⁴<https://www.w3.org/Provider/Style/URI>

One way to mitigate identifier issues is to introduce PIDs, which aim to fulfill the following principles [Sol12]: longevity, scalability, extensibility, and security. As noted in [KB22], the important to ensure longevity of persistent identifiers is that “persistence is purely a matter of service”. Which means, that persistent identifiers can only remain persistent if someone is committed to ensuring they stay accessible to users. This requires an engagement or a service level agreement for PID availability, in contrast to URIs, where no such service level agreement exists. As [Cou+21] notes, PIDs can be resolved via a URI, which follows the first principle of the Den Haag Manifesto from 2011 [Tre11]. An example is using a DOI such as 10.1145/3543873.3587689 to construct a URL such as <https://doi.org/10.1145/3543873.3587689>, which can be browsed using HTTP GET and content negotiation. With this alignment, Semantic Web tools, standards, and concepts such as schema.org, vocabularies, ontologies, RDF, JSON-LD [Spo+20], RDFa⁴⁵, R2RML [DSC12]/RML [Dim+14], SPARQL [Ara+13], and CSV on the Web⁴⁶ to link, map, query, and integrate different data formats and data sources, and KGs become usable for dataspace based on the FAIR data principles.

FAIR Principles

The FAIR principles were proposed in 2016 [Wil+16] and published on the Web by Force11⁴⁷. The acronym FAIR expands to the following principles for data: **F**indable, **A**ccessible, **I**nteroperable, and **R**e-usable. The FAIR principles initially focused on the area of scientific data management, but received widespread adoption in dataspace efforts (cf. section 2.1.1) and the European data strategy [Eur20a].

- **Findable:** Data should be easily discoverable and identifiable with rich metadata and persistent identifiers. This includes providing a unique and resolvable identifier, descriptive metadata, and clear accessibility information.
- **Accessible:** Data should be openly available to both humans and machines. It should be retrievable using standard protocols, and access should be granted with as few restrictions as possible, while respecting privacy and security concerns.
- **Interoperable:** Data should be structured in a way that facilitates integration and interoperability. This involves using standardized data formats, vocabularies, and ontologies to enable data exchange and interoperability between different systems and disciplines.
- **Reusable:** Data should be well-described, with clear and machine-readable license information. It should be accompanied by relevant documentation and provenance information, enabling others to understand and reuse the data for diverse purposes.

The FAIR principles aim to enhance the value and impact of data by promoting its discoverability, accessibility, interoperability, and reusability. By adhering to these principles, data can be more effectively shared, combined, and analyzed, fostering collaboration, innovation, and reproducibility across various domains and disciplines.

⁴⁵<https://www.w3.org/TR/rdfa-primer/>

⁴⁶<https://www.w3.org/TR/tabular-data-primer/>

⁴⁷<https://www.force11.org/group/fairgroup/fairprinciples>

Knowledge Graphs

Although the term *Knowledge Graph* has been used in academic literature since about fifty years ago [Sch73], the modern usage came with the 2012 announcement of the *Google Knowledge Graph* [Sin+12]. KGs are a recent advancement in Semantic Web technologies, offering a sophisticated approach to organizing and structuring data on the Web. Rooted in Linked Data principles and Semantic Web standards, KGs provide a framework for representing knowledge in a machine-readable format [BHB11]. By encoding information as interconnected nodes and edges, KGs establish a rich network of relationships between entities, enabling a more detailed and contextual understanding of data. This interconnectedness enables KGs to capture complex semantic relationships such as hierarchies, classifications, and associations that are fundamental to the Semantic Web paradigm. As a result, KGs facilitate the creation of intelligent and context-aware applications that can derive meaningful insights and relationships from large volumes of data [Pau17]. In addition, KGs play a central role in enabling advanced capabilities such as semantic search, entity resolution [Wan+17], and data integration across disparate sources. As a fundamental component of the Semantic Web, KGs are helping to unlock the potential of Linked Data. They provide a scalable and flexible framework for representing and navigating knowledge on the Web, driving innovation in areas ranging from artificial intelligence to data analytics and beyond.

A recent review on *Knowledge Graphs* was published by Hogan et al. in 2021 [Hog+21]. After reviewing more than a dozen tertiary literature on KGs, they define a KG “as a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities. The graph of data (a.k.a. data graph) conforms to a graph-based data model, which may be a directed edge-labeled graph, a heterogeneous graph, a property graph, and so on.” [Hog+21]. The research article covers (i) graph data models and languages for query and validation, (ii) deductive formalism and entailment, (iii) inductive techniques and knowledge extraction, and (iv) ongoing and future research directions. Deductive formalism and entailment for example builds on vocabularies and ontologies (cf. section 2.2.1).

Hogan and others identify the trends to use KGs for data integration from divers sources, and the combination of deductive (rules, ontologies) and inductive methods (machine learning, analytics). Their research article is accompanied by a GitHub repository with examples⁴⁸.

2.3 Application Domains of Dataspaces

This section presents application domains of dataspace [The+23], which act as the research context of this thesis, including manufacturing, mobility, and research data.

Dataspace in general are a central aspect in the research context of this thesis. Most of the concepts and solutions proposed in this thesis are applicable to all of the dataspace initiatives presented in section 2.1.1. We have performed concrete development and evaluation in the scope of the initiatives IDS, Gaia-X, NFDI, and the Mobility Data Space. This is due to the given project landscape as well as to their cutting-edge progress and collaborative communities. In the following, we present three application domains that all have a significant number of use cases and dataspace in the *dataspace radar* [Mer23] of the IDSA and the Data Spaces Support Centre: Manufacturing, mobility and research.

⁴⁸<https://github.com/knowledge-graphs-tutorial/examples>

Manufacturing

In the manufacturing domain, which is also called *production*, there is a wide variance in data sources (e.g., sensors and actuators from different vendors), physical location of assets (e.g., a distribution across multiple shop floors or production sites), and technology readiness levels (e.g., legacy and modern systems).

The so-called clusters of excellence establish internationally visible and competitive research institutions at German university locations and enable scientifically required networking and cooperation. In addition, they create excellent funding and career conditions for young scientists. The Excellence Strategy of the Federal Government and the Länder is intended to continue and further develop the top achievements, profiling and cooperation in the science system. Within the framework of the Excellence Strategy, project-related funding is provided for clusters of excellence in internationally competitive fields of research. Clusters of excellence enable scientific networking and interdisciplinary cooperation. They are an important component of the university's strategic plan and serve to significantly raise its profile. Since January 2019, a total of 57 clusters of excellence have been funded for an initial period of seven years, each with an average of 6.5 million euros per year, including three at RWTH Aachen University.

This thesis focuses on the *Internet of Production* [Pen+19], which visions to transform the entire world of manufacturing into one lab: *“The vision of the Internet of Production is to enable a new level of cross-domain collaboration by providing semantically adequate and context-aware data from production, development and usage in real-time, on an adequate level of granularity.”* [RWT19]. The project covers from infrastructure over materials, production technology and management, up to integrated usage.

We in the following adopt the approach of sections 2.1 and 2.2 and concentrate on semantics in manufacturing. Semantic interoperability within the Internet of Production is crucial, connecting data, models, systems, and knowledge across manufacturing stages. While Semantic Web technologies are ideal for this, their potential remains underutilized due to limited practical implementation, partly attributed to a lack of expertise among practitioners. Engineers lack Semantic Web knowledge, and Semantic Web experts lack domain knowledge. This section highlights five vital semantics application areas and matching examples in the Internet of Production to showcase research gaps and potentials.

In production technology, the concept of semantics has garnered attention, primarily in academic research projects. However, these ideas have yet to find broad acceptance or implementation within industry. The main hindrances are limited ontology knowledge among employees, lack of tool support, unclear benefits in real-world scenarios, and uncertain returns on the investment needed for extensive modeling. These challenges span various levels and domains within the industry, impacting engineers, domain experts, ontology engineers, and senior managers. Consequently, despite interesting concepts and demonstrated feasibility in demo setups, productive implementation remains rare, with most production applications remaining experimental.

Nonetheless, compelling reasons exist to persist in researching Semantic Web technologies for production engineering. The achievements of the *Industry 4.0* vision have laid the foundation for networked production. Modern automation technology products now possess requisite interfaces and communication protocols for distributed, data-driven applications. While data is more accessible, achieving the vision of the Internet of Production entails extensive system networking and data-driven optimization across the production process.

2 Background and Research Context

This is especially crucial as knowledge-based applications like artificial intelligence mature, necessitating the transformation of basic data into meaningful knowledge. The demand for semantic technologies and semantics aligns with the growing attention to protocols like Open Platform Communications Unified Architecture (OPC UA) [HSK08], which introduce semantics to data interactions and enhance interoperability. The need for operational semantic technologies within the Internet of Production becomes evident due to two factors: Improved data accessibility facilitated by advanced production system infrastructure, and the rising requirement for semantically annotated data to support novel applications.

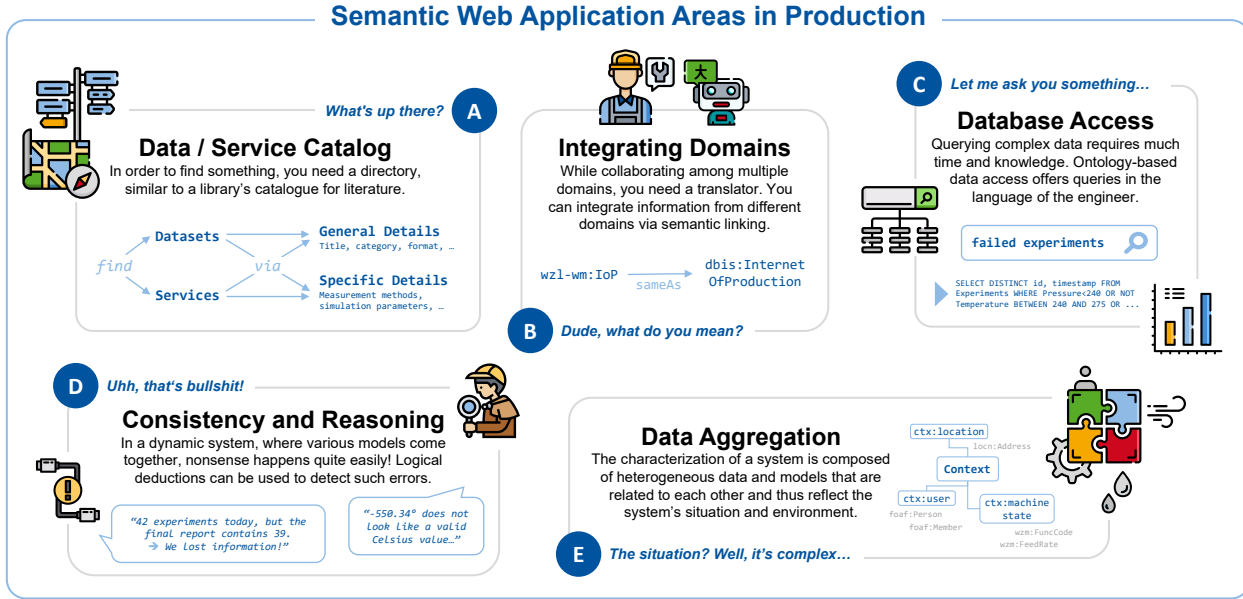


Figure 2.2: Illustration of the main Semantic Web application areas within the Internet of Production project [LS20].

Figure 2.2 depicts five application areas for semantics in the Internet of Production:

- (A) A widely used application is the **Data / Service Catalog**. It serves as a comprehensive directory of various data sources, covering datasets, services, individuals, projects, and sensors. This catalog helps users locate information using specific search criteria. Examples include open data portals like [Föd13; Pub12] or the real-time management of semantically defined functions for production processes [Los+11; Bre+18].
- (B) The domain of **Integrating Domains** serves both human comprehension and machine-level interoperability. Instead of reinventing solutions for each small domain, individuals blend insights from diverse areas to optimally represent their specific use cases. Merging segments of various knowledge sources, such as ontologies, frequently results in intersections or overlaps that are unclear for both humans and machines. Utilizing Semantic Web technologies (cf. section 2.2) can address these challenges by introducing relationships between concepts originating from different domains.
- (C) Ontology-based data access offers the opportunity for straightforward **Database Access** at a semantic tier. By establishing fundamental notions like *failed experiments* within a given context, domain experts (even without profound database expertise) can effortlessly formulate intricate queries. The objective is to disentangle user-friendly query expressions from the underlying database architecture.

- (D) The capability to verify semantically annotated data for **Consistency** and to draw conclusions via **Reasoning** proves advantageous within the intricate realm of production technology. Swift errors may arise during transitions between diverse applications, systems from various manufacturers, and different standards throughout a product’s life-cycle. Moreover, this approach can even expose logical inconsistencies within datasets.
- (E) Enabling the synthesis of intricate scenarios and environmental contexts from heterogeneous data sources, the **Aggregation of Data** and models (E) becomes valuable. Semantic connections among disparate facets that define a situation permit the abstraction of concepts alongside their distinct representations, even when encoded within different concepts.

In summary, the Internet of Production is a valuable application domain for this thesis, and its intense focus on semantics are beneficial for our research context.

Mobility

The mobility sector is a heterogeneous and often scattered ecosystem with many different stakeholders following similar goals, but with different perspectives and approaches. The exchange of mobility data between the different parties – passengers, mobility service operators, infrastructure suppliers, and many others – is a vital prerequisite for improved and new mobility services aimed at safe, efficient, and convenient mobility. The goal and need to build trusted ecosystems for secure data exchange applies to many domains including mobility. The European Commission identified this challenge in their Data Strategy [Eur20b] and proposed the creation of a European wide data market. Dedicated European dataspace are envisaged for 9 key fields, one being mobility. They envision that *“personal as well as non-personal data, including sensitive business data, are secure and businesses also have easy access to an almost infinite amount of high-quality industrial data”* [Eur20b]. In other words, data providers can be confident that their data assets will only be shared with and exploited by third parties in a regulated manner and under well-defined conditions.

Fortunately, a first implementation, the Mobility Data Space, is untying this knot by adopting dataspace concepts from the IDSA. Core components complement the decentral concept and foster discoverability, accessibility, interoperability, and trustworthiness. In a German project, a minimal viable demonstrator has been implemented, which includes core components, various data sources and first applications. It proves the feasibility of the dataspace concepts for the mobility sector and paves the way for its business evolution.

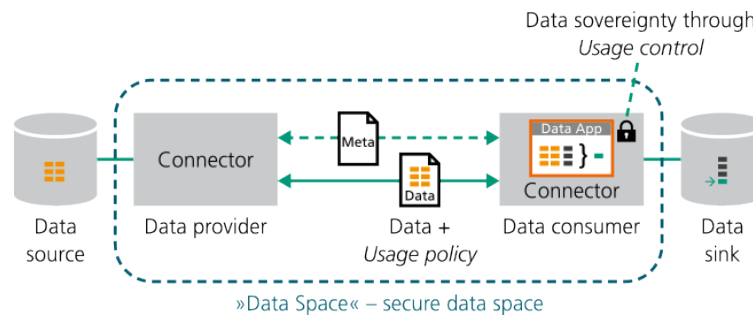


Figure 2.3: The Mobility Data Space promotes data sovereignty through data usage control mechanisms [Dre+21].

2 Background and Research Context

We in the following present the Mobility Data Space [Dre+21; Fra19] as a driver and representative for the mobility sector. The initiative applies dataspace concepts of the IDSA [Int16a; Ott+19; Int23; Bad+20b] to the mobility domain with the goal to conveniently provide mobility data and support of *National Access Points* for the European States [CMT11]. The Mobility Data Space targets to address a broad user group while conforming to data sovereignty methods and sustainability strategies [Eur20c]. Basic concepts of the Mobility Data Space are available through a white paper [Pre+20], and the project yielded the following results:

- Data sovereignty through usage control (see figure 2.3): Dataspace connectors establish connection and validate previously agreed-upon rules (*usage policies*) before they communicate. The technical sovereignty solutions build on the IDS Policy Language [Ste+21], which extends the Open Digital Rights Language [Ian+18], and usage contracts described in the IDS information model [Mad+19].
- Distributed system with central directories: Published data sources and services can be queried by integrating existing regional and national mobility data platforms like the Mobility Data Marketplace⁴⁹ or the mCLOUD⁵⁰.
- Other applied dataspace concepts include: Metadata directories, vocabulary and identity providers, data apps, and a clearing house (see [Ott+19; Int23] for details).

Figure 2.4 depicts the project’s planned ecosystem and architecture, covering public and private stakeholders and respective platforms.

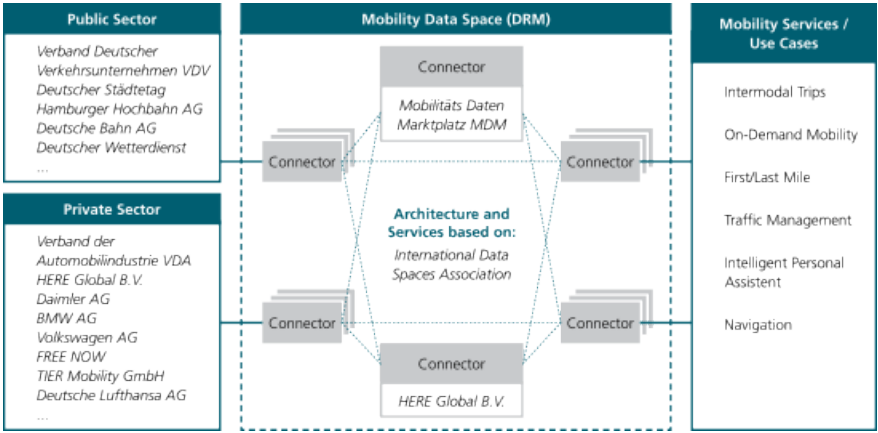


Figure 2.4: Connected participants and their data offerings in the Mobility Data Space project [Dre+21].

Research Data

Research data is the foundation of scientific investigation, providing the evidence needed to support findings and advance knowledge. Effective research data management ensures that these data are properly organized, stored, and preserved to improve accessibility and reproducibility. The research data sector faces several specific challenges that can impact the effective management, sharing, and utilization of research data:

⁴⁹<https://www.mdm-portal.de/?lang=en>

⁵⁰<https://www.mcloud.de/>, will be superseded by <https://mobilithek.info/>

- **Data Management and Curation:** Ensuring proper data organization, documentation, and metadata creation is a challenge. Researchers must invest time and effort in structuring and annotating their data to make it understandable and reusable by others.
- **Data Quality and Integrity:** Maintaining the quality and integrity of research data is crucial. Inaccuracies, errors, and inconsistencies in data can lead to unreliable research outcomes and hinder reproducibility.
- **Data Sharing and Access:** Balancing the need for open data sharing with privacy concerns, intellectual property rights, and ethical considerations is a delicate challenge. Researchers must find ways to share data while respecting confidentiality and legal boundaries.
- **Interoperability:** Research data often come in various formats, structures, and standards. Achieving interoperability among different datasets and systems is essential for effective data integration and reuse.
- **Long-Term Preservation:** Ensuring the long-term preservation and accessibility of research data is a challenge. Technological obsolescence, changes in data formats, and storage costs can threaten the sustainability of data over time.
- **Data Security:** Protecting sensitive or confidential research data from unauthorized access, breaches, and cyberattacks requires robust security measures and protocols.
- **Funding and Resources:** Adequate funding for data infrastructure, storage, and maintenance can be a challenge, particularly for individual researchers or smaller institutions.
- **Data Sharing Culture:** Building a culture of data sharing and collaboration within the research community can be challenging. Encouraging researchers to share their data openly requires incentives, recognition, and support.
- **Legal and Ethical Considerations:** Compliance with data protection regulations, ethics committees, and institutional policies adds complexity to data sharing, especially when dealing with sensitive or human subject data.
- **Technical Infrastructure:** Developing and maintaining the necessary technical infrastructure for data storage, analysis, and sharing requires continuous investment in hardware, software, and expertise.
- **Data Literacy and Training:** Researchers need to be educated about best practices in data management, including data documentation, sharing, and reuse. A lack of data literacy can hinder effective data management and collaboration.
- **Lack of Standards:** The absence of standardized protocols for data documentation, metadata, and formats can lead to inconsistency and hinder data interoperability.

Addressing these challenges requires collaboration among researchers, institutions, funding agencies, and policymakers to establish clear guidelines, provide resources, and promote a culture of responsible and open research data practices. The [NFDI](#) is a significant initiative aimed at approaching these challenges through the establishment of a comprehensive data infrastructure. See [section 2.1.1](#) for details, which include supporting open science principles, fostering interdisciplinary cooperation, and contributing to the advancement of knowledge across diverse domains within the academic realm in Germany.

3 Developing a Data Lifecycle Model for Dataspaces

This chapter addresses [RQ 1](#), which questions: *What are the key stages and challenges in the data lifecycle within dataspaces, and how can a comprehensive data lifecycle model address these challenges?*

Data and metadata are central components in dataspaces. Tackling the topic of data in dataspaces generally would be too broad and imprecise; a structured approach is required. A data lifecycle in generally involves the stages of data throughout its life as a series of iterative steps, and we refer to a model that represents this as [DLCM](#). We propose the use of [DLCMs](#) in dataspaces because they divide the data handling into stages, thereby simplifying a dedicated investigation and improvement of data handling.

There exists no consistent [DLCM](#) for dataspaces yet. This chapter closes that research gap by proposing one. We build on existing [DLCMs](#) from relevant contexts and construct one dedicated for dataspaces. It leverages all achievements from research on [DLCMs](#) and at the same time meets the requirements of modern dataspaces, which we propose in this section as well.

Chapter Outline

This chapter aims to develop a [DLCM](#) for dataspaces and is structured as follows. We first present our methodology and the requirements specific for dataspaces. [Section 3.2](#) subsequently discusses state-of-the-art [DLCMs](#), which mainly focus on data and metadata, and evaluates the defined requirements. We investigate how [DLCMs](#) are considered in existing dataspaces initiatives, before we propose a [DLCM](#) that best suits dataspaces and fine-tuning it for maximum impact in dataspaces.

3.1 Methodology and Requirements

This section presents our approach to propose a [DLCM](#) for dataspaces and introduces dataspaces-specific requirements for such a model.

Methodology

This chapter's approach is as follows and is depicted in [figure 3.1](#). First, we introduce requirements that shall be met by a [DLCM](#) to be successful in dataspaces. We then evaluate these requirements by consulting related documents, including the scientific publications as well as websites, reports, or other resources provided. Our analysis covers two aspects:

1. **DLCMs** in general: We present common **DLCMs**, which are designed to describe data and metadata in general, and evaluate how they meet our dataspace-specific requirements. We create abstract representations of each **DLCM** to simplify comparison.
2. **DLCMs** in dataspace initiatives: We investigate if and how the dataspace initiatives presented in [section 2.1.1](#) consider a **DLCM**, either explicitly or implicitly.

Finally, [section 3.3](#) builds on this state-of-the-art analysis, selects the **DLCM** candidate that best meets our requirements and fine-tunes it for optimal use in the context of dataspaces.

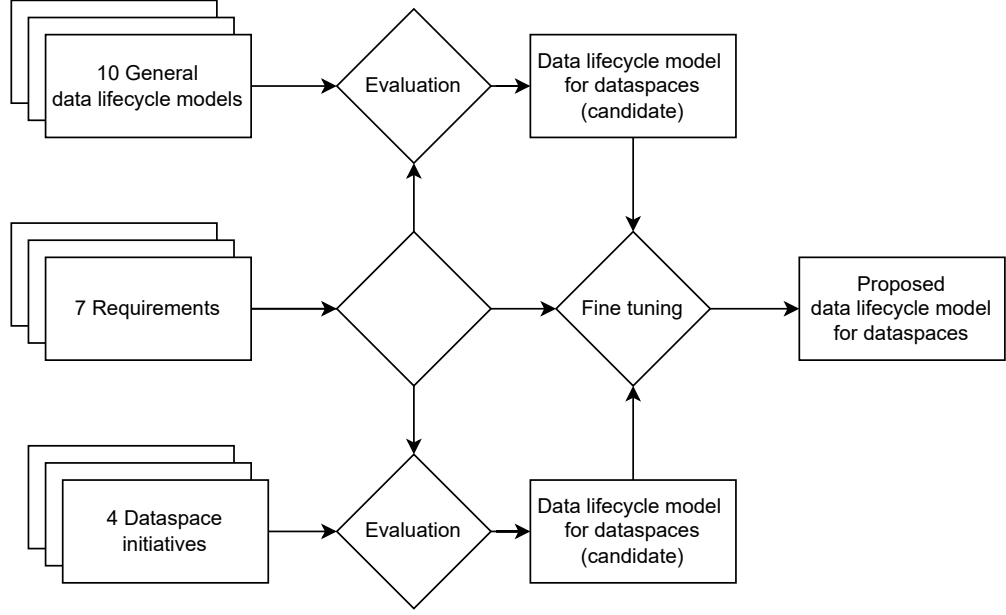


Figure 3.1: Our methodology for developing a **DLCM** for dataspaces.

Dataspace-specific Requirements to the Data Lifecycle

In this section, we propose dataspace-specific requirements for a **DLCM** and thereby lay a foundation for a **DLCM** for dataspaces. We derive these requirements mainly from these contexts: From the background on dataspaces, namely the real-world initiatives presented in [section 2.1.1](#) and the (scientific) foundation of dataspaces discussed in [section 2.1.2](#). And from the application domains of this thesis, which are illustrated in [section 2.3](#). The dataspace-specific requirements for a **DLCM** are listed below. Note that **DLCM-R7** is a non-functional requirements, while the others are functional requirements:

DLCM-R1 Keep data at source: Follow this dataspace principle [HFM06] by leaving heterogeneous data in their source systems until query or access.

DLCM-R2 Separate discovery and access: Support data sovereignty by separating discovery and access (cf. the five essential services for enabling sovereign data exchange¹ in the **IDS** [OJ19; Ott+19; Int23], data sovereignty in Gaia-X [Pla19; GAI22] or the **EOSC** [EOS22]).

¹<https://internationaldataspaces.org/adopt/essential-services/>

- DLCM-R3** Metadata submission: Submit descriptions in the form of metadata to an indexing service, enabling query and understanding for others (cf. the **IDS** Metadata Broker [Bad+20a] and the Gaia-X Catalogue [GAI22]).
- DLCM-R4** Vertical integration: Do not only describe data but also services and actors. See the **IDS** information model [Bad+20b] for descriptions of actors, interactions between them, resources, or data usage restrictions. See the Gaia-X self descriptions [SL22] for concepts such as resources, services, or participants.
- DLCM-R5** Vocabulary standardization: Synchronization of vocabularies is important to foster common understanding and machine-interpretable descriptions (cf. the **IDS** information model, the Gaia-X self description ontology [SL22], or common vocabularies in the **NFDI**² such as the Metadata4Ing ontology [Arn+22]).
- DLCM-R6** Identifiers: Use globally-unique, persistent identifiers for data and metadata (derived as technical prerequisite from the **FAIR** principles [Wil+16] and supported by projects such as the **EOSC** and the **NFDI**).
- DLCM-R7** Active development: This **DLCM** is not only utilized in projects but also benefits from an engaged development community. This contributes to the **DLCM**'s continued relevance and serves as evidence of its applicability in real-world scenarios.

3.2 Analyzing the State of the Art

We in this section investigate state-of-the-art **DLCMs**. This on the one hand includes *general* **DLCMs**, which were designed for rather generic data and metadata management but are still highly relevant for application in the scope of dataspace. On the other hand, we review the dataspace initiatives from section 2.1.1 with a focus on **DLCMs** or hints on such.

3.2.1 General Data Lifecycle Models

This section investigates general **DLCMs** from both academia and projects. Table 3.1 gives an overview of all ten considered **DLCMs**, which we explain in detail below. We in the following present each **DLCM**, create an abstract, unified representation to simplify comparison, and evaluate the **DLCM** based on the dataspace-specific requirements from the previous section. Our results on how each **DLCM** meets our requirements are summarized in table 3.2 at the end of this section.

ANDS Data Sharing Verbs

The goal of the *Data Sharing Verbs* [BT09] of the *Australian National Data Service* is to enable and simplify the reuse of research data (for scientists). To enable this, a structure is proposed from the acquisition of data in a scientific project, to its completion and eventual reuse through Data Sharing Verbs. The focus will be on reusability and data sharing and others will be added as appropriate [BT09]. This **DLCM** is circular and consists of the following eight verbs (stages):

²<https://www.nfdi.de/section-meta/?lang=en>

Table 3.1: Overview of general [DLCMs](#) included in our survey.

Name	Reference
ANDS Data Sharing Verbs	[BT09]
Data Documentation Initiative	[Gro04]
Linked Data Life Cycle	[ALN11]
Capability Maturity Model	[CQ11]
DataOne	[MJ12]
Digital Curation Centre	[Hig08 ; McK+12]
GFBio	[GFB22]
Data Lifecycle for MSE Projects	[Rüe+14]
UK Data Archive	[Eyn13]
Data Science Institute of Columbia University	[Win18]

- **Create:** While the pure generation of data objects is not considered, all processes related to it and also the compilation of already existing data are considered. In addition, the generation of metadata is already recommended in this stages.
- **Store:** Describes the long-term storage of data, which are currently institutional repositories and data stores as well as national data centers.
- **Describe:** The more meaningful the description or the metadata, the better the usability of the data determined in the previous stages. The description enables the storage, preservation, discovery, access and exploitation of the data. This description should include the people, organizations, and activities involved in collecting the data.
- **Identify:** The Identify stages provides a unique identification of the collected data. This stages can be applied either before or after the Describe step. Three identifier types are supported, namely standards or community-assigned, the ones provided by the project itself, or [DOIs](#).
- **Register:** Contains all the ways to convey the descriptions and information to the outside world, and show their existence. E.g. by adding semantic links. The project provides a *collections registry* to automatically share this information in different browse, discovery, query and search environments. Described data collections, for example, can be registered manually by the data collector via a web interface or automatically via [HTTP](#) GET requests.
- **Discover:** Locating information using data discovery services based on the descriptions from the *Describe* stage.
- **Access:** The project offers the possibility to make the data available in open- and closed-access, but recommends the former. The search use is without registration and the access control for closed-access data is enforced by the data store. If data is not available in digital form, contact information will be provided by those responsible for the data.
- **Exploit:** Here the respective data objects are reused. This stages is heavily dependent on quality metadata and meaningful description. It also includes information about the context and the observations made during the collection of the data.

3 Developing a Data Lifecycle Model for Dataspaces

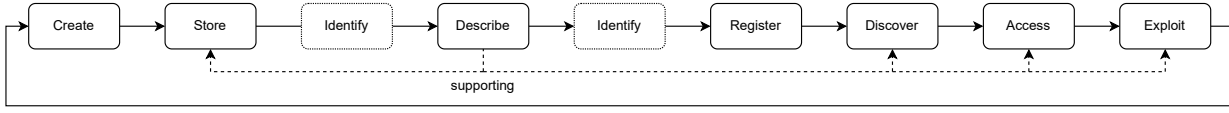


Figure 3.2: Abstract representation of this [DLCM](#) based on the *Data Sharing Verbs* [BT09] by the Australian National Data Service.

The eight stages and their interrelationship is shown in [figure 3.2](#). The requirements [requirements DLCM-R1](#) and [DLCM-R2](#) are met by the phases *Store*, *Discover*, and *Access*. With its *Describe* phase, this proposal meets [requirement DLCM-R3](#) by storing metadata and [requirement DLCM-R4](#) by explicitly including assets like people, organizations, and activities. This project does not directly foster standardization of shared vocabularies ([requirement DLCM-R5](#)), but fully matches [requirement DLCM-R6](#) with multiple identification methods in its *Identify* phase. [Requirement DLCM-R7](#) is not met, because it lacks evidence for an ongoing active development.

Data Documentation Initiative Combined Life Cycle Model

The combined lifecycle model version 3.0 developed by the Data Documentation Initiative focuses on the acquisition, creation, and reuse of metadata in the different stages of the research data lifecycle [Zen12; Gro04]. It is shown in [figure 3.3](#) and is a linear [DLCM](#) with two branches from data processing to data archiving and data distribution. The study concept and data collection stages are traversed only once and are not part of the lifecycle loop.

This [DLCM](#) involves the following stages:

- **Study Concept:** Planning and design of the research questions, methodology and data processing.
- **Data Collection:** Data collection including primary and secondary data sources as well as their differentiation.
- **Data Processing:** Input data is processed and analyzed to produce output data in machine or human readable form.
- **Data Archiving:** Archive the entire research data, including metadata, to enable long-term access to it.
- **Data Distribution:** Distributing the data via data archives, libraries or directly to other users.
- **Data Discovery:** Publish the data via books, publications, or online services.
- **Data Analysis:** Evaluation of the data and the use of the data of others in the context of the original conceptualization.
- **Repurposing:** Reuse of the data for a secondary use.

This [DLCM](#) only meets three out of seven requirements. It does not instruct keeping data at source ([requirement DLCM-R1](#)) or separating discovery and access ([requirement DLCM-R2](#)). The *Data Documentation Initiative's* [DLCM](#) supports basic metadata submission to meet [requirement DLCM-R3](#), but does not include other assets than data sets in it (cf.

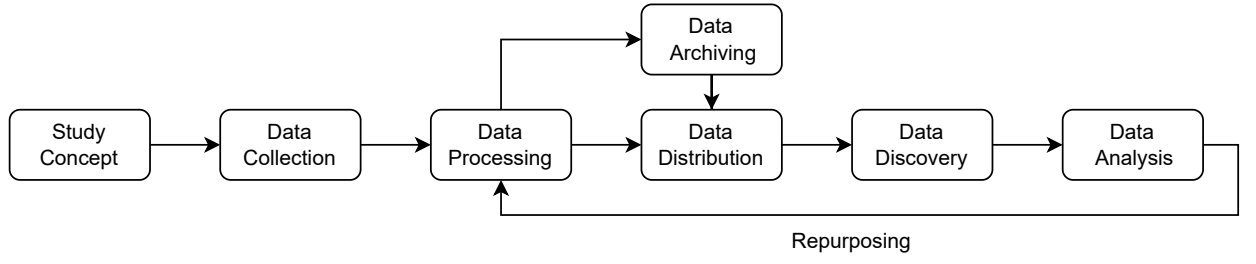


Figure 3.3: Abstract representation of this [DLCM](#) based on the *Data Documentation Initiative's combined lifecycle model* [Zen12; Gro04].

[requirement DLCM-R4](#)). It meets [requirement DLCM-R5](#) by encouraging users to apply standardized vocabularies, but does not advise how to properly identify things ([requirement DLCM-R6](#)). We verify active development ([requirement DLCM-R7](#)) through its up-to-date website³, which lists ongoing research, events, and workshops.

Linked Data Lifecycle

Auer et al. in [ALN11] proposed a *Linked Data Lifecycle*, which was later applied for a collaborative integration, publishing, and analysis of distributed scholarly metadata in the dissertation of Vahdati [Sah19]. The authors mention that different stages of the Linked Data lifecycle are interdependent, and executive tasks (e.g., schema mapping) often cover more than one stage. The stages of this [DLCM](#) (see [figure 3.4](#)) are defined as follows:

- **Extraction:** Map information to the [RDF](#) data model
- **Storage & Querying:** Implement mechanisms to store, index, and query [RDF](#) data
- **Authoring:** Users can create new information or correct/extend existing ones
- **Linking:** Link information about the same or related entities
- **Enrichment:** Enrich data with higher level structures such as structure and schema information
- **Quality Analysis:** Assess the quality of published data
- **Evolution & Repair:** Repair problems to support the evolution of Linked Data
- **Search, Browsing & Exploration:** Enable users to browse, search, and explore information in a fast and user-friendly manner

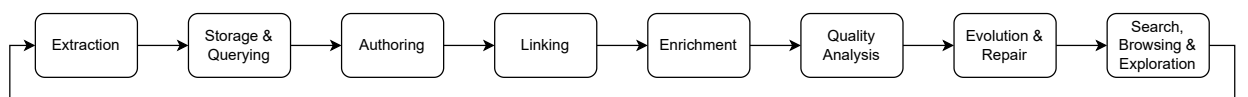


Figure 3.4: Abstract representation of this [DLCM](#) based on the *Linked Data lifecycle* [ALN11].

The *Linked Data Lifecycle* does not directly address storage of data itself, without con-

³<https://ddialliance.org/>

verting it to the [RDF](#) data model, and therefore does not meet [requirements DLCM-R1](#) and [DLCM-R2](#). It however meets [requirements DLCM-R3](#) to [DLCM-R6](#) by instructing metadata submission and supporting vertical integration in the *Extraction* and *Enrichment* steps, by reusing common vocabularies, and by linking to proper identifiers following the Linked Data principles. A website⁴ documents both a continuous research agenda and on-going events to meet [requirement DLCM-R7](#).

Capability Maturity Model for Scientific Data Management

[CQ11] proposes a *Capability Maturity Model* for scientific data management based on a review of the literature, see [figure 3.5](#). They discuss the importance of effective data management in scientific research, present their [DLCM](#) as a tool for assessing and improving data management practices in research organizations and introduce a life cycle consisting of four key areas. The authors also provide a detailed description of their [DLCM](#), including its five maturity levels of data management practices, from the *zeroth* level, where the recommended procedure is not followed, to an ad hoc realization in the *Initial* level, to the *Optimizing* level, where all stages are implemented.

- **Data acquisition, processing and quality assurance:** The reliable acquisition and description of scientific data in a manner that facilitates preservation and reuse. This stages includes but is not limited to data processing and preparation for storage, analysis, and distribution, and data quality assurance (e.g., data validation and verification).
- **Data description and representation:** Create high-quality metadata for functions such as data discovery, preservation, and provenance. This requires developing and applying metadata specifications and schemas, contextualizing and describing data, documenting software and sensors, creating descriptive and semantic metadata for datasets, designing mechanisms for linking datasets to publications, ensuring interoperability with data and metadata standards, and ensuring compliance with standards.
- **Data dissemination:** Create interfaces that enable users to retrieve and interact with data. It emphasizes the importance of identifying and managing data products, encouraging data sharing, distributing data, and granting access through service models.
- **Repository services/preservation:** Preserve collected data for long-term use, which includes storing, backing up, and securing data, managing archival schedules, correcting data, performing data migrations, building a digital preservation network, validating data archives, and packaging and delivering data archives.

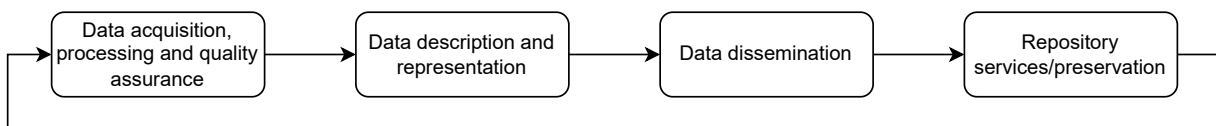


Figure 3.5: Abstract representation of this [DLCM](#) based on the *capability maturity model for scientific data management* [CQ11].

⁴<https://aksw.org/Projects/LOD2.html>

As the previous *Linked Data Lifecycle* [DLCM](#), the *Capability Maturity Model* does not sufficiently address data storage, discovery, and access ([requirements DLCM-R1](#) and [DLCM-R2](#)). Its *Data description and representation* phase however meets both [requirements DLCM-R3](#) and [DLCM-R4](#), while [requirement DLCM-R5](#) is met by standardized vocabularies in multiple phases and the need for proper identifier use (cf. [requirement DLCM-R6](#)) is covered by its *data dissemination* and *repository services/preservation* stages. There is no sufficient evidence for active development, leaving [requirement DLCM-R7](#) unmet.

DataONE

The *DataONE* project [[Mic+12](#)] claims to be a “research platform that is specifically designed to support the full data lifecycle and to enable new, data-intensive science” [[MJ12](#)]. They introduce a [DLCM](#) and differentiate between two types of projects: First, data collection projects typically proceed through plan, collect, assure, describe, and preserve, and then go directly to analysis. Second, metadata analysis can initially start at discover and proceed to integrate and analyze. The authors however claim that the following steps of their [DLCM](#) (see [figure 3.6](#)) neither are necessarily exclusive nor need to be sequential:

- **Plan:** Comprehensively plan data management, use supportive tools when possible, recognize that data management plans as living documents that change in response to project needs and new technologies.
- **Collect:** Might include manual recording as well as automated data collection. Use commonly agreed approaches and tools for capturing both data and metadata. Leverage technologies for robust calculations, quality assurance, and visualization.
- **Assure:** Prior to data collection, quality assurance can validate data formats, standards, codes, measurement units, and metadata. Applies to all levels ranging from individual data points to high-level descriptions.
- **Describe:** For data, add metadata to describe who created, collected, and maintains it, its content and format, when it was collected, where it was collected and stored, how it was generated, processed, secured, and analyzed, and why it was generated. Apply metadata standards and tools to provide consistency.
- **Preserve:** Deposit data and metadata with replication and curation over time. Assign identifiers such as [DOIs](#) to data products.
- **Discover:** Recognize that data is a valuable product that needs to be made available for broader use. Support sophisticated, user-friendly search tools using faceted search techniques. Controlled vocabularies and community thesauri are useful for discovery, while data models and ontologies can automatically expand searches for better results.
- **Integrate:** Encode the semantics of data and then reason across these semantics to semi-automate the process of data integration. Bottom-up semantic models explic-

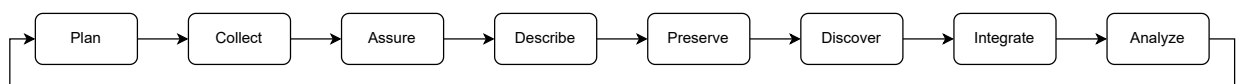


Figure 3.6: Abstract representation of this [DLCM](#) based on the *DataOne project* [[MJ12](#)].

3 Developing a Data Lifecycle Model for Dataspaces

itly capture semantics of data to enable others to understand it. Use standardized approaches for integration, transformation and homogenization.

- **Analyze:** Use both general-purpose and specialty tools as well as simulation and analytical models to derive conclusions about the data. Consider metadata such as provenance or the relationship between data for analysis.

The *DataONE DLCM* supports a distributed data storage approach and thereby fulfills requirement *DLCM-R1*, but does not encourage separated treatment of discovery and access (cf. requirement *DLCM-R2*). It not only supports metadata submission (requirement *DLCM-R3*) but also assuring its quality in an automated manner, which however does not apply to multiple assets or layers (cf. requirement *DLCM-R4*). This *DLCM* meets requirement *DLCM-R5* by explicitly encouraging reuse of metadata standards and tools for consistency, and in the *preserve* phase clearly includes proper identifiers (cf. requirement *DLCM-R6*) throughout the lifecycle. Both a project website⁵ and a GitHub repository⁶ demonstrate ongoing development to meet requirement *DLCM-R7*.

Digital Curation Centre - Curation Lifecycle Model

The Digital Curation Centre's *Curation Lifecycle Model* [Hig08] outlines the various stages involved in curating and preserving digital data. These stages include planning, acquisition, appraisal, archiving, and deployment of data. This *DLCM* assists in ensuring that all necessary steps have been taken and that data has been adequately documented and archived to maintain its integrity and availability over time.

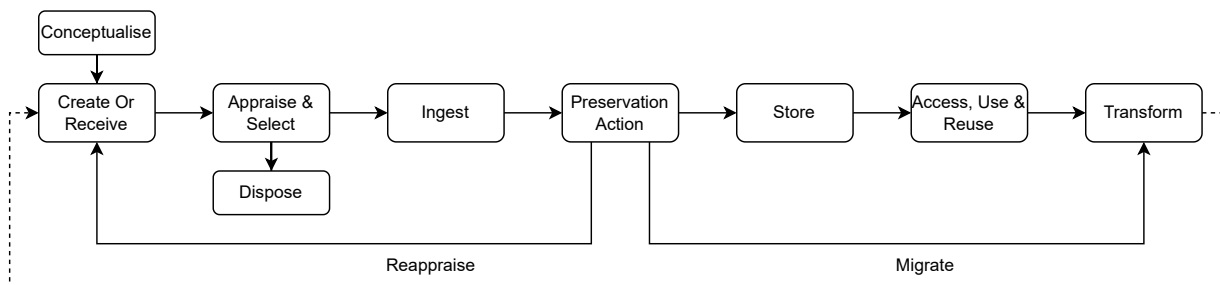


Figure 3.7: Abstract representation of this *DLCM* based on the *Digital Curation Centre's Curation Lifecycle Model* [Hig08].

- **Conceptualize:** Conceptualize and plan for data creation, including collection methods and archiving options.
- **Create or Receive:** Data should be created with administrative, descriptive, structural, and technical metadata, along with preservation metadata if necessary. Data can be received from creators or other archives and should be assigned appropriate metadata based on documented collecting policies.
- **Appraise & Select:** Evaluate and select data for long-term curation and preservation. Comply with documented guidelines, policies, or regulatory requirements.

⁵<https://www.dataone.org/>

⁶<https://github.com/DataONEorg>

- **Dispose:** Data not selected for long-term preservation should be disposed of in accordance with documented policies and legal requirements. For example, by transferring the data to another archive or repository, or by secure destruction.
- **Ingest:** Transfer data to a designated custodian such as an archive, repository or data center. Ensure that the transfer follows established guidelines, policies and legal requirements.
- **Preservation Action:** To ensure that data remains authoritative, reliable, and usable, it is important to take steps to preserve it for the long term. These include cleaning and validating data, assigning metadata and presentation information, and ensuring acceptable data structures or file formats.
- **Store:** Store the data in a secure manner that complies with relevant standards.
- **Access, Use & Reuse:** Make sure that data is easily accessible to designated users and re-users, possibly through public release. In addition, implement strong access controls and methods of authentication.
- **Transform:** Create new data from the original, for example, by migrating it to a different format or creating a subset through selection or query to derive new results, potentially for publication.
- **Reappraise:** For further evaluation and selection, retrieve data that did not pass validation procedures.
- **Migrate:** To adapt to the storage environment or to protect against hardware or software obsolescence, data should be converted to another format.

The Research360 project “*aimed to develop human and technical infrastructure to facilitate management of [...] research data*”⁷. This project ran between 2011 and 2013, and proposed a research lifecycle based on the above-mentioned **DLCM**, which incorporates data as output of research and intends to make data citable, linked, and reusable [McK+12]. Its stages involve plan and design, collect and capture, interpret and analyze, manage and preserve, release and publish, and discover and reuse. Due to the general similarity to the **DLCM** by the Digital Curation Centre, we jointly inspect these two as one (i.e., the one depicted in figure 3.7).

The *Curation Lifecycle Model* meets requirements **DLCM-R1** and **DLCM-R2** in its *ingest* and *store* stages. In its phases *create or receive*, it assigns valuable metadata to data sets and thereby fulfills requirement **DLCM-R3**, but does not support describing services or actors (cf. requirement **DLCM-R4**). The metadata in this **DLCM** builds on documented collecting policies by the Digital Curation Centre, which incorporates sufficient vocabulary standardization to meet requirement **DLCM-R5**. The initial **DLCM** does not explicitly instruct the use of proper identifiers (cf. requirement **DLCM-R6**). Its active development meets requirement **DLCM-R7** with an up-to-date website⁸ that, among other research and community activities, also encourages the adoption of the **FAIR** principles, which lay a strong foundation for identification methodologies.

⁷<http://www.ukoln.ac.uk/projects/research360/>

⁸<https://www.dcc.ac.uk/events>

GFBio

GFBio represents the *German Federation for Biological Data*, which deals with the management and standardization of biological and environmental data from creation up to reuse, in particular with a focus on the entire data lifecycle [GFB22]. It aims at a sustainable long term provision of its core services and is illustrated in figure 3.8. These services include tools, standards and platforms for acquisition, quality control and standardization, long-term storage facilities in data centers for archiving and publication, a portal for access, analysis and visualization services for interpretation and analysis, and educational materials for research data management [GFB22].

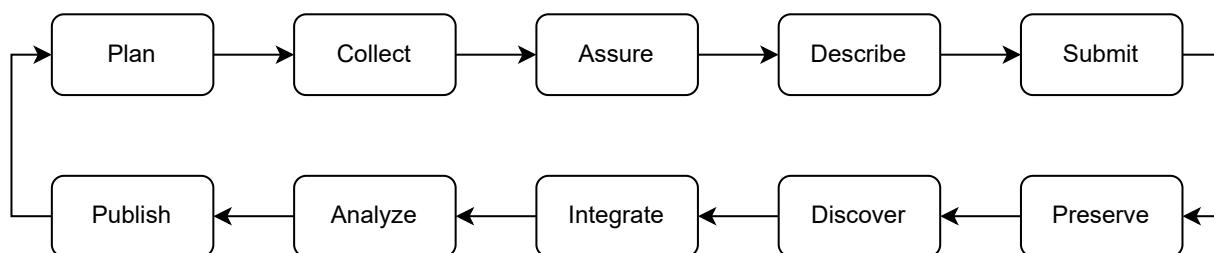


Figure 3.8: Abstract representation of this [DLCM](#) based on the *GFBio* [GFB22].

- **Plan:** Create a plan for how the data can be managed, shared, and reused. Consideration should also be given to how and what data will be collected, how it will be preserved, and whether data already exists on this topic which can be reused.
- **Collect:** A strategy should be developed and, in addition, a record kept of how and what data is collected. The long-term compatibility of the collected and generated data with established standards needs to be considered. Open software and non-proprietary formats have to be used and backup plans on distributed systems should be considered.
- **Assure:** It must be ensured that the collected data is complete, its quality is high and it does not contain any errors. Versioning the data is particularly helpful for this. This state should be maintained over the entire life cycle.
- **Describe:** The data should be provided with meaningful information to facilitate retrieval or reuse. This metadata should be created in a standardized format and answer the following questions: “WHY were the data generated? WHO created the data? WHERE and WHEN were the data collected? WHAT is the content of the data? HOW were the data assessed?”
- **Submit:** The process of submitting data to a shared research area. Select useful data, check for spyware, ensure accurate metadata and use common keywords. Prepare the data to meet archival standards. Select the data availability status, upload and obtain a [PID](#).
- **Preserve:** Preservation involves measures to ensure the accessibility, authenticity and longevity of data. Digital objects are fragile and susceptible to *data rot*, and preservation ensures that datasets are stored, discovered, accessed, and reused in the best shape possible, using methods such as migration, emulation, digital archaeology, and technology preservation.
- **Discover:** To find useful data for research, look for accessible data with metadata that

has been submitted to a data infrastructure and ideally has a [PID](#).

- **Integrate:** Integrate datasets from different sources to merge multiple datasets into a larger dataset, either manually or automatically, using a common syntax and terminology. When reusing data from other authors, credit them using robust data citation practices, preferably with a [PID](#).
- **Analyze:** To ensure reproducibility, document analysis and visualization workflows in detail, use appropriate open source software, and normalize or transform raw data into meaningful values. Discover patterns or outliers through data mining and plots, and choose the statistical model that best fits the data for analysis.
- **Publish:** To promote data publication and citation, give preference to journals and data centers. Provide [PIDs](#) for data sets, indicate in publications where data sets can be discovered, accessed, and visualized, and update archived data sets as newer versions become available.

The *GFBio* [DLCM](#) meets [requirements DLCM-R1](#) and [DLCM-R2](#) in many phases like *collect*, *submit*, *integrate*, and *publish*. The *describe* stage adds meaningful information (cf. [requirement DLCM-R3](#)), but only covers conventional data sets, not covering data services and therefore not meeting [requirement DLCM-R4](#). This [DLCM](#) strongly encourages and supports the use of standardized vocabularies and [PIDs](#), which lead to a superior conformation with [requirements DLCM-R5](#) and [DLCM-R6](#). A project website with somewhat recent events⁹, a GitHub repository¹⁰, a wiki¹¹, training opportunities¹², and an integration into the [NFDI](#) consortium on *biodiversity*¹³ prove active development and therefore fulfills [requirement DLCM-R7](#).

Data Lifecycle for MSE Projects

Rüegg et al. presented a [DLCM](#) in the scope of macrosystems ecology, or *MSE* for short, research [[Rüe+14](#)]. This [DLCM](#) differentiates between traditional projects, which range from planning to analysis and where data often ends up on a desktop computer, and projects relying on existing data, which include extra steps such as additional data discovery and integration. The two final steps *data documentation with metadata* and *data archiving* in a publicly accessible repository complete this [DLCM](#). The idea of this [DLCM](#) was later used in the NASA project *the Arctic Boreal Vulnerability Experiment*, in short ABoVE, which proposed an adopted version as part of their implementation plan¹⁴.

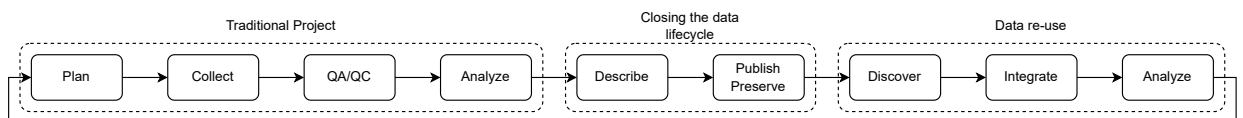


Figure 3.9: Abstract representation of this [DLCM](#) based on the *data lifecycle for macrosystems ecology projects* [[Rüe+14](#)].

⁹<https://www.gfbio.org/events/>

¹⁰<https://github.com/gfbio>

¹¹https://gfbio.biowikifarm.net/wiki/Main_Page

¹²<https://www.gfbio.org/training/>

¹³<https://www.nfdi4biodiversity.org/>

¹⁴https://above.nasa.gov/implementation_plan/data_cycle.html

3 Developing a Data Lifecycle Model for Dataspaces

The [DLCM](#) proposed by Rüegg et al. is illustrated in [figure 3.9](#) and consists of:

- **Plan:** Integrate information management early on, especially for complex data, when many people are involved, or when data should be collected over an extended time period.
- **Collect:** Use scientifically-proven methods for data collection, which includes common data and metadata models.
- **QA/QC:** Transparently document quality control and data aggregation approaches; use standardized vocabularies for labeling.
- **Analyze:** Generate insights and integrate them from local observations to models and back.
- **Describe:** Follow community-developed and well-documented standards for converting and modeling data. Provide controlled vocabularies to ensure clarity and support automation. Document provenance.
- **Publish & Preserve:** Provide preprocessed data along with metadata, and enable access and subsequent use.
- **Discover:** Query large-scale data from different sources and with different resolution and accuracy. Use common data models and metadata as well as interpolation/aggregation for search.
- **Integrate:** Integrate data into a common data model and format. Provide workflow components for data processing to simplify subsequent analysis.

While this [DLCM](#), which was designed for macrosystems ecology projects, does not envisage a distributed data storage and therefore does not meet [requirement DLCM-R1](#), it still fulfills [requirement DLCM-R2](#) by separating discovery and access in its stages *publish & preserve* and *discover*. Basic metadata submission ([requirement DLCM-R3](#)) is supported, but not considered for any deeper vertical integration ([requirement DLCM-R4](#)). Its phases *collect*, *discover*, and *integrate* particularly advocate the application of common (meta)data models and formats, which fulfills [requirement DLCM-R5](#). [Requirements DLCM-R6](#) and [DLCM-R7](#) are not met, because there is neither a sophisticated identifier system incorporated nor any evidence for ongoing development or use of this [DLCM](#).

UK Data Archive - Research Data Lifecycle

The *UK Data Archive* published their *Research Data Lifecycle* in 2013, which has later become part of the *UK Data Service*'s research data management strategy [[UK 18](#)]. This [DLCM](#) describes stages for creating research data, including planning research, collecting data, processing and analyzing data, publishing and sharing data, preserving data, and reusing data. Their [DLCM](#) is illustrated as a unidirectional cycle that begins with data creation. The [DLCM](#) addresses issues such as data sharing, data licensing, and migration.

The stages of the UK Data Archive [DLCM](#) are illustrated in [figure 3.10](#) and are defined as follows:

- **Create:** Design research, plan data management (e.g., formats and storage), plan consent for sharing, locate existing data, collect data, capture and create metadata

- **Process:** Enter/digitize/transcribe/translate data, check/validate/clean data, anonymize data, describe data, manage and store data
- **Analyze:** Interpret and derive data, produce research outputs, author publications, prepare data for preservation
- **Preserve:** Migrate data to best format and suitable medium, back-up and store data, create metadata and documentation, archive data
- **Give access:** Distribute and share data, control access, establish copyright, promote data
- **Reuse:** Follow-up and new research, undertake research reviews, scrutinize findings, teach and learn

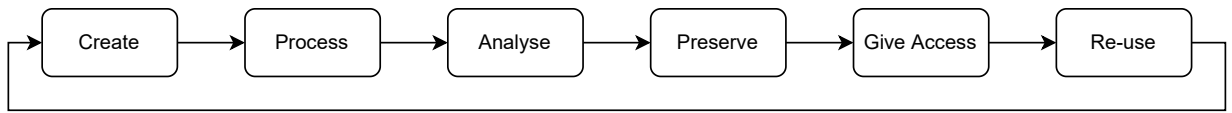


Figure 3.10: Abstract representation of this [DLCM](#) based on the *UK Data Archive* [Eyn13].

This [DLCM](#) supports the dataspace principle of *keep data at source* and thereby meets [requirement DLCM-R1](#) through its stages *preserve* and *give access*. It does not separate discovery and access by design ([requirement DLCM-R2](#)), but includes additional information in the form of metadata in its stages *process* and *preserve*. Due to the facts that metadata is only added to data sets and that no standard vocabularies or identifiers are emphasized, [requirements DLCM-R4](#) to [DLCM-R6](#) are not met by this [DLCM](#), respectively. An active development however is proven by the *UK Data Archive* website driven by the University of Essex¹⁵, which links to an up-to-date data catalog¹⁶ and ongoing training events¹⁷.

Columbia University DLCM

Wing in [Win18] proposed a [DLCM](#) consisting of several stages and ending with the human. According to Wing, this human could be a scientist, a policymaker, or in medicine, finance, law, or business. This [DLCM](#) considers a more general approach to data sharing and data science without exploring it in more depth. The stages this [DLCM](#) are shown in [figure 3.11](#) and involve:

- **Generation:** Every day, vast amounts of data are generated by more and more sensors to monitor the health and condition of buildings, Internet of Things applications, etc. Walmart, for example, collects 2.5 petabytes of unstructured customer data per hour.
- **Collection:** Only a portion of the data generated is actually collected. This may be due to large amounts of data, time constraints or personal decisions that the data is not needed.
- **Processing:** Entire process of data processing from formatting through compression to encryption.

¹⁵<https://www.data-archive.ac.uk/>

¹⁶<https://beta.ukdataservice.ac.uk/datacatalogue/studies/>

¹⁷<https://ukdataservice.ac.uk/training-events/>

3 Developing a Data Lifecycle Model for Dataspaces

- **Storage:** Any form of storage especially concerning long-term capabilities.
- **Management:** Storing data in such a way that it can be retrieved as easily and universally as possible. In order for structured and unstructured as well as different data types to be discoverable, particularly meaningful metadata is important.
- **Analysis:** The computational and statistical procedures used to analyze data to gain knowledge or insight about the data.
- **Visualization:** A visual presentation of the analyzed data that is easy to understand for humans.
- **Interpretation:** Explain and classify the visualized images and the results of the data analysis.
- **Human:** The person who discovers the data and uses it further.



Figure 3.11: Abstract representation of this [DLCM](#) based on the *Data Science Institute of Columbia University* [[Win18](#)].

This [DLCM](#) is rather a high-level proposal than an in-depth solution. It covers the concepts of keeping data at source, meeting [requirement DLCM-R1](#), and meaningful metadata for enabling discovery, thus addressing [requirement DLCM-R3](#). The rest however, namely [requirements DLCM-R2](#) and [DLCM-R4](#) to [DLCM-R7](#) are not addressed by this approach.

Table 3.2: Evaluation of which [DLCMs](#) meet which dataspace requirements (R1 is short for [requirement DLCM-R1](#)).

General Data Lifecycle Model	R1	R2	R3	R4	R5	R6	R7
ANDS Data Sharing Verbs	x	x	x	x		x	
Data Documentation Initiative			x		x		x
Linked Data Life Cycle			x	x	x	x	x
Capability Maturity Model			x	x	x	x	
DataONE	x		x		x	x	x
Digital Curation Centre	x	x	x		x		x
GFBio	x	x	x		x	x	x
Data Lifecycle for MSE Projects		x	x		x		
UK Data Archive		x	x				x
Data Science Institute of Columbia University	x		x				

Conclusion

We summarize the above analysis of ten general **DLCMs** for their applicability in the scope of dataspace. Our investigation takes into account the seven dataspace-specific requirements defined in [section 3.1](#) and [table 3.2](#) illustrates our results. It depicts which **DLCM** meets which dataspace requirement and highlights *GFBio*, because it meets the most requirements. The *GFBio* **DLCM** therefore is a promising candidate and basis for proposing a **DLCM** for dataspace in [section 3.3](#).

3.2.2 Data Lifecycle Models in Dataspace Initiatives

This section investigates how the dataspace initiatives presented in [section 2.1.1](#) and [table 2.1](#) meet our requirements for a **DLCM** for dataspace proposed in [section 3.1](#). In the following, we analyze the dataspace initiatives and either present their respective **DLCMs** or, if none is explicit, derive information from the individual components of the dataspace. Our results are summarized in [table 3.3](#) at the end of this section. The rating ranges from *not addressed* (-) over *proposed* (empty cell) to *implemented* (+).

IDS

The **IDSA** explicitly proposes a **DLCM** for objects in the **IDS** in their reference architecture model [[Int23](#)], which should be applied for both data and metadata (e.g., in the form of self descriptions):

1. **Create**: Create an object, e.g. software by programming or data from reading a sensor.
2. **Own**: Own an object or hold the corresponding license or right according to local rules and regulations.
3. **Certify/verify**: E.g. certify software according to the **IDS** certification scheme or verify authenticity of data.
4. **Publish**: Share metadata on objects such as data, apps, or services.
5. **Provide**: Technically provide the object.
6. **Consume**: Technically receive the object.
7. **Use**: Make use of an object in a business model that does not consist of an intermediary function.
8. **Delete**: Delete, eliminate, or turn an object off.

This **DLCM** along with other fundamental architectures and concepts from the **IDS** reference architecture model [[Int23](#)] meet our requirements as follows. **Requirements DLCM-R1** and **DLCM-R2** are met by the **IDS**' separation of discovery and access into the to above-mentioned phases *publish* and *provide*, which is implemented in their components like the *Connector*. This **DLCM** fulfills **requirements DLCM-R3** to **DLCM-R5** with its *Connectors* and the **IDS** information model [[Bad+20b](#)] (cf. [section 2.1.1](#)), because it addresses data and metadata with self descriptions, which are used to describe not only data sets but also participants, apps, or services. However, the **IDS** information model redefines some concepts even though they have already been standardized elsewhere, or copies them into the **IDS**

namespace. Recent activities correct this approach by referencing external ontologies and URIs directly¹⁸, but a full implementation is still pending. The *IDS Identity Provider*¹⁹ provides trusted identities but does not include PIDs in its design or implementation. Identifiers for connectors or objects served by them are by design *unique* but neither persistent nor proper PIDs either, which is insufficient to fulfill requirement *DLCM-R6*. Active development (cf. requirement *DLCM-R7*) is proven by the up-to-date *IDS* website [Int16a], which lists ongoing events²⁰, and the demonstration of *IDS*-based dataspaces in the continuously updated *dataspace radar* [Mer23]. The Mobility Data Space (see section 2.3), an example of a domain-specific application of the *IDS*, applies the concepts and implementations of the *IDS* to the mobility sector and fulfills the same requirements as the *IDS*.

Gaia-X

Gaia-X in their *Architecture Document* [GAI22] do not define an explicit *DLCM* but only one for self descriptions [SL22], which covers the following states²¹:

- **Expired:** If the expiration datetime is older than current datetime or the certificate containing the key used to sign the claim has expired.
- **Revoked:** If the key used to sign the array is revoked.
- **Deprecated:** If another verifiable credential with the same identifier and the same signature issuer has a newer issuance datetime.
- **User-Defined:** By using `credentialStatus` from the *W3C Verifiable Credentials Data Model* [Spo+19; GAI22].
- **Active:** Only if none of the above. Does not imply compliance.

This *DLCM* along with other fundamental architectures and concepts from the Gaia-X *Architecture Document* meet our requirements as follows. Requirements *DLCM-R1* and *DLCM-R2* are met in Gaia-X by separating discovery and access within their connector design and implementation. Self descriptions in Gaia-X provide a sophisticated solution to semantically describe many objects like data offerings, participants and services, which meets requirements *DLCM-R3* and *DLCM-R4*. The use of standardized vocabularies (requirement *DLCM-R5*) is intended and encouraged, but not yet fully implemented. Identifiers in Gaia-X ensure uniqueness, but a complete integration with PID-based global systems is still pending, leaving requirement *DLCM-R6* partially met. The Gaia-X website lists a wide range of ongoing international events and Gaia-X hubs as local ambassadors²², and deep conceptual and technical demonstrations in their lighthouse projects²³. Active code repositories on GitLab²⁴ and the *Eclipse Foundation*²⁵ additionally prove the active development demanded by requirement *DLCM-R7*.

¹⁸<https://github.com/International-Data-Spaces-Association/InformationModel/pull/504>

¹⁹<https://github.com/International-Data-Spaces-Association/omejdn-daps>

²⁰<https://internationaldataspaces.org/calendar/>

²¹<https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/self-description/#self-description-lifecycle>

²²<https://gaia-x.eu/events/> and <https://gaia-x.eu/who-we-are/hubs/>

²³<https://gaia-x.eu/who-we-are/lighthouse-projects/>

²⁴<https://gitlab.com/gaia-x>

²⁵<https://projects.eclipse.org/projects/technology.edc>

NFDI and EOSC

The **NFDI** does not provide an explicit **DLCM** yet. The *NFDI4DataScience* consortium aims to “support all steps of the [...] research data lifecycle in Data Science and Artificial Intelligence”²⁶, but has not published such a **DLCM** yet. The *NFDI4Cat* consortium in their 2021 publication [Hor+21] reports on their initial outcomes, which include an integration of their repository architecture in a **DLCM**. The mentioned **DLCM** has the phases *collect & create, process, analyze, preserve, access, and reuse*.

With respect to this model and the general other components and consortia of **NFDI**, we evaluate our requirements as follows. **NFDI**’s shared and networked information infrastructures [DSE21] follow the dataspace principles *keep data at source* and at the same time support metadata submission to a semantic storage, which thereby fulfills **requirements DLCM-R1** and **DLCM-R3**. However a separation of discovery and access is not elaborated on sufficiently (**requirement DLCM-R2**) and **NFDI** is currently limited to describe data without vertical integration (cf. **requirement DLCM-R4**). Vocabulary standardization (**requirement DLCM-R5**) is met by the analysis, usage, and development of vocabularies and ontologies for different domains, such as the *Metadata4Ing* ontology [Arn+22]. **NFDI** manages a registered identifier namespace²⁷, which is until now only used to describe groups, roles, and permissions. That is insufficient to fulfill **requirement DLCM-R6**, but a recently confirmed funding of a new **NFDI basic service for PIDs and terminologies**²⁸ will eventually tackle **requirement DLCM-R6** in the near future. A range of cross-consortia activities as well as the integration into various application areas and other initiatives such as the *FAIR Data Spaces*²⁹ or the **EOSC**³⁰ [GML21] prove active development and meet **requirement DLCM-R7**. Due to this conceptual and organizational integration of **NFDI** and the **EOSC**, this evaluation of requirements holds for both altogether. The **EOSC** in their declaration³¹ identify the fact that there is a data lifecycle, but do not (yet) elaborate on such.

Conclusion

The results of our evaluation of **DLCMs** in dataspace initiatives is depicted in [table 3.3](#). The rating in this table ranges from *not addressed* (-) over *proposed* (empty cell) to *implemented* (+). If dataspace initiatives do not explicitly propose a **DLCM**, we make partial reference to their concepts. The table lists the **IDSA** as representative for the **IDS** and **IDS**-based dataspace such as the *Mobility Data Space*, *Gaia-X*, and the two initiatives **NFDI** and **EOSC** for research data management. The evaluation criteria are the requirements specified in [section 3.1](#), which are abbreviated as *R1* to *R7* for better readability. Please note that this evaluation is a snapshot of the current state of the art, which might change in the near or far future. New dataspace initiatives might arise, existing ones might change, merge, evolve or even be disregarded.

The **IDS** is the only dataspace initiative that has published an explicit **DLCM**. *Gaia-X* at least proposes one for its metadata elements called *self descriptions*, but not for data. In such cases, we instead refer to individual components to form an overall picture for our evaluation.

²⁶<https://www.nfdi.de/consortia-nfdi4datascience/?lang=en>

²⁷<https://www.nfdi.de/persistent-identifiers/?lang=en>

²⁸<https://www.nfdi.de/funding-of-new-nfdi-basic-services-for-persistent-identifiers-and-terminologies/>

²⁹<https://www.nfdi.de/fair-data-spaces/?lang=en>

³⁰<https://eosc.eu/members/nationale-forschungsdateninfrastruktur-nfdi-e-v>

³¹https://eosc-portal.eu/sites/default/files/eosc_declaration.pdf

Table 3.3: Evaluation of which dataspace initiatives meet which **DLCM** requirements (R1 is short for **requirement DLCM-R1**).

Dataspace Initiative	R1	R2	R3	R4	R5	R6	R7
IDS	x		x	x			x
Gaia-X	x		x				x
NFDI and EOSC	x		x		x		x

Our measure ranges from *not addressed* (-) over conceptually *proposed* (empty cell) to addressed or *implemented* (+). While both the **IDS** and Gaia-X meet most requirements, they do not properly implement standardization of vocabularies (cf. **requirement DLCM-R5**) and **PIDs** (cf. **requirement DLCM-R6**). The **NFDI** on the other hand addresses these two: They provide vocabulary standardization through consortia, services, and sophisticated ontologies. And proper identifier management through a to-be-established basic service, which will eventually update the empty cells to a + in column R6 of **table 3.3**. **Requirements DLCM-R2** and **DLCM-R4** however are currently not fulfilled by the **NFDI**.

In summary, the investigated dataspace initiatives meet most requirements either through an explicit **DLCM** (for the **IDS**) or implicit building blocks (for Gaia-X, the **NFDI** and the **EOSC**). Their conceptual proposals appear promising, but their implementation, demonstration, and consolidation leave room for improvement in meeting the requirements. This may hopefully change in the future with their continued active development. For our research on a **DLCM** for dataspaces, the proposals of these initiatives are certainly promising.

3.3 Towards a Data Lifecycle Model for Dataspaces

This section addresses **RQ 1** and proposes a representative **DLCM** for dataspaces, which is based on established general **DLCMs** (cf. **section 3.2.1**) and those from dataspace initiatives (cf. **section 3.2.2**). We base on *GFBio*, because out of the investigated general **DLCMs** it meets the most requirements, and evolve it with the help of findings from dataspace initiatives. The result is depicted in **figure 3.12**. The ten stages remain the same but extended descriptions pay attention to the dataspace-specific requirements, whose impacts on the model stages are highlighted with gray rectangles. Note that stages such as *describe* are affected by more requirements than others. The rest of this thesis refers to this **DLCM** as the **DLCM for dataspaces**.

In the following, we present a series of modifications to our proposed **DLCM** for dataspaces. These modifications are designed to ensure comprehensive fulfillment of all requirements. They include the incorporation of additional guidelines into each phase of the data lifecycle model, which addresses the complexities of handling data within dataspaces. The additions to the phases of our proposed **DLCM** for dataspaces are:

Addition to **DLCM-R1**: While the *keep data at source* paradigm also affects stages like *collect*, it mainly affects the stages *submit* and *preserve*. The data itself therefore is not to be loaded into a dataspace, but kept in adjacent systems.

Addition to **DLCM-R2**: The requirement *separate discovery and access* mainly tackles the stages *discover* and *integrate*. Queries are evaluated (centrally) on metadata and (decentrally) on the distributed systems holding the data. After finding a resource, ac-

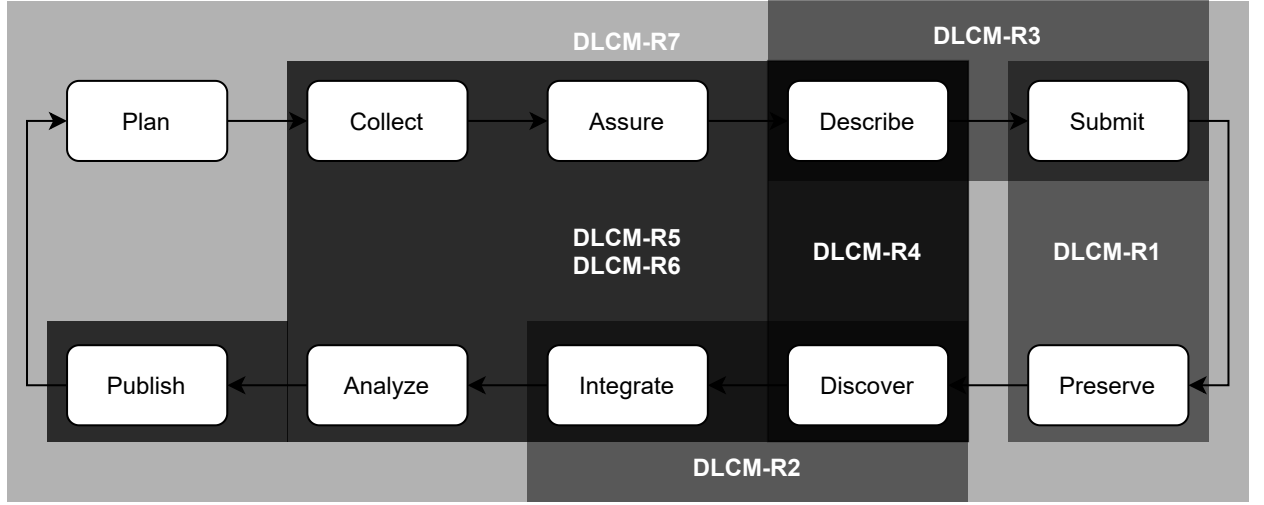


Figure 3.12: Abstract representation of our proposed **DLCM** based on *GFBio* [GFB22] (see section 3.2.1 for details).

cessing it becomes a dedicated second steps (cf. the **FAIR** principles).

Addition to **DLCM-R3**: The requirement *metadata submission* essentially affects the stages *describe* and *submit* but marginally also the *collect* stage, where metadata might be gathered.

Addition to **DLCM-R4**: While the *GFBio* **DLCM** interprets the term *data* as data from research projects (e.g., measurement results), its methods can be applied to any thing. Such an extension to include objects like actors or services enables the requirement *vertical integration*. It is mainly used in the stages *describe* and *discover*, but marginally tackles most other stages as well.

Addition to **DLCM-R5**: Apply the *vocabulary standardization* paradigm, which imposes the synchronization of vocabularies and thus enables a common understanding for human and machine, to all stages of the **DLCM**.

Addition to **DLCM-R6**: Using globally-unique, persistent *identifiers* also affects nearly all stages. This can also include the *submit* stage, if the design of the dataspace foresees the generation of **PIDs** there.

Addition to **DLCM-R7**: *Active development* affects the entire lifecycle as whole, because active use in applications proves applicability and an engaged development community boosts correctness and up-to-dateness.

In summary, this chapter developed a **DLCM** for dataspaces. We introduced seven requirements to be met and evaluated these for both general **DLCMs** and dataspace initiatives. **Requirement DLCM-R3** addresses metadata submission and is met by all ten investigated models, and most other requirements were fulfilled by some models. The decentralized approach of dataspaces to keep data at source is only met by half of the models, which also applies for a proper usage of identifiers. A support for vertical integration (cf. **requirement DLCM-R4**), which demands descriptions to cover not only data but also other assets such as services and actors, is only met by three general **DLCMs**. The *GFBio* model meets the most requirements and therefore is a good candidate for an adoption to dataspaces.

3 Developing a Data Lifecycle Model for Dataspaces

Our analysis of [DLCMs](#) in dataspace initiatives covered the [IDS](#), Gaia-X, the [NFDI](#) and the [EOSC](#). While only the [IDS](#) proposes an explicit [DLCM](#), the others provide individual components that we composed into an overall picture for our evaluation. Most dataspace initiatives meet the majority of our requirements with conceptual approaches, but leave room for improvement when it comes to technical implementation or in-depth explorations. We combined the findings from general [DLCMs](#) and dataspace initiatives to propose a [DLCM](#) for dataspaces. [Figure 3.12](#) illustrates our proposal, which is based on the *GFBio* model and was improved with extended phase descriptions so that it considers all given requirements and is properly suited for application in dataspaces.

In the following chapters of this thesis, the phases of the [DLCM](#) for dataspaces are addressed at different layers (see gray highlighted boxes in [figure 3.13](#)). We propose this [DLCM](#) layer model based on the structure of the International Organization for Standardization’s Open Systems Interconnection OSI model [[Inf94](#)] and follow an approach by Decker et al. [[DHH11](#)] to integrate a semantic layer just below the application layer. [Chapter 5](#) addresses the *data layer* by making data from heterogeneous sources usable for dataspaces. The *dataspace layer* is covered in [chapters 3](#) and [4](#) and enables communication in dataspaces. Finally, the *domain layer* addressed in [chapter 6](#) ensures that data consumers in dataspaces can actually understand and use this data.

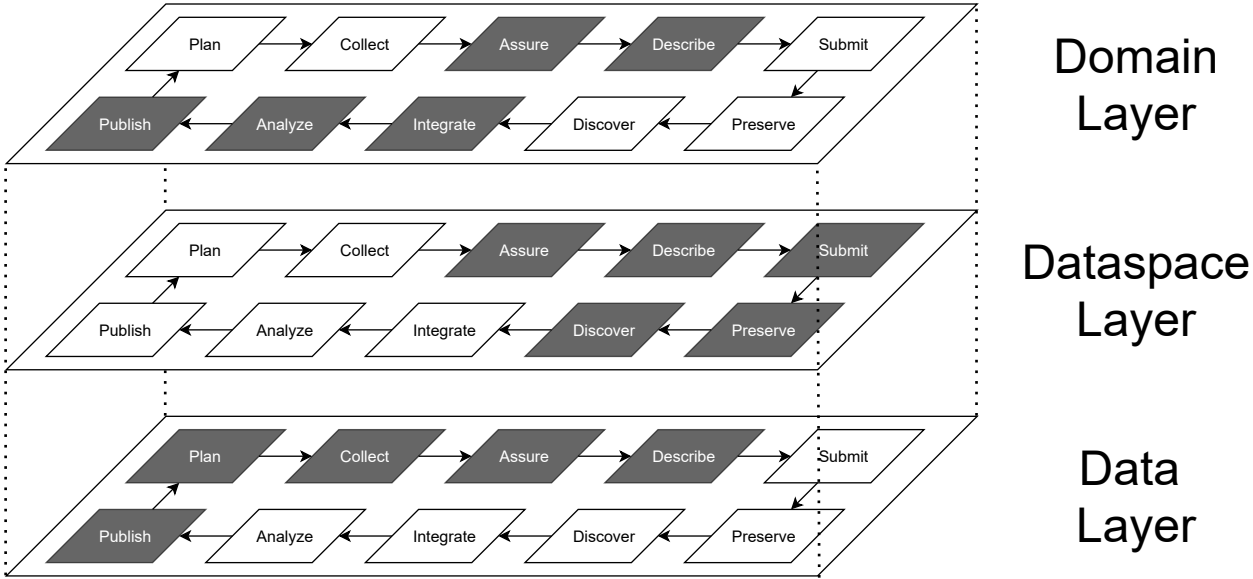


Figure 3.13: The proposed [DLCM](#) for dataspaces is addressed on three layers throughout this thesis.

4 Information Models for Dataspaces

This chapter addresses [RQ 2](#), which questions: *How can both human and machine understanding be achieved in dataspace and what are useful information models to agree on general concepts within dataspace?*

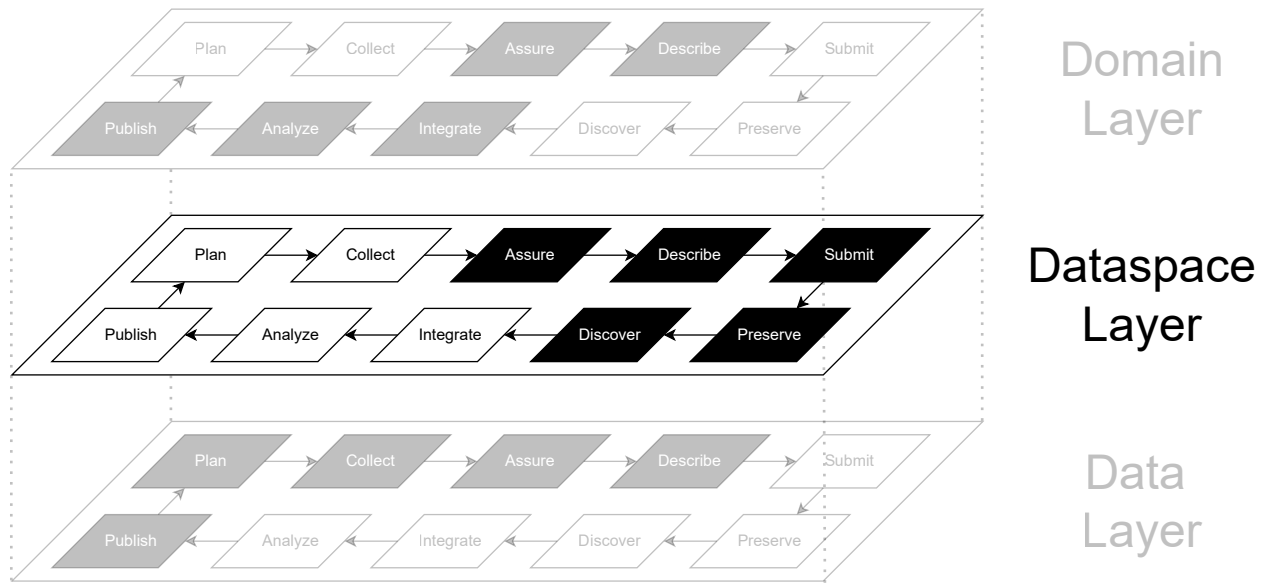


Figure 4.1: This chapter focuses on the *dataspace layer* of the [DLCM](#) layer model presented in [figure 3.13](#).

Following the [DLCM](#) layer model presented in [figure 3.13](#), this chapter explores information models to facilitate common agreement on general concepts on the *dataspace layer* (see [figure 4.1](#)) and emphasizes the importance of conceptual representations and standardized languages. This enables an interoperable implementation of dataspace features such as search and data exchange between participants. A goal of dataspace is to provide an architecture that enables sovereign data exchange between participants in a decentralized manner (see [section 2.1](#)). To achieve this goal, the participants must be able to communicate, collaborate and trust each other, which requires effective descriptions of data, services, participants, and the interactions between them. This leads to a need for a structured approach to address these requirements and establish robust mechanisms for facilitating data governance, sovereignty, and interoperability in dataspace. Information models, which we refer to as semantic descriptions at the dataspace layer, can be used to address the complexities of describing participants, data, and interactions at this layer. While the common understanding of specific domains is formed continuously among individual groups of dataspace participants (see [chapter 6](#)), the information models as the common core of dataspace are usually defined only once.

In the following, we present the principles of dataspace information models and two examples that meet the requirements, and discuss their practical application and value in dataspace.

4.1 Definition and Principles of Information Models

We refer to dataspace information models as a structured approach to effectively describe common core aspects of dataspace that are required for operation. These aspects may include but are not limited to data, services, participants, or the interactions between them. Note that information models focus on the dataspace layer itself, abstracting away individual data or service formats. The following requirements for dataspace information models are derived from the definition of dataspace (see [section 2.1](#)):

1. **Scoped Information:** Derived from the aspect of data federation in dataspace, which emphasizes tailoring data access and integration to specific contexts. Scoped information ensures that the information model includes data relevant to all participants within a particular dataspace, enhancing collaboration by providing a shared context without overwhelming users with irrelevant data.
2. **Data Sovereignty:** Directly derived from the data sovereignty aspect of dataspace, which emphasizes the control and ownership of data by its original creators or owners, even when shared within dataspace. The information model must respect this sovereignty by ensuring that participants retain control over their data and can define access permissions and usage policies.
3. **Semantic Interoperability:** Derived from the aspect of common language and interoperability in dataspace. By utilizing standardized ontologies or vocabularies, the information model ensures that stakeholders can communicate and interpret fundamental concepts consistently across the dataspace ecosystem. This promotes semantic interoperability, enabling effective data exchange and collaboration among diverse participants.
4. **Agreed-Upon Format:** Derived from the aspect of data integration and interoperability in dataspace. The adoption of a common representation format, such as RDF, is essential for seamless communication and data exchange between participants within dataspace. This ensures that all participants natively use a standardized format, which is consistent with the goal of integrating data from heterogeneous sources.
5. **Traceability:** Derived from the aspect of data provenance and lineage tracking in dataspace. Traceability allows participants to trace the origin and flow of data within the dataspace ecosystem, enhancing transparency and accountability. The information model should support mechanisms for documenting and tracking data lineage to ensure its reliability and integrity.
6. **Documentation and Evolution:** Derived from the aspect of continuous improvement and metadata management in dataspace. Proper documentation of the information model, along with a process for continuous evolution and quality assurance, ensures its relevance and up-to-dateness. This is consistent with the goal of effectively managing metadata and facilitating continuous improvement and adaptation within dataspace.

Note that the exact nature of these requirements may vary depending on the individual dataspace. For example, if a dataspace places a lot of emphasis on trust and interactions between participants, the corresponding information model must cover these areas well. If, on the other hand, a dataspace focuses primarily on its content, such as data and services and their context, the information model must be tailored to this. In the following sections, we

address these requirements and propose two information models for dataspace as examples. Both information models have been developed by teams of researchers and practitioners, and our work has contributed to the design and implementation outcomes aimed at promoting the semantic interoperability of the models, use of best practices, and conformance to core aspects of dataspace.

4.2 Example 1: The IDS Information Model

The [IDSA](#) is establishing an ecosystem to facilitate secure, sovereign, and semantically interoperable data exchange (see [section 2.1.1](#)). The *IDS Information Model* is an [RDFS/OWL](#) ontology that defines core concepts for describing actors within a dataspace, their interactions, resources exchanged, and restrictions on data use. The latest major version 4.0.0 was released in August 2020¹ and the latest minor version in September 2020². A sophisticated documentation including several serializations and a visualization is available³. This *IDS Information Model* ontology serves two purposes: It is a catalog of machine-readable terms and data schema for IDS components, and a common language for all participants. See [figure 4.2](#) and an example instance on the *IDS Information Model* and [table 4.1](#) for general key facts. This sample includes: A participant of the *IDS* (bottom left) maintains a technical solution called a *connector* to offer resources in a catalog. These data resources (top right) may have different representations (e.g., *CSV*) and may have usage policies encoded in contract offers (bottom center).

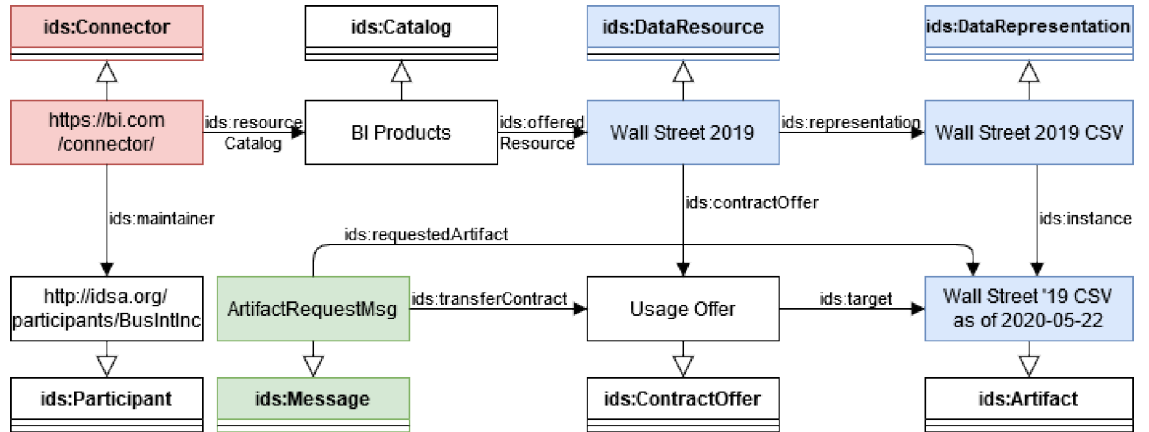


Figure 4.2: A sample instance of the *IDS Information Model* [Bad+20b].

[Table 4.2](#) summarizes our evaluation of the *IDS Information Model* against the above requirements and we discuss relevant details below. For a comparison with the other information model example, see [table 4.4](#).

Scoped Information: The *IDS Information Model* is designed for the *IDS* to describe actors, their interactions, the resources exchanged by them, trust, and data usage restrictions in order to support sovereign and secure data exchange in the *IDS*. It defines general concepts of dataspace, based on the central dataspace concerns shown in [figure 4.3](#). These concerns, represented by concepts in both the documentation and ontology of this information model,

¹<https://github.com/International-Data-Spaces-Association/InformationModel/releases/tag/v4.0.0>

²<https://github.com/International-Data-Spaces-Association/InformationModel/releases/tag/v4.2.0>

³<https://w3id.org/idsa/core>

Table 4.1: Key facts about the [IDS](#) Information Model version 4.0.0 and related resources [\[Bad+20b\]](#).

General	License	Apache License 2.0
	Size	278 classes, 149 object properties, 115 data properties, 684 individuals
	Total size	3912 triples
Reuse	Reused ontologies	CC, DCAT, DCMI Terms, FOAF, ODRL, OWL-Time, VoID, etc.
Documentation	Ontology docs	https://w3id.org/idsa/core/
	Element description	Using <code>rdfs:label</code> , <code>rdfs:comment</code>
Availability	Namespace	ids: https://w3id.org/idsa/core/
	Serializations	Turtle, RDF/XML , JSON-LD , N-Triples
	Source code	IDSA/InformationModel on GitHub
	VoCol [Hal+16] in- stance	http://vocol.iais.fraunhofer.de/ids/

Table 4.2: Evaluation of the [IDS](#) Information Model with the approach and requirements from this chapter. Scores range from (–) to (++).

Requirement	Score	Comment
Scoped Information	++	Sovereign and secure data exchange between participants in international dataspace, including actors, their interactions, the resources exchanged by them, trust, and data usage restrictions.
Data Sovereignty	+	Sovereignty is an essential element in the IDS . They include trust and data usage rules, but enforcement of such is not fully implemented.
Semantic Interoperability	+	Extensive reuse of existing standards and best practices, often defining only new aspects.
Agreed-Upon Format	++	Human- and machine-readable common formats such as HTML , JSON-LD , or RDF/Turtle .
Traceability	+	Provenance and policies can be modeled, but not fully enforced.
Documentation and Evolution	+	Open and closed source repositories with fairly transparent release cycles.

neatly mirror all information areas that this information model should address in the [IDS](#). These concerns include the complex interplay of resource *contents* that are semantically linked to *concepts* to promote interpretability among different participants, as well as the consideration of multidimensional *contexts* such as geospatial information. In addition, the model includes mechanisms for facilitating seamless *communication* among participants using messages between defined endpoints, ensuring the *commodity* of resources such as provenance and quality, and fostering a robust *community of trust* within the expansive ecosystem of dataspace. In particular, the use of gray boxes signifies the careful integration and reuse of established standards, thus promoting interoperability and coherence within the framework. Note that the [IDS](#) Information Model as such is domain agnostic, but can be extended for domain specific applications while leaving its common core unchanged for interoperability.

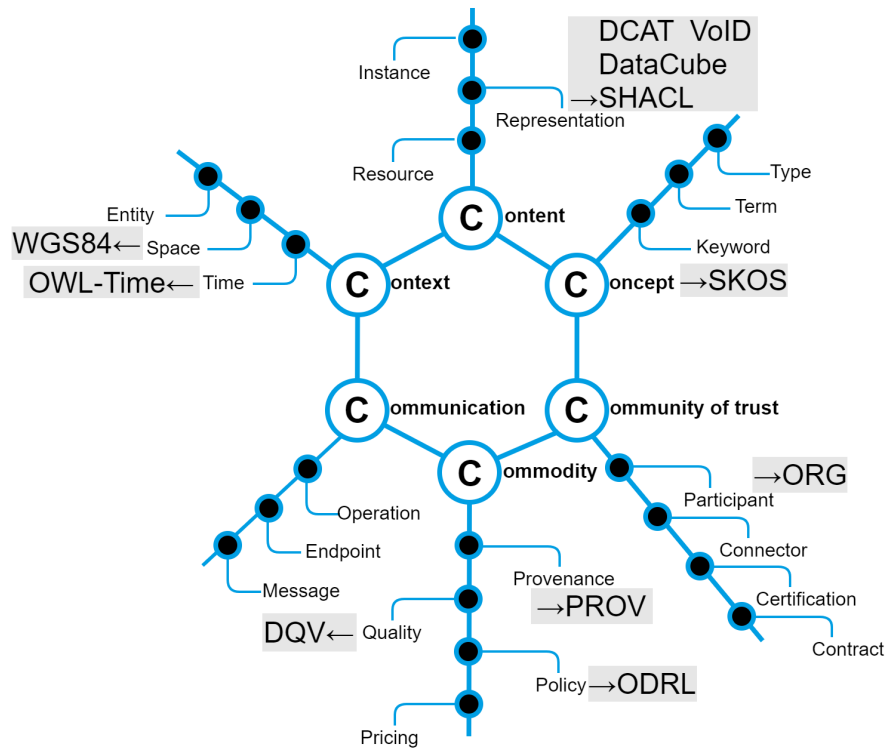


Figure 4.3: The [IDS](#) Information Model addresses several dataspace concerns [Bad+20b]. Gray boxes indicate the reuse of existing standards.

Data Sovereignty: The concerns *community of trust* and *commodity* shown in [figure 4.3](#) are used to address data sovereignty in the [IDS](#), enabling data providers retain control over their data even when sharing it. The so-called [IDS Connectors](#) act as communication interfaces that interpret and enforce any issued policies. The layers of implementation range from rough contracts between participants up to technical solutions for enforcement of agreed-upon data usage policies. The [IDS](#) Information Model is understood by all participants in the [IDS](#) and uses terms from the Open Digital Rights Language [Ian+18] or *MyData*⁴ to formulate human- and machine-readable usage control statements. The close integration of this information model with software component of the [IDS](#) transforms descriptive statements into legally binding and enforceable statements and thus, for example, can grant or deny access to individual resources. [Listing 4.1](#) shows an example policy for a data resource expressed with the [IDS](#) Information Model version 4.0.0, granting read access to members of an organization for

⁴<https://www.mydata-control.de/>

a specified price. Note that in the more recent version 4.2.0, terms such as `ids:leftOperand` are directly used from the Open Digital Rights Language (`odrl:leftOperand`) to increase native interoperability.

```

1 _:StockDataOffer a ids:ContractOffer ;
2   ids:permission [ ids:target _:StockData ;
3   ids:action idsc:READ ;
4   ids:constraint [
5     ids:leftOperand idsc:USER;
6     ids:operator idsc:MEMBER_OF;
7     ids:rightOperandReference <whiterock-invest.com/> ];
8   ids:postDuty [
9     ids:action [ ids:includedIn idsc:COMPENSATE ;
10    ids:actionRefinement [
11      ids:leftOperand idsc:PAY_AMOUNT ;
12      ids:operator idsc:EQ ;
13      ids:rightOperand "5000"^^xsd:double ] ] ] [...] ]

```

Listing 4.1: Example policy (`ids:ContractOffer`) that grants read access to members of a particular organization for a specified price [Bad+20b]. Some details are omitted for readability.

Semantic Interoperability: The **IDS** Information Model follows various best practices known from the Semantic Web, which include:

- **Reuse:** Reuse existing work wherever possible by using or refining terms from standard vocabularies, many of which are W3C recommendations.
- **Linked Data:** It is published under a stable namespace, in common **RDF** serializations, along with human-readable documentation, and linked to external resources.
- **FAIR:** The whole ontology follows the **FAIR** principles.

Existing standards and widely used vocabularies are integrated to this information model in various ways, including extensions of classes (`rdfs:subClassOf`) or properties (`rdfs:subPropertyOf`), or with direct use (e.g. in `rdfs:range`). Reused vocabularies include (see figure 4.3) SKOS [MB09], PROV-O [Leb+13], dcterms [KB07], DCAT [Alb+19], VoID [Ale+11], DataCube [CRT14], SHACL [KK17], WGS84 [Bri04], OWL-Time [Cox+17], ORG [Rey12], and ODRL [Ian+18]. Through this strong interlinking with standards and widely used ontologies, participants of the **IDS** and beyond can understand and interpret the same information consistently.

Agreed-Upon Format: The **IDS** Information Model is available in common representation formats (see table 4.1), including **RDF/Turtle**, **RDF/XML**, **JSON-LD** and **N-Triples**. Its documentation and namespace are hosted under the same **URL**⁵, which follows the **W3C** working group note on best practice recipes for publishing **RDF** vocabularies [BP08b], because it allows dynamic access for both human and machine based on **HTTP** content-negotiation. Both a Java **API** for creating instances and a graphical instance manager are available to minimize entry barriers, especially for non-experts [Bad+20b]. The common representation formats of the **IDS** Information Model and supportive extras allow dataspace participants to seamlessly communicate.

⁵<https://w3id.org/idsa/core>

Traceability: The IDS Information Model, mainly in the concerns *commodity* and *community of trust* (see figure 4.3), provides means for tracing and tracking in the process of data provisioning and (re)use. Terms from the *PROV-O* [Leb+13] ontology, which is a W3C recommendation, are used to provide traceability of the provenance, access, and use of data exchanged in the IDS. Software components based on central certification authorities (*community of trust*) can be used to add technical accountability in the IDS as a decentralized ecosystem.

Documentation and Evolution: The IDS Information Model is documented in several ways, including a GitHub repository with examples⁶, the IDS Knowledge Base⁷, and a documentation page⁸ offering information, serializations, and visualization for all versions of this information model released so far. The continuous development is accompanied by *continuous integration and continuous deployment* mechanisms with automated quality control. The quality of the ontology’s RDF/Turtle source code is assessed with syntax checks, reasoner-based logical consistency checks, and *RDFUnit*⁹ test cases [Bad+20b]. Evolution of the IDS Information Model is achieved via collaboration in a GitHub repository and continuous work of a sub-working group, which leads to regular releases on a *stable* and *develop* branch (see figure 4.4): Participants can raise issues (left) in the GitHub repository for enhancements or bug reports, which are then addressed according to software development best practices. Release cycles (right) combine a release plan with community proposals, are implemented by the component owner (currently Fraunhofer), evaluated by *Sub-Working Group 4: Information Model*, and finally released as open source. The proper documentation of this information model and its clean process for continuous evolution ensures that the model remains relevant and usable for a broad range of users.

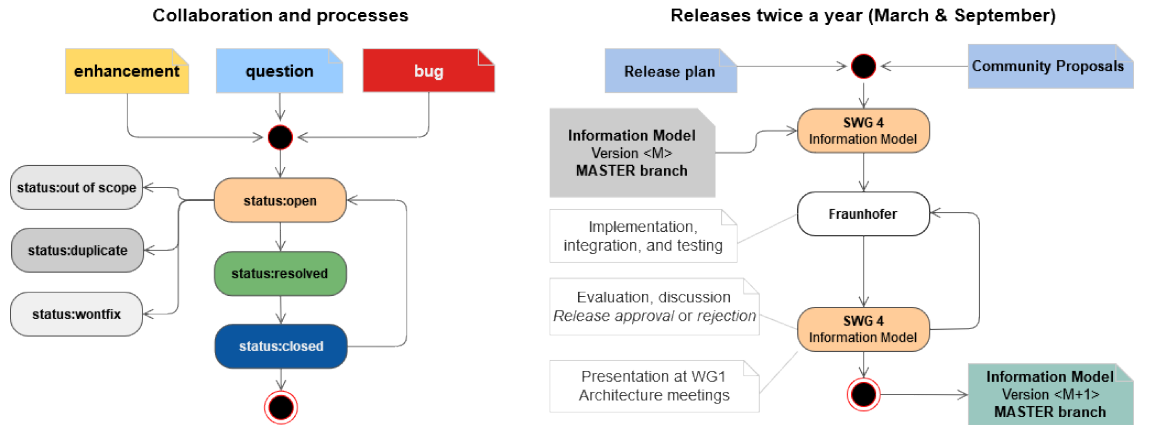


Figure 4.4: The update and release process of the IDS Information Model [Bad+20b].

In summary, the IDS Information Model, driven by the IDSA, is a domain-agnostic, mature, and sophisticated information model for the IDS dataspace and satisfies all requirements for dataspace information models (see section 4.1 and table 4.2).

⁶<https://w3id.org/idsa/core>

⁷<https://docs.internationaldataspaces.org/ids-knowledgebase/>

⁸<https://w3id.org/idsa/core>

⁹<https://github.com/AKSW/RDFUnit>

4.3 Example 2: The Digital Shadow Reference Model

This second example of an information model for dataspace comes from the production and manufacturing domain (see [section 2.3](#)) and is being developed in the cluster of excellence *Internet of Production*¹⁰ [Pen+19]. The *Digital Shadow Reference Model* was first proposed in [Bec+21] and later refined in [Mic+23]. This information model addresses the challenge of the ever-increasing amount of heterogeneous data and models within the production domain, which requires a precise and standardized understanding of their foundations, structure, forms of aggregation, and most importantly, their use. A central aspect of this model is the concept of *Digital Shadows*, which is defined as “a set of contextual data traces and their aggregation and abstraction collected concerning a system for a specific purpose with respect to the original system” [Bec+21]. These digital shadows can be used to share data, for example in digital twin applications. [Figure 4.5](#) shows the Digital Shadow Reference Model in the form of a class diagram: This information model uses digital shadows (top center) as key elements and includes information about their purpose (top right) and models of various types (right). It refers to the actual data content via data traces (center), which contain not only data points and metadata, but also the source system (left) from which they originate. The source of data can be the results of any kind of data generation (bottom left), including the reuse of other data, or any real-world asset (left), such as production machinery.

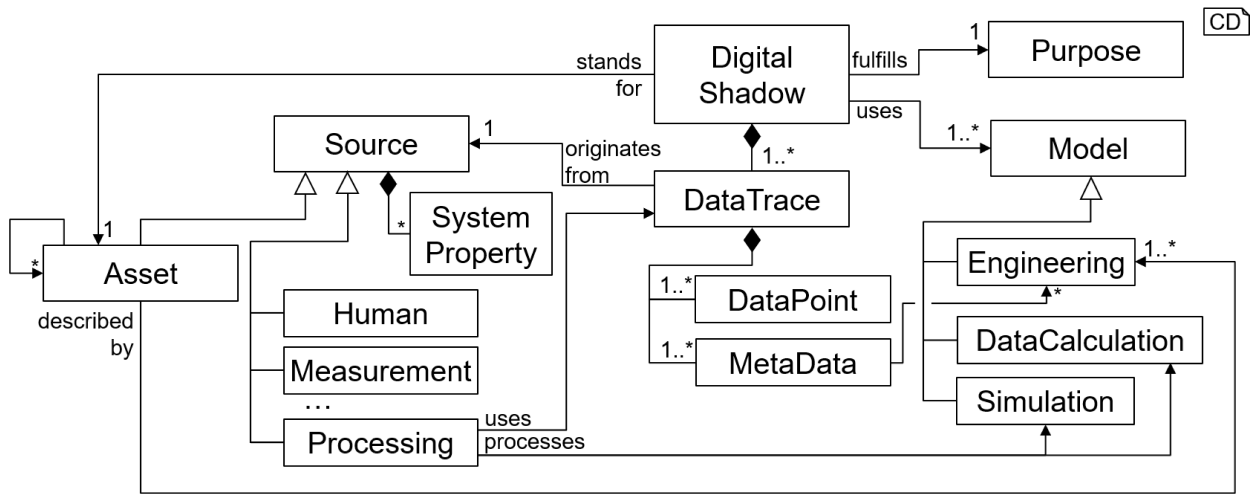


Figure 4.5: The refined *Digital Shadow Reference Model* [Mic+23].

[Table 4.3](#) summarizes our evaluation of the *Digital Shadow Reference Model* against the above requirements and we discuss relevant details in the following. For a comparison with the other information model example, see [table 4.4](#).

Scoped Information: The Digital Shadow Reference Model aims to contribute to a standardized information architecture of relevant data in the manufacturing domain. With the vision that this entire domain will become a globally connected production laboratory, researchers in the Internet of Production are working to harmonize the respective notion space. This requires common descriptions of digital shadows, data structures for leveraging data and models, and relevant contextual information. This information model focuses on the use of data from digital shadows for decision making. These digital shadows (see [figure 4.5](#)) are augmented with a variety of additional information, ranging from real-world assets to models

¹⁰<https://iop.rwth-aachen.de/>

Table 4.3: Evaluation of the *Digital Shadow Reference Model* with the approach and requirements from this chapter. Scores range from (–) to (++).

Requirement	Score	Comment
Scoped Information	++	Purposefully collect, aggregate, and abstract production data in the form of digital shadows, enriched with meta-information for rapid decision-making.
Data Sovereignty	-	All data in this research project is currently treated as free open access, no sovereignty techniques are applied.
Semantic Interoperability	++	Reuse of existing standards and best practices, defining only new aspects.
Agreed-Upon Format	+	Mainly provided as a high-level model only, but an RDF / Turtle ontology is available.
Traceability	+	Traceability is provided only for pure data (data lineage) by referencing data traces, processing steps, and links between them.
Documentation and Evolution	-	Documentation through scientific publications only, no transparent approach for continuous evolution.

of various types to interlinks with actual data traces and other digital shadows. Note that the Digital Shadow Reference Model is tailored to the production domain and the data within it. It is not intended to contain sophisticated information about dataspace participants beyond the general information contained in its `purpose` or `model` fields.

Data Sovereignty: The Internet of Production researchers developing the Digital Shadow Reference Model treat all dataspace participants equally and assume that all data contained in digital shadows should be available to everyone. This means that this information model does not distinguish between different participants in a dataspace and does not include mechanisms for data sovereignty, authorization, access control, or usage policies. Note that this information model follows the dataspace principle of decentralized data storage, which still allows access control to be implemented on another layer. From a technical perspective, this means that the references to `data traces` (see [figure 4.5](#)) within this model do not automatically grant access to the actual data.

Semantic Interoperability: The Digital Shadow Reference Model is based on standards and common languages that define fundamental concepts. These include [UML](#) class diagrams for layout descriptions, [SHACL](#) and the Object Constraint Language for value restrictions, ontologies in general and [SPARQL](#) for various metadata fields, the Systems Modeling Language for asset structures, MontiArc [\[Dal+20\]](#) for architecture modeling, or MatLab Simulink [\[Doc20\]](#) for simulating an asset’s behavior. The definition of an asset as “*an item, thing or entity that has potential or actual value to an organization*” is based on the norms *DIN SPEC 91345* [\[Int16b\]](#) and *DIN ISO 55000* [\[Int17\]](#). By adhering to these standards and commonly used languages and solutions, this information model helps different dataspace participants to understand and interpret the same information consistently.

Agreed-Upon Format: The Digital Shadow Reference Model is available as a [UML](#) model and as a [RDF](#)/[Turtle](#) ontology; other representation formats are not present. These pub-

lished formats are documented, but only in the form of scientific publications. There are no programmatic solutions or graphical applications yet. No endpoint with content negotiation is maintained, which makes machine retrieval of this information model difficult. The provided common formats [URL](#) and [RDF](#) allow basic usage, but are only small steps towards seamless communication between dataspace participants.

Traceability: This information model provides traceability for pure data, but cannot provide full transparency or accountability within the dataspace. Data lineage can be constructed, but not enforced, by referencing data traces, processing steps, and links between them (see [figure 4.5](#)). Documentation of the source of data and the processing steps that modify it can be facilitated across multiple digital shadows or assets. However, proper provenance, including participants, certifications, and trust, is not included in the Digital Shadow Reference Model.

Documentation and Evolution: Documentation of the Digital Shadow Reference Model is provided through scientific publications [[Bec+21](#); [Mic+23](#)] only, which allows the reader to understand its general concept and even important details but limits its use. Several examples in real-world use cases (see [figure 4.6](#)) demonstrate its applicability but are not available free open access for reproduction: The referenced injection molding machine (top left) includes the machine itself, a mold, material, and a human operator (left). The data traces contain process data and additional metadata (bottom center). Based on this digital shadow, a digital twin (top center) controls the cavity pressure (center) using an appropriate computational model (right). However, the scientific nature of the authors of this information model does not motivate a plan for continuous evolution of this model or a methodology of such.

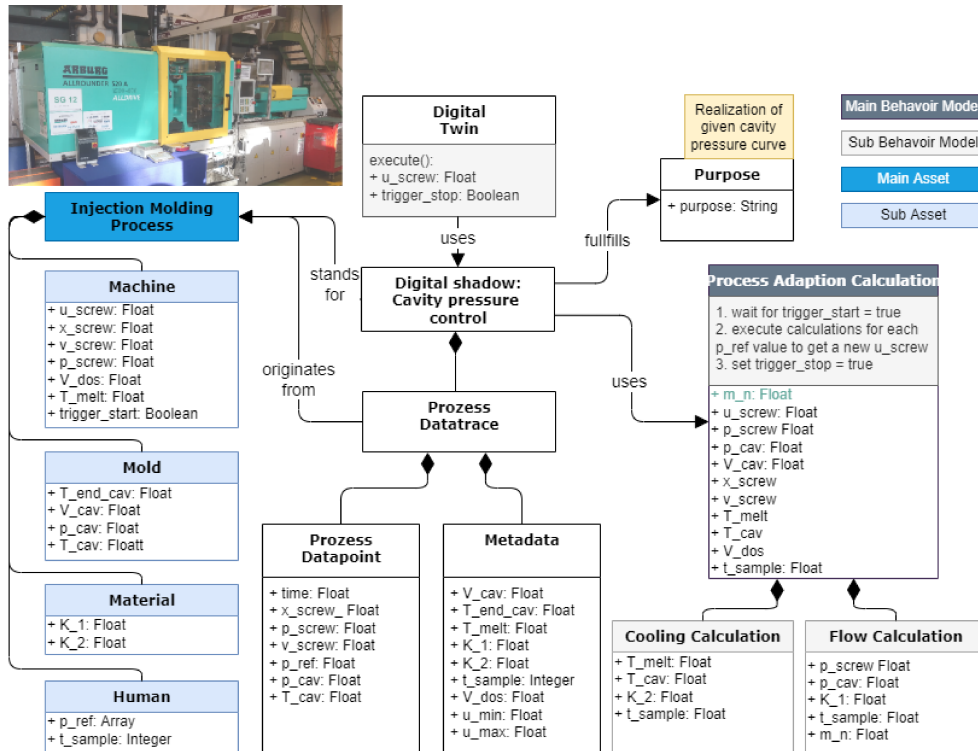


Figure 4.6: Example application of the Digital Shadow Reference Model to a process control use case in injection molding [[Bec+21](#); [Mic+23](#)].

In summary, the Digital Shadow Reference Model developed by researchers in the *Internet of Production* research project is an information model focused on supporting decision making in the manufacturing domain (see [section 2.3](#)). It fully satisfies two requirements for dataspace information models, partially satisfies two, and does not satisfy two (see [section 4.1](#) and [table 4.3](#)). Compared to the [IDS](#) Information Model, it is less mature at both the conceptual and implementation levels, but has sophisticated features tailored to a specific domain. At its core, digital shadows provide all relevant information, including cross-references to other digital shadows, in order to gather and provide relevant information for defined purposes that support informed decisions by dataspace participants.

4.4 Discussion and Conclusion

Exploring information models in the dataspace layer addresses the challenges of human and machine understanding while establishing a common base for interoperability and data exchange. This chapter explored [RQ 2](#) and the nature of information models in facilitating a common understanding of general concepts within dataspaces.

Our key discussions and conclusions based on the principles and examples we presented are as follows. The chapter emphasized the key role of information models in enabling effective communication, collaboration, and trust among participants in decentralized dataspace ecosystems. By providing structured representations of data, services, participants, and interactions, information models serve as a foundation for establishing robust mechanisms for data governance, sovereignty, and interoperability. The adoption of semantic descriptions at the dataspace layer additionally contributes to a common base for communication in dataspaces. We presented two illustrative examples of information models designed to meet the needs and requirements of dataspace environments. Through these examples, the chapter demonstrated the practical application and usefulness of information models for dataspaces.

Table 4.4: Summary of how the two example information models satisfy the requirements for dataspace information models.

	IDS Information Model	Digital Shadow Reference Model
Scoped Information	++	++
Data Sovereignty	+	-
Semantic Interoperability	+	++
Agreed-Upon Format	++	+
Traceability	+	+
Documentation and Evolution	+	-

[Table 4.4](#) summarizes how the two information models satisfy key requirements within dataspace environments, with scores ranging from (-) to (++). The [IDS](#) Information Model satisfies all of the requirements fairly well (indicated by + or ++): scoped information, data sovereignty, semantic interoperability, agreed-upon format, traceability, and documentation and evolution. On the other hand, the Digital Shadow Reference Model partially satisfies four (+ or ++) requirements: of scoped information, semantic interoperability, agreed-upon format, and traceability. This model does not fulfill two requirements (-): data sovereignty and documentation and evolution. This may indicate potential areas for improvement if the

4 Information Models for Dataspaces

Digital Shadow Reference Model is to be used in a general dataspace. However, it is tailored to the manufacturing domain using digital shadows to provide relevant information, and may still be the preferred model depending on the scope of the dataspace (e.g., what information participants need to make informed decisions).

In summary, the principle of dataspace information models offers a promising way to address the complexity of dataspace environments and advance the realization of sovereign data exchange. Information models lay the foundation for the development of interoperable dataspace ecosystems by providing a common framework for defining core assets and promoting unified semantic understanding. Further research could refine existing information models, explore new methodologies, and address emerging needs in dataspace to further improve common understanding on the dataspace layer and the use of general concepts in dataspace.

5 Principles of Heterogeneous Data Access for Dataspaces

This chapter addresses [RQ 3](#), which is: *How can data access from heterogeneous sources, such as manufacturing systems, be effectively facilitated within dataspaces to support seamless integration and utilization of diverse data?*

In today's dataspace landscape, there are a multitude of different dataspaces, each tailored to specific domains, communities, and solutions. However, within this diversity lies a significant challenge: ensuring interoperability within and across these different dataspaces. This challenge is magnified by the diverse nature of the heterogeneity found within these dataspaces, which includes differences in data formats, access methods, speeds, resource limitations, evolving component definitions, and variable metadata availability across data sources. To address this challenge, we propose mapping specific standards to more widely accepted standards, such as common Web standards like [RDF](#). The goal of this chapter is to explore the core of dataspaces: *the data*. By examining methodologies for accessing and integrating relevant data from sources despite various aspects of heterogeneity, this chapter aims to improve interoperability and collaboration within and across dataspaces.

Following the [DLCM](#) layer model presented in [figure 3.13](#), this chapter explores strategies and approaches that facilitate data availability in dataspaces at the *data layer* (cf. [figure 5.1](#)). Our contributions allow the upper layers to assume data availability and readiness for use in dataspaces, enabling them to leverage any kind of data on a more abstract level. The *data layer* of the [DLCM](#) mainly addresses the *plan* and *collect* phases by making data from heterogeneous sources available. It also addresses the *assure* and *describe* phases by providing quality assurance/control and understandable descriptions for data. Therefore, the goal of this chapter is to explore principles of heterogeneous data access to provide a layer that makes this data available for use in dataspaces.

Chapter Outline

This chapter aims to prepare all relevant data from potentially heterogeneous sources for integrated use in dataspaces and is structured as follows:

- [Section 5.1](#) first presents the requirements and our methodology.
- [Section 5.2](#) discusses techniques for collecting data and metadata from sources that have heterogeneous data formats and access to the data.
- [Section 5.3](#) then examines and overcomes the resource limitations of source systems, such as data velocity imposed by limited network bandwidth.
- [Section 5.4](#) presents a solution for confidently managing changes of individual system components and their consequences for semantically correct communication.

- [Section 5.5](#) proposes methods to generate more valuable metadata than is natively provided by the heterogeneous source systems. This contributes to a better understanding of the associated data in dataspace.
- [Section 5.6](#) finally discusses and concludes this chapter, aiming to identify methods for making data from heterogeneous sources available for integrated use in dataspace.

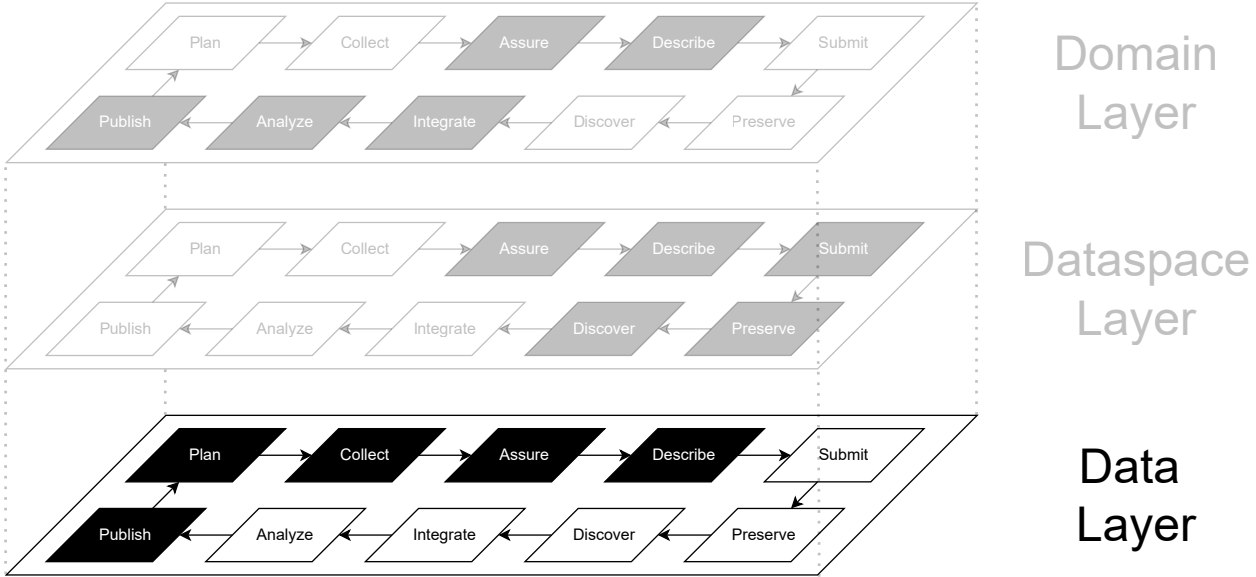


Figure 5.1: This chapter focuses on the *data layer* of the [DLCM](#) layer model presented in [figure 3.13](#). We explore principles for accessing data from heterogeneous sources to make relevant data available for integrated use in dataspace.

5.1 Requirements and Approach

Data is a core aspect of dataspace and is created in many different ways by many different sources. The availability of data is constantly growing, for example, as more sensors and other data sources are deployed. The need for data is also growing, as more and more decision support systems and machine learning or artificial intelligence solutions are deployed. At the same time, the world is constantly changing, and keeping data source systems interoperable and up-to-date has become a challenge. Although one of the dataspace principles is to *keep data at source and distributed* (see [section 2.1.2](#)), some preparation is still required to enable interoperable data exchange in dataspace. This includes making the data, which may be accessible only through complex interfaces, available and accessible to dataspace (see the *plan* and *collect* phases of [figure 5.1](#)). And, assuming fundamental access to such data source systems, being able to precisely control what and how data should be collected, handling certain constraints imposed by these systems or their environments (*collect* and *assure* phases). Subsequently, the ever-changing nature of data sources and their contexts requires confident management of real-world changes to digital descriptions to maintain correctness of shared understanding (*assure* and *describe* phases). Finally, even data source systems with little or poor semantic description may be relevant to dataspace, and thus their metadata must be appropriately enriched to enable seamless integration into dataspace (*describe* phase).

This leads to the following requirements for creating access to data from heterogeneous sources on the *data layer* in dataspace:

Data-R1 Data Accessibility: Data from diverse sources, even those with complex interfaces or query languages, must be made available and accessible to dataspace. Ensuring data accessibility involves making these data sources available and easily accessible to dataspace.

Data-R2 Controlled Data Collection: There should be precise control over what data is collected and how it is collected, while respecting resource constraints imposed by data source systems or their environments. Controlled data collection helps ensure that only relevant and reliable data is used in dataspace and that the velocity of added data stays manageable.

Data-R3 Management of Data Changes: Data sources are dynamic and subject to changes in structure, content, and context. Data change management involves implementing strategies and tools to track, monitor, and manage these changes effectively. Effective management of data changes ensures the consistency, accuracy, and reliability of data within dataspace, even as data sources evolve over time.

Data-R4 Enriched Metadata: Metadata provides essential context and information about the data, including its structure, semantics, and provenance (see the [FAIR](#) principles in [section 2.2.2](#)). Enriching metadata involves augmenting existing metadata with additional information to improve its usefulness and relevance within dataspace. Especially, but not only, source systems with limited semantic description can benefit from metadata enrichment and thus be better understood and used in dataspace.

Our methodology in this chapter is to address the above derived [requirements Data-R1](#) to [Data-R4](#) in the subsequent [sections 5.2](#) to [5.5](#), respectively, and to propose individual but modular solutions to overall contribute to a better support for data from heterogeneous systems in dataspace.

5.2 Enabling Data Accessibility from Heterogeneous Systems

This section addresses [requirement Data-R1](#) and aims to make data from heterogeneous sources accessible to dataspace. The availability of data is increasing because more data sources are becoming available (e.g., sensors are being connected or data processing is being set up; scale out) and because these data sources are providing more data (e.g., due to more complex production machines or more sophisticated data reuse; scale up). At the same time, the need for data increases [[LS20](#)] as more and more decision support is required in the complex world we live in (scale out) and as such systems become more complex (e.g., deep learning in artificial intelligence or large language models; scale up). Data availability does not automatically mean data accessibility. Even if the data is generated, e.g. by a production machine, dataspace participants still need a convenient way to collect this data in order to use it in dataspace, and a convenient way to collect relevant metadata to improve common understanding with other dataspace participants (at all levels of the data lifecycle model, see [figure 3.13](#)). Because different, specific dataspace intend to use data from different, specific

data sources, different, specific data access solutions are required. Our goal is to reduce this broad landscape of specialized data access solutions to fewer, standardized solutions.

In 2011 [KLW11] started a debate about Germany’s competitiveness as a high-wage manufacturing country in a globally connected world, and autonomous communication became a core aspect [Pet+17]. A common vocabulary [Pet+17] and shared semantics to properly understand the communication [VDI20] are required for the required machine-to-machine communication. OPC UA [HSK08] and its extensions such as IEC 62541 [IEC20] were designed by the *OPC Foundation* to enable autonomous and semantically interpretable information exchange and is now a widely used protocol. More and more manufacturers are adding OPC UA servers to their assets, but are not utilizing its full potential. Similar machines from different vendors are often not uniformly configured [Gön+19], or common data extractions do not include valuable metadata such as units of measure. Data collection is typically customized for each data source, and OPC UA queries often have to be written manually, which is time-consuming and error-prone (see the example in figure 5.2): This sample shows a screenshot of the *UaExpert* OPC UA client from *Unified Automation* with red boxes highlighting values of interest: identifier (top), sensor value (middle), and link to engineering units (bottom). Even with such graphical interface clients, this manual approach remains time-consuming and error-prone. More expressive, dynamic, and convenient query languages exist, such as XPath [RDS17a] (for XML). Higher *expressiveness* in this context refers to better usability, greater portability, and a higher level of abstraction. These could improve accessibility for dataspace participants, but cannot be used natively on OPC UA servers.

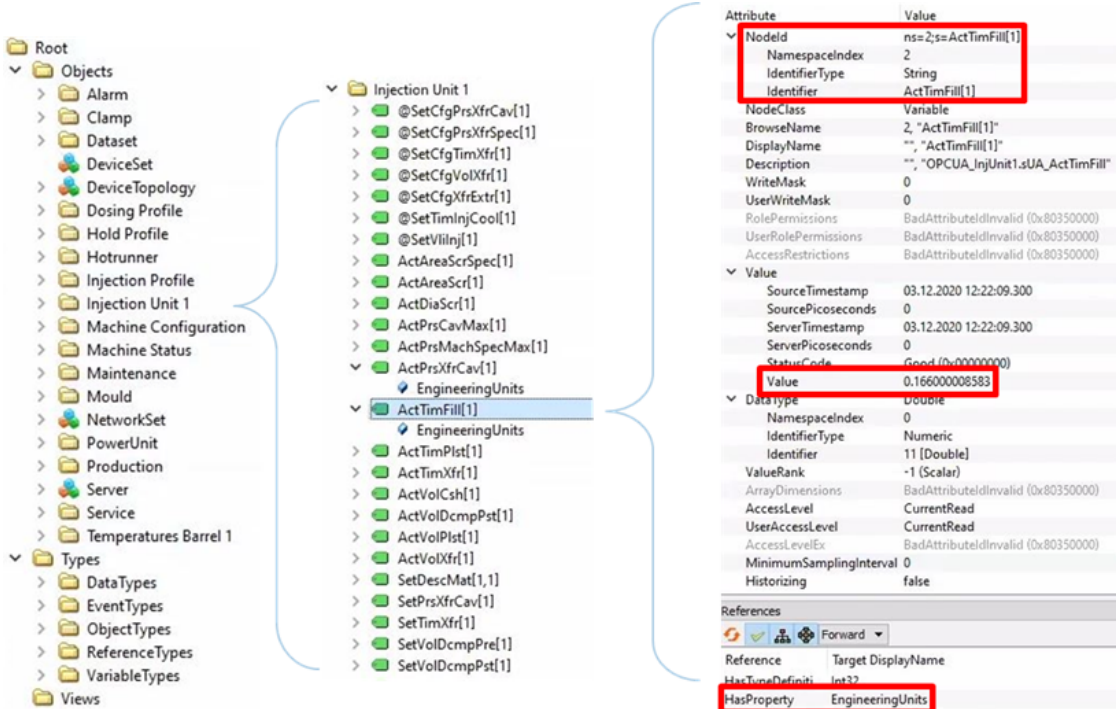


Figure 5.2: Real-world example demonstrating the high manual effort required to inspect and manage data and metadata from OPC UA [The+22a; Koc21].

Therefore, we propose an approach to use expressive query languages (such as XPath) on data sources with limited APIs (such as OPC UA) using a transformation process. This approach bridges the gap between highly expressive query languages and less expressive ones

through a transformation layer, promotes common understanding among dataspace participants by ensuring that relevant metadata accompanies the data, and improves unified access through unified queries that can be applied to heterogeneous data sources, which ultimately contributes to improved data accessibility from heterogeneous sources. Specifically, this includes designing an algorithm for mapping between the data structure of many data sources (graph format) and the data structure of the Web standard XML (tree format).

5.2.1 Related Work and Background

This section explores the background and related work on structuring and transferring data in OPC UA as well as XPath [RDS17a]. It establishes connections among various aspects such as XPath, the concepts of Semantic Web, OPC UA, or the extraction of sensor data that have been previously discussed in works like [LGD20; Lip+20; Lip+21c]. Researchers have utilized XQuery as a language for data integration [RG15] or introduced a syntax for translating to and from non-XML documents. While [LSV11] presents methods to integrate semantic technologies with OPC UA, [Sch+19] and [SFK19] focus on transforming OPC UA information models into OWL. Additionally, [SFK19] retrieves OPC UA sensor values through user-defined functions using SPARQL, and XSPARQL [Bis+12] can merge RDF with mappings from XPath to OPC UA.

OPC UA defines *information models*, which are not to be confused with the dataspace information models introduced in chapter 4. They denote a directed graph that describes the data structure of an OPC UA server, encapsulating *nodes*, *references*, and *attributes*. The *address space* constitutes the entire information provided by an OPC UA server [LIB14]. Typically, a server hosts multiple *namespaces*, and new nodes can be appended via unique *NodeIds* to custom namespaces defined by the prefix URIs. To enable OPC UA clients to access nodes and extract information, the two fundamental methods, *Browse* and *Read*, are used. The *Browse* method allows step-by-step traversal of the address space, and *Read* retrieves the attributes of individual nodes [IEC20]. More complex operations are not available by default and require composition by the client.

5.2.2 Approach

Low-level query and data collection interfaces are widely used in data sources and provide a high *availability* of data. Examples include basic text filtering with regular expressions, operations on CSV files, or OPC UA when only basic traversal methods such as following references and reading the current node are available. Query languages that are more expressive, abstract, and usable are more convenient for dataspace participants and provide more flexibility, expressiveness, and interoperability. Examples include XPath [RDS17a], XQuery [RDS17b], and SPARQL [Ara+13]. Our approach is to cross the bridge between these to turn data *availability* into data *accessibility* for dataspaces.



Figure 5.3: Our approach [The+22a; Koc21] involves mappings between differently expressive query languages and their underlying data models.

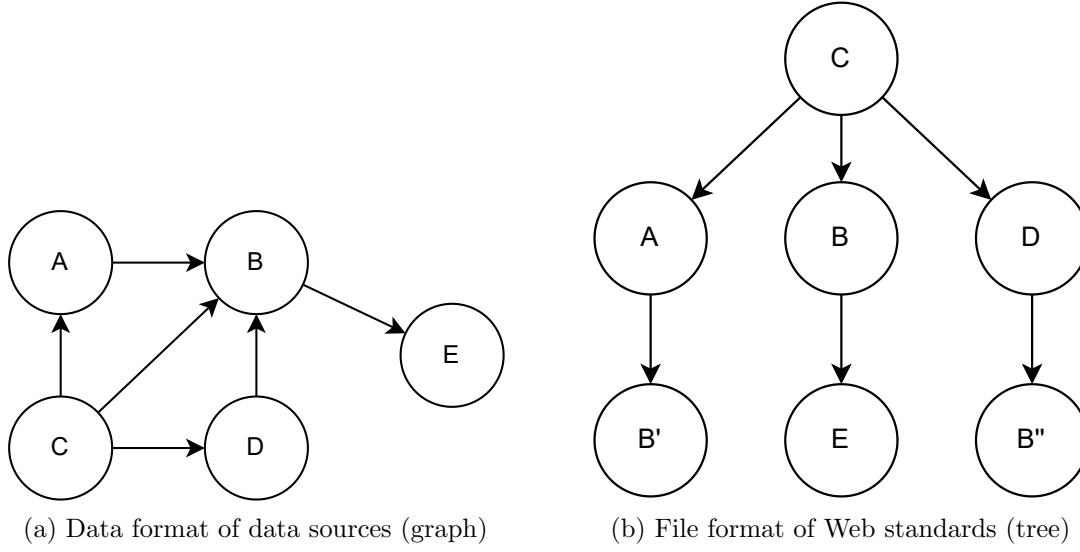


Figure 5.4: Implementing our approach involves mapping the graph data format (a) to the tree data format (b).

This methodology involves transforming queries formulated in highly expressive query languages into less expressive ones (see [figure 5.3](#)) to enable flexible access to data metadata in existing setups. This approach makes low-level query functionality (e.g., [OPC UA](#)) accessible via more convenient, higher-level query languages (e.g., [XPath](#)). Implementing this involves mapping the underlying data model of many source systems (graph format) to a data model supported by Web standards such as [XML](#) (tree format), see [figure 5.4](#). While this transformation theoretically leads to exponential complexity and runtime, the input data in the graph model usually has a low degree of interconnectedness, which in practice leads to reasonable complexity and runtime. The approach of mapping the data models and the query languages allows the user to benefit from the higher usability and abstraction of one language when formulating queries, but also allows those queries to be executed on a variety of systems that only support the other query language. [Figure 5.5](#) shows how the principle of generic translation between query languages is demonstrated in a common real-world [OPC UA](#) use case from the manufacturing domain (see [section 2.3](#)) with mappings from the expressive query language [XPath](#). The proposed workflow is as follows: Export the [OPC UA](#) information model to [XML](#), convert it to an [XPath](#)-compatible XML, design [XPath](#) expressions, and finally translate and execute them in [OPC UA](#) with a modified client. Our approach replaces individual, machine-dependent solutions with universal ones that reduce time and complexity for dataspace participants and are portable to other data sources such as machines. Specifically, we enable dataspace participants to formulate convenient [XPath](#) expressions based on data exports and to execute these queries on live [OPC UA](#) servers.

5.2.3 Results

In this section, we present our results for bridging different expressive query languages to improve data accessibility for dataspaces by implementing our approach shown in [figure 5.3](#). We demonstrate this transformation process on [XPath](#) as a more expressive query language and [OPC UA](#) as a less expressive one. Following the workflow illustrated in [figure 5.5](#), our solution is to export information models from [OPC UA](#) servers, post-process them locally

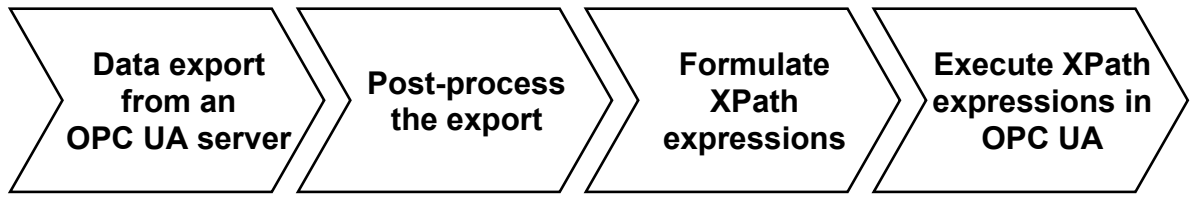


Figure 5.5: We demonstrate our methodology by proposing this workflow for a common manufacturing use case (see [section 2.3](#)), mapping [XPath](#) to [OPC UA](#) [[The+22a](#); [Koc21](#)].

to make them [XPath](#)-compatible, formulate [XPath](#) expressions, and finally translate and execute these queries on an [OPC UA](#) server using a modified [OPC UA](#) client.

Our implemented solution is available on GitHub [[Koc+21](#)] and uses Python 3.9.0 and the asynchronous [OPC UA](#) client *asynua* 0.9.0 from FreeOpcUa¹. The software implementation has two parts: Export creation and post-processing (steps one and two below), and an extended [OPC UA](#) client that allows execution of any of the [XPath](#) axes on any [OPC UA](#) server, based on [OPC UA](#)’s native methods *Browse* and *Read* (step four below). It allows any [XPath](#) expression formulated by dataspace participants to be transformed to [OPC UA](#) and executed, and returns the native [OPC UA](#) result.

Data Export from an OPC UA Server

[OPC UA](#) is a standardized protocol that is supported by both server and client. Any client implementation should be able to access the data in any server implementation. A general approach to creating an export of an [OPC UA](#) server is to start at the root node and iteratively follow links to all reachable nodes. This can be done using any search algorithm, including breadth first search and depth first search. A complete snapshot of the contents of the server at that time is obtained by repeating this procedure for each namespace of the server. The export is in the form of a directed graph and can be serialized to [XML](#), for example.

Post-process the Export

[XPath](#) requires [XML](#) documents to be in a cycle-free tree structure. However, the exported [OPC UA](#) server address spaces are directed graphs of nodes, which usually contain cycles. These two models (graph vs. tree) are not identical and therefore the export is not suitable as input for [XPath](#). We overcome this challenge as shown in [figure 5.6](#): We handle the differences in the data structures of [OPC UA](#) (left, graph structure) and [XML](#) (right, tree structure) with a flattening approach using node duplication. Using an extended breadth-first search, this node duplication expands the [OPC UA](#) address space graph to a valid [XML](#) tree [XPath](#), allowing [XML](#) to handle data exports from [OPC UA](#) servers. A second challenge arises because children are ordered in [XML](#) but not in [OPC UA](#). Our solution is to order them lexicographically on the basis of their unique *NodeId*. Due to the node duplication introduced in the extended [XML](#) representation, node identifiers may also be duplicated. These node identifiers are kept intact, and for consistency during [XPath](#) query evaluation, we continuously expand the current selection to include any duplicated nodes with the same

¹<https://github.com/FreeOpcUa/opcua-asyncio>

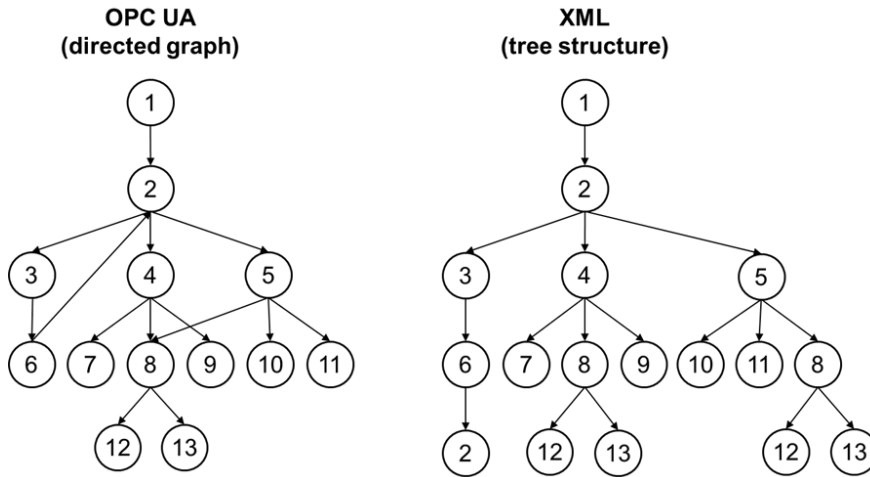


Figure 5.6: Our approach to overcoming the differences in the data structures graph and tree [The+22a; Koc21].

identifiers. The result of these two steps (see figure 5.5) is an [XPath](#) compatible export from the [OPC UA](#) server. Technically spoken, our Python tool connects to an [OPC UA](#) server and yields a post-processed [XML](#) export for [XPath](#) usage.

Formulate XPath Expressions

We use the expressiveness of [XPath](#) to allow dataspace participants to access data and meta-data such as units or ranges. They design [XPath](#) expressions for post-processed [OPC UA](#) export that capture all the information of interest. This approach is more convenient and scalable than existing complex and static machine-based [OPC UA](#) extractions. The same query can be applied to any number of nodes simultaneously, such as reading values, units, and ranges for a particular node. The output of this step is an [XPath](#) expression to extract the desired information from the extended [XML](#) version of the [OPC UA](#) export.

Execute XPath Expressions in OPC UA

[XPath](#) expressions cannot be executed natively on [OPC UA](#) servers, so a transformation approach is required. While [XPath](#) is more expressive and provides twelve usable axes, [OPC UA](#) only provides the two methods *Browse* and *Read*. Therefore, transforming the former into the latter requires representing [XPath](#) axes with native [OPC UA](#) methods. We propose a transformation that uses a combination of parameterized atomic calls with more sophisticated composite calls. For example, we transform the [XPath](#) axes *child* and *parents* into a call to [OPC UA](#)’s *Browse*, which is parameterized as forward and backward browsing, respectively. The resulting transformations are shown in Figure 5.7. In this mapping, all twelve [XPath](#) axes (white boxes) are represented using the two native [OPC UA](#) methods *Browse* and *Read* (black boxes). While axes such as *parent* or *child* are represented directly by parameterized (forward/backward) calls to the *Browse* method, others require more complex composition. The result of this fourth and final step is a representation of all relevant [XPath](#) axes in [OPC UA](#), so that dataspace participants can use the more expressive [XPath](#) query language in [OPC UA](#) environments.

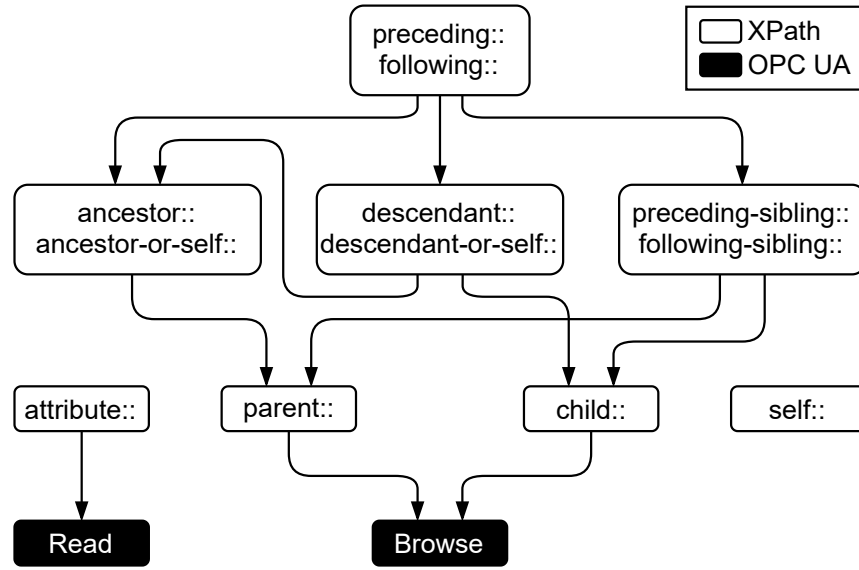


Figure 5.7: Our proposed mapping [The+22a; Koc21] of all twelve [XPath](#) axes to the two native [OPC UA](#) methods.

Technically speaking, our extended [OPC UA](#) client application supports the use of transformed [XPath](#) expressions to query data from an [OPC UA](#) server either online or offline from its export. While direct data extraction requires a connection to the server, all non-server-bound calculations can be performed independently of the [OPC UA](#) server based on the export (offline). The result is a patched [OPC UA](#) client that allows the addition of custom methods. We have implemented all twelve [XPath](#) axes as new [OPC UA](#) methods in our client to provide the full expressiveness of [XPath](#). See [figure 5.8](#) for an example transformation from an [XPath](#) query (top) for selecting the descendants of a specified node to executable code using our patched [OPC UA](#) client. This example returns the node identifiers of all descendants of the node with identifier 84. See [figure 5.7](#) for all supported [XPath](#) axes.

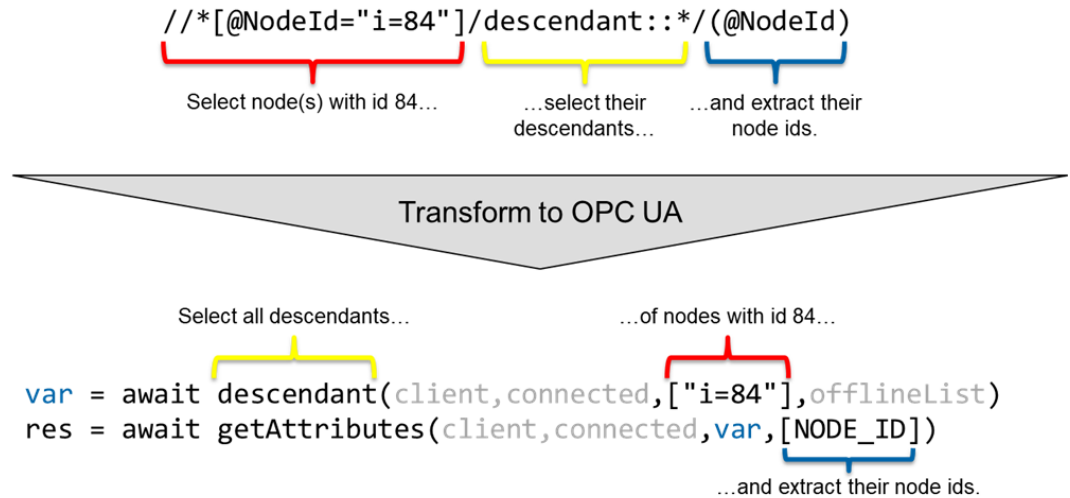


Figure 5.8: Example of transforming an [XPath](#) query into executable code for [OPC UA](#) [The+22a; Koc21].

These base directly on [OPC UA](#)’s native *Browse* and *Read* methods (see [IEC20]) or compose more complex combinations of them. As shown in [figure 5.7](#), the [XPath](#) axes are realized with *Browse* and the *attribute* axis with *Read*. While the black terms represent the [XPath](#) expressions, the white terms represent the native [OPC UA](#) service sets. The result is a patched *FreeOpcUa* Python client that provides convenient methods for all twelve [XPath](#) axes, allowing them to be executed in [OPC UA](#).

5.2.4 Evaluation

This section evaluates the effectiveness and practicality of the proposed approach for improving data accessibility from heterogeneous sources for dataspace participants. It covers all steps of our approach (see [figure 5.5](#)), including creating data exports from [OPC UA](#) servers, post-processing the exports, manually formulating [XPath](#) queries, and finally executing them on [OPC UA](#) servers. In the following, we first motivate the evaluation criteria and then present the evaluation results, structured according to the four steps of our approach.

Evaluation Criteria and Scenarios

The first critical aspect of the evaluation concerns the performance implications of the proposed approach. We evaluate the latency and throughput of data retrieval operations for both export creation and query execution. In addition, we evaluate the scalability of the proposed solution to handle increasing volumes of [OPC UA](#) data nodes. By quantifying the performance overhead and scalability limitations, we identify potential optimizations and caching strategies to enhance the efficiency of the transformation process.

Beyond technical considerations, we assess the practical implications of implementing the proposed approach in real-world dataspace environments. This evaluation includes analysis of factors such as the ease of integration, usability, and maintainability of the transformation solution. We obtain feedback from dataspace participants to examine the economic feasibility of deploying the transformation process at scale, considering factors such as manual effort and computational and network resource overhead.

Table 5.1: Evaluation scenarios representing different use cases for [OPC UA](#) access.

Scenario	Route	Hop Count	Latency (ms)	Bandwidth (Mbps)
UA client	Same machine	1	< 1	1000+
UA local	Local network	2	4	100
UA remote	Internet	11	37	20

We examine three different evaluation scenarios to represent common use cases from the manufacturing domain (see [section 2.3](#)), which differ in the number of network hops, latency, and bandwidth between the [OPC UA](#) clients and servers. The scenarios (see [table 5.1](#)) are: The **UA client** scenario represents direct access to a production machine (e.g. during maintenance) and runs both client and server on the same machine (1 hop, <1 ms, 1000+ Mbps). The **UA local** scenario represents access from a (secured) production network and runs both client and server on different machines on the same network (2 hops, 4 ms, 100 Mbps). The **UA remote** scenario represents remote access to a production machine over

the Internet, which may be necessary for global enterprises, remote production sites, or restrictions during the Covid-19 pandemic (11 hops, 37 ms, 20 Mbps). Note that the resource specifications are averages that demonstrate a typical setup and are subject to change.

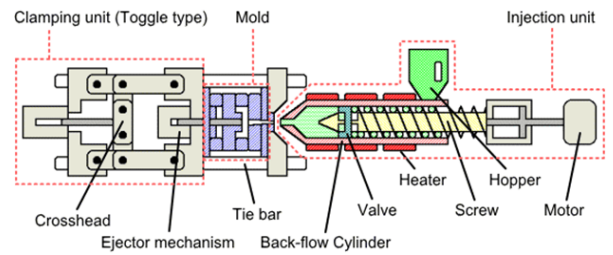
Evaluation Use Cases for Practical Demonstration

This evaluation consists of two use cases: A simulation environment for accurate, repeatable results, and a real-world use case for practical impact.

We implement the simulation environment with the publicly available *Prosys OPC UA Simulation Server*² as technical server implementation to maximize reproducibility. We run all experiments on version 5.0.4-284 of this simulation server and the *Prosys* software development kit version 4.4.2-1266 on a 64-bit Windows 10 Intel i7-6700HQ CPU @ 2.60GHz with 16 GB memory.



(a) Photo of an injection molding machine (published by the European Commission at <https://een.ec.europa.eu/partners/quality-plastic-manufacturing-complex-package-services-offered-hungarian-sme-frame>)



(b) Cross section of an injection molding machine (image copyright ©Polyplastics Co., Ltd.)

Figure 5.9: Photo of an injection molding machine (a) and a cross section of the machine (b).

We address practical impact by assessing our presented solution in an *injection molding* use case within the *Internet of Production* project (see section 2.3). In the injection molding of plastic parts, the plasticized material is injected under high pressure into a mold whose cavity is the negative of the part, yielding the final molded part after it solidifies [RRR00]. See figure 5.9 for an illustration of the machine (a) and its cross section (b). The machine heats material and injects it into a mold that forms the desired product. Our evaluation use case involves injection molding machines at the *Institute for Plastics Processing* of the *RWTH Aachen University*. Until now, engineers and data scientists have had to manually inspect each machine's embedded *OPC UA* server and determine the required data and metadata fields for extraction one by one (up to dozens or hundreds). See figure 5.2 for an example. We apply our solution to this manual process, enabling researchers to collect units, datatypes, and value ranges from sensor nodes using *XPath* queries. In this evaluation, the machine's *OPC UA* server contains 3343 nodes, and the query to be designed should extract 166 sets of nodes with metadata (966 values in total).

²<https://prosypoc.com/products/opc-ua-simulation-server/>

Evaluation Results

In the following, we present the evaluation results for all steps of our approach (see [figure 5.5](#)) using the above evaluation criteria, which specifically refer to execution time and manual effort. This includes experiment execution for the three scenarios (UA client, UA local, UA remote) for both use cases (simulation and real-world).

Data Export from an OPC UA Server The [OPC UA](#) demonstration server used in these experiments contains a total of 4329 nodes with their respective attributes, which we call *node sets*. Browsing the entire server and storing all node sets in a local representation took different amounts of time for the three evaluation scenarios introduced in [table 5.1](#). The results are shown in [figure 5.10](#) and their execution times scale with the network constraints. While the network constraints do not limit the local machine/network scenarios too much (23 and 77 seconds), they slow down the export over the Internet (UA remote) to about ten minutes. The extraction time of individual node sets (orange bars) scales linearly with the total number of nodes (blue bars). All of these execution times are considered acceptable for one-time extractions.

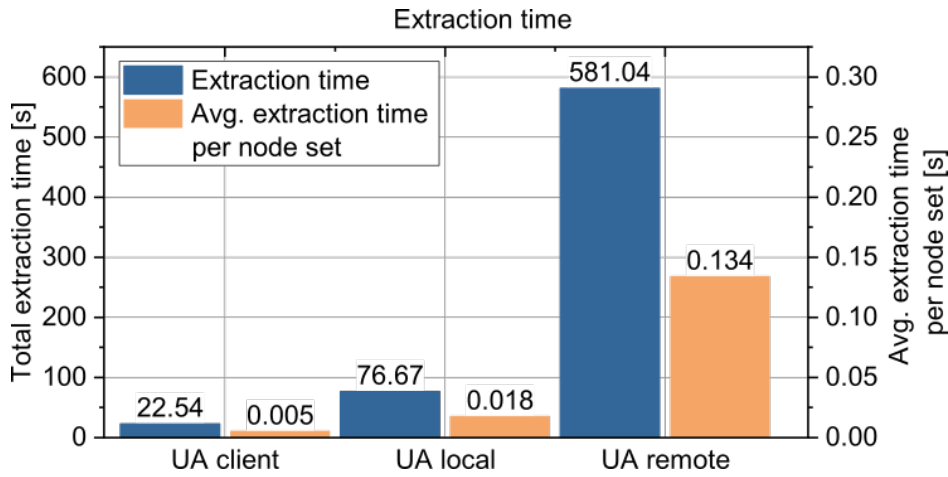


Figure 5.10: Execution times of the [OPC UA](#) export [[The+22a](#); [Koc21](#)] for the evaluation scenarios from [table 5.1](#).

Post-process the Export Post-processing the export generated from the [OPC UA](#) server was done locally on the client and includes to transform the export’s graph structure into an [XPath](#)-compatible [XML](#) tree (see [figure 5.6](#)). In this step, a breadth-first search is applied to the directed [OPC UA](#) graph representation, following all paths and duplicating nodes to remove any cycles.

The initial evaluation results of the post-processing are shown in [figure 5.11](#) on the left (a). The *number of loop iterations* on the x-axis denote the computing steps. *Paths in list* denotes the number of paths stored by the breadth-first search algorithm that it still needs to browse. *Paths found* accumulates the total number of (expanded) paths in the [XPath](#)-compatible output format. *Runtime* tracks the time required for computing the result. The results on the left (a) in [figure 5.11](#) revealed significant problems of feasibility: Even after more than 20 hours of local computation, about seven million paths were constructed and the *paths in list* still grew, indicating that search algorithm would not finish any time soon.

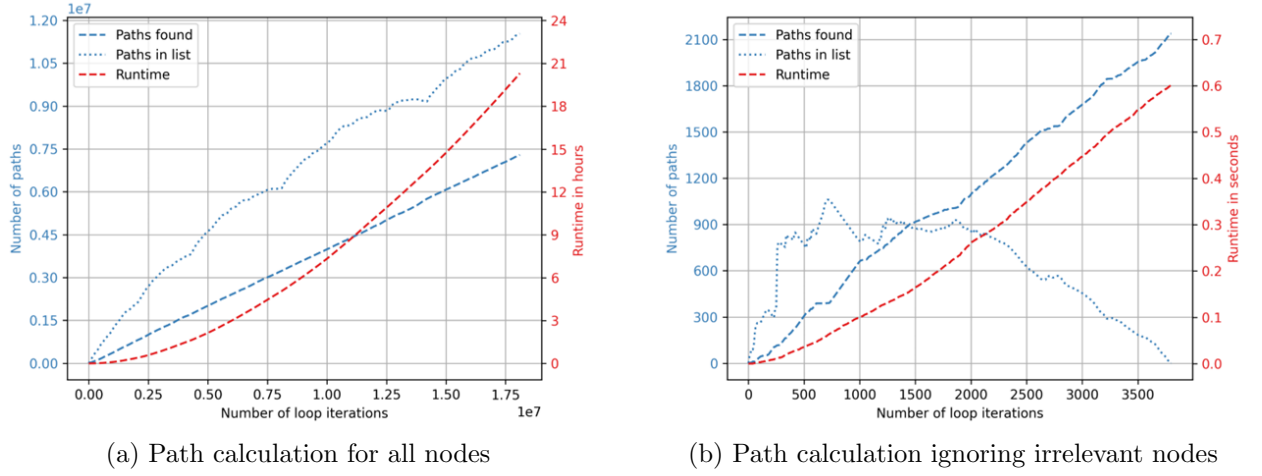


Figure 5.11: Post-processing evaluation results from an OPC UA export to XPath-compatible XML [The+22a; Koc21].

An investigation led to the insight that the used *Prosys* simulation server by default is not configured well for our evaluation, because it contains a massive amount of irrelevant nodes with numerous links between them.

We identified all irrelevant nodes and updated the client implementation to ignore these during the post-processing step. Note that this does not weaken the impact of our evaluation results, because this setup is special just for this OPC UA simulation server. The final results, after identifying and filtering out irrelevant nodes, are shown on the right (b) of figure 5.11. The *paths in list* grows and first as the search algorithm discovers new paths, and, from some point in time on, continuously shrink as these are being processed by the search algorithm.

The total execution time of the post-processing from the OPC UA export to XPath-compatible XML finally demonstrates a streamlined process with an improved runtime of only 0.6 seconds.

Formulate XPath Expressions Selecting the data and metadata to be included in subsequent data collection was a time-consuming manual task (see figure 5.2). This information of interest includes raw sensor values and associated metadata such as unit, scale, and precision. Our evaluation task was to set up a collection of all these node values including their metadata such as unit ranges expressed via `UnitRangeMin` and `UnitRangeMax`.

Dataspace participants in the *injection molding* use case took about 60–90 seconds to manually set up each node. Extrapolating this to the total of 166 nodes of interest results in a total of 2–3 hours of manual effort. Using our approach, dataspace participants instead formulated the XPath query shown in listing 5.1 only once to collect all this information simultaneously. Depending on their expertise, it took them 5–10 minutes to design this query. Our approach reduced the manual effort for this step from 2–3 hours to 5–10 minutes.

```
Object/Object[@NodeId="i=85"]//*[DataType="EUInformation"]/parent::*/child::*/((parent::*/
  @NodeId)|@NodeId|@DataType|@UnitID|@UnitRangeMin|@UnitRangeMax)
```

Listing 5.1: XPath expression to extract all sensor nodes and their metadata at once.

Execute XPath Expressions in OPC UA Assessing the execution times of *XPath* queries on an *OPC UA* server is the final part of our evaluation. We cover both use cases (simulation and real-world) in all three scenarios (*UA client*, *UA local*, *UA remote*). We also investigate a possible improvement by assuming that the structure of an *OPC UA* server does not change too often. This allows us to take advantage of local caches by introducing *offline* and *combined* variants of the scenarios that rely entirely on *XML* export or only read attributes (node values) from the live server.

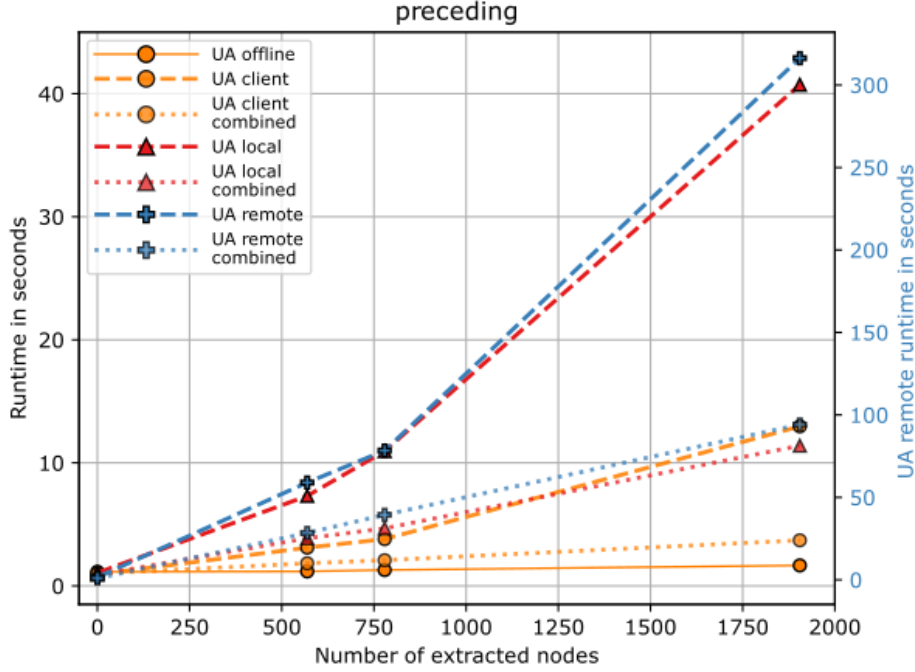


Figure 5.12: Average execution times for *preceding* on exemplary nodes in the different scenarios [The+22a; Koc21].

For the simulation use case, we assess the *preceding* function with *attribute* to include all twelve *XPath* axes (see figure 5.7) in this evaluation. Figure 5.12 shows the execution time of all scenarios for different numbers of extracted nodes, each as an average of five repeated runs. The offline variant takes about one second for extracting all nodes, and the runtime of all other variants scales linearly with the number of nodes. The *UA client* scenario took four seconds and the *UA local* scenario took twelve seconds, even with the maximum number of nodes extracted. The *UA remote* scenario has stricter network constraints and took up to several minutes. For all scenarios, the combined variants reduce the number of network calls and the total time by about 50 %. The local and remote execution times scale proportionally with network bandwidth and latency, and even the maximum time of 5 minutes is suitable for one-time executions. The other functions *descendant*, *ancestor*, and *following* (see figure 5.7) yield similar results.

For the real-world use case, we assess the *XPath* query designed by dataspace participants from the injection molding domain (see listing 5.1). The *OPC UA* server is a real asset in the injection molding machine (see figure 5.9). In this use case, the *OPC UA* server can be accessed by clients from a production network, which corresponds to the *UA local* scenario. Our solution automatically transforms, based on the export of this server, the formulated *XPath* query to the commands shown in listing 5.2. Running these commands with our

```

1 descendants = await descendantOrSelf(client,connected,['i=85'],offlineList)
2 attributeList = await getAttributes(client,connected,descendants,[1,14])
3 nodes = set()
4 for attribute in attributeList:
5     if attribute[1] == "EUInformation":
6         nodes.add(attribute[0])
7 siblings = await child(client,connected,await parent(client,connected,nodes,offlineList),
8     offlineList)
9 connected2 = nodeX[1]
10 attrsPar = await getAttributes(client,connected2,await parent(client,connected2,siblings,
11     offlineList),[1])
12 attrsRest = await getAttributes(client,connected2,siblings,[1,14,101,102,103])

```

Listing 5.2: For the [XPath](#) expression from [listing 5.1](#), our approach generates executable Python code for our patched [OPC UA](#) client.

patched [OPC UA](#) client extracts the same information from the [OPC UA](#) server that the [XPath](#) query would extract from its data export. These results are shown in [figure 5.13](#) as the average runtime over five repeated executions of the [OPC UA](#)-compliant code generated from the [XPath](#) expression of dataspace participants shown in [listing 5.1](#). *IKV* is the German acronym for *Institute for Plastics Processing*, which provides the use case. Direct execution of the [XPath](#) query took milliseconds for 166 node sets. The offline scenario took about half a second, and the online scenario (UA local, see [table 5.1](#)) took about 48 seconds. The combined variant of this scenario uses local caches and reduces the runtime to about 19 seconds. In summary, these measurements confirm the results from the simulation use case above and also prove the applicability of the cache-based *combined* variant in practice, saving an additional 61 % execution time. Similar results were also obtained in another evaluation in [[The+22b](#)] with six [OPC UA](#) servers from the foundry/castings domain.

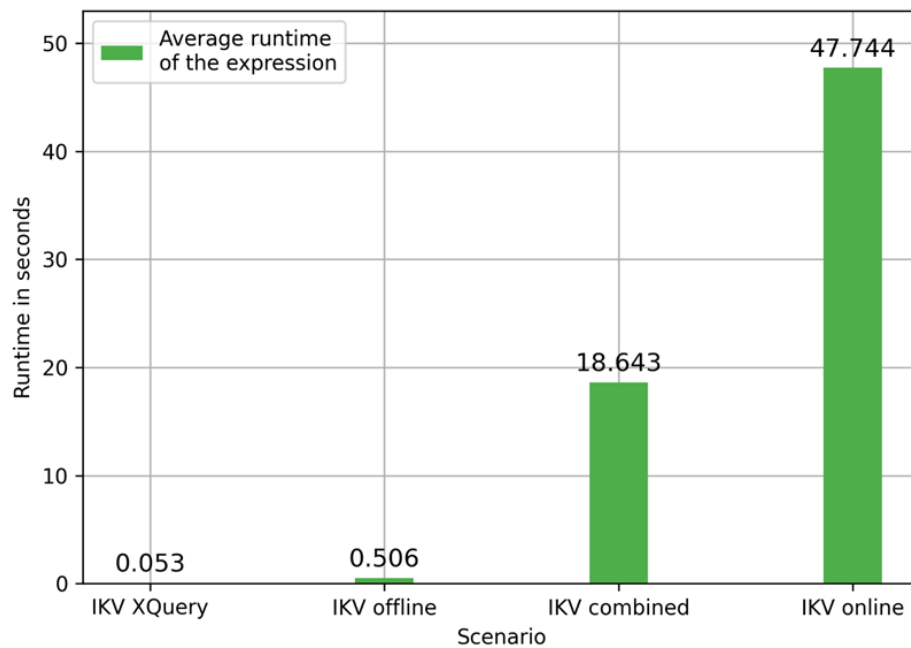


Figure 5.13: Average runtime of the [OPC UA](#)-compliant code in [listing 5.2](#) generated from the [XPath](#) expression of dataspace participants shown in [listing 5.1](#).

Evaluation Summary

Evaluation in representative scenarios, including a real-world injection molding use case, demonstrated full support for all [XPath](#) axes and resulted in promising execution times and significantly reduced manual effort. Exporting all information from an [OPC UA](#) server took about one minute within the same network or about ten minutes over the Internet. The post-processing export to [XPath](#)-compliant [XML](#) is performed offline and takes a negligible short time of less than one second. Formulating a complex [XPath](#) expression to query a desired set of nodes took dataspace participants less than ten minutes. Translating this into executable code to query an [OPC UA](#) server takes a negligible amount of time, a few milliseconds. Executing these commands in [OPC UA](#) on the same (production) network for a real-world use case took less than a minute. The total time of our approach is about 10 minutes of manual work and a few minutes of automatic processing, which is a significant improvement over the baseline, which required 2–3 hours of manual effort.

Contributions

In conclusion, this section has effectively addressed the challenge outlined in [requirement Data-R1](#) by proposing a solution to improve data accessibility from heterogeneous systems within dataspaces. By bridging the gap between low-level query interfaces and higher-level query languages, we have significantly improved the efficiency and flexibility of data retrieval and metadata collection processes.

The contributions of this section include:

- **Methodology for Query Language Transformation:** We have developed a comprehensive methodology for transforming queries between differently expressive query languages, facilitating seamless access to (meta)data across diverse sources.
- **Development of Tools and Utilities:** Our contributions extend to the creation of practical tools and utilities, including an [OPC UA](#) export tool, conversion mechanisms for [XPath](#)-compatible [XML](#) trees, and Python implementations of [OPC UA](#) methods, ensuring the accessibility and usability of our proposed approach.
- **Evaluation in Real-world Scenarios:** Through rigorous evaluation in laboratory and real-world conditions, we have demonstrated the effectiveness and performance of our methodology, validating its practical applicability and impact on data accessibility within dataspaces. Our approach reduced the manual effort from 2–3 hours to about 15 minutes in a real-world use case.

These contributions collectively advance the field of data accessibility within dataspaces, fostering collaboration, interoperability, and efficiency in complex decision-making processes. By providing open-source solutions and methodologies [[Koc+21](#)], we help dataspace participants overcome the challenges posed by heterogeneous systems and leverage the full potential of data in dataspaces.

5.3 Adaptive Data Load Management

This section addresses [requirement Data-R2](#), focusing on the imperative of controlled data collection within dataspace. In an era of rapidly expanding data availability, the challenge is not only to leverage the richness of data from diverse and heterogeneous sources, but also to ensure the accuracy and efficiency of the data collection processes. Data is exchanged in dataspace for a specific reason, and data sources (such as manufacturing machines) produce more data in both depth and breadth than we can offer in dataspace. Thus, a controlled and adaptive data collection is becoming an essential need, requiring careful control over the selection and acquisition of data while considering the resource constraints imposed by the data source systems and their environments.

The following motivation comes from the manufacturing domain (see [section 2.3](#)), but can be applied to many other domains. The massive connectivity of machines, sensors, and systems in general is supported by protocols such as *Message Queuing Telemetry Transport* [SH99] and [OPC UA](#) [HSK08], which are standardized and openly available, allowing flexible data acquisition independent of vendor specifications. Production machines typically provide numerous nodes via [OPC UA](#) that expose machine values to connected clients. A major problem with traditional data acquisition methods is that they are set up a priori and remain constant throughout the data acquisition process. Integrating more and more production machines leads to an uncontrollable increase of network traffic, because detailed analysis of observed processes requires a fine granularity and therefore a high frequency and precision of collected data. A higher collection frequency (linearly) increases the data load and quickly reaches the limits of available network and processing resources. However, high-frequency data is rarely needed for an entire process cycle, but rather for short periods. This means that even though high-frequency values are of minor importance in some process phases, they are recorded over a long period of time. To keep the data manageable for use in dataspace, an optimal, adaptive trade-off between bandwidth utilization and information transferred is required. This raises the need for methods that adaptively adjust data acquisition rates based on data source specifications (e.g., observed production cycles). Incorporating domain knowledge can help optimize such data collection strategies. In the remainder of this section, we propose adaptive data load management that uses domain knowledge to offer dataspace participants the right data at the right time.

5.3.1 Motivating Domain Example

This section presents a motivating example from the domain of High Pressure Die Casting ([HPDC](#)) within the manufacturing sector. [HPDC](#) processes are critical to the production of complex light metal components, such as automotive structural parts or powertrain components [Lip+20]. This manufacturing process operates in a discontinuous, iterative fashion, typically producing a casting within 40–60 seconds, depending on the specific machine and tool configuration. In the case of the cold chamber variant of [HPDC](#), molten metal is injected into a cavity using a plunger system, then solidified under high pressure and ejected from the machine.

An [HPDC](#) production cell includes not only the [HPDC](#) machine (see [figure 5.14](#)), but also additional auxiliary systems, which are crucial for enabling a highly automated production system that relies on a robust information network [Rix+16]. A standard machine includes the shot end, which is responsible for injecting metal into the cavity, and the locking unit,



Figure 5.14: *Frech K-Series* cold chamber **HPDC** machine (image copyright ©Oskar Frech GmbH + Co. KG).

which keeps the mold closed during injection. Auxiliary systems facilitate automatic part production, including a molten metal holding furnace, a metering unit for metal transfer, several thermal regulators for mold cooling, a vacuum system for cavity evacuation prior to injection, a spray head for mold surface treatment after ejection, and a plunger lubrication system. These auxiliary systems operate at different stages of the overall cycle as required.

While most sensors provide continuous data over the entire cycle, the need to maintain a constant sampling rate regardless of the machine's sub-cycle status limits data acquisition (see [figure 5.15](#)). This figure shows the **HPDC** machine from our use case, with 13 temperature sensors highlighted with red arrows as a selection of the many sensors on the machine. During *dosing*, the bottom right ones are most relevant for analysis, but during *cooling*, the top ones in the mold are most relevant instead. The duration of the entire cycle and its phases varies due to process control adjustments or production interruptions caused by unstable process conditions. In addition, different sub-cycles require different sampling rates to accurately capture the relevant sensor information. Therefore, it is beneficial to detect transitions between sub-cycles and adjust sampling rates based on machine status. For example, the plunger injection phase consists of a slow shot that typically lasts 750–1500 milliseconds and a fast shot that completes within 25–100 milliseconds. The programmable logic controller of real-time controlled cold chamber **HPDC** machines aims to reproduce programmed process parameters with minimal deviation. However, various factors such as machine wear, operator, or environment can introduce variations into production cycles and affect process data. By employing information density appropriate to the machine state, domain experts or mathematical models can effectively identify the root cause of process instability. Achieving this requires a shift from constant sampling rates to a more sophisticated, data-driven approach that accounts for sub-cycle phases. This research focus also tackles the computer science research area of *data streams* and *edge computing*.

Even with a proposed solution, ensuring usability for dataspace participants remains critical for practical adoption. The convergence of digital and physical worlds, especially in manufacturing, is challenging because engineers typically specialize in either software or metallurgy.

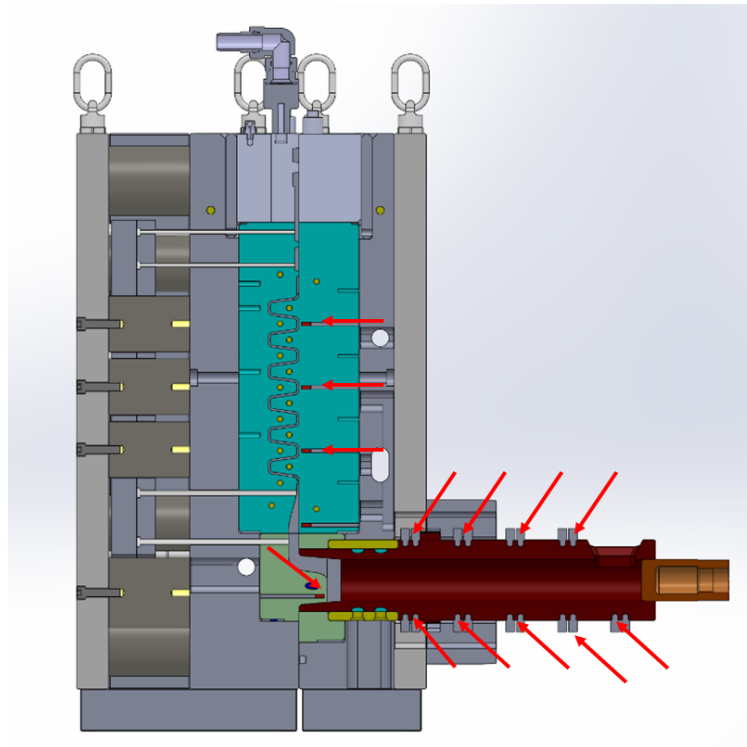


Figure 5.15: The HPDC machine from our use case has many sensors, including 13 temperature sensors [Lip+20].

Not all software integration challenges require a computer science background, as many engineers are familiar with using proprietary software to control machines and collect data. Laboratory measurement systems offer a high degree of configurability, with accompanying software providing user-friendly interfaces for connecting logic blocks. This familiarity is preferred by engineers and saves them from extensive coding. Open source visual programming tools offer a promising solution, simplifying data processing tasks for domain experts without advanced programming skills. Modern production machines often use open, standardized protocols such as OPC UA, which can be used in visual programming tools to effortlessly design data collection for these machines. Therefore, the transition from custom OPC UA client-based systems to visual programming tools is essential for wider adoption.

5.3.2 Background and Related Work

This related work sections extends the state of the art for OPC UA data collection we presented in [Lip+20] with the completely new aspect of usability for domain experts. First of all, we revisit both the environment and the requirements for data collection again, followed by a section dedicated to investigations on visual programming environments in order to increase the usability for users.

Data Load Infrastructure

The traditional extract-transform-load process and the more modern extract-load-transform process are shown in figure 5.16 for the technical context of this section. Solutions with relational data models, such as data warehouses, first transform the extracted data into

their intended schema and then load it [Qui03; QHV16; VS09]. In contrast, data lakes and other NoSQL systems make data available quickly by first loading it and then later transforming it in-place (extract-load-transform). This is also a principle for dataspace, which extract the data but do not load it a priori into a central location (see section 2.1.2). However, this section focuses solely on the extraction phase and is therefore independent of the chosen methodology, since both start with extraction [Lip+20]. Therefore, we will focus on related work on the extraction phase, particularly as applied to OPC UA, and skip the not directly related load and transform steps in this section.

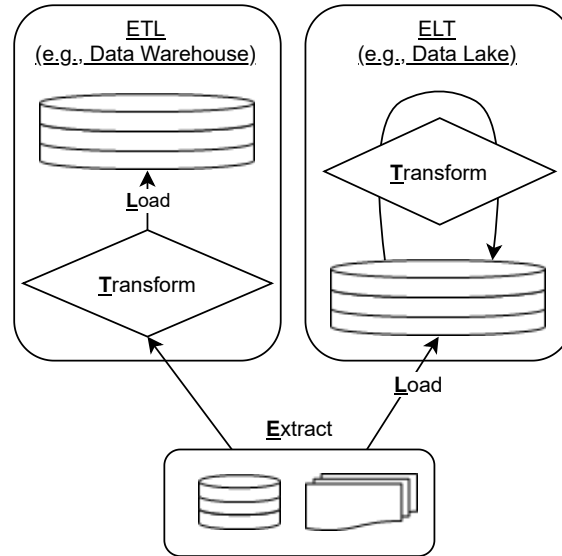


Figure 5.16: The common data integration processes extract-transform-load and extract-load-transform [Lip+20].

Data Collection via OPC UA

Classical data collection via OPC UA [HSK08] (see also section 5.2) uses either periodic polling or subscriptions where changes are pushed to the client on demand. With polling, the client sets a fixed time interval and contacts the server at each interval to read specified data. However, this approach can be inefficient due to potential missed value changes if the interval is too long, or excessive data collection if the interval is too short. In contrast, OPC UA subscriptions provide an advanced solution as described in OPC UA Part 1 [OPC17a]. Here, the client registers subscriptions on nodes of interest to the server, which monitors them. Value changes that exceed a threshold trigger updates to the client, thus addressing the drawbacks of polling. This functionality, called filtering, was introduced in OPC UA Part 4 [OPC17b]. OPC UA Part 14 [OPC18] goes even further and introduces *PubSub*, a more sophisticated approach where any publisher can distribute data to interested subscribers via a central broker. Unfortunately, the PubSub features are not yet implemented on many production machines, including our use case. However, given its growing adoption in the industry, future research could explore its potential once it is more widely implemented. Note that while the PubSub mechanism in OPC UA controls when updates are propagated to clients, it does not support adaptive changes. Clients are responsible for monitoring the state of the production process and updating subscription rules accordingly, limiting its potential for adaptive data load optimization.

Usability and Involvement of Non-Experts

With the increasing availability and complexity of data (see [section 5.2](#)), there is a growing need for high usability solutions that can be used by non-experts. Easy-to-use, low-code programming environments, especially visual programming environments, are a solution to this demand. Visual tools can help bridge the gap between software technologies and dataspace participants in various domains such as manufacturing. This section briefly evaluates the state of the art for visual programming environments needed to achieve broad acceptance and application of adaptive data load management. Dataflow programming began as a research topic in the 1970s, and Johnston et al. in [\[JHM04\]](#) reviewed its evolution and subsequently discussed the trend toward visual programming languages. Note that this paper appeared in 2004, and several approaches and implementations have been developed since then. The popularity of visual programming among people other than pure software developers remains high and is likely to increase as various participants in dataspace or the Internet of Production are increasingly tasked with data management responsibilities. [Table 5.2](#) summarizes our following evaluation of frameworks and tools from this area.

Table 5.2: Overview of frameworks and tools for visual programming [\[Lip+21c\]](#).

n8n.io [n8n19]	Workflow automation tool designed for connecting APIs through an intuitive web interface.
Microsoft Power Automate [Mic16]	Proprietary tool for automating rule-based business processes and office tasks.
OPC Router [inr03]	Middle-software facilitating connections between databases and sources in industrial automation, promoting Industry 4.0.
Apache StreamPipes [The18]	Java-based project focusing on real-time data collection in Internet of Things systems, integrating with industry standards like OPC UA .
Node-RED [Nic13]	Browser-based visual programming environment ideal for Internet of Things applications, supporting easy data flow modeling with a wide range of available nodes.

The open source workflow automation tool *n8n.io* (nodemation) [\[n8n19\]](#), developed by n8n GmbH in 2019, aims to provide an easy-to-use and highly customizable means of integrating [APIs](#) through a user-friendly web interface. Written in Typescript, it provides seamless connectivity. Originally introduced in October 2016 as *Microsoft Flow*, *Microsoft Power Automate* underwent a name change in November 2019 [\[Mic16\]](#). This proprietary tool helps users automate rule-based business and office processes. *OPC Router*, developed and marketed by inray Industriesoftware GmbH, serves as middleware that facilitates connectivity between databases and various sources, thereby advancing the concepts of the Internet of Things and Industry 4.0 [\[inr03\]](#). Its visual interface enables the management of data flow between components such as [OPC UA](#), databases, manufacturing execution systems, and supervisory control and data acquisition systems. The Java-based *Apache SteamPipes* project, initiated by the FZI Forschungszentrum Informatik in Karlsruhe, Germany, operates as a project under the Apache Software Foundation [\[The18\]](#) since 2020. This project focuses on real-time data acquisition within (Industrial) Internet of Things systems and is designed to seamlessly integrate with industry standards such as [OPC UA](#), as well as popular Apache ecosystem tools such as *Apache Kafka* and *Apache PLC4x*. Originally developed by IBM and

now maintained by the *OpenJS* Foundation, *Node-RED* [Nic13] is a browser-based visual programming environment. It facilitates data processing through drag-and-drop functionality, with nodes representing data sources, processing units, and sinks. Node-RED's flow-based architecture enables seamless data flow through interconnected nodes, allowing users to efficiently process different types of data. Its extensive library provides a wealth of predefined nodes and complete flows, with additional user-generated nodes available through the Node-RED Library. Node-RED has been used in numerous (Industrial) Internet of Things scenarios. Toc et al. [TK18] used Node-RED-driven middleware to develop a Modbus-**OPC UA** wrapper for industry applications. Newman et al. [New+19] used Node-RED on an embedded Linux platform to monitor **OPC UA** devices and analyze machine health metrics, while Sosa et al. [SKR18] investigated offloading performance (transfer of resource intensive computational tasks to a separate processor) on Raspberry Pis running Node-RED.

5.3.3 Approach

Our approach to facilitating controlled and adaptive data collection within dataspace is as follows. We propose a model that divides data generation into phases and a method that allows adaptive data collection depending on these phases. This includes modeling transitions, data collection rules, and start/end conditions of phases. We then show how this approach can be integrated into existing data collection pipelines to make it available to a wide range of dataspace participants.

[Listing 5.3](#) shows an extract of our proposed data model for dividing data collection processes into phases. A *phase* in this context represents phases of any process that generates data, such as a real-world manufacturing process. Phases may run sequentially or even in parallel. The level of detail of the phases is flexible and depends on the scope of the dataspace participants. We define a data collection (process) phase as a collection of

- a name and/or description
- a list of data of interest (what)
- rules for collecting the data (how),
- one or more start conditions, and
- one or more end conditions.

For example, a manufacturing process may start with one or more phases, and may be cyclic if the start condition for the first phase includes the end of the last phase. This model allows dataspace participants to divide their (real-world) data generation process into phases. For each phase, they can define exactly what data to collect and how to collect it. Finally, the model allows them to specify the transitions between phases in a data-driven manner, allowing for changing data collection strategies.

We enable the practical use of the introduced phase model by introducing the concept of deploying either a physical or logical edge device in direct proximity to the data source, such as a manufacturing machine (see [figure 5.17](#)). This device serves a critical role in dynamically adapting data collection strategies based on the evolving requirements outlined in the phase model from [listing 5.3](#) [Lip+20] By leveraging this adaptive capability, our proposal increases the efficiency and flexibility of data extraction processes, ensuring that

```

1 "phases": [
2   {
3     "name": "Phase 1",
4     "whatToCollect": [...],
5     "howToCollect": [...],
6     "startConditions": [...],
7     "endConditions": [...]
8   },
9   [...]
10 ]

```

Listing 5.3: Our proposed model for dividing data collection into process phases [Lip+20]. Phases can be interpreted as data collection strategies that can be changed adaptively.

relevant data is effectively captured at the source. This makes the right data available to dataspace participants at the right time.

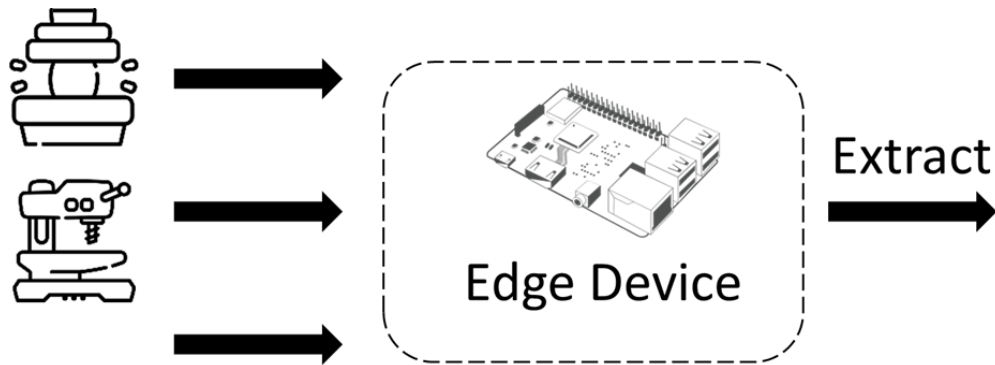


Figure 5.17: Our proposed extension to the *extract* data collection stage (see figure 5.16) is based on the phase model from listing 5.3 [Lip+20].

We overcome the final challenge of such sophisticated solutions being too technical and complex for many dataspace participants by simplifying their setup. This is important because their deep knowledge of the processes behind data sources can be a powerful advantage in optimizing data collection. Both easy-to-use tool support and intuitive user interfaces are needed to achieve broad acceptance among dataspace participants. We propose a user-friendly integration wizard that simplifies access to our proposal for defining, deploying, and maintaining adaptive data integration pipelines. This completes the above phase model for data acquisition with a step that integrates it into common, user-friendly workflows.

5.3.4 Results

This section presents our results for an adaptive and easy-to-integrate data collection for dataspaces. We present three results, namely an implementation of the proposed phase model (see listing 5.3) into an OPC UA client, an integration of this solution to edge devices in close proximity of data sources, and an embedding of its configuration into user-friendly visual programming environments. We present several results in the form of specific examples

from a horizontal cold chamber HPDC machine (see section 5.3.1) to clearly show all the details that would be left out in a generic presentation.

Implementing Adaptive Data Load We enable adaptive data collection for a data source as follows. An implementation of the proposed phase model (see listing 5.3) consists of two parts, a serialization of a model instance and an extended client software capable of performing the required steps. This includes interpreting the adaptive data loading configuration encoded in the phase model instance, changing the data collection settings in the data source as needed, and collecting the right data at the right time as specified.

Our implementation addresses data collection from manufacturing processes (see section 2.3) managed by OPC UA (see section 5.2.1). It includes a JSON configuration file that serializes the proposed phase model and an extended OPC UA Java client that provides all the necessary functionality, such as interpreting the JSON phase model and adapting the data collection from an OPC UA server as specified.

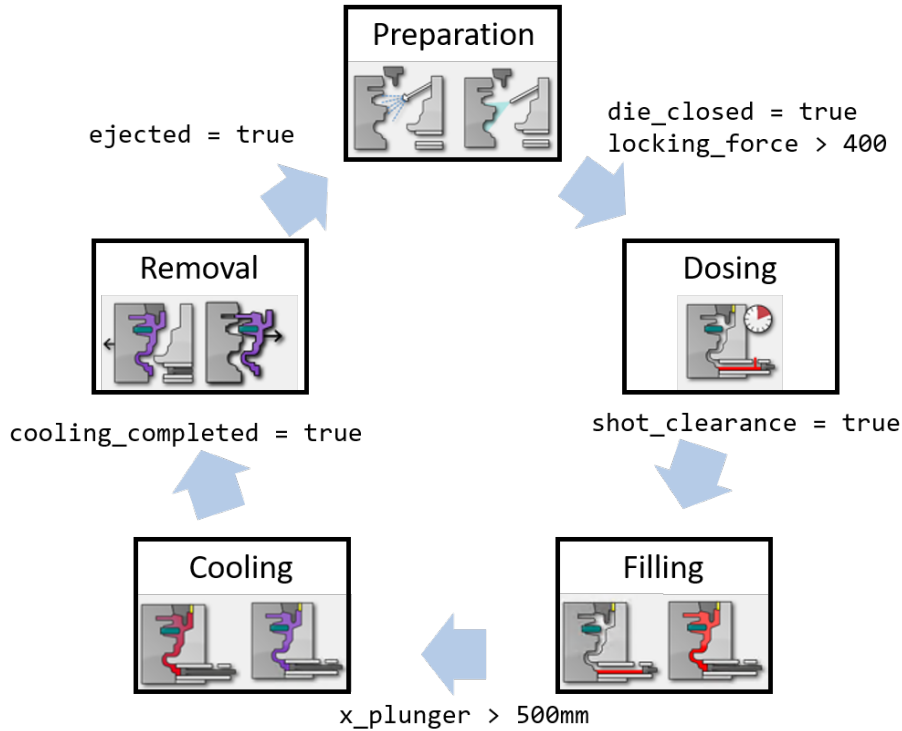


Figure 5.18: Our analysis of a sample HPDC process (see section 5.3.1) yields phases and transition conditions [Lip+20]. Image copyright ©Magmasoft 5.4 [MAG19].

Figure 5.18 illustrates five process phases and transition conditions of an HPDC production cycle (see section 5.3.1 for details). The HPDC production cycle begins with the *preparation* phase, during which mainly the injection unit and the ejection system are active until the preparation is completed. Finally, the die is closed and a locking force is applied to keep it shut. Once the die is securely closed with sufficient force of 400 metric tons, indicated by `die_closed = true` and `locking_force > 400`, the process is ready to move to the *dosing* stage. In this stage, the material is introduced into the die and after ensuring that the shot is prepared (`shot_clearance = true`), the process transitions to the *filling* stage. Here the material fills the mold and once the plunger is on its terminal position (`x_plunger > 500mm`), indicating that the mold is sufficiently filled, the process moves to the *cooling*

stage. During cooling, the material in the mold is allowed to cool, and the transition to the *removal* stage occurs after a specified period of time, when cooling is assumed to be complete (`cooling_completed = true`). Finally, in the removal stage, the finished product is ejected from the mold. The ejection of the product (`ejected = true`) signifies the completion of the production cycle and readiness for the next cycle to begin.

Following our proposed phase model, this process description translates to the JSON serialization shown in [listing 5.4](#). It contains the five phases indicated by the `name` keys and information for each of them. Since OPC UA is used to collect data in the HPDC use case, the `whatToCollect` key lists all OPC UA nodes that contain data of interest during a phase. The identifiers of these nodes are omitted for readability. Different interval values for poll-based data collection are determined by domain experts (`howToCollect`). While some phases may require a data collection frequency of 4 seconds for selected temperature sensors [Lip+20], other phases may require a significantly reduced interval of 200 milliseconds. See below for subscription-based data collection strategies in addition to the interval-based polling used here in all five phases. Transitions between phases are modeled using start and end conditions. This HPDC process is sequential, so all phases start when the previous phase ends (e.g., `PHASE_END:"Preparation"`). The phases end according to the defined transition conditions illustrated in [figure 5.18](#). The process is cyclic because the first *preparation* phase is modeled to start after the last *removal* phase ends (`PHASE_END:"Removal"`). Note that other combinations of start and end conditions allow modeling phases that overlap or run in parallel. Possible extensions using more complex rules could represent advanced conditions as such in process diagrams or Petri nets [Pet77]) and trigger the start of one phase when several others have ended [Lip+20].

The above example from the HPDC use case uses only interval-based data collection, which is indicated by `interval:200` in the `howToCollect` fields. Our enhanced OPC UA client supports multiple data collection strategies [Lip+20], including:

- interval-based (polling): Read values based on a fixed time interval, such as 300 milliseconds,
- subscription-based (absolute): The current value is fixed and data collection is triggered whenever the current value exceeds the specified absolute limits, e.g. when the temperature changes by more than 7 degrees Celsius,
- subscription-based (relative): Same as absolute, but changes are calculated relative instead of absolute.

Start and end conditions in our implementation cover observing values (boolean, integer, double, etc.) for both polling and subscriptions. The joint result is a serialization of the proposed phase model and an enhanced software client capable of processing it, thus enabling adaptive data collection for a data source.

Network Integration We enable practical use and scale-out of the implemented solution by deploying either a physical or logical edge device in close proximity to the data source (see [figure 5.17](#)). [Figure 5.19](#) shows the three-zone network architecture of our running use case example of a cold chamber HPDC machine: The *production network* resides in the factory floor itself and is securely connected to the *office network* through a firewall. The office network allows user access from workstations and has an uplink to a self-managed *cloud* with a Kubernetes [Clo14] cluster as possible connection to dataspace.

```

1 "phases":[
2 {
3     "name":           "Preparation",
4     "whatToCollect":  [...],
5     "howToCollect":   {interval:4000},
6     "startConditions": [PHASE_END:"Removal"],
7     "endConditions":  [die_closed=true, locking_force>400]
8 }, {
9     "name":           "Dosing",
10    "whatToCollect":   [...],
11    "howToCollect":    {interval:200},
12    "startConditions": [PHASE_END:"Preparation"],
13    "endConditions":   [shot_clearance=true]
14 }, {
15    "name":           "Filling",
16    "whatToCollect":   [...],
17    "howToCollect":    {interval:200},
18    "startConditions": [PHASE_END:"Dosing"],
19    "endConditions":   [x_plunger>500]
20 }, {
21    "name":           "Cooling",
22    "whatToCollect":   [...],
23    "howToCollect":    {interval:200},
24    "startConditions": [PHASE_END:"Filling"],
25    "endConditions":   [cooling_completed=true]
26 }, {
27    "name":           "Removal",
28    "whatToCollect":   [...],
29    "howToCollect":    {interval:4000},
30    "startConditions": [PHASE_END:"Cooling"],
31    "endConditions":   [ejected=true]
32 }]

```

Listing 5.4: Example JSON config file for the HPDC use case based on the phases from figure 5.18 [Lip+21c].

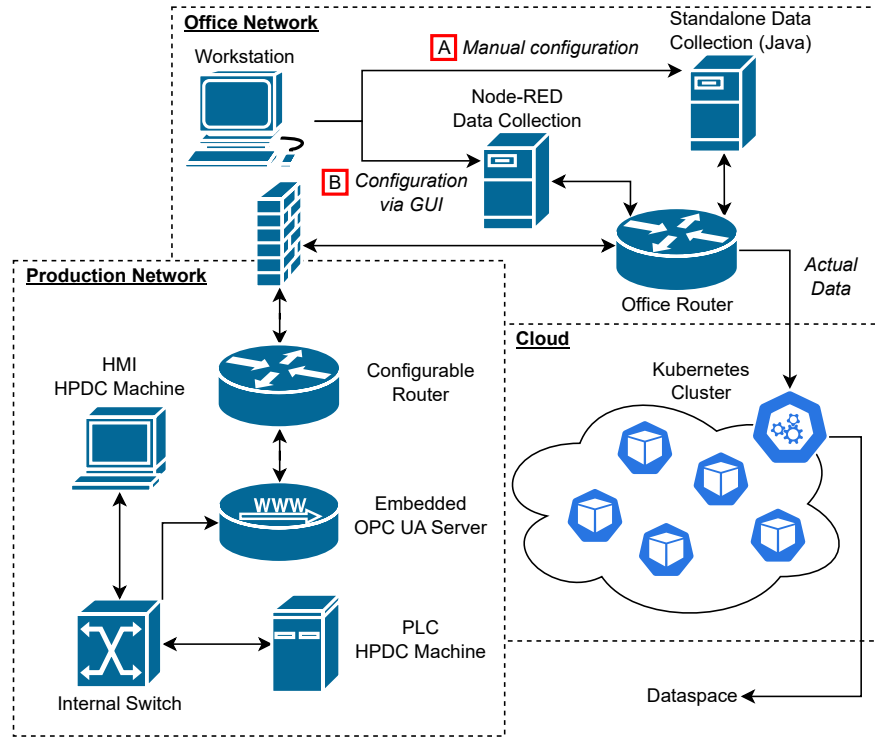


Figure 5.19: Three-zone network architecture for the HPDC machine use case [Lip+21c].

The production network on the lower left of figure 5.19 includes the HPDC machine itself and its components. Its programmable logic controller holds all the data values for this machine, which can be accessed in two ways. First, a human-machine interaction panel provides maintenance access and experimental data exploration for preconfigured values or setups. Second, an embedded OPC UA server can be freely configured to provide data access to the underlying programmable logic controller and additional sensors via an Ethernet port. The OPC UA server supports data access over a signed and encrypted connection that can be authenticated with username and password or signed X.509 certificates. A router connects this setup to the local production network. A firewall secures the connection to and from the office network, which includes employee workstations and a more powerful router that manages this office network. For several reasons, including security, flexibility, performance, and accessibility, this network is best suited to implement our solution. We use *Raspberry Pi 4* units as physical edge devices, equipped with a Broadcom quad-core 64-bit SoC @ 1.5GHz, 4GB SDRAM, Ethernet connectivity, and a 32 GB SD card. These edge devices provide two options for integrating our adaptive data collection approach, highlighted by (A) and (B) at the top of figure 5.19. The first option is a manual configuration of an edge device (A) using our previous contribution, a JSON file with the phase model, and an extended OPC UA client capable of performing the adaptive data collection defined in it. Note that this OPC UA data collection is performed via a connection to the OPC UA server embedded in the HPDC machine (bottom left). The second option (B) is a more user-friendly, visual configuration of data collection for dataspace participants using Node-RED, which we describe in detail below. Both options are set up and maintained by a dataspace participant from a workstation. In the final network zone, the Kubernetes cluster in a self-managed cloud, all data is properly maintained for use by various dataspace applications. The result is a practical use of our proposed solution in the network architectures of real-world use cases, enabling adaptive data load management for dataspaces.

Visual Configuration for Dataspace Participants To achieve widespread adoption among dataspace participants, we implement the final step of our approach, easy-to-use tool support and intuitive user interfaces. We integrate our solution into visual programming environments to leverage their benefits for dataspace participants. Among the visual programming tools analyzed (see [section 5.3.2](#)), Node-RED is the most suitable due to its mature state of development, detailed documentation, extensive community, and ease of use. We therefore demonstrate a user-friendly practical application of our approach by providing a Node-RED embedding for it, and finally provide an exemplary application to our running example use case (see [section 5.3.1](#)).

We implement the functionality of phase-based data collection as a custom node within the Node-RED framework. Each node within this framework represents a distinct process phase, serving as a forwarding mechanism for incoming messages based on predefined rules. These rules are configurable by the domain expert via an input interface. The expert specifies the activation criteria for the node, as well as the commencement and cessation of data transmission, including the frequency of transmission. Notably, the node does not dictate the content of the transmitted data; rather, it merely forwards incoming messages, thereby leaving the content determination to the messages injected into the node. [Figure 5.20](#) illustrates the aforementioned input masks, where an arbitrary number of rules can be defined to specify the conditions for starting and ending phases. In this example (a) the phase starts when the previous phase sends an end signal and the incoming message’s `payload` attribute is greater than zero. It ends (b) when the variable `myVar` equals `true` and the incoming message’s `payload` is less than or equal to zero. Consistent with the definition of process phases, dataspace participants can specify whether data should be collected at fixed intervals (polling) or on a subscription basis with absolute or relative trigger variants (see [figure 5.21](#)). The example shown in this figure uses a fixed interval of 4 seconds.

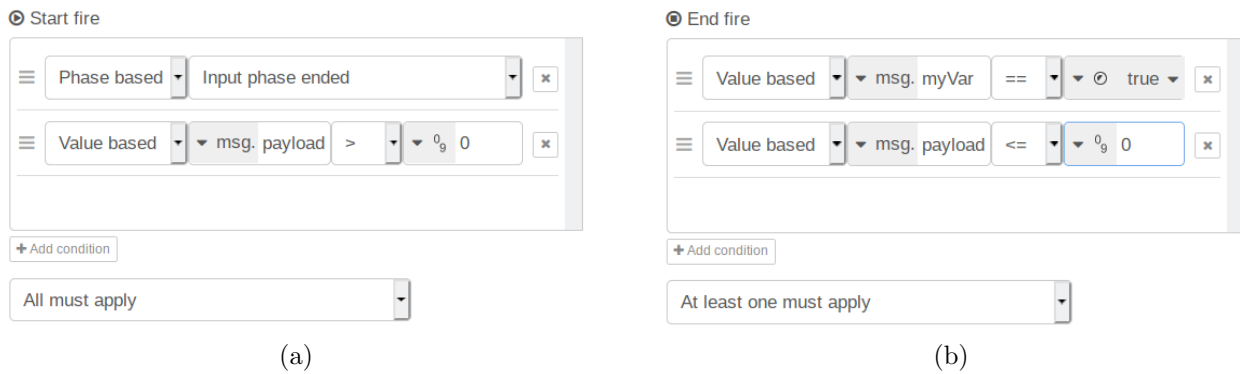


Figure 5.20: Window for configuring the start and end conditions of a process phase in Node-RED [\[Lip+21c\]](#).

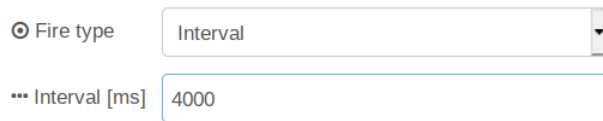


Figure 5.21: Configuration window of the data collection strategy [\[Lip+21c\]](#).

Each Node-RED node, representing a process phase, is equipped with one input and two output connectors. Messages received from the previous node are processed according to predefined rules and then sent to the first output connector. The second output connector is used to transmit specific messages, signaling the start or end of the node's phase, or prompting a data source node to change its subscription pattern. Multiple nodes can be interconnected, as shown in [figure 5.22](#), or organized into complex networks. In this figure, two nodes named *Phase 1* and *Phase 2* represent phases from the proposed phase model. The nodes route incoming messages from data sources according to predefined rules. The activity of Phase 2 can be triggered based on the behavior of Phase 1 or its data source *Source 2*. In general, the activation of one node can consequently trigger the activation of another node by sending start and end signals through its second output port.

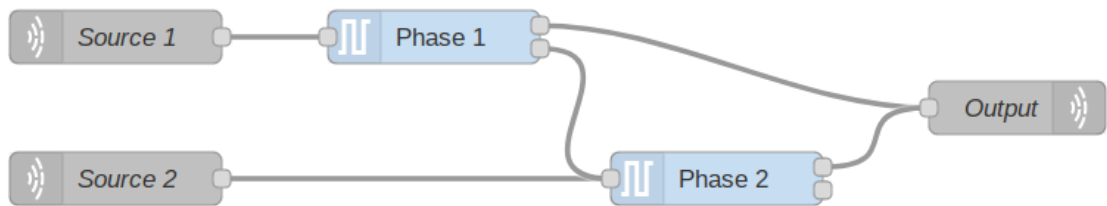


Figure 5.22: Example Node-RED setup with multiple nodes and data sources [[Lip+21c](#)].

Repeating this approach in Node-RED for all designated phases yields a complete configuration of the underlying phase model and thus enables adaptive data collection for dataspaces. The *source* nodes are implemented with data connectors to the desired data sources, and the *output* nodes to desired data storage for making them available to dataspaces. We in the following demonstrate this step and the entire workflow by implementing it for the running [HPDC](#) use case (see [section 5.3.1](#)).

Three elements are required to set up adaptive data collection from the [HPDC](#) use case in Node-RED: An [OPC UA](#) client as a data source, the Node-RED process phase node introduced above, and a data sink of choice. As data sources in Node-RED, publicly available nodes are available that implement the functionality of [OPC UA](#) clients³. These nodes connect and subscribe to the [OPC UA](#) server of the [HPDC](#) machine (see [figure 5.19](#)). They generate messages containing machine data and inject these messages into the Node-RED flow. Control messages injected into the Node-RED [OPC UA](#) nodes can modify which [OPC UA](#) nodes to subscribe to or which data collection strategy (polling vs. subscriptions) to use.

[Figure 5.23](#) presents an implementation of all five [HPDC](#) phases preparation, dosing, filling, cooling and removal in Node-RED. In this figure, the exact definition of data connections to [OPC UA](#) sources and outputs is omitted for clarity. The two small gray arrows indicate *Link in/out* nodes that are connected to make the process cyclic. Using and configuring the generic phase node (see [figures 5.20](#) and [5.21](#)) for each of these process phases and connecting them in a sequential manner completes their representation in Node-RED. Messages (data collection) generated by [OPC UA](#) nodes are injected into the corresponding process phase node, where they are handled depending on the activity of the phase.

³<https://flows.nodered.org/node/node-red-contrib-opcua>

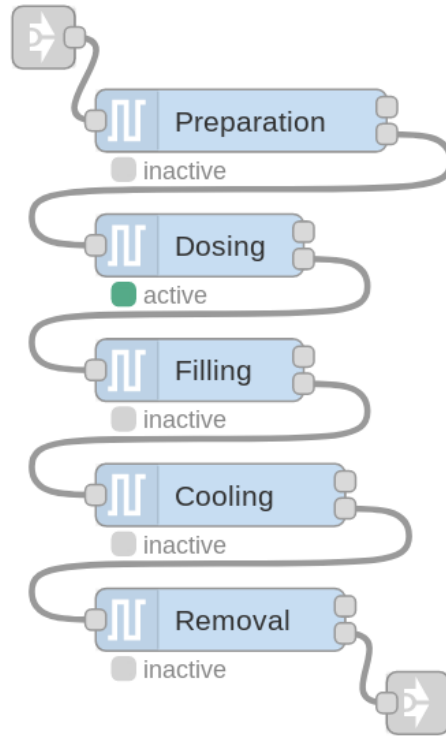


Figure 5.23: Overview of the HPDC process modeled as a Node-RED flow [Lip+21c].

As the third component of the sample Node-RED implementation, the data sink, we use an available node for the *Advanced Message Queuing Protocol*⁴. This protocol is an open standard for passing data messages between applications and supports Kubernetes, the data storage infrastructure of the running HPDC use case. This node is connected to all five phase nodes and centralizes the sending of their messages to the Kubernetes cluster, from which the data can be used in dataspace.

The complete Node-RED flow consisting of sources, phase nodes and sinks is deployed to an installation of Node-RED on an edge device, as indicated by “(B) Node-RED data collection” in the office network in figure 5.19. The flow can be maintained by dataspace participants through a graphical Node-RED Web interface from anywhere within the office network, without any additional network configuration or programming efforts. The joint result is a user-friendly solution using the Node-RED visual programming environment to define, deploy, and maintain our solution for adaptive data collection.

5.3.5 Evaluation

In this section, we evaluate the effectiveness and performance of the proposed adaptive data load management system. First, we define the metrics used to assess the performance of the system.

The evaluation is primarily based on the *number of data points* collected and transmitted, which serves as a basic indicator of the amount of data processed and transmitted for dataspace data collection. This single metric serves as a general indicator that encompasses

⁴<https://flows.nodered.org/node/node-red-contrib-amqp>

various aspects of system performance. By measuring the number of data points collected from sources and transferred to the dataspace, it inherently reflects bandwidth utilization, resource consumption, and information transfer efficiency. Higher numbers of data points collected and transferred indicate higher bandwidth consumption, resource utilization, and information transfer, while lower numbers indicate more efficient data collection.

The experimental setup for evaluating the adaptive data collection system is provided by the running use case example (see [section 5.3.1](#)). The HPDC process consists of the five process phases shown in [figure 5.18](#) and [listing 5.4](#). The network architecture of the evaluation setup is illustrated in [figure 5.19](#), where our proposed extension with Raspberry Pi 4 edge devices in the office network is indicated. Note that the processing power of the edge device does not affect the results, as the primary evaluation metric is *number of data points transmitted*.

The *baseline* to which we compare our adaptive data collection strategy is a non-adaptive (static) data collection. It represents the current state of the art for many data collection setups in manufacturing, which are defined a priori based on domain knowledge. [Table 5.3](#) shows expert knowledge about the phases of the HPDC production process. For each of the five phases (see [figure 5.18](#)), the *average duration* indicates how many seconds the phase typically takes.

Table 5.3: Expert knowledge about the phases of the HPDC production process [[Lip+20](#)].

	Preparation	Dosing	Filling	Cooling	Removal
Average duration [s]	15	6.5	1.2	20	12
Min. collection interval [ms]	4000	200	200	200	4000

And the *minimum collection interval* indicates the frequency at which data (sensor values) are needed to produce appropriate results (e.g., for analysis or decision making). For *dosing*, *filling* and *cooling*, the minimum collection rate is high and requires data points every 200 milliseconds. For *preparation* and *removal* it is lower and requires data points only every 4 seconds. Since the data collection in the baseline scenario must be defined a priori and remains static throughout the production cycle, the highest required rate of 200 milliseconds must be chosen to meet the demands of each phase. Our adaptive data load solution, on the other hand, can change the data collection intervals (e.g., from 200 ms to 4000 ms) based on the needs of the current HPDC process phase. Both scenarios are performed via OPC UA and within the same work shift to ensure direct comparability of measurement results.

[Table 5.4](#) presents the result of this evaluation. It illustrates the number of data points collected for a specific value of interest during an HPDC production cycle. The average duration of the five phases of the HPDC process, as detailed in [table 5.3](#), adds up to almost one minute per production cycle. A comparison is made between a baseline scenario using a static data collection interval of 200 milliseconds and our proposed solution, which adapts the collection interval based on the current phase. Note that the baseline constantly collects five data points per second, so the number of data points transmitted (rounded up) scales linearly with the duration of the phases. During the *dosing*, *filling*, and *cooling* phases, the number of data points collected is the same for both scenarios. However, a significant reduction in the number of collected data points is observed during the *preparation* and *removal* phases, which is achieved by adjusting the collection interval from 200 milliseconds to 4 seconds as needed. For *preparation*, for example, our solution only transmits 4 data points (1/4 per second over 15 seconds, rounded up) but the baseline transmits 75 data points (5 per second

over 15 seconds). Notably, this **reduces the number of transmitted data points by 46 %** from 146 to 274, demonstrating the effectiveness of our approach in optimizing data load management. Note that these measurements describe data collection for one single **OPC UA** node (e.g., one sensor value), but they scale from one **OPC UA** node to thousands, or from one production cycle to any number of iterations.

Table 5.4: The measured number of data points collected for a value of interest during a **HPDC** production cycle [Lip+20].

HPDC Phase	Duration [s]	Static Collection (Baseline)		Our Adaptive Solution	
		Interval [s]	Data Points	Interval [s]	Data Points
Preparation	15.0	0.2	75	4	4
Dosing	6.5	0.2	33	0.2	33
Filling	1.2	0.2	6	0.2	6
Cooling	20.0	0.2	100	0.2	100
Removal	12.0	0.2	60	4	3
Total	54.7		274		146

In conclusion, we evaluated the effectiveness and performance of the proposed adaptive data load management system using the *number of data points* collected and transmitted as the primary evaluation metric. This general indicator directly or indirectly includes aspects of system performance, including bandwidth utilization and resource consumption. Execution in a real-world **HPDC** production setup with **OPC UA** showed that the proposed adaptive solution reduces the number of collected and transmitted data points by 46 % compared to a static data collection baseline, optimizing data load management for dataspaces.

Contributions

In conclusion, this section has effectively addressed the challenge outlined in [requirement Data-R2](#) by proposing a solution for adaptive data load management for dataspaces. By transforming static data collection into adaptive data collection, we have significantly improved the efficiency and adaptability of data and metadata collection processes.

The contributions of this section include:

- **Data Requirements Model:** We have proposed a phase model to divide data collection processes into phases, and for each phase to express precisely what data needs to be collected, when, and how. With a flexible level of detail and flexible configuration of such phases, we facilitate an accurate specification of data requirements that is agnostic to implementations or communication protocols.
- **Support for Data Collection Strategies:** Our solution builds on, but is not limited to, the different data collection strategies of **OPC UA**. Support for both interval-based (polling) and subscription-based (absolute and relative) data collection allows adaptive data load management to accurately match the actual data source.
- **Integration into Arbitrary Use Cases:** We have enabled practicality and scalability of the proposed solution by deploying physical or logical edge devices in close proximity to data sources. These provide support for data requirements specified using our phase model and ensure correct execution of adaptive data loading in diverse environments.

- **Ease of Configuration:** We have leveraged visual programming environments such as Node-RED to configure and manage our solution, providing easy-to-use tool support and intuitive user interfaces. This enables broad adoption by dataspace participants with any level of technical expertise and further simplifies large-scale deployments across many data sources.
- **Evaluation in Real-World Scenarios:** Through evaluation in a real-world [HPDC](#) production use case, we demonstrated the effectiveness and performance of our solution, validating its practical applicability and impact on data load within dataspaces. Our approach reduced the number of data points collected and transmitted by 46 %, which can dramatically optimize bandwidth utilization and resource consumption.

Together, these contributions make data collection adaptive, thus advancing the field of data collection within dataspaces, overcoming resource limitations, and improving data efficiency. By providing a phase model for describing data needs, user-friendly ways to manage these needs, and solutions to serve them adaptively, we help dataspace participants overcome the challenges posed by rapidly expanding data availability.

5.4 Managing Data and Metadata Changes

This section addresses [requirement Data-R3](#), focusing on the critical aspect of managing data changes within dataspaces. Ensuring the stability and reliability of data within dataspaces necessitates robust strategies for managing dynamic data sources. Data sources are inherently subject to changes in structure, content, and context, necessitating proactive measures to monitor and manage these changes effectively. The management of data changes is crucial for maintaining data consistency, accuracy, and reliability within dataspaces, particularly as data sources evolve over time. This necessitates the effective implementation of data change management strategies and tools to uphold the integrity and usability of data across diverse applications and analyses within dataspaces.

In the modern software architecture landscape, interfaces such as service-oriented architectures ([SOAs](#)) play a critical role in enabling modular and scalable communication between different components of a system or between different systems. [SOAs](#) and their subset *remote procedure calls* facilitate the development of distributed systems by allowing different software services to interact over a network, promoting reusability and interoperability. However, while [SOAs](#) are good at checking the syntax of the communication between services, they often fall short when it comes to verifying the semantics of the data being exchanged. This limitation poses significant challenges, as it can lead to misunderstandings and errors when relying solely on syntax validation for communication between services. To address this problem, *Semantic Web Services* [[MSZ01](#)] emerged as an extension to the Semantic Web [[BHL01](#)] in 2001. Semantic Web Services aimed to enhance traditional Web services by incorporating semantic meaning into service descriptions, enabling more intelligent service discovery, composition, and invocation. However, Semantic Web Services faced several challenges at that time, including complexity in implementation and adoption, as well as issues with data availability and syntax validation. Fast forward to the present, and many of the obstacles that hindered Semantic Web services in the past have been overcome, particularly in terms of data availability and syntax validation. However, the need for robust validation of both syntax and semantics in communication interfaces such as [SOAs](#) remains an open

challenge. Therefore, there is an urgent need for a lightweight version of Semantic Web Services tailored to common communication interfaces such as [SOAs](#). This lightweight solution would bridge the gap between syntax and semantics, allowing validation of both aspects to ensure secure and reliable communication for dataspaces. By addressing this challenge, we can improve the effectiveness and efficiency of communication interfaces such as [SOAs](#). This paves the way for more robust communication that can manage changes in both data and metadata.

Even assuming that there is a solution for validating both syntax and semantics in communication interfaces such as [SOAs](#), several challenges remain. It is not sufficient to validate the interfaces only during the initialization of the communication, because changes can occur during runtime. Therefore, continuous validation and monitoring is required to ensure the ongoing integrity and reliability of communication between services. In addition, the validation techniques employed must be sufficiently sophisticated to meet the diverse requirements of participants in dataspaces. This complexity can include domain-specific details such as engineering units, as well as general constraints such as value ranges, complex data types, cardinalities, or cross-attribute dependencies. Addressing these challenges requires a comprehensive approach that integrates dynamic validation processes into the operational lifecycle of the [SOAs](#), ensuring that communication remains robust, adaptable, and aligned with the evolving needs of attached systems and stakeholders of dataspaces.

5.4.1 Motivating Domain Example

The following motivating example comes from the manufacturing domain (see [section 2.3](#)), but can be applied to many other domains. It motivates the extension of [SOAs](#) with semantic capabilities based on Semantic Web Services by presenting an example from ultrashort pulse ([USP](#)) laser system development and showcasing concrete challenges.

Semantic Web and laser experts are collaborating to create a [USP](#) laser system (see [figure 5.24](#)) based on a [SOA](#) for the *Internet of Production* research project [[Pen+19](#)] (see [section 2.3](#)). This involves assembling machine parts from different manufacturers and establishing interoperable communication between them. The concept of [USP](#) can be described as a form of *reverse 3D printing*, where high amplitude laser pulses are used to incrementally remove material and shape the final product. A [USP](#) laser consists of multiple components, including a scanner, motion system, and camera. Client applications, such as a central controller, make remote service calls to these components to configure the laser and perform actions. The syntax of the service calls is validated using *Protobuf*, but semantic mismatches are currently overlooked and not detected.

A scanner moves the laser based on float values for `position.x` and `position.y`, but the interpretation of units is determined by the implementation of the service. [Figure 5.25](#) demonstrates a dangerous scenario where the old scanner hardware interprets position values as millimeters, while a new component might interpret the same data value as centimeters, potentially causing incorrect laser movement (2 centimeters instead of 2 millimeters) or product damage. Similar challenges arise with other variables like laser heating temperatures and laser speeds. The current solution uses gRPC [[Goo16](#)] for network communication, allowing data serialization using Protobuf [[Goo15b](#)]. Bazel [[Goo15a](#)] is used as the build tool to manage dependencies, build source files, and execute the code. Protobuf files define the input and output parameters of service calls and the data types for client-server communication. However, semantic information such as units cannot be included in these definitions. De-

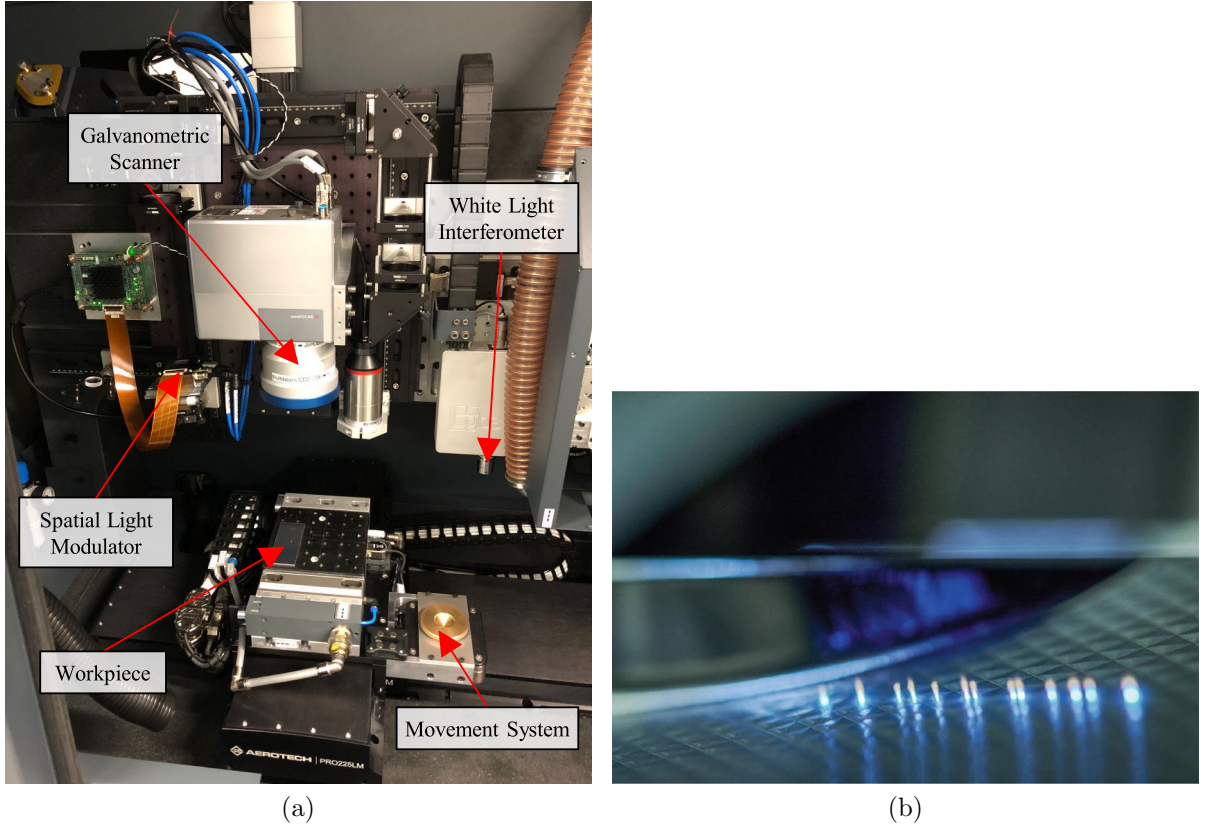


Figure 5.24: A [USP](#) laser system (a) consists of several components that may be frequently replaced [\[Krö+24\]](#) and forms objects by (b) selectively removing material layer by layer [\[Mic+23\]](#).

velopers often use comments and variable names in Protobuf files to make units visible to readers temporarily. However, this approach is error-prone and not machine-readable. It is important to accurately define and validate units at service interfaces to avoid semantic mismatches.

While detecting semantic mismatches in an industrial [SOA](#) framework helps prevent accidental errors when integrating new machine components, it does not prevent runtime errors or malicious messages. Specific rule sets can be formulated to validate messages and prevent these errors. An example of a hard runtime error is a `MoveToPosition` message where the requested position is out of bounds. Currently, the system detects these errors on the server side because physically moving to that position is impossible. These system-specific rules protect the system from internal damage. A domain-specific language for formulating these rules may be advantageous. However, defining softer runtime errors holds more potential. These soft runtime errors are typically producible but may not be preferable in terms of quality or long-term maintainability. These rules can be *cross-platform* and applied to multiple services implementing the same interface. The formulation of these constraints is more complex, requiring an expressive constraint language.

In the USP production domain, ablation at the 0/0 position of the scanner system is an example of a softer runtime error. At this point, the laser's angle of incidence is perpendicular to the workpiece surface, causing reflected laser light to potentially damage components due to overheating [\[TD16\]](#). Another soft constraint is the combination of laser power and the corresponding velocity of the laser beam, which determines the delivered energy per area

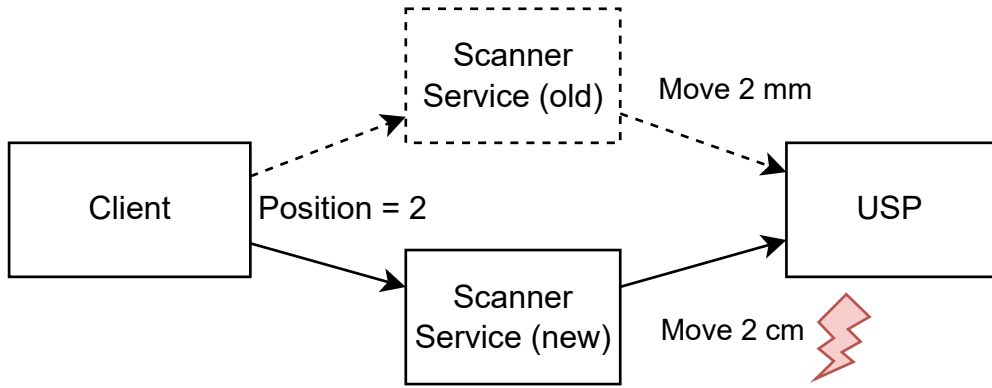


Figure 5.25: Example of a semantic mismatch that occurs during the replacement of components on a [USP](#) laser system [Lip+21d].

(fluence). This constraint depends on the material used, including the melting point and absorption rate. If the fluence is too low, the material will not vaporize. If it is too high, the temperature of the material may rise, leading to thermal deformation [Bys+10].

These errors should be avoided during a [USP](#) ablation process. However, the scanner component does not restrict this behavior. Including this rule in the scanner service would be detrimental because it would directly tie the general *scanner system* component service, which is used in various manufacturing machines, to the [USP](#) process. Currently, machine operators mitigate these errors through their experience. They apply their knowledge during the design process of the manufactured part and the setup of the machining process. The main goal of this motivating domain use case is to overcome semantic mismatches and erroneous messages in [SOA](#) frameworks.

Enhancing communication frameworks, such as [SOAs](#), requires effective semantic capabilities to address message communication complexities. This involves extending validation approaches to include semantic and entity validation for mitigating inconsistencies and ensuring message accuracy. Developers need flexibility in configuring semantic information for both clients and servers to enhance predictability and reliability. Automation is crucial in proactively identifying erroneous messages and reducing errors. The system should support extensible semantic message rules for dynamic updates. Ensuring backward compatibility is essential for seamless integration with (distributed) [SOA](#) frameworks. Accessibility to a wide range of dataspace participants, automation, and adherence to software best practices strengthen the proposed solution. The goal is to deliver a field-proven and easy-to-deploy solution that meets diverse dataspace use cases for [SOA](#) environments.

5.4.2 Related Work and Background

This section explores the background in terms of relevant ontologies and semantic service descriptions. The area of data lifting and mapping is covered in [sections 2.2.1 and 5.5](#), and the area of constraint-based validation in [sections 2.2.1 and 6.7](#).

Relevant Ontologies

Our survey addresses relevant ontologies related to sensors, units, and measures, leveraging existing concepts wherever possible, as advocated in [GSA15]. Our goal is to identify ontolo-

gies that can be used in a modular and flexible manner. Metrics for evaluating ontologies include:

- **Relevance:** Determines how well the ontology scope matches our work and the motivating domain use case.
- **Coverage:** Rates the applicability of the ontology’s concepts to our needs. The metric evaluates available programmatic extensions such as unit conversions.
- **Flexibility:** Rates how well one can tailor the concepts from the ontology to specific applications, e.g., being able to build custom units with prefixes like `milli`.

Note that these scores are specific to our requirements and are not universal ontology scores. A summary of the evaluated ontologies is provided by [table 5.5](#). The symbols (- / o / +) represent a low / medium / high value of the given metrics. QUDT ranks first and OM ranks second. Note that this ranking is specific to our needs and therefore not universal.

Table 5.5: Evaluation of relevant ontologies in the areas of sensors and engineering units based on the requirements from [section 5.4.1](#) [[Lip+21d](#)].

Ontology	Relevance	Coverage	Features	Flexibility
SDO [ELE07]	-	o	-	-
OntoSensor [RKT05]	-	o	-	-
SSN [Com+12]	o	+	+	o
OM [RWT11 ; RAT13]	+	+	+	o
QUDT [Ray14]	+	+	+	+
UCUM [ELE07 ; SM09]	+	o	+	o

In the area of sensor ontologies, the article [[Sch+13](#)] reviewed existing sensor data ontologies to decide if they can be reused for an ontology for perception sensors in the manufacturing industry. The result of their work is a review and an ontology. Among the ontologies mentioned are the *Sensor Data Ontology* (SDO) [[ELE07](#)], which uses the *Suggested Upper Merged Ontology* [[NP01a](#)], the *OntoSensor* ontology [[RKT05](#)], and especially the *Semantic Sensor Network* (SSN) ontology [[Com+12](#)]. This ontology seems promising for our work and has several features: Data discovery and linking, device discovery and selection, provenance and diagnosis, and device operation, tasking, and programming [[Com+12](#)]. It allows focusing on sensors, observed data, system, or feature and property. The Semantic Sensor Network ontology describes sensor networks well, but needs to be combined with other ontologies for describing units, such as the ontology *Library for Quantity Kinds and Units* [[Kon05](#)].

In the area of engineering units, the following ontologies are relevant. Developed during the development of the Ontology of Quantitative Research [[RTM09](#)], the *Ontology of Units of Measure* (OM) [[RWT11](#); [RAT13](#)] allows the description of units with all their details and relationships to each other. It is based on existing measurement standards such as [[Tay95](#)] and includes units, prefixes, quantities, scales, measures, systems, and dimensions. Each quantity is defined by a measurement scale, which is a mapping of categories and points to quantities. Units can then be further scaled with prefixes, which make it easier to represent values for a base unit. For example, millimeter is defined as a prefixed unit with meter

and the prefix milli. Quantities and units have dimensions and systems of units to organize them. A system of units defines a set of base dimensions that can then be used to express any other possible dimension as a combination of certain base units, such as the International System of Units [TT08]. These base units can then be used to compute all other dimensions. The *Quantities, Units, Dimensions, and Types* (QUDT) ontology [Ray14] follows a similar approach, building modularly on top of several ontologies. It covers fewer quantity types and units per application domain than OM, but allows more flexible conversion multipliers and offsets. QUDT offers a mature SPARQL integration and provides flexibility by linking to OM and UCUM via additional identifiers within the ontology. The *Unified Code for Units of Measure* (UCUM), published in [ELE07; SM09], covers virtually all of the units used in science and engineering, and assigns unique identifiers to each of them. A web service⁵ allows to validate and convert datatypes. The UCUM unit codes are referenced in the QUDT ontology mentioned above and support these unit conversions. However, UCUM specifies less comprehensive units and scales compared to OM.

At the end of the ontology evaluation (see table 5.5), QUDT and OM rank first and second and are therefore promising candidates for our work.

Communication and Service Description

Semantic Web Services [MSZ01], their organized peer-to-peer extensions [Sch+02], and similar approaches semantically enrich Web services and provide machine-readable markup. Required descriptions can be written in the Web Service Description Language. This language has been extended with semantic annotations in [Kop+07]. These annotations refer to ontologies and support lifting and lowering mappings between XML messages. However, a complex feature set including informational, functional, behavioral, and non-functional semantics complicates the lightweight application that is needed in dataspace. Semantic Web Service descriptions with slightly reduced complexity have been proposed in [Fen+11] and [Rom+15], which are also available as W3C submission [Fen+10]. But they still include functional, non-functional, and behavioral semantics. [Ben19] introduces a so-called descriptor that adds operation, link, non-functional, and service descriptions to representational state transfer (REST) services using an ontology-based approach.

Remote procedure calls allow services to be called remotely and parameters to be passed [BN84]. This concept has recently become an active research topic again, although it was introduced nearly four decades ago. Google offers gRPC [Goo16], an open source, high performance remote procedure call framework that has gained a lot of popularity. Advantages include high scalability, low latency distributed systems, and support for developed programming languages. It is recommended to use Protobuf to describe the syntax of expected inputs and outputs of services [Goo15b]. A concrete example is the extension of a closed source platform for deployment, integration and orchestration of digital services with semantics about engineering units [Mar+20]. Proposed features include data contextualization by enriching data with (semantic) information that facilitates the understanding of the data and its context.

We conclude that Semantic Web Services was an active research area between 2001 and about 2008, but has lost its momentum. This is mainly due to very ambitious goals that were not yet applicable in practice. Later developments based on remote procedure calls

⁵UCUM Unit Conversion Web Service: <https://ucum.nlm.nih.gov/ucum-service.html#conversion>

have solved important fundamental problems. Therefore, there is now a chance to properly combine these research areas.

5.4.3 Approach

The fact that data sources are inherently subject to changes in structure, content, and context requires proactive measures to effectively manage these data changes within dataspace. While the management of syntactic changes is mainly solved in state-of-the-art solutions, the management of semantic changes is still an open challenge (see [figure 5.25](#)). Our approach is to integrate concepts from the Semantic Web and Semantic Web Services into communication environments to complement existing syntactic checks with semantic ones. We address the requirements of the motivating domain example presented in [section 5.4.1](#) by proposing *Lightweight Semantic Web Services for Units*, which continuously monitors communication and reacts to changes in data or metadata. See [table 5.6](#) for an overview of how we address the requirements that are derived and generalized from this use case. In this table, the *illustration* column refers to figures and sections that describe our solutions in detail. The details of our approach are presented below.

Table 5.6: A summary of our approach to addressing requirements derived from the motivating use case (see [section 5.4.1](#)).

Requirement	Approach	Illustration
Semantic capabilities	Semantic units with links to ontology PIDs or URIs	Figure 5.26
Validation	Initial and continuous validation	Figures 5.27 to 5.29
Flexible configuration	Configuration files with automatic error avoidance	Figures 5.26 and 5.27
Automation	Automated processes once deployed	Figures 5.27 to 5.29
Backwards compatibility	Semantic units designed as a modular addition	Figure 5.26
Field-proven	Evaluation in both laboratory and real-world environments	Section 5.4.5
Easy to deploy	Simple deployment and configuration of semantic units	Figures 5.26 and 5.27

[Figure 5.26](#) shows a state-of-the-art setup using gPRC on the left, which serves as both the communication technology and the interface definition, typically encapsulated in Protobuf files. In the baseline configuration shown, a client initiates contact with a server via a service call, prompting the server to validate the syntax of the call before executing the service upon successful verification. We add semantic validations to the [SOA](#) framework, as shown on the right side of [figure 5.26](#). This enhancement involves integrating semantic units into both client and server components that reference [URIs](#) from ontologies to uniquely identify semantics such as engineering units. Our extended pre-communication validation now includes a semantic check that compares the ontology [URIs](#) of both communicating parties and allows service execution only if they match. In addition, the server is extended

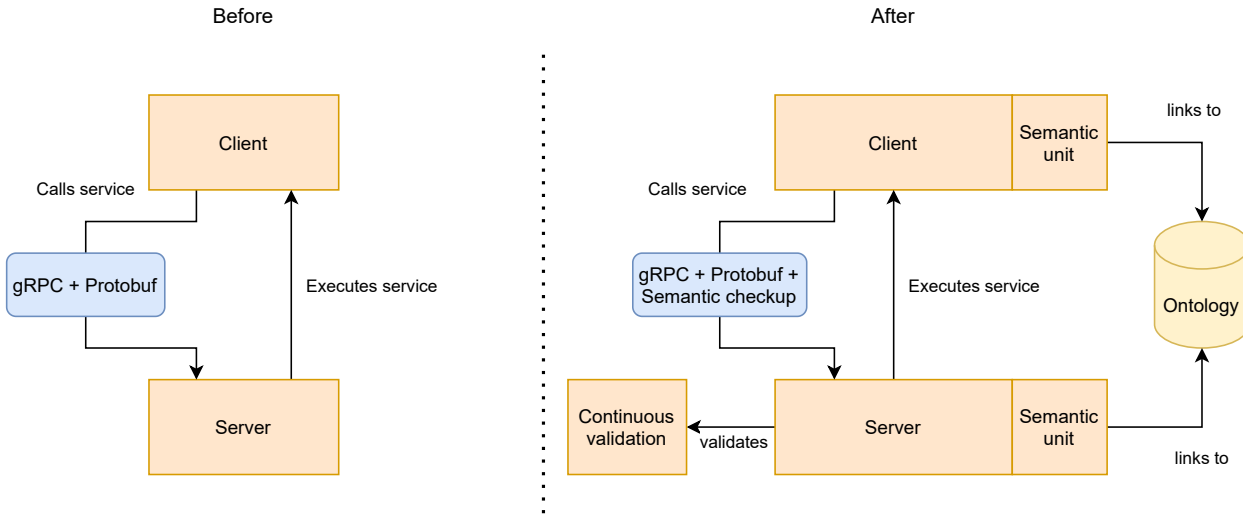


Figure 5.26: Our contribution (right) adds semantic units and validation based on ontology [URIs](#) to the baseline (left) [The+22c].

with continuous validation mechanisms to ensure compliance with defined constraints for continuous further communication.

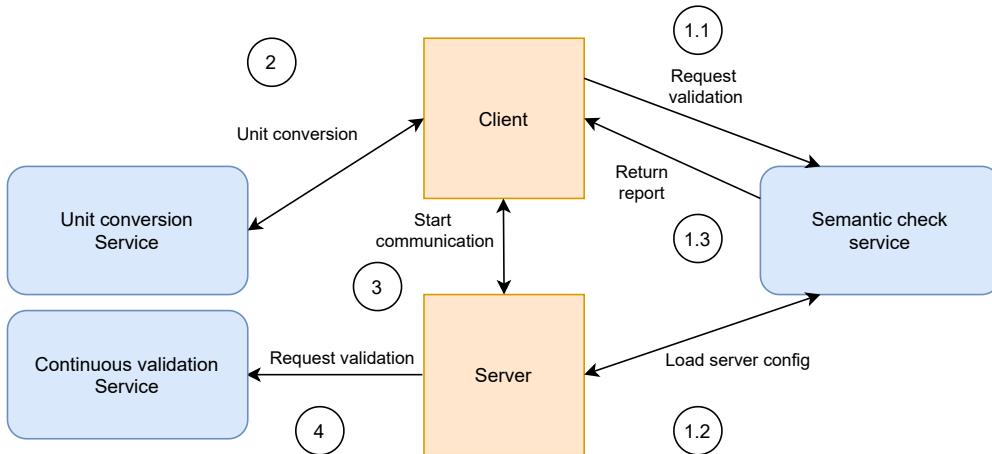


Figure 5.27: Our approach for more reliable communication between two communication partners (e.g. client and server) [The+22c].

Our approach for the configuration of semantic units, ontology references, and semantic validations (all see [figure 5.26](#)) is as follows. We add mapping from the interface definitions, such as gRPC for services, to the corresponding ontology [URIs](#). This information can then be used to validate semantic information such as engineering units between client and server and only initiate communication if they match. The proposed workflow of such a semantic validation is shown in [figure 5.27](#) and is an addition to the existing workflow of the communication architecture such as [SOA](#). In this figure, rounded blue rectangles indicate newly introduced services. A client initiating a service call first undergoes a semantic check, which compares the client’s semantic specifications with those of the server. The client (1.1) requests validation from the *semantic check service* by submitting its semantic configuration. This service (1.2) retrieves the corresponding configuration from the server, compares it, and (1.3) returns a validation report to the client. The client then interprets this report, see the

next section for details. If the report contains problems, the client must fix them or abort the communication. For example, it may (2) contact a unit conversion service to correct unit mismatches. In addition, (3) the client initiates the actual communication with the server. Finally, (4) the server frequently requests validation of the currently sent data by a continuous validation service, for example, based on a fixed time or message count interval.

A client (step 1.1) receives a validation report from the semantic check service during communication preparation. Figure 5.28 depicts an activity diagram of our proposed initial semantic validation workflow, which detects and recovers semantic mismatches to establish predictable communication. In this figure, empty diamonds join alternative paths. The workflow begins by loading the client and server interface configurations, and depending on the report from the semantic check service, one or more actions may be required at the client:

- **Client configuration incomplete:** The supplied configuration lacks properties required by the server. The client must correct it by adding missing definitions, usually manually.
- **Unit dimension mismatch:** Fix the dimensions and units used and try again. Hard mismatch, e.g. *speed in m/s* used but *temperature in Celsius* expected. Usually fixed manually.
- **Unit mismatch:** The correct dimension is used but the wrong unit, e.g. *m/s* instead of *km/h*. Use the provided unit conversion service to automatically convert units.
- **All semantics match:** Begin communication.

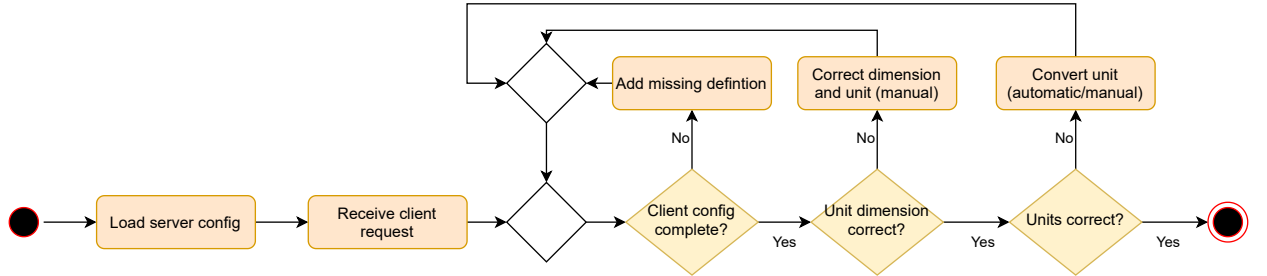


Figure 5.28: Activity diagram of our proposed initial semantic validation workflow (steps 1 to 3 in figure 5.27), which detects and recovers semantic mismatches [Lip+21d].

All relevant issues related to semantic mismatches are detected in this workflow. Critical issues such as incomplete configurations or dimensional mismatches typically require manual resolution. However, we can automatically resolve unit mismatches within the correct dimensions.

Our approach to continuous semantic validation is illustrated in figure 5.29. Depending on performance, some messages of a data stream may not be validated and the workflow is terminated immediately (bottom left). A server periodically submits a portion of the continuously read data stream to the continuous validation service. Since the targeted validation languages may not inherently support the native format used for (SOA) communication, the data may undergo a transformation process. It is first represented in a non-binary format before being transformed into a semantic representation using PIDs, such as ontology URIs (see section 5.5), which facilitates effective validation. This semantic representation, such as

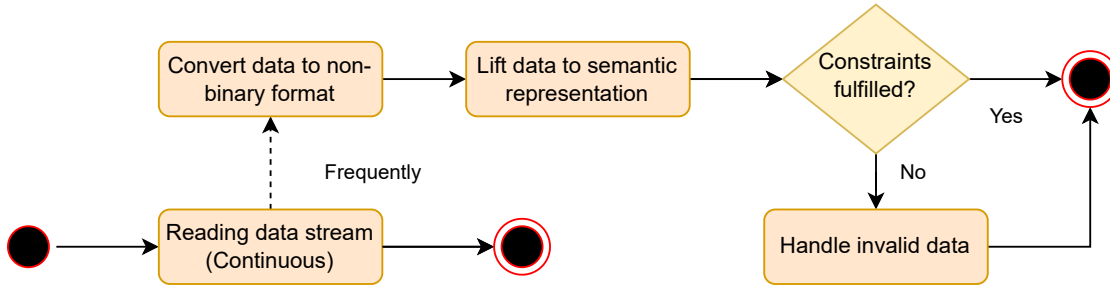


Figure 5.29: The proposed workflow for continuous monitoring of data streams [The+22c].

RDF, allows verification that the defined constraints are met. In cases where the constraints are not met, the approach proposes a handler implementation, which could manifest as a logging or recovery action, to ensure that any discrepancies are appropriately addressed. This comprehensive workflow ensures the integrity and reliability of communication processes in dynamic dataspace environments.

In summary, managing data and metadata changes within dataspace requires proactive strategies due to the inherent variability in structure, content, and context of data sources. While existing solutions effectively handle syntactic changes, addressing semantic changes remains a persistent challenge. To bridge this gap, our approach integrates Semantic Web and Semantic Web Services concepts into communication environments, complementing syntactic checks with semantic validations. In particular, our proposal of Lightweight Semantic Web Services for Units addresses the needs of dynamic domains, such as the one outlined in our motivating use case. By integrating semantic information such as engineering units, continuous monitoring, and validation mechanisms, we ensure robustness, flexibility, and reliability in data communication processes. By addressing issues ranging from incomplete configurations to semantic mismatches, and by incorporating continuous validation workflows, our approach supports the effective management of data and metadata changes in the dynamic environments of dataspace.

5.4.4 Results

This section presents the results of the approach described above. This includes justifying a selection of concrete tools based on the requirements derived from the motivating domain use case (see section 5.4.1). We present our results in detail below:

1. An initial semantic validation service, including a method for specifying semantics at both client and server.
2. A unit conversion service that automatically corrects unit inconsistencies within a dimension.
3. A continuous semantic validation service, including a method for validating streaming data against given requirements after lifting heterogeneous (semi-)structured data.
4. User interaction and workflow suggestions.

The source code of our solution is available as open source on GitHub [LSK21], including code snippets and an example.

Tool Selection Based on Requirements

There are several ontologies, data description formats and additional software to choose from (see [section 5.4.2](#)). The tool selection for our implementation is based on requirements derived from the motivating domain use case (see [section 5.4.1](#)) and is as follows. We use *Protocol Buffer* and *gRPC* to syntactically model the interface of services, and introduce semantic descriptions, such as engineering units, via additional [JSON](#) files stored at the client and server. Reasons for choosing [JSON](#) include its high expressiveness and improved human readability, as well as its easy integration with many programming languages. We integrate *Bazel* to automate the creation and testing of these descriptions.

Based on the results of our ontology evaluation, summarized in [table 5.5](#), we select *QUDT* as the main ontology for the semantic representation of units of measure, while maintaining support for *OM*. We do not need *SSN* or any of the other ontologies in this use case, since *QUDT* and *OM* already cover all the necessary concepts. However, the semantic descriptions link to additional ontologies for certain types of data, such as the representation of *UNIX timestamps*. We chose *QUDT* because it covers all the relevant units of measure we need and provides good support for unit conversions via [SPARQL](#) queries. This can be done using the Python library *rdflib* [[Kre02](#)] or other programming languages like Java or Go.

To export [SOA](#) messages from Protobuf, we use the [JSON](#) data format. We choose the [RDF Mapping Language](#) (see [section 5.5](#) for more details) to formulate data mappings to [RDF](#), and use its reference implementation *RMLMapper* as a Java library to execute them. Using it from the command line makes integrating easy and modular. Any [SHACL](#) validator can automatically validate the generated [RDF](#) files.

Semantic Validation Service

Our implementation solution for an initial semantic validation service that compares the semantics of two communicating parties a priori is as follows. We implement the architectural approach presented in [figure 5.26](#) as semantic units that link to ontology [URIs](#) for validation. The semantic units consist of [JSON](#) files containing the semantic descriptions of the client and server, and Python software capable of interpreting them. See [listing 5.5](#) for an example server configuration with references to ontology [URIs](#) from Wikidata and *QUDT*. The prefix *QUDT* is short for <http://qudt.org/schema/qudt/> and the prefix *wiki* is short for <http://www.wikidata.org/wiki/>. This basic Python code is extended with program logic for the initial validation service introduced as *steps 1.1 – 1.3* in [figure 5.27](#). Finally, we implement the communication protocol proposed in [figure 5.28](#) in the Python software on both the client and the server, so that the client handles the server feedback correctly.

The Python solution uses Protobuf’s `*.proto` files to define service calls and message types. [JSON](#) files are used to extend the existing (syntactic) interface descriptions of server and client with semantics by linking to ontology [URIs](#). The example shown in [listing 5.5](#) defines a service interface from the [USP](#) use case (see [section 5.4.1](#)). The first entry (`scannerservice`) defines the name of the *Protobuf* file, which is extended with the following semantics. The (`Position` and `Time`) entries specify the Protobuf messages that are referenced. This example contains a two-dimensional *X–Y* position with engineering units *millimeter* from the *QUDT* ontology, and a timestamp in Unix time (i.e., seconds since January 1, 1970) defined in Wikidata. We use `qudt:Millim` to refer to millimeters from the *QUDT* ontology. For the *OM* ontology, this would be the [URI](#) `om:millimetre` instead. See [table 5.7](#) for more

```
1 {
2     "scannerservice": {
3         "Position": {
4             "x": "qudt:MilliM",
5             "y": "qudt:MilliM"
6         },
7         "Time": {
8             "timestamp": "wiki:Q14654"
9     }
}
```

Listing 5.5: Sample of a server configuration that adds semantic information via ontology URIs [Lip+21d].

example variables, with their (syntactic) data types and (semantic) engineering units, from the motivating domain use case. These references to [RDF](#) resources enable semantic features such as [SPARQL](#) queries to access more details about engineering units.

Table 5.7: Example variables with their data types and units of measure for the real-world use case [Lip+21d].

variable name	datatype	unit of measure
preheatingTemp	double	qudt:DEG_C
laserSpeed	float	qudt:MilliM-PER-SEC
position.x/y	float	qudt:MilliM
shotTime	int64	wiki:Q14654

During the actual validation, all entries and their ontology [URIs](#) are compared and a response with the validation result is sent to the client. Calls to the semantic validation service and their responses are also defined with Protobuf files. These are executed with gRPC calls that accept [JSON](#) configurations as string input and return a response message with various information: A server response with the validation result (**True** or **False**), a string with a detailed description of any violations, and the history of previous units and unit changes on the server.

Unit Conversion Service

The approach presented in [figure 5.27](#) includes a service for converting between engineering units. This service is used in the last step of the activity diagram of the proposed semantic validation workflow (see [figure 5.28](#)) to automatically correct unit mismatches within the same unit dimensions.

In the implementation, a non-empty conversion list in the response from the semantic validation service triggers the client to call the unit conversion service. It queries engineering units from the QUDT ontology via [SPARQL](#) to extract various properties of the units from the [JSON](#) configuration files. The unit conversion service implementation uses the structure of the QUDT ontology to calculate conversions, including the *conversionMultiplier* and *conversionOffset*. Note that conversions from a source unit to a target unit are only possible within the same dimensions. Our solution describes dimensions using the *International System of*

Units, also known as *SI units*. Their unique identifiers ensure correct execution: While the conversion from grams to pounds is valid for two masses, the conversion from grams (force) to pounds (mass) is prevented.

Technically, the unit conversion service processes a list of values and units to be converted, each with a [URI](#) of the source and target units. For each conversion task, our solution extracts the multiplier and offset of the source unit (e.g. grams) to its base unit (e.g. kilograms), converts the source value to that base unit, and finally converts the value from the base unit to the target unit (e.g. pounds). The unit conversion service finally returns a list of all converted values in the target units.

Continuous Semantic Validation Service

The implementation of the continuous semantic validation service proposed as step 4 of the approach depicted in [figure 5.27](#) follows the workflow shown in [figure 5.29](#). It is implemented as a modular extension to existing [SOA](#) solutions. While receiving a data stream, the server frequently selects messages for validation, depending on resource limitations or simple user-defined specifications such as *validate every nth message* or *validate a message every n seconds*. The validation process consists of two parts, namely lifting binary data into semantic representations and actually validating them.

The lifting of data to semantic representations uses concepts that are explained in more detail in [section 5.5](#).

We use [SHACL](#) to validate (see [section 2.2.1](#)) Protobuf messages from the data stream. This requires lifting these messages into [RDF](#) (see [sections 2.2.1](#) and [5.5](#) for details). Protobuf provides the *JsonFormatter*⁶ for exporting Protobuf messages to [JSON](#), which is a supported input format for the *RMLMapper*. We implement a set of [RML](#) rules to define the appropriate mappings from the Protobuf messages to [RDF](#), including unique identifiers (e.g., for units), properties, and structure. Note that this [RML](#) rule file only needs to be designed once, and can then be applied continuously to every message in the given stream. We integrate the *RMLMapper* via command line calls to automatically generate corresponding [RDF](#) files, which are then compatible with [SHACL](#) for validation. An explicit call could be `java -jar rmlmapper.jar -s turtle -m mapping.ttl -o output.ttl`, or the method can be called from a Java program with the same arguments.

The constraint validation process, represented by the diamond in [figure 5.29](#), is implemented using [SHACL](#). This involves the formulation of [SHACL](#) forms to define various constraints. These range from unit quantity types such as length for meters or kilometers to specific requirements for values such as ranges or filters. We use the QUDT ontology to reference these. The QUDT ontology is then appended to the [RDF](#) data graph resulting from data lifting to make all references from this ontology available for validation (see [section 6.7](#) for more details). These [SHACL](#) validation methods are versatile and can be applied to various programming languages such as Java, JavaScript, .NET or Python, thus ensuring portability. Finally, the data extracted from the data stream is validated against the defined [SHACL](#) schema using the QUDT ontology. Constraint validations activate a configurable handler that can perform actions ranging from simple logging to more complex functionality such as intercepting incorrect client interactions.

⁶<https://developers.google.com/protocol-buffers/docs/proto3#json>

User Interaction and Workflow

Incorporating these new services extends the workflow for designing and implementing new applications within the communications system. This extended workflow for dataspace participants is as follows. Initially, the system requires a server configuration for each newly added server, as well as client configuration files for each newly added client. When a dataspace participant adds a new client, they must complete their configuration file and include a section in their code to load their configuration and invoke the semantic validation service. After that, typical client operations are performed unless there are mismatches. If there are mismatches, they are either automatically corrected by the unit conversion service. Or, communication is halted and the dataspace participant is prompted to manually adjust the configuration file and values based on a detailed error message. This iterative process continues until all errors and mismatches are resolved, and ultimately ensuring predictable and secure communication.

5.4.5 Evaluation

This section evaluates the effectiveness and practicality of the proposed approach for managing data and metadata changes in dataspace. It covers the two main steps of our approach (see [figure 5.26](#)), including the one-time initial validations and the continuous validation of data streams. In the following, we first motivate the evaluation criteria and then present the evaluation results, structured according to the two steps of our approach.

Evaluation Criteria and Scenarios

The main goal of evaluating the semantic validation service is to measure the predictability and reliability of communication in terms of mutual understanding of the communicated content by both communication parties (e.g., client and server). If both communication parties interpret the content in the same way, the receiver of messages will behave as expected by its sender, and they can communicate with confidence. If they interpret the content differently, they certainly should not communicate. If the interpretation of one or both of them is unknown, they can communicate, but at the risk of misunderstanding. A metric for measuring the correctness of the receiver's response is a gold standard, where the actual response or behavior is compared to the expected one. The results are either a match or a mismatch with the expected values. The expected behaviors for the different scenarios range from *communicate* to *block* to *warn*. For the second part of the evaluation, the continuous semantic validation service, execution time is the main evaluation criterion because it is the most limiting factor when monitoring potentially high data loads on data streams.

The evaluation scenario setup for the initial semantic validation workflow is as follows. We directly compare the proposed extended system with the previous state as a baseline. Based on the motivating use case, the task during this evaluation is to call the `JumpToPosition` function, which instructs the scanner to move the laser beam to the specified position on the scan field. Its input is a 2-dimensional (X,Y) position and its response is the completion time after successful execution. The evaluation setup includes the implementation of clients (senders) that connect to the scanner services (servers / receivers) and call the `JumpToPosition` function. This setup includes several different configurations of clients and servers based on the following combinations of mutual (syntactic and semantic) interpretation that are expected to result in either *communicate*, *block*, or *warn*:

- **Scenario 1 - Syntax match, semantics unspecified:** The syntax specifications in the client and server protobuf files match, as implemented by the baseline system. However, there is no configuration of semantic interpretation at either the client or the server, leading to possible semantic misunderstandings between the two. For example, the client might implicitly treat the position values as *centimeters*, but the server might implicitly interpret them as *millimeters*. Communication is possible, but without confidence, so the expected response for this scenario is a warning (*warn*).
- **Scenario 2 - Syntax mismatch, semantics unspecified:** Same as scenario 1, but even the explicit syntax from the baseline system does not match between client and server. Receiving different data types than expected will certainly break the mutual interpretation on the receiver side, so the expected response is to not accept this communication (*block*).
- **Scenario 3 - Syntax match, semantics match:** This scenario uses our proposed system for expressing semantics by adding configuration files to both client and server. This is the ideal scenario. It even covers inconsistencies within the same unit dimension, such as millimeters vs. centimeters, because these can be automatically corrected using our unit conversion service. Since the client and server configurations match in both aspects, syntax and semantics, they can communicate confidentially and the expected response is to do so (*communicate*).
- **Scenario 4 - Syntax match, semantics mismatch:** Same as scenario 3, but with at least one semantic configuration mismatch that cannot be automatically resolved. For example, the server explicitly expects a number (integer) value for a position, and the client sends exactly that. But the server explicitly expects the unit *millimeter*, but the client sends *kilogram*. This can lead to critical malfunctions such as production errors or even incidents. Since the server will certainly misinterpret the messages, it is expected not to accept this communication (*block*).

```

1 {   "workPlanes": [{
2       "vectorBlocks": [{
3           "points": [0.585648, 0.648475, -0.05 ],
4           "markingParamsKey": 1 }]],
5 "jobMetaData": { "jobName": "Batsymbol_2mm" },
6 "markingParamsMap": {
7     "laserSpeedInMmPerS": 100.0,
8     "jumpSpeedInMmS": 1000.0,
9     "jumpDelayInUs": 100.0,
10    "laserOffDelayInUs": 100.0,
11    "laserOnDelayInUs": 100.0 },
12 "numWorkPlanes": 1 }

```

Listing 5.6: Example of non-binary JSON data extracted from Protobuf messages in the use case [The+22c]. It includes a 3D point and laser movement speed.

The scenario for the second part of this evaluation, the continuous semantic validation service, is as follows. Messages in binary format are intercepted from the data stream and lifted to the non-binary JSON format so that SHACL validation can be applied to them. An

example message from the motivating use case, converted to [JSON](#), is shown in [listing 5.6](#). It contains, among other values, points and marking parameters such as movement speed. Note that some of these have semantic information such as units implicitly in their names (e.g., `laserSpeedInMmPerS`), but not explicitly.

```

1 :PointMap a rr:TriplesMap; rml:logicalSource :PointSource;
2 rr:subjectMap [rr:class :Point; rr:termType rr:BlankNode];
3 rr:predicateObjectMap
4 [rr:predicate ex:x;
5 rr:objectMap [rr:parentTriplesMap :PointXMap]],
6 [rr:predicate ex:y;
7 rr:objectMap [rr:parentTriplesMap :PointYMap]],
8 [rr:predicate ex:z;
9 rr:objectMap [rr:parentTriplesMap :PointZMap]].
10
11 :PointXMap a rr:TriplesMap; rml:logicalSource :PointSource;
12 rr:subjectMap [rr:class qudt:Quantity;
13 rr:termType rr:BlankNode];
14 rr:predicateObjectMap
15 [rr:predicate qudt:unit; rr:object unit:MilliM],
16 [rr:predicate qudt:value; rr:objectMap [rml:reference "x"; rr:
17     datatype xsd:decimal]].
18 # PointYMap and PointZMap are implemented similarly.

```

Listing 5.7: Sample of an [RML](#) mapping definition for lifting [JSON](#) [The+22c].

A Java program based on the `RMLMapper` maps this [JSON](#) data to [RDF](#) using an [RML](#) mapping, shown as a snippet in [listing 5.7](#). The upper part of the mapping, from line 1 to line 10, defines the arrangement of the three dimensions X, Y, and Z (see the `points` in line 5 in [listing 5.6](#)). The lower part of the mapping, from line 12 to line 18, defines the mapping target for the dimension X, including quantities (`qudt:Quantity`) and units like millimeters (`unit:MilliM`) from the respective ontologies. In short, this assigns the unit millimeter to the X-point from line 5 of [listing 5.6](#). The rest of the [JSON](#) data is lifted in a similar way. See [section 5.5](#) for more details and more sophisticated data-uplifting methods. Now that the intercepted messages from the data stream are available in [RDF](#), dataspace participants define validation constraints. An excerpt of the resulting [SHACL](#) file is shown in [listing 5.8](#). It covers the definition of mandatory values (`minCount 1` in line 5), valid value ranges for data points (lines 14 – 15), and data types (`xsd:decimal` in line 16). Note that [SHACL](#) can model complex constraints including logical operators and cross-attribute relations, too. Other examples from the motivating use case include zero-checks, where x and y coordinates of a point are not allowed to be zero (see [listing 5.9](#) for an example [SHACL](#) constraint file). Therefore, the shapes in [listing 5.9](#) define this kind of constraint. See [section 6.7](#) for more details and more sophisticated validation solutions.

Finally, this scenario is run for the second part of this evaluation, the continuous semantic validation service, and the execution time of data lifting using [RML](#) mappings is evaluated.

```

1 ex:Position a rdfs:Class, sh:NodeShape ;
2   sh:property [
3     sh:path ex:x ;
4     sh:minCount 1 ;
5     sh:maxCount 1 ;
6     sh:node ex:XMeterQuantity ; ] .
7
8 ex:XMeterQuantity a sh:NodeShape ;
9   # Additionally references to quantitykind:Length
10  sh:property [
11    sh:path qudt:value ;
12    sh:minExclusive -10 ;
13    sh:maxExclusive 10 ;
14    sh:datatype xsd:decimal ; ] .

```

Listing 5.8: Sample of shapes defining the constraint that x-values of positions must be within -10 and 10 [The+22c].

```

1 ex:Position
2   a rdfs:Class, sh:NodeShape ;
3   sh:not [
4     sh:and (
5       [ sh:path ex:x ; sh:node ex:ZeroValue ; ]
6       [ sh:path ex:y ; sh:node ex:ZeroValue ; ]
7     )] .
8
9 ex:ZeroValue
10  a sh:NodeShape ;
11  sh:property [
12    sh:path qudt:value ;
13    sh:minInclusive 0 ;
14    sh:maxInclusive 0 ;
15    sh:datatype xsd:decimal ; ] .

```

Listing 5.9: Sample of shapes that fulfill the constraint that the x and y paths of a position cannot both have values equal to 0 [The+22c].

Results for the Semantic Validation Service

According to the above evaluation criteria and scenarios, the results for the semantic validation service are shown in [table 5.8](#) and are as follows. This table shows the four evaluation scenarios defined above and their characteristics in terms of whether the syntax and semantics of the two communication parties match. The column *expected* indicates the expected response for these scenarios (block, warn, or communicate) that were derived above. Finally, the columns *baseline* and *our solution* show the results of the execution. Scenarios 1 and 2 represent the previous state of the system, before our semantic extensions were added, and thus decide whether to communicate or not based on syntax only. This leads to communication (*comm.*) for the baseline in scenario 1, while our solution also warns the dataspace participants about possible semantic mismatches, as expected for this scenario. In scenario 2, both compared solutions work as expected. Scenario 3 has matching syntax and semantics, and the expected behavior is to communicate. Our solution does this as expected, and the baseline luckily does so without checking the semantics. Scenario 4 has matching syntax but mismatching semantics and is the most critical because it certainly involves different understanding and interpretation of the communicated content between the two communication parties. The expected behavior is to block this communication to avoid misbehavior of the systems. The baseline only validates the syntax and communicates although no communication is expected. Our solution detects the semantic mismatch and correctly blocks the communication. The system should only initiate communication between two communicating parties when both have the same understanding of syntax and semantics. Our solution provides this in all four scenarios, while the baseline misses the expectation in two of these.

Table 5.8: Comparison of our solution and the baseline with expected values for the four evaluation scenarios [\[Lip+21d\]](#).

Scenario	Syntax / Semantics Match	Expected	Baseline	Our Solution
1	✓ / ?	warn	comm.	warn
2	✗ / ?	block	block	block
3	✓ / ✓	comm.	comm.	comm.
4	✓ / ✗	block	comm.	block

There are two typical situations in practice where the results of [table 5.8](#) play a crucial role. Firstly, when a new client application is introduced, potentially leading to incomplete configurations or incorrect usage of units. Semantic checks identify such issues and notify developers. Secondly, when hardware components change, necessitating adjustments in server code to align with the new hardware. While applications without semantics, like the baseline, might fail, our solution detects errors and prompts dataspace participants to update definitions to meet new system requirements. To conclude the first part of this evaluation, our semantic validation service efficiently prevents and corrects unit mismatches, thereby improving communication predictability. Its backward compatibility facilitates seamless integration with large existing systems. Even in ambiguous cases without semantic definition, warnings are issued while functionality is maintained. Our solution detects and notifies clients of semantic mismatches and stops communication if necessary. Additionally, the human understanding of the services and their code bases is increased with links to ontologies that provide detailed explanations, descriptive comments, and interlinks to others concepts.

Results for the Continuous Semantic Validation Service

According to the above evaluation criteria and scenario, the results for the continuous semantic validation service are shown in [figure 5.30](#) and are as follows.

We performed time measurements for the above data mapping scenario from the motivating use case, which yielded the following results on a 64-bit Windows 10 Intel i7-8650U CPU @ 1.90GHz with 32GB of RAM. The experimental mapping file contained 18 triple maps that extracted up to 24 different predicates, such as constructions of xyz-points or unit assignments. Using Java’s precise `System.nanoTime()` method, we performed and measured five repetitive runs for different numbers of input files. The median runtime was 101 ms for 1 file, 366 ms for 10 files, and 2196 ms for 100 files. The standard deviation was relatively high (360 ms / 278 ms / 830 ms), which we attribute to Java’s one-time class loading overhead, as the first runs always took much longer. Putting these numbers in the context of the real-world use case, the mapping execution times ranged between 22 and 101 ms and needed further improvement. This was due to the high frequency of SOA messages in the USP use case: `JumpToPosition` often occurred at 1-100 Hz, but specialized products with many sub-commands reached over 1 kHz. Sensor readings may have been sent at 100-300 Hz, while camera images and other sensors could reach several kHz. We conclude from these time measurements that our current periodic sampling approach is suitable for experimental validation, but that execution times need to be improved (e.g., by caching mapping files) for use in production.

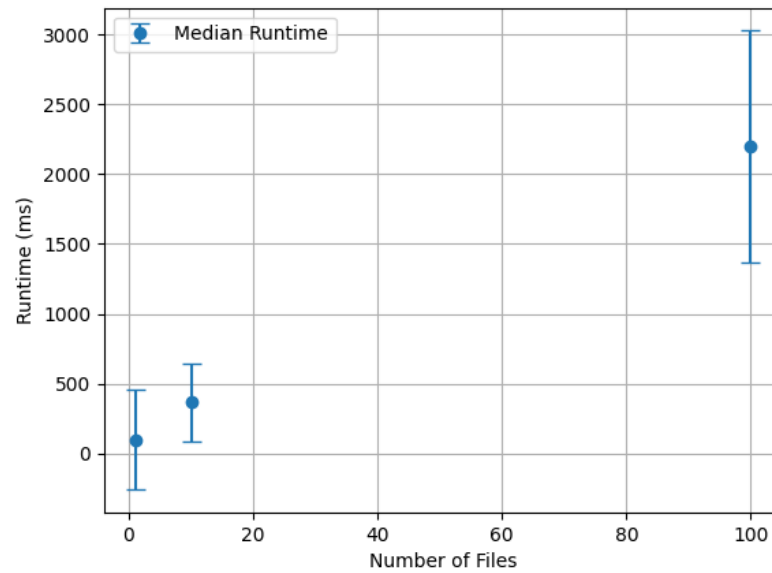


Figure 5.30: Experimental results for mapping data from streams in the continuous semantic validation service [[The+22c](#)].

Evaluation Summary

The evaluation of our proposed approach for managing data and metadata changes in datas-paces focused on two key aspects: the initial semantic validation workflow and the continuous semantic validation service. In the initial semantic validation workflow, our approach

demonstrated its effectiveness in detecting and resolving semantic mismatches between communicating parties, thereby improving the predictability of communication. By comparing our solution to the baseline, we identified scenarios where our approach provided warnings or blocked communication as expected, ensuring mutual understanding between the parties. In addition, our solution could be integrated with existing systems and provide valuable insights through links to ontologies.

In the continuous semantic validation service, our approach showed promising results in mapping data from streams to [RDF](#) format. While execution times showed room for improvement, especially for production use with high-frequency data streams, our periodic sampling approach proved suitable for experimental validation. Further optimizations, such as caching mapping files, may improve the performance of the service for real-world applications.

Overall, our evaluation underscores the effectiveness and practicality of our approach to managing data and metadata changes in dataspace, providing improved communication reliability, and facilitating integration with existing systems. Looking to the future, the evaluated validation system has potential for multiple applications in ensuring high-quality and error-free production systems. With the ability to implement various scenarios, such as online monitoring of production data streams and the creation of manufacturing firewalls for industrial services, it offers flexibility to adapt to different operational needs and improve overall system reliability and efficiency.

Contributions

In summary, this section has effectively addressed the challenge outlined in [requirement Data-R3](#) by proposing a solution for managing dynamic data changes within dataspace. We have emphasized the need for robust strategies to proactively monitor and manage data and metadata changes to ensure data consistency and reliability for evolving data sources in dataspace.

The contributions of this section include:

- **Semantic Validation Service:** These services ensure that exchanged data maintains semantic integrity. The initial validation service checks for consistency prior to communication, thus avoiding misunderstandings between communication parties.
- **Unit Conversion Service:** Automatically corrects unit inconsistencies across data sources, improving interoperability and usability.
- **Continuous Semantic Validation Service:** Monitors and validates data streams against semantic constraints in real time, ensuring continuous data integrity.
- **User Interaction and Workflow Enhancements:** Enhances user experience and workflow processes to facilitate efficient configuration and communication within dataspace.

These contributions collectively advance the field of managing data and metadata changes within dataspace, fostering predictable, confidential, and interoperable communication. By addressing the challenges of common syntax and semantics in communication interfaces such as [SOAs](#), we have paved the way for more predictable and reliable communication, ensuring the continued integrity and usability of data within dataspace.

5.5 Generating Rich Metadata

This section addresses [requirement Data-R4](#) and emphasizes the importance of rich metadata within dataspace. Metadata serves as a critical component, providing essential context and information about the data, including its structure, semantics, and provenance. Due to the decentralized nature of data management in data spaces, most queries will be executed on metadata that is explicitly provided by dataspace participants. The execution of such queries usually does not have access to the actual content of the data, which can lead to incomplete results. Mechanisms that provide selected information from the data itself to the shared metadata of that data, while preserving data sovereignty, could thus improve the accuracy, completeness, and quality of metadata-based queries. We call this process *metadata enriching* or *data uplifting*, because existing basic metadata is enriched with additional information. This process involves mapping pieces of data to [RDF](#), which enhances their utility and relevance within dataspace through the use of common languages and formats. Common languages are achieved through globally unique identifiers in the form of the ontology [URIs](#) and common formats through the web standard [RDF](#). This process is particularly beneficial for source systems with limited semantic descriptions, as it facilitates a deeper understanding and use of data within dataspace. The research area of data uplifting is not new, but it is limited to a fixed set of (semi-)structured input formats such as [CSV](#) or [JSON](#).

Because dataspace manage many data sources with different data formats and a variety of participants with different technical expertise, a research gap remains: Supporting the variety of data formats in data mappings. We address this research gap by exploring existing data uplifting languages and focusing on extending support for arbitrary data formats as input to the mappings. This allows data uplifting methods to be applied to all types of data formats that may occur in dataspace. Specifically, we analyze existing data uplifting languages, propose a methodology for extending them to support arbitrary new input formats, and demonstrate the effectiveness of this methodology by providing and evaluating two example implementations. Applying our contributions to dataspace promotes better understanding and integration of data sources, ultimately improving their interoperability, usability, and value.

5.5.1 Related Work and Background

For a more detailed background on data mapping and data uplifting, see [section 2.2.1](#). In the following, we introduce the data mapping language [RML](#) and two sample data formats from the manufacturing domain (see [section 2.3](#)), which will be used later to demonstrate our proposed solution. These examples include the Industry Foundation Classes ([IFC](#)) format, which is a data schema and exchange file format structure for *Building Information Modeling* data, and [UML](#) class diagrams, which are used to visually represent the structure and relationships of classes in software architectures or data models.

RML

Established as a widely used mapping language, [RML](#) [[Dim+14](#)] builds on W3C recommendations such as *R2RML* [[DSC12](#)] and is influenced by related research such as *RML-star* [[Igl+22](#)]. [RML](#) is driven in part by Dimou et al. [[Dim+13](#); [Dim+20](#)] and allows users to

convert heterogeneous data formats such as [CSV/SQL/XML](#) into [RDF](#) triples. [RML](#) mappings are typically defined in [RDF](#) format using terms from the [RML](#) ontology⁷ and executed by [RML](#) processors. A mapping consists of one or more *TriplesMaps* defining mappings from a source file to [RDF](#) triples. The resulting nodes can be joined to create linked data from heterogeneous sources. While most of a mapping is source-independent and semantically describes the creation of the output graph, source-dependent parts include *iterators* and *references* that access the contents of the input file during the mapping process. [Listing 5.10](#) provides an example [RML](#) mapping that extracts data from a [JSON](#) file by iterating over each “contact” object, creating triples with the contact’s name and phone field.

```

1 <#PhoneMapping> a rr:TriplesMap ;
2   rml:logicalSource [
3     rml:source "phone-numbers.json" ;
4     rml:referenceFormulation ql:JSONPath ;
5     rml:iterator "\$.contacts[*]";
6   rr:subjectMap [
7     rr:template "http://example.com/{name}";
8     rr:class foaf:Person];
9   rr:predicateObjectMap [
10    rr:predicate foaf:phone ;
11    rr:objectMap [ rml:reference "phone" ]].

```

Listing 5.10: Example [RML](#) mapping with an iterator (line 5) and two references (`{name}` in line 7 and `phone` in line 11).

The reference implementation for executing [RML](#) mappings is the *RMLMapper*, an open source Java library⁸ that executes [RML](#) rules from the command line. A tool overview⁹ provides a sophisticated list of processors, graphical user interfaces, wrappers, and supporting tools.

Example Source Format 1: IFC

Defined in the international standard ISO 16739-1:2018, the buildingSMART organization maintains IFC as an open specification for representing 3D building and factory data for construction and facility models throughout their lifecycle. To enable information sharing and collaboration between users and software applications, the standard specifies a data schema for representing facilities with 3D models and their associated metadata. The data schema conforms to the EXPRESS data specification language used to define industrial product data [Int18]. As an example, [listing 5.11](#) contains an excerpt of the definition of a factory door in IFC format. The graph-based structure contains information about the entity, such as height and width, and relationships to other building elements.

```

1 ENTITY IfcDoor
2   SUPERTYPE OF (ONEOF (IfcDoorStandardCase))
3   SUBTYPE OF (IfcBuiltElement);

```

⁷<http://semweb.mmlab.be/ns/rml>

⁸<https://github.com/RMLio/rmlmapper-java>

⁹<https://rml.io/tools/>

```

4 OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
5 OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
6 [...]
7 END_ENTITY;

```

Listing 5.11: Extract from the EXPRESS definition for a `IfcDoor` entity in IFC 4.3.rc.2.

Software support for IFC is provided, for example, by the BIMserver software package [Bee+10], which offers parsing and a Java API to use the contained entities as Java objects. Using the Eclipse Modeling Framework¹⁰, the API is generated directly from the EXPRESS schema. The *IfcOWL* ontology [PT16] was defined by Pauwels and Terkaj to represent IFC data in RDF. Directly related to the EXPRESS schema, it contains classes for each entity type and properties for each relationship.

Several IFC data query languages have been proposed. They aim to provide shortcuts and abstractions from the IFC data model to make it more accessible to authors. BimSPARQL is an extension of SPARQL that allows users to query IfcOWL data more conveniently by defining special operators and geometric algorithms for common use cases [ZBV18]. The *Building Information Modeling* query language *BimQL* [MB13] is a domain-specific language with syntax similar to SQL and SPARQL. It defines shortcuts to hide the technicalities of the EXPRESS data model and allows binding of variables, aggregation and comparison of data. BIMserver comes with a simple filter language¹¹ based on JSON that can filter entities based on their type, attributes and relationships. However, it does not support data transformations such as variable binding or arithmetic operators. Graph-based search algorithms have been proposed to evaluate a query on IFC data [TBS16; INS17].

Example Source Format 2: UML Class Diagrams

The basics of modeling with UML, including the language, concepts, and methods, are described by Rumpe in [Rum16]. In the following, we will focus on the connection between UML and RDF, as these are the source and target of the data mappings in this second example. To support model-driven development processes, the transformation of UML class diagram data into RDF or OWL is of particular interest to reduce modeling effort. Transformation rules for creating OWL ontologies from UML class diagrams have been proposed in [VH20]. Similarly, existing tools such as *UMLtoOWL*¹² allow for automated conversion. However, the available solutions require specific source formats, mostly XML-based, which are already supported by current RML mappers. In practice, manufacturing dataspace participants often design their class diagrams using end-user software such as *Draw.io*¹³ or the *yEd* graph editor¹⁴, which provide their own data formats. This fact on the different data representations was found in an unpublished internal survey in the *Internet of Production* project (see section 2.3). An implementation that converts visualizations of ontologies designed in Draw.io to RDF/OWL was presented by Chávez-Feria et al. [FGP21].

¹⁰<https://eclipse.dev/modeling/emf/>

¹¹<https://github.com/opensourceBIM/BIMserver/wiki/JSON-Queries>

¹²<https://www.sfu.ca/~dgasevic/projects/UMLtoOWL/>

¹³<https://draw.io> or <https://app.diagrams.net/>

¹⁴<https://www.yworks.com/products/yed>

5.5.2 Approach

Our approach to the first research gap, supporting the variety of data formats in data mappings, is visualized in [figure 5.31](#). To address this first research gap, we focus on [RML](#) because it is a widely used mapping language, builds on web standards such as [RDF](#), is a superset of the [W3C](#)-standardized mapping language [R2RML](#) [[DSC12](#)], and is well maintained with rich documentation resources and tools. It involves analyzing the [RML](#) specification¹⁵ and mapper implementations¹⁶, designing and implementing interfaces for the two example source formats mentioned above, and finally proposing a general workflow for extending [RML](#) with new formats.

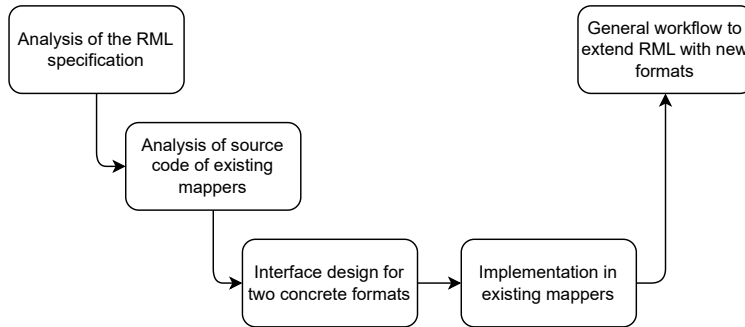


Figure 5.31: Our approach to proposing a general workflow for extending [RML](#) with new formats [[Hoc22](#); [The+24](#)].

The first step, the analysis of the [RML](#) specification, involves a sophisticated examination of the documentation and a (formal) derivation of the language’s core functionalities. The approach to step two, analyzing the implementation of [RML](#), is similar. The demonstrative interface design for the example formats (step three) builds on the results of the previous steps and involves applying the derived general concepts of [RML](#) to the two example formats. The implementation of these two new input formats (step four) then follows these proposed designs. Finally, based on the general [RML](#) analysis and the insights from the two exemplary new input formats, we derive a general workflow that can be used to add any new source format to [RML](#). The goal of this solution is to allow dataspace participants to add any data format to the list of supported [RML](#) data formats, thus enabling the generation of rich metadata for all data sources with this new format.

Our approach to the second research gap, the overwhelming landscape of data mapping languages and tools that overloads many dataspace participants, is to provide an overview of the available solutions. This involves conducting a survey of languages and tools for mapping data to [RDF](#) and presenting the results in a manageable way. Dataspace participants could then, with knowledge of the requirements of their use case, use the overviews provided by us to decide which mapping languages and tools to use.

5.5.3 Results

This section presents the results of our research on generating rich metadata for dataspace through data mappings. We propose an extension to [RML](#) with two new input formats and

¹⁵<https://rml.io/specs/rml/>

¹⁶<https://rml.io/tools/>

a general workflow for extending [RML](#) with new formats. Our results follow the approach illustrated in [figure 5.31](#) and consist of five steps: Analyzing the [RML](#) specification, analyzing the [RML](#) Mapper source code, designing the interfaces of two new example formats, implementing them in a fork of the [RML](#) Mapper code repository, and finally derive a general workflow for extending [RML](#) with new input formats. The results of these five steps are presented below.

Step 1 – Analysis of the RML Specification: In the following, *document* refers to the [RML](#) input file and *mapping* refers to the [RML](#) mapping file. From the scientific publications about [RML](#) mentioned above and from its documentation website¹⁷, we aim to learn a general interface description for accessing data from documents. In [RML](#), documents are composed of *records*. To read and later map data from documents, the functions *I* (iterator) and *E* (reference extractor) define the semantics of two formal languages that read and interpret records from documents. The iterator language *I* is used to specify the records to be converted. Each record is converted into a [RDF](#) node during the iteration loop of [RML](#). Both *I* and *E* are specific to each input format, as are the underlying traversal and query languages are. For example, the iterator *I* for tabular documents such as [CSV](#) files is empty, because there is no record to select, and the reference extractor *E* simply refers to the column names. For [XML](#), I_{XML} and E_{XML} are both defined using *XPath* to traverse through the records within the [XML](#) document and perform selections such as reading attributes within individual records. We suggest the following formal specification. Let D denote the set of documents and R_d denote the set of records contained in a document $d \in D$. Then, *I* maps a document d to a subset of records using a string specified as `rml:iterator` in `rml:LogicalSource`:

$$I(d, i) \in \mathfrak{P}(R_d), \quad d \in D, \quad i \in \text{Strings}$$

I maps a string and a document context to a list of records [[The+24](#)].

After *I* has selected records to map, these records are still in the data format of the document and must be interpreted before they can be mapped to [RDF](#). A *reference extractor E* does this by operating on these selected records and extracting a list of strings using a reference text (or *reference*, for short). The evaluation of a reference represents the actual conversion between data models, as it transforms information from the document’s records into labels, predicates of [URIs](#) in [RDF](#). The translated information from the input document’s data model, in the form of a list of strings, allows direct insertion into [RDF](#):

$$E(r, e) \in \mathfrak{P}(\text{Strings}), \quad r \in R_d, \quad e \in \text{Strings}$$

E maps a string with context of a record to a list of strings [[The+24](#)].

Step 2 – Analysis of Source Code of Existing Mappers: This step consists of researching [RML](#) mapping solutions and selecting one that best meets our requirements for high quality and extensibility of source code, rich documentation, and broad support for [RML](#) features.

¹⁷<https://rml.io/docs/>

We analyzed these [RML](#) mapping tools based on these requirements: The *RMLMapper*¹⁸ as the reference implementation, *RocketRML*¹⁹, and the *SDM-RDFizer* [Igl+20]. In terms of extensibility and code quality, RMLMapper and RocketRML are similarly good and outperform SDM-RDFizer. Interfaces for *I* and *E* can be identified in the source code of both RMLMapper and RocketRML, which successfully decouples the general logic of [RML](#) from support for individual input formats. The reference implementation RMLMapper is guaranteed to cover all [RML](#) features and is therefore chosen for our work. We will fork the open source Java code repository of the RMLMapper²⁰ to a new open source repository²¹ and apply the extensions from the next step in this repository.

Steps 3a and 4a – Extending RML with the Example Format IFC: Our results of extending [RML](#) and the RMLMapper with the first example format, [IFC](#) (see [section 5.5.1](#)), are as follows. [IFC](#) is a structured file format in which entities, their attributes, and interlinks are defined using the EXPRESS schema. To represent such information about entities and application details, a small ontology *ifcrml*²² is introduced (also see [Hoc22]). Following the previous two steps, the definition of *I* and *E* is required for this new input format, which we do by designing and implementing both I_{IFC} and E_{IFC} .

Creating the I_{IFC} iterator involves examining [IFC](#) files and establishing methods to access the records within them. In the context of [IFC](#), we construct [RML](#) records that correspond to the entities defined in the EXPRESS schema. Typically, [IFC](#) files contain many entities, only some of which are relevant to the mapping process. We use the BimQL query language to traverse the relevant records. We use the compatible BIMserver solution to enable support for a [JSON](#)-based filter language designed to select IFC entities. While BimQL is expressive, its limited development and adoption present challenges. To overcome these limitations, we leverage the full expressiveness of Java using the *Function Ontology* [De + 20] to implement a filter function. This Java function receives lists of entities and other parameters and returns processed and mapped IFC entities. The Function Ontology can describe function signatures associated with Java methods, which we combine with the introduced ifcrml ontology for parameter assignment.

Creating the E_{IFC} reference extractor involves extracting information from the records that are returned from the just defined I_{IFC} iterator. Entities in [IFC](#)’s EXPRESS schema, and thus also the [RML](#) records we defined for representing these, contain attributes such as labels or links to other entities via their identifiers. We in [Hoc22; The+24] defined a syntax for extracting such textual information (labels) from attributes by their name. For label attributes, containing values in the form of simple data types, the reference extracts these values. For attributes in the form of cross-references to other entities, these are repetitively followed as defined in the query statement, until a basic label is reached. For the reference extractor E_{IFC} , we suggest to separate references with period (.) characters. Additionally, a pseudo attribute `IfcType` has been implemented that returns the record type by reading the entity’s EXPRESS class name. This can be used with [RML](#) templates to structure [URIs](#) of entities based on their classes, such as the last section of the example `rr:template` `"https://standards.buildingsmart.org/IFC/DEV/IFC2x3/FINAL/OWL/IfcType"`. The

¹⁸<https://github.com/RMLio/rmlmapper-java>

¹⁹<https://github.com/semantifyit/RocketRML>

²⁰<https://github.com/RMLio/rmlmapper-java>

²¹<https://github.com/PHochmann/rmlmapper-java>

²²<https://git.rwth-aachen.de/philipp.hochmann/ifcrml-ontology/-/blob/main/ifcrml.rdf>

described functionality was implemented using BIMserver, which operates on IFC files. We propose a communication architecture for our extended RMLMapper and any generic BIMserver instance as illustrated in figure 5.32. Our extended version of the RMLMapper executes RML mappings with the input format IFC by transforming *.ifc files from a local file system to the BIMserver [Bee+10] for parsing. We implemented an optional filtering via provided queries. The responses from the BIMserver are IFC objects in the *Eclipse Modeling Framework* format (abbreviated with EMF in figure 5.32), which are parsed by the BIMserver client API package in our extended RMLMapper. The designed and implemented Java application finally transforms these received IFC objects to RDF, which completes the RML mapping.

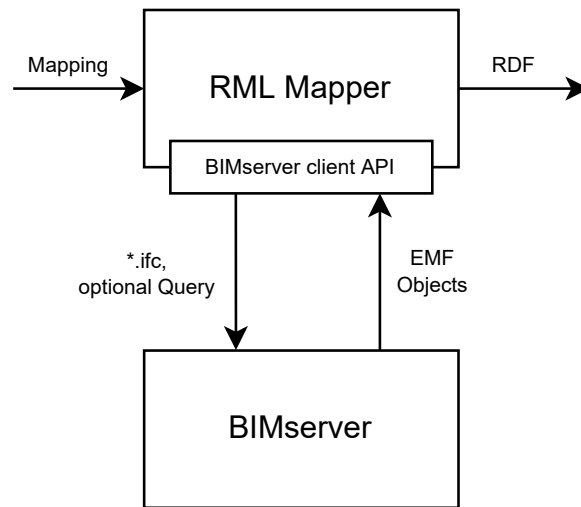


Figure 5.32: Architecture of our extended RMLMapper, which enables the communication of IFC files, user-defined queries, and a final mapping to RDF output.

Steps 3b and 4b – Extending RML with the Example Format Draw.io Class Diagrams:

Our results of extending RML and the RMLMapper with the second example format, UML class diagrams (see section 5.5.1), are as follows. We use the *Draw.io* editor, which supports modeling of UML class diagrams and allows supporting diagrams to XML. The fact that classes in UML diagrams have names, attributes and functions, and links to and from other classes is respected in this export functionality. Thus, the exported XML files contain this information in a machine-readable graph structure. RML already supports the XML format, but does not yet support the data model of Draw.io’s UML class diagrams. To select elements from XML, the XPath language can be used. However, one challenge remains: Class diagrams are generic graphs, but XML documents are tree-structured (see also section 5.2). This complicates the direct extraction of information using XPath, since some application logic is needed to interpret the list of edges in the graph structure in order to follow the arrows of the class diagram. A second challenge arises because the sources and destinations of class diagram arrows are not explicitly expressed in the XML export, but must be computed based on their coordinates. To address both challenges, we will propose a class diagram query language to hide the intricacies of XML serialization from the mapping author.

Creating the I_{UML} iterator involves adapting its design to the input format of class diagrams in XML. A fundamental aspect of this process is the identification of records within the data model. Essentially, the role of the iterator is to access these records and facilitate their

$$\begin{aligned}
S &\rightarrow [\text{Prefix}] \text{ClassSelection} [\text{WhereClause}] \\
\text{Prefix} &\rightarrow \text{attributes\_of\_} \\
\text{Prefix} &\rightarrow \text{usages\_} (\text{of\_} \mid \text{by\_}) \\
\text{WhereClause} &\rightarrow \_ \text{where\_} \text{Restriction} \\
\text{Restriction} &\rightarrow \text{Reference PropOp String} \\
&\quad [\text{PropOp Restriction}]
\end{aligned}$$

Figure 5.33: Extract from the production rules of the proposed context-free grammar for an [RML](#) iterator for [UML](#) class diagrams [[The+24](#); [Hoc22](#)].

availability to the [RML](#) mapping engine. In the context of [UML](#) class diagrams, the data model has similarities to a generic graph, in that classes can be connected by arrows indicating inheritance or dependencies. At the heart of a class diagram are the classes themselves, along with their attributes and the relationships between them. However, relying solely on classes as datasets can lack intuitive clarity. For example, when mapping attributes that satisfy some condition *COND*, dataspace participants must first select classes that potentially contain those attributes. They could then use a reference such as "**attributes.name where COND**" to establish a link between the containing class and the labels of its attributes. While this approach allows multiple values to be extracted with a single reference, linking the class to other resources requires repeating the constraint in each reference. Furthermore, the limitations of [RML](#) become apparent in its inability to later link these attributes through joins, as only nodes originating from records can be joined, potentially compromising the quality of the resulting linked data.

To address these challenges, a simplified data model for class diagrams is proposed, consisting of the record types *classes*, *attributes*, and *arrows* with a limited number of properties. The iterator is conceptualized as a context-free language. An excerpt of the producing grammar is shown in [figure 5.33](#). For the complete language, see [[Hoc22](#)]. The start rule *S* specifies that each iterator consists of a *class selection* and, optionally, a prefix and a where clause. A class selection embodies a recursive statement that encodes a breadth-first search of the diagram, returning a set of selected classes. It serves as the basis for selecting either the classes themselves as records, their attributes, or arrows. The prefixes **attributes of** and **usages of/by** modify the record type. If the prefix is omitted, the classes themselves are extracted as records. A *WhereClause* allows filtering based on individual properties of the record. A simple example is **usages of service** to select all arrows that indicate usage of the *service* class.

Our implementation of these concepts for adding support for *Draw.io* class diagrams exported as [XML](#) files to [RML](#) is as follows. We use a standalone [XML](#) parser to deserialize the exported class diagrams. The different data models (tree vs. graph) are reconciled in our extended Java implementation of the *RMLMapper* by adding application logic to interpret the sets of vertices and edges based on identifiers. As an additional challenge, the [XML](#) export feature of *Draw.io* focuses on the graphical representation of the diagram using shapes, which makes logical connections from classes to their attributes or functions only implicit. We overcome this by introducing geometric algorithms to detect which attributes and functions belong to which class based on their coordinates. Our solution includes tol-

erant geometric detection by adding a threshold for supporting arrows between classes that are nearly connected but not properly connected.

Step 5 – General Workflow to Extend RML With New Formats: Our results for deriving a general workflow for adding new formats to [RML](#) from our results for the two example formats [IFC](#) and [UML](#) class diagrams are as follows. The results of the previous steps showed that the methods *I* and *E* must be defined and implemented to add a new format to [RML](#). Both *I* and *E* depend heavily on the input format and its data model – the iterator *I* for selecting records in documents and the reference extractor *E* for reading information from those records. Note that the form and scope of the records must also be defined, depending on both the format and the requirements of the target application. Our workflow for adding a new format to [RML](#) takes these circumstances into account. It starts with a feasibility analysis, before defining both *I* and *E*, and finally implementing them. The proposed workflow unfolds in three fundamental stages:

Feasibility Analysis: The feasibility analysis is the first step in our proposed workflow. Its primary goal is to evaluate the suitability of a given format for integration with [RML](#). This evaluation includes an examination of both the semantics and the syntax of the format. Semantically, the analysis focuses on the data model inherent in the format, examining its structure to ensure that it conforms to the requirements of [RML](#). Key considerations include the presence of well-defined, enumerable records within the data model, as well as the meaningfulness of the data contained therein for conversion to [RDF](#) resources. In addition, the analysis addresses the syntactic aspects of the format, considering how its syntax affects the technical implementation and effort required for integration with [RML](#). These evaluations serve as basic criteria for determining the feasibility of adding a format to [RML](#).

Design of *I* and *E*: After the feasibility analysis, the focus shifts to the design of the iterator *I* and the reference extractor *E*. These components should be carefully designed to effectively navigate through documents and extract meaningful information from selected records. The iterator language is tailored to balance expressiveness with ease of use, allowing efficient navigation and filtering of records based on their structure. Conversely, the reference extractor language should prioritize simplicity to improve maintainability, focusing on extracting textual attributes without requiring complex navigation within records.

Implementation: With the designs of *I* and *E* established, the final stage involves their implementation. This phase depends on factors such as document serialization and the complexity of the designed languages. First, a parser is used to deserialize the document into objects in a programming language that reflect its data model. Next, the iterator *I* is used to select records, with post-processing as necessary for non-standard record selection. Finally, the reference extractor *E* is implemented to extract text from selected records, completing the integration of the new format into [RML](#).

5.5.4 Assessment of Two Example Extensions

This section presents the results of an assessment of the two example formats [IFC](#) and [UML](#) class diagrams. For each format, a task was derived from a use case in the manufacturing domain (see [section 2.3](#)), our proposed [RML](#) extensions were used to define and execute mappings, and finally observations were made. The metrics evaluated through qualitative observations included usability in terms of how easily mappings can be defined, expressiveness

in terms of whether all required information can be extracted and mapped, and performance in terms of mapping execution time. We measured runtime on an AMD Ryzen 7 PRO 3700U @2.7GHz, Linux 5.10.84-1-MANJARO with OpenJDK 17.0.1 and the `StopWatch`²³ Java class from the Apache Commons *lang3.time* package.

Example Source Format 1: IFC

For the first example source format, **IFC** (see section 5.5.1), an IFC input file is provided by a real-world use case from the *Internet of Production* project (see section 2.3). The provided file represents a factory building, and a visual representation of it is shown in figure 5.34. It uses a three-dimensional structure to describe all the assets of the factory building, their attributes, and links to other assets. The task derived from the use case is to identify all coverings, such as windows or doors, of a given wall and map them to **RDF** using our proposed **RML** extension. The wall is identified by its **GlobalID** attribute, and all coverings should be mapped to named nodes, including a list of attributes of value to the use case, such as dimension, type, material, and transparency.

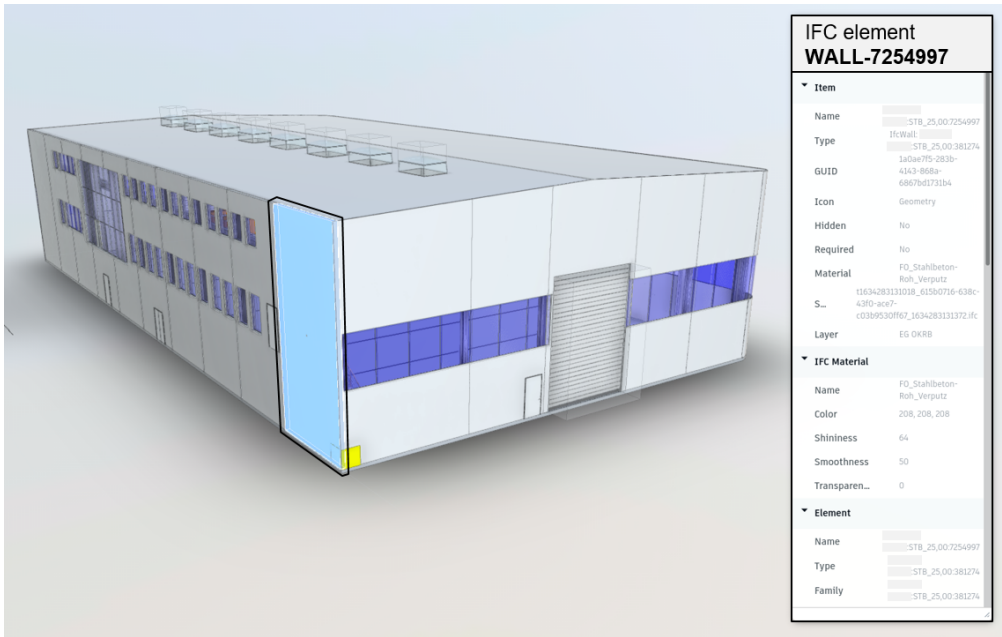


Figure 5.34: Visual representation of an IFC file representing a factory building [The+24; Hoc22].

Usability was demonstrated by the ease of extracting records for RDF mapping, allowing in-depth analysis of pertinent information within milliseconds. Despite challenges such as the limited expressiveness of the BIMserver filter language, the approach facilitated the exploration of use-case specific data. *Expressiveness*, although constrained by the limitations of the BIMserver filter language, was demonstrated by the comprehensive extraction of relevant records for RDF mapping. The approach efficiently handled complex entities such as `IfcOpeningElement` and `RelatedBuildingElement`, demonstrating its ability to capture diverse data elements within the specified use case. *Performance* measurements provided

²³<https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/time/StopWatch.html>

insight into runtime operations. For example, the initial check-in of an 8 MB IFC file containing 140253 entities took 46 seconds, indicating room for optimization. Subsequent operations showed efficiency, with runtime operations taking milliseconds for further steps such as executing the actual mapping. These observations highlight the potential of our proposed RML extension in mapping IFC files to RDF, thus enabling the application of the Semantic Web. However, due to the limited expressiveness of the BIMserver filter language and the complexity of the Java filter function, industrial application may still require specialized expertise.

Example Source Format 2: UML Class Diagrams

Our assessment of support for UML class diagrams includes the task of mapping a diagram to an equivalent ontology in the form of a RDF file. A real-world use case from the *Internet of Production* project serves as the basis for comparison, allowing an assessment of compatibility with existing ontology editors such as Protégé. Although the semantics of the diagram differ from the standard UML class diagram, as a metamodel with a class concept, it shows strong similarities. UML classes correspond to OWL classes, arrow relationships correspond to ObjectProperties, and attributes correspond to DatatypeProperties, with arrows labeled with corresponding ObjectProperty names and attributes specifying DatatypeProperty names.

The *usability* of the RML mapping process proved effective, requiring only basic OWL knowledge and about an hour to formulate the mapping. This minimized the end-user workload by eliminating the need for manual ontology creation in Protégé. Despite its simplicity, the process facilitated in-depth analysis and understanding of the ontology structure. The *expressiveness* of the resulting ontology was satisfactory, capturing the essential components of the class diagram. However, the output was limited to components describable by recognized UML syntax elements, excluding ontology meta-information. *Performance* measurements showed efficient runtime operations, with the mapping process taking about an hour to formulate and a runtime of about 100 milliseconds. Note that the (long) formulation of mappings only needs to be done once, while the (short) execution is usually done many times. This indicates a relatively fast execution, which facilitates further steps such as ontology analysis and application integration. Overall, the use of RML for class diagram conversion was successful in the application scenario, providing a streamlined approach to ontology creation. However, the limited expressiveness of the resulting ontology and the exclusion of meta-information may be limiting in certain contexts.

Contributions

In conclusion, this section has addressed the challenge of enriching metadata within dataspace by proposing a methodology for generating rich metadata through data mappings. By extending RML with support for new input formats, we enable the extraction of metadata from a wide range of data sources, contributing to improved interoperability, usability, and value of dataspace.

The contributions of this section include:

- **Design of a Generic Data Mapping Interface:** Our contributions extend to the design and implementation of interfaces for iterator (*I*) and reference extractor (*E*), tailored to efficiently navigate and extract meaningful information from selected records within various input formats.

- **Extension of RML with New Input Formats:** We have developed a methodology for extending RML to support arbitrary new input formats, enabling the extraction of metadata from diverse data sources, including IFC files and Draw.io class diagrams.
- **Demonstration of Practical Applicability:** TBy implementing and evaluating our methodology on two example formats (IFC and Draw.io UML class diagrams), we have demonstrated its practical applicability in real-world scenarios, highlighting its effectiveness and performance in enriching metadata within dataspace.

These contributions collectively advance the field of metadata enrichment within data spaces by providing a workflow for extending data mappings to support arbitrary data formats. By addressing the challenges posed by heterogeneous data sources, our work improves the availability of rich metadata, thereby facilitating better understanding and integration of data sources. This, in turn, enhances collaboration, interoperability, and decision-making processes in data spaces, ultimately contributing to more efficient and effective use of data for various applications.

5.6 Discussion and Conclusion

This chapter has explored the complex area of heterogeneous data access within dataspace, in line with the challenge outlined in RQ 3: *How can data access from heterogeneous sources, such as manufacturing systems, be effectively facilitated within dataspace to support seamless integration and utilization of diverse data?*

The focus on data as the cornerstone of dataspace necessitated an exploration of methods for seamlessly integrating data from heterogeneous sources, each of which presents unique challenges in terms of format, accessibility, velocity, or metadata availability. From the perspective of the DLCM, specifically the data layer (see figure 5.1), this chapter has aimed to pave the way for a smoother integration process of heterogeneous data into dataspace ecosystems.

We began our journey by clarifying the requirements and methodological approach (section 5.1), recognizing the imperative of ensuring data accessibility, controlled collection, effective change management, and enriched metadata to enhance the usability and relevance of heterogeneous information within dataspace.

Each section of this chapter has contributed distinct insights and solutions to address these requirements, underscoring the multifaceted nature of the challenge at hand:

- **Enabling Data Accessibility from Heterogeneous Systems (section 5.2):** Proposed a solution to improve data accessibility from heterogeneous systems within dataspace. Bridged the gap between low-level query interfaces and higher-level query languages, significantly improving the efficiency and flexibility of data retrieval and metadata collection processes.
- **Adaptive Data Load Management (section 5.3):** Addressed the challenge of managing resource constraints by proposing an adaptive data load management solution for dataspace. Transformed static data collection into adaptive data collection, significantly improving the efficiency and adaptability of data and metadata collection processes.

- **Managing Data and Metadata Changes (section 5.4):** Proposed a solution for managing dynamic data changes within dataspace. Emphasized the need for robust strategies to proactively monitor and manage data and metadata changes to ensure data consistency and reliability for evolving data sources in dataspace.
- **Generating Rich Metadata (section 5.5):** Addressed the challenge of enriching metadata within dataspace by proposing a methodology for generating rich metadata through data mappings. Proposed a methodology for extending RML to support new input formats, enabling the extraction of metadata from a wide range of data sources and contributing to improved interoperability, usability, and value of dataspace.

Together, these efforts are designed to promote the seamless integration and utilization of heterogeneous data within dataspace, aligning with the overarching goal of promoting a cohesive ecosystem of shared knowledge and resources. While significant progress has been made in addressing the challenges outlined in RQ 3, the journey is not yet complete. Future efforts may include rigorous usability evaluations, real-world application validations, and further refinement of proposed solutions. Furthermore, extending our methods to broader domains and diverse participant profiles could enhance the inclusivity and scalability of our approach. In conclusion, this chapter represents a significant step forward in realizing the vision of dataspace as dynamic systems of integrated knowledge and resources. By addressing the complexity of heterogeneous data and developing tailored solutions, we aim to unlock the full potential of data for dataspace. This will enable dataspace and their participants and services to properly manage the core asset, *data*, with confidence and competence.

6 Creating a Common Domain Understanding in Dataspaces

This chapter addresses [RQ 4](#), which questions: *How can reuse of data and knowledge within dataspaces be achieved, and how can recommendations, best practices, and validation mechanisms enhance data and metadata quality to support this reuse?*

Following the [DLCM](#) layer model presented in [figure 3.13](#), this chapter explores strategies and approaches that facilitate the valuable reuse of data and knowledge within dataspaces on the *domain layer* (see [figure 6.1](#)). This figure shows that the phases of the [DLCM](#) on this layer use those from the underneath *dataspace layer*, such as submission, preservation, and discovery of data and services within dataspaces. We also assume that data is available in a dataspace (data layer). The results of the previous chapters particularly include solutions for heterogeneous data access and solutions for sharing such data in dataspaces. Data (re)users in dataspaces need to receive data with appropriate metadata in order to process it properly within their specific applications or domains. The dataspace layer (cf. [figure 3.13](#)) presented in [chapter 4](#) provides basic and domain-agnostic metadata such as general information about participants, data or services within dataspaces. However, proper analysis and (re)use of such content requires domain knowledge, which is represented as the top layer in the figure. An important building block for representing (domain) knowledge are ontologies, see [section 2.2.1](#). Therefore, this chapter aims at a common understanding among stakeholders in dataspaces, as an additional layer on top of the general information a dataspace provides to them.

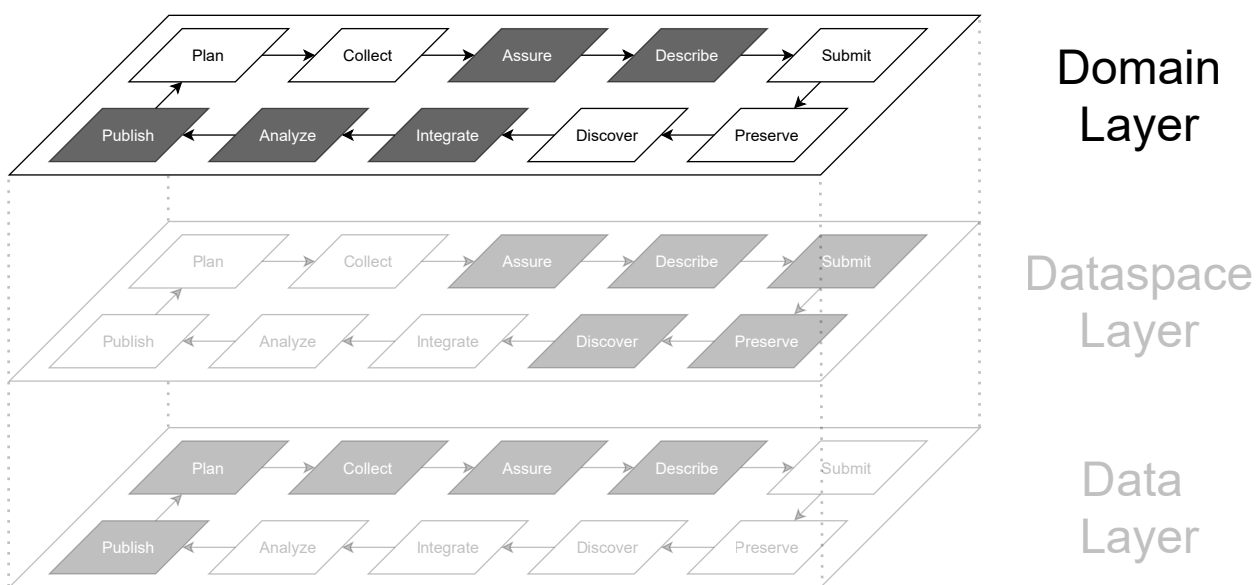


Figure 6.1: This chapter focuses on the *domain layer* of the [DLCM](#) layer model presented in [figure 3.13](#).

Chapter Outline

This chapter aims to create and improve common domain understanding among participants in dataspace and is structured as follows:

- [Section 6.1](#) first presents the requirements and our methodology.
- [Section 6.2](#) discusses techniques for improving reuse, which includes a separation of names and identifiers since common understanding builds on common languages (and technically spoken *common identifiers* for things).
- [Section 6.3](#) subsequently collects best practices to make creation of common vocabularies more intuitive and available for a broad user base. This provides a common foundation.
- [Section 6.4](#) presents a tool for easy vocabulary drafts for non-experts and explores novel technologies to conveniently capture as much human knowledge as possible. This provides helpful tooling.
- [Section 6.5](#) contains recommendations as addition to the process of vocabulary creation, which can promote the reuse of existing (standardized) concepts and identifiers instead of reinventing them. This provides a common use of tools.
- [Section 6.6](#) evolves vocabularies and semantic annotations, providing a common enhancement and interoperability.
- [Section 6.7](#) proposes a technique for enhanced validation, which provides common quality assurance.
- [Section 6.8](#) finally discusses and concludes this chapter, aiming to identify methods that can improve metadata quality and promote the valuable reuse of data and knowledge within dataspace through proper common domain understanding.

6.1 Requirements and Approach

[Chapters 4](#) and [5](#) proposed solutions for the *data* and *dataspace* layer depicted in [figure 6.1](#). The established status quo for the scope of this chapter therefore includes dataspace, which allow sovereign exchange of data: The *data layer* enables access to data from heterogeneous sources, and the *dataspace layer* enables providing, discovering, and exchanging data within dataspace. A major challenge however remains. Data, which participants receive from others in dataspace, must be understood to enable its benefits and make further processing possible. Metadata that accompanies the data can include such extra information to foster a (common) understanding.

Referring to [figure 6.1](#), the **DLCM** phases of interest are *assure*, *describe* and *publish* to (as a provider) properly describe resources to be submitted to others, and *integrate* and *analyze* to (as a consumer) understand resources provided by others. Starting from the goal of creating a common understanding (among participants) in dataspace, we derive the need for common vocabularies. Ontologies (see [section 2.2.1](#)) are commonly referred to as “a formal, explicit specification of a shared conceptualization” [Gru+93]. **RDFS** and **OWL** are the most commonly used ontology languages to express and share (domain) knowledge, which especially applies to the phases *assure*, *describe* and *integrate* (see [figure 6.1](#) and [chapter 6](#)).

Although these two ontology languages have many features that can contribute to a common understanding (among participants), they are not sufficient. In the following, we derive requirements for creating a common understanding on the *domain layer* in dataspace:

Domain-R1 Common Understanding of Identifiers: Ontologies represent shared understanding within a domain, but dataspace contains participants from multiple domains. Interoperability is needed between different domains and even within one domain. This requires a common understanding of what identifiers are (descriptive labels, tags, abbreviations, internal codes, [URIs](#), [URLs](#), [DOIs](#), ...).

Domain-R2 Common Usage of Identifiers: Given a common understanding of identifiers, there are still various ways to use them. This leads to ambiguity and confusion, especially in communication between participants from different domains in a dataspace. A common usage of identifiers, ontologies, and other principles is required. This includes best practices and guidelines for both expert and non-expert participants in dataspace.

Domain-R3 Knowledge Representation for Non-Experts: Widespread adoption of dataspace automatically leads to a broad user base that includes participants who may not be familiar with technical solutions, application domains, or the Semantic Web. Often, only ontology experts are able to adequately express and share knowledge. Application experts, who are the only ones with the required domain knowledge, depend on close and time-consuming collaboration with ontology experts to model information. This leads to the requirement that knowledge expression must be made accessible to all participants in dataspace, including non-experts.

Domain-R4 Common Representation of Knowledge: Dataspace can have many participants and domains, and there are numerous identifiers or ontologies to choose from when representing knowledge¹. Such large-scale and distributed development of vocabularies increasingly causes their contents to drift apart, leading to ambiguities. One solution to this problem is to use (quasi-)standardized vocabularies whenever possible. Given the large number of available identifiers and terms to be reused, methodologies are needed to help users make the right choices.

Domain-R5 Support for Evolution: Semantic annotations of resources, whether created manually or (partially) automated, initially represent only a snapshot of the moment. In an ever-changing and complex world, we need methods to continuously extend, improve, and combine semantic annotations of data and other resources in dataspace.

Domain-R6 Common Shape of Information: In dataspace, both the descriptive vocabularies and the descriptions of the resources themselves are created decentrally and exchanged between participants. However, an integrated use of data (e.g. in mature software solutions) demands precise and correct data. This requires validation principles to ensure that the potentially very different information exchanged in dataspace conforms to certain specifications. Especially on the *domain layer* (cf. [figure 6.1](#)) good but at the same time convenient methods for validation are needed.

¹For example, a repository search for *ontology* on GitHub.com returns 12.4k results, or BioPortal and Linked Open Vocabularies list hundreds of ontologies with hundreds of thousands of terms.

Our methodology in this chapter is to address the above derived requirements [requirements Domain-R1](#) to [Domain-R6](#) in the subsequent [sections 6.2](#) to [6.7](#), respectively, and to propose individual but modular solutions to overall contribute to a better common domain understanding in dataspace.

6.2 Identification Techniques

In this section, we address [requirement Domain-R1](#) and discuss the situation of identifiers for names, validation, and reasoning. This mainly concerns the *describe* and *assure* phases of the domain layer depicted in [figure 6.1](#), but partially also the other phases highlighted there. For general information on identifiers as a technology in the Semantic Web, see [section 2.2.1](#). In the following, we will examine and discuss a rather niche aspect, but one that has a large impact in practice. We investigate how identifiers are commonly used as [URLs](#) in ontologies, which does not sufficiently satisfy [requirement Domain-R1](#). We are working towards meeting this requirement by proposing changes to the current state-of-the-art use of identifiers.

[RDFS](#) and [OWL](#) ontologies (cf. [section 2.2.1](#)) are a common way of expressing information. They simultaneously define concept identifiers, syntactic data structures and relationships, and reasoning rules. This strong coupling significantly hinders the reusability of both ontological concepts and annotated data, due to logical pitfalls in [RDFS](#) and [OWL](#) semantics. To improve reusability, we revisit Tom Gruber’s basic ontology definition [[Gru+93](#)] and reimagine a more decoupled, layered design that independently defines (i) atomic *terms*, (ii) syntactic *shapes*, and (iii) reasoning *rules*.

This decoupling facilitates the appropriate reuse of terms in both shapes and rules, while maintaining flexibility and semantic consistency intact. This enables a precise use in shapes for data validation (e.g. at system interfaces) and highly controllable local reasoning rules that build on existing rules or define new ones. Our approach thus contributes to better term reusability and identifier handling in dataspace by helping to share the right information with others. We present a revised semantic layer model that considers the three layers *terms*, *shapes* and *rules* individually, allowing for flexibility and correctness at the same time. Finally, [section 6.2.3](#) discusses the strengths, weaknesses, opportunities, and threats of our approach in a *SWOT* analysis.

6.2.1 Background and Motivation

Gruber in 1993 defines an ontology as “an explicit specification of a conceptualization”, and subsequently a conceptualization as “an abstract, simplified view of the world we wish to represent for some purpose” (see [[Gru+93](#)] and [section 2.2.1](#)). Simplification with respect to some *purpose* is typically an essential part of limiting the scope of such a specification and aligning it with a given set of requirements. However, it also limits the reusability of ontologies. Taking an ontology designed for one purpose and using it for another often leads to inaccuracies or contradictions. To improve reusability, the Semantic Web community uses modular upper-level ontologies, which typically define easily reusable higher-level concepts, and more specific domain ontologies, which extend upper-level concepts for applications in specific domains. This is however insufficient, especially when mixing multiple ontologies or reusing individual concepts from ontologies with a similar purpose.

There are many reasons why people develop ontologies. In this section, we focus on the core goals of knowledge sharing, validation, and reasoning. The former aims to establish a common understanding of a domain, mostly between humans (cf. [section 2.2.1](#)). Validation ensures that data entered into a system conforms to defined formats and other criteria such as structure or cardinalities (cf. [section 2.2.1](#)). The latter, namely reasoning, infers logical consequences from asserted axioms and thus creates new knowledge (cf. [section 2.2.1](#)).

[RDFS](#) and [OWL](#) ontologies have evolved into one-size-fits-all solutions that try to fulfill all of these tasks at once. [RDFS](#) introduces a class concept as well as hierarchies, and can share knowledge via human-readable properties such as `rdfs:label` or `rdfs:comment`, and [OWL](#) adds additional elements such as annotations, deprecation, named individuals and many more. Both appear to support validation, but actually do not properly: [RDFS](#) introduces restrictions via `rdfs:subClassOf` and `rdfs:domain` and `rdfs:range`, as well as datatype definitions. Reasoning tools can, for example, apply parent class restrictions to a subclass too, or evaluate the correct use of domain, range, and data types. [OWL](#) supports specifying cardinalities, union/intersection/disjointness, equality for both classes and properties, and even complex restrictions, although it operates under an open world assumption which affects its validation capabilities. However, these features lead to issues when applied in a different context (cf. [figure 6.2](#)), which we will elaborate on in the second next paragraph. All of these can, in theory, be evaluated to determine whether a given instance conforms to defined rules. Finally, reasoners can leverage both the domain/range specifications of properties and the hierarchy established via `subClassOf` to infer logical consequences when applying these rules. [OWL](#) also supports a number of axioms that a reasoner can use, including symmetric properties, member relations, and equivalent classes and properties. Note that these ontology elements serve multiple of the above ontology tasks, and thus [RDFS](#) and [OWL](#) ontologies perform all three of these tasks at once, in a very tangled way.

Importing an [RDFS](#) or [OWL](#) ontology poses problems, see the comment for [OWL 2](#) itself: *“The information provided by this ontology may be misleading if not used with care. This ontology SHOULD NOT be imported into OWL ontologies. Importing this file [...] may have other, unexpected, consequences”*². Even without explicitly declaring ontology equivalences, the mere use of multiple ontologies together implicitly imports shapes and rules, and thus in practice forms a global ontology as a by-product. This entailment produces results that may be logically inconsistent or incorrect, leading to an inconsistent knowledge base. Note that the amount of implicitly imported reasoning rules, and thus the inconsistency, multiplies with each new ontology that is reused.

The challenges that hinder the reuse of even simple ontologies are illustrated by the following example. Unknown to many practitioners, even the use of the basic [RDFS](#) domain and domain range predicates can lead to many practical pitfalls and result in factually incorrect statements. [Figure 6.2](#) shows a trivial [RDFS Automotive Ontology](#), which defines the classes `Person` and `Car` and an ownership relationship `owns` between them, and a [Corporate Ontology](#), which initially defines only the classes `Company` and `Machine`. Now suppose you want to express that the individual `ex:eGO` (of type `Company`) owns the `Machine` `ex:CNCMill`. Then it seems reasonable to reuse the existing `a:owns` relationship, defined in the [Automotive Ontology](#), but this entails that `ex:eGO` is also of type `Person` and `ex:CNCMill` of type `Car`, which probably contradicts the intended semantics.

Note that there are efforts in the Semantic Web community to address one of the above chal-

²<https://www.w3.org/2002/07/owl#>

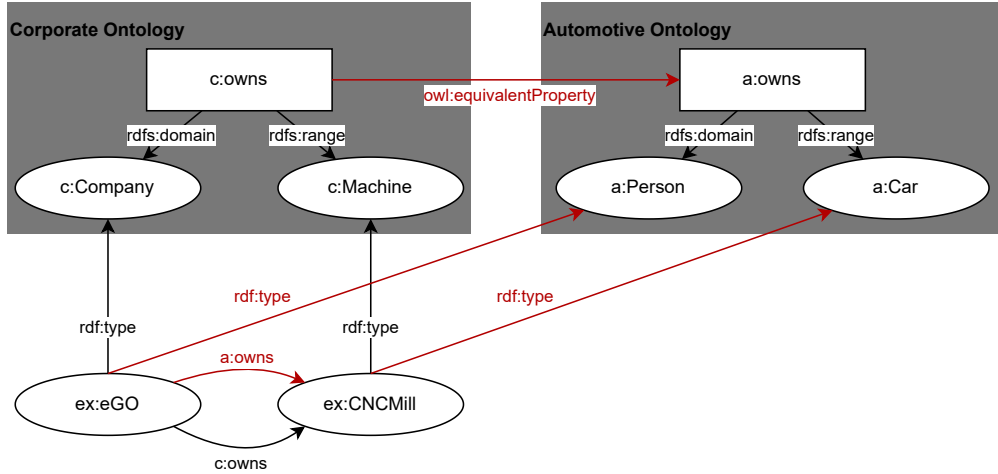


Figure 6.2: Semantic pitfalls of [RDFS](#) and [OWL](#) ontology reuse, illustrated by example of domain and range entailment [LGD20].

lenges by weakening the semantic strictness of domain and range. `schema:domainIncludes`³ and `schema:rangeIncludes` were introduced and, after being a widely used de facto standard for a while, became proper DCMI standards [KB07], see `dcam:domainIncludes`⁴ and `dcam:rangeIncludes`.

To address this challenge, the Semantic Web community introduced the concept of *Graph Shapes* to describe and validate [RDF](#) graphs, namely [ShEx](#) [PLS14] and [SHACL](#) [KK17] (see [section 2.2.1](#)). Currently, there is a need for further development and redesign of key components due to the second rise of the Semantic Web [BHL01], driven by the Internet of Production [Pen+19] and Industry 4.0. In contrast to the original design of the Semantic Web, new requirements have been added by the machine-centered *Internet of Things* and decentralized concepts such as dataspace.

When reusing terms from other ontologies, [RDFS](#) and [OWL](#) automatically adopt any axioms defined there, which acts as a backdoor leading to inconsistencies and problems, as shown in our example. In the following, we present a revised semantic layer model that considers the three layers *terms*, *shapes*, and *rules* individually, thus allowing flexibility and correctness at the same time.

6.2.2 Approach

In the following, we analyze the degree of reusability for the concepts *terms*, *shapes*, and *reasoning*, and then present an approach that decouples these three to increase reusability. We strive for global concepts, shared shapes, and individual reasoning. [Table 6.1](#) defines the above concepts to be decoupled and rates their reusability from high to low. Note that we have grouped [PIDs](#) with terms and synonym sets (synsets), which we describe in detail together with our approach below. The first row refers to any semantic meaning of a thing, such as synsets from WordNet [Mil95] or identifiers from WikiData [VK14] or any other ontology [URI](#) or [PID](#) that is intended to be reused many times. The second row describes the use of shapes that define the structure of a given class (in this context: synset), such

³<https://schema.org/domainIncludes>

⁴<http://purl.org/dc/dcam/domainIncludes>

6 Creating a Common Domain Understanding in Dataspaces

as [ShEx](#) [PLS14] and [SHACL](#) [KK17] (cf. [section 2.2.1](#)), whose reusability is restricted to similar contexts. The last row represents reasoning and inference rules, e.g. with OWL, which are generally not suitable for reuse (cf. [section 6.2.1](#)).

Component	Definition	Reusability
Synsets, terms, PIDs	An atomic concept’s meaning, without reasoning	High
Structure	Structural shape definitions on a syntactical level	Medium
Reasoning	Reasoning and inference rules	Low

Table 6.1: Reusability of the individual concepts terms, structure, and reasoning [LGD20].

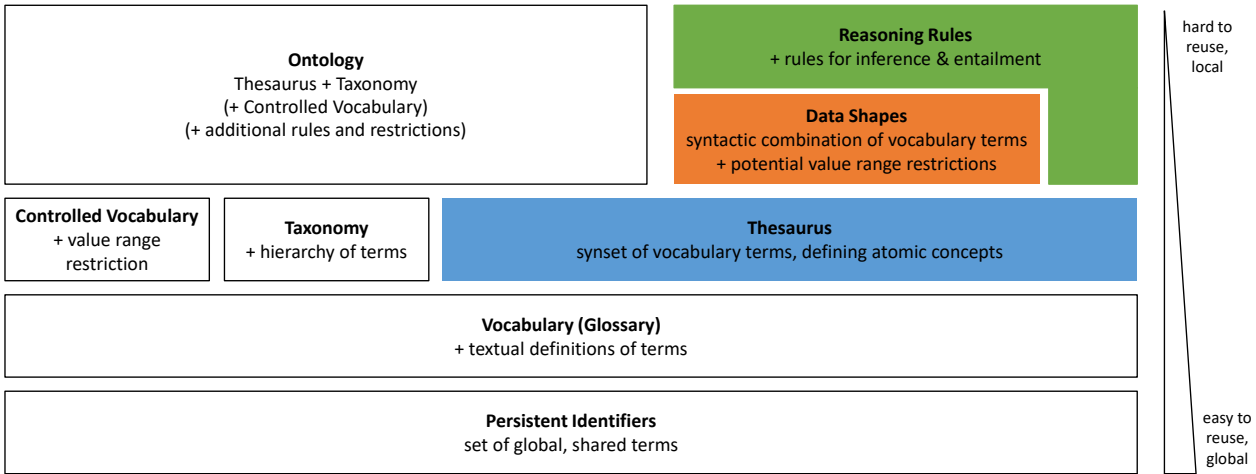


Figure 6.3: Our proposed modifications (marked with colors) to the semantic layer model so that it no longer merges terms, shapes, and rules into ontologies, but makes them available independently [LS20].

[Table 6.1](#) and [figure 6.3](#) show our proposed Semantic Web layer model, with our adaptation marked with colors. We build on state-of-the-art [PIDs](#), vocabularies with textual definitions, and thesauri that group these vocabulary terms into *synsets*. Our envision is that synsets are defined in an open, community-driven, global repository that allows for flexible and collaborative addition and definition. Corresponding maturity or relevance ratings should further guide users in term selection, to promote convergence and reuse of established terms. On top of this layer, we let users describe the syntactic structure of data through shapes, which others can reuse or extend for their applications. Finally, users define reasoning rules as usual, but independent of the shapes and without the necessity of sharing them coupled with the above shapes.

Our proposal includes the suggestion to share only the synsets (e.g. classes/concepts) without any structure or reasoning rules. That is, use identifiers for [WikiData](#) or similar knowledge bases to uniquely identify the meaning, but disregard any [OWL](#) (or other) reasoning rules here. For a particular use case or application, one defines the expected structure for that class via shapes. These shapes should be shared by all parties that will be communicating data that should be validated with these shapes. These may be internal or distributed components of a single application, or even external agents or user interfaces.

In practice, it may be useful to uniquely identify and share created shapes with others for interoperability or collaborative development purposes. We therefore motivate the implementation of a shape directory that allows both humans and machines to query existing

shapes for any concept, e.g. from WikiData. For a synset [PID](#), this query would return a list of shapes previously defined by the community, and could later be extended to include ranking features. Note that we do not typically recommend sharing instances of the third layer, the reasoning rules, between participants, as these face significant complications when shared across domains (cf. [Section 6.2.1](#)) - but users can still do so if desired. This approach facilitates proper reuse of synsets, shapes, and reasoning rules without implicitly importing unwanted axioms from the other domain.

6.2.3 Discussion

To dissect the strengths, weaknesses, opportunities, and threats of the proposed approach, we analyze it using the *SWOT* methodology [[Hum05](#)]:

Strengths

- Enables proper reasoning
- Full support for already existing shapes
- Reusing ontology classes without burden

Weaknesses

- Missing tool support
- Reasoning rules are not inherited and thus need to be defined explicitly

Opportunities

- Quick drafts without the need to fully understand all relevant ontologies
- Simple, dedicated tool support
- Further adoption in applications
- Community-driven shape development

Threats

- Low acceptance in the community, which mostly focuses on well-known standards

In terms of **strengths**, the approach demonstrates an ability to facilitate thoughtful analysis, provide comprehensive support for existing shapes, and allow for the reuse of ontology classes without unnecessary complexity. This aspect contributes to the potential effectiveness of our approach in fostering a robust foundation for domain understanding. However, it is important to acknowledge **weaknesses**, such as the current lack of robust tool support and the need to explicitly define reasoning rules, which potentially introduces a layer of complexity that deserves consideration.

In exploring the **opportunities**, the methodology offers the advantage of generating quick drafts without requiring a full understanding of all relevant ontologies. In addition, the potential for streamlined, dedicated tool support enhances its appeal, suggesting broader applicability across diverse domains. The envisioned increased adoption in various applications and the prospect of community-driven shape development further underscore potential avenues for growth and refinement.

However, inherent **threats** remain, most notably the potential for low adoption within the Semantic Web community that largely favors well-established standards. This underscores the importance of overcoming potential resistance and building a case for the relevance and effectiveness of the approach in the broader context of dataspaces.

To address the weaknesses and threats identified in the SWOT analysis, a multi-faceted approach can be taken. Improving tool support is essential, and this can be achieved by developing and integrating user-friendly tools that simplify the process of defining reasoning rules and ensure that they are seamlessly offered. Encouraging community involvement and collaboration can mitigate the risk of low adoption by raising awareness and demonstrating the practical benefits of the approach. In addition, fostering partnerships with key stakeholders and industry leaders to endorse and adopt the system can further strengthen its standing and acceptance within the community and ensure that it complements and enhances existing standards.

In summary, this SWOT analysis provides an examination of the proposed approach, offering valuable insights into both its potential benefits and challenges. These considerations are central to a balanced understanding of the viability of our proposed approach and its positioning within dataspaces.

Contributions

In this section, we have addressed [requirement Domain-R1](#). We identified the strong coupling of identifiers, structure, and reasoning in [RDFS](#) and [OWL](#) ontologies as a critical pitfall when sharing them. This is because it implies false assertions. The main contribution of this section is a revised layer model for the Semantic Web that separates these three concepts into individual layers, allowing them to be used independently. In this proposed layer model, identifiers, terms or synsets are atomic concepts without any reasoning and should be shared globally and collaboratively. Shapes describing the data structure are application specific and should only be shared and adapted in similar domains. A repository listing available shapes for given synsets could support usability. Finally, in this model, reasoning rules are defined and used for specific use cases, independent of the structure layer. This allows for independent reuse or adoption of either concept in other applications, without breaking the intended semantics through unintended implicit imports. Our proposal still needs further discussion and revision, as discussed in a *SWOT* analysis. But it is a good step toward [requirement Domain-R1](#) because it improves the perception and use of identifiers, which is beneficial for dataspaces.

6.3 Best Practices for Vocabulary Creation

This section deals with [requirement Domain-R2](#), which is about the need for best practices and guidelines for the handling of vocabularies, especially for non-experts. This mainly concerns the *describe* and *assure* phases of the domain layer illustrated in [figure 6.1](#), but partially also the other phases highlighted there. The knowledge that is created, shared, and (re)used in these phases is usually expressed by formal descriptions based on ontologies. Information in dataspaces is shared in a common domain-agnostic core (see [chapter 4](#)) and in domain-specific extensions such as the manufacturing domain (see [section 2.3](#)). Both, but especially the latter, require two groups of actors, namely domain experts and knowledge

engineers (or Semantic Web experts), to collaborate and complement each other to create semantic models that are sound in both fields. The domain experts usually do not have sufficient knowledge to create proper semantic models, and the knowledge engineers lack proper expertise in the specific domain. This creates a cycle of knowledge dependency that is even worse when one of these roles is underrepresented in organizations.

The challenges of ontology development tasks are various: First, there are too many ontology development methodologies to choose the right one in a user-friendly, non-expert oriented way [Hit21]. Second, existing tools are not easily compatible in some cases, and fundamental aspects such as usability and performance are usually overlooked [Vig+14]. Last, ontologies are just now yet widely adopted in many domains such as manufacturing or mobility (see [section 2.3](#)) because of various reasons [GP09; PDK22; Vig+14]:

1. Domain experts or decision makers do not see the benefits of using ontologies to model knowledge.
2. Hidden best practices in the literature
3. Lack of adequate tools to support ontology development and maintenance (see [section 6.4](#))
4. Limited support for creating or reusing ontologies
5. The [OWL](#) modeling language is complicated and also has a broader scope than semantic modeling, e.g. ontology reasoning, which is not often used in projects where only a semantic model needs to be created (see [section 6.2](#)).
6. Issues related to collaborative ontology development, such as difficulties in reaching consensus
7. Ontologies are often large and segmented, and therefore difficult to understand.

In the following, we will research on best practices for vocabulary creation in this context. We first cite and contextualize related work in this research field. We then describe our research methodology and our data collection, before we present our results and evaluation. The outcomes of this section includes a guideline booklet for the process of ontology development that includes a descriptive workflow, best practices, templates, and tool recommendations. This contribution addresses [requirement Domain-R2](#) and helps domain experts and knowledge engineers to collaborate more efficiently to create semantic models.

6.3.1 Related Work

The related work in this section consists of two parts: First, related work on the process of ontology development and guidelines for it, which also includes an extension of the background chapter on ontology development tools, modeling languages, validation or reasoning (see [section 2.2.1](#), respectively). And second, considerations in the domain of the Internet of Production (see [section 2.3](#)), where we will integrate and evaluate our results.

Ontology Development Guidelines

In general, the process of ontology development involves four main steps [NM+01; Rud+16; SS15]: (i) identify the purpose of the ontology, (ii) build the ontology, (iii) integrate it

with existing ontologies, and (iv) document the ontology. Figure 6.4 shows an iterative process with seven steps, which was introduced by Noy and McGuinness in 2001 [NM+01]. Similarly, Rudnicki et al. in [Rud+16] introduced a set of steps for creating an ontology. Both approaches first involve preliminary work to define the scope and then include defining and refining terms, setting the class hierarchy, modeling properties, and modeling constraints.

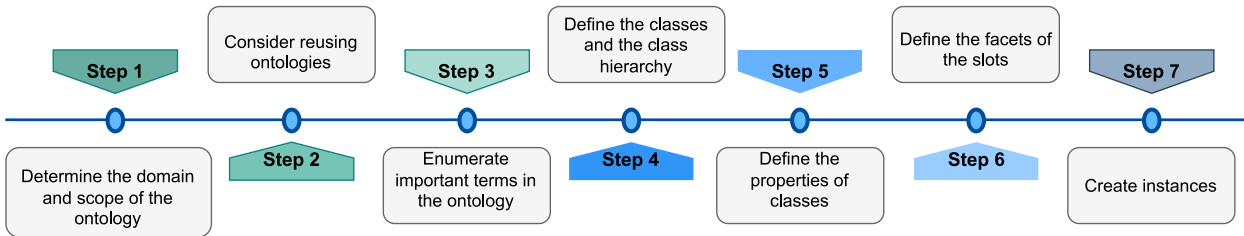


Figure 6.4: Based on [NM+01], the iterative steps required to develop an ontology [Com22a].

Besides this iterative process, ontology development also concerns how ontologies are created, which is *top down*, *bottom up*, or a *combination* of these [NM+01; Zho+21]. Though all these approaches aim at building an ontology, the result may be different depending on the strategy chosen, because they approach class definition from different perspectives. In the *top-down* approach, the most general concepts in the domain are first defined as a data schema and then a gradual refinement of terms is applied. In contrast, in the *bottom-up* approach, the most specific classes are first defined and then grouped into more general concepts. Finally, in the *combination* approach, one first models the concepts that stand out and then applies generalization and specialization to these terms according to their suitability.

Decisions in the development of an ontology can depend on the classification of such. The classification usually corresponds to the level of detail and logical formalism in which the ontologies describe concepts [Ram+21]. Ocker et al. in [OVP19] presented a comprehensive classification based on [ASS15; LDU10], in which the ontology types are: *top-level*, *reference*, *application* and *domain* ontologies, see figure 6.5.

A final consideration, which may affect the modularization of ontologies, is the use of ontologies [Wac+01]: (i) Single ontology, which uses a global ontology as a common vocabulary, (ii) multiple ontologies, in which each information source is described by an independent ontology, and (iii) hybrid, which is a combination of the two.

In this context of ontology development, the hybrid approach is usually preferred, as it allows the addition of new information sources without changing the mapping, and at the same time supports the acquisition and evolution of the ontology [PM13].

Noy and McGuinness [NM+01] emphasize that ontology modeling is a dynamic, iterative process where the representation of classes should closely align with their physical or logical counterparts. To enhance ontology modeling across various domains, several key considerations include:

1. Evaluating the necessity of introducing new classes.
2. Choosing between introducing a new class or a property value.
3. Defining and modeling class disjointness.
4. Ensuring the correctness of class hierarchies.

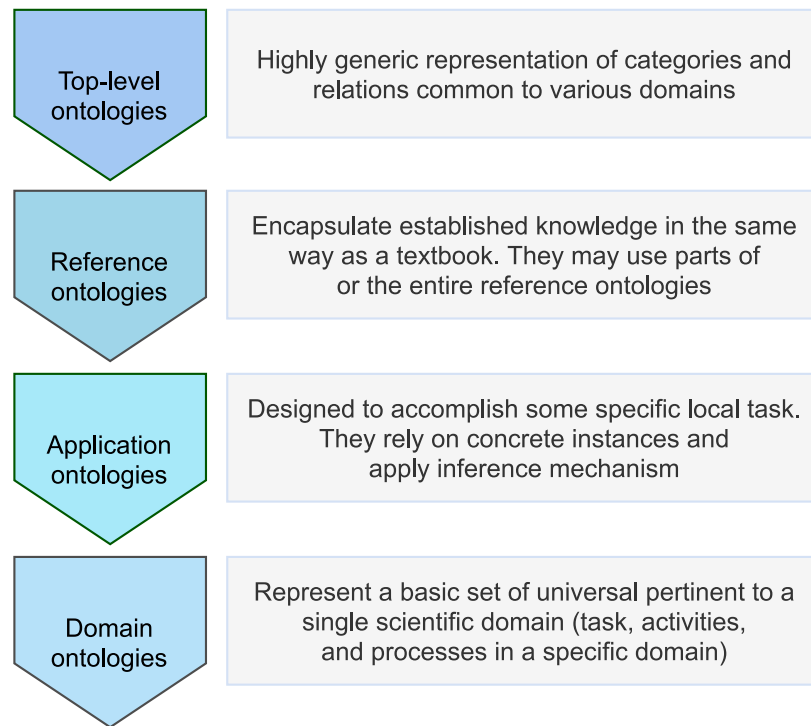


Figure 6.5: Based on [OVP19], the classification of ontologies [Com22a].

Concerning the fourth point and the ontology classifications outlined in the previous section, it is advisable to adopt the *Basic Formal Ontology* as the top-level ontology [Rud+16]. Understanding the intentional structure of ontological descriptions is fundamental, as “ontology” is rooted in philosophical concepts. Based on [Neu+22], these intentional structures encompass: (i) The intention to identify things (continuants) and (ii) the intention to explain things (occurents).

Furthermore, [Ins20] recommends to prioritize the reuse of ontologies over the construction of completely new ontologies from scratch, as this saves time and improves interoperability with existing ontologies. In this context, [GP09] advocates the use of *ontology design patterns*, as they serve as reusable, modular building blocks. Such patterns help to identify the entities to be modeled and provide versatile solutions to recurring modeling challenges [JMK15; SHH19]. To ensure practicality, flexibility, and usability, the focus should be on easily understandable and adaptable ontology design patterns [Cam+22]. However, it’s worth noting that despite their potential, these patterns are underutilized due to the variety of design patterns and lack of comprehensive documentation [JMK15]. There are additional best practices to consider, such as those presented in [JBV98; Rud+16; Zho+21], to help avoid the creation of heterogeneous and non-reusable ontologies by providing a supportive framework. On a collaborative level, [Vig+14] suggests a best practice where a knowledge engineer supports domain experts in modeling, at least at the beginning of the modeling process.

Methodologies for ontology development can be classified based on the collaboration of actors (*non-collaborative*, *collaborative*, and *custom*) or based on the approach underlying the methodology (*waterfall*, *lifecycles*, and *agile*) [SP21]. Moreover, there are methodologies which focus on ontology maintenance or on the preliminary tasks of the engineering process, including knowledge acquisition and scope. Additionally, and in line with the classification presented in [SP21], there are also ontologies covering the entire process of ontology develop-

ment, while others tackle the development process from a traditional perspective following an iterative or waterfall-like approach [GP09; JBV98; SS15; SSS09; YBA21]. On the other hand, some methodologies favor the agile perspective [ADH19; AH06; Hog+21; OPP19; Per16; SHH21]. The main features of agile methodologies are their flexibility, adaptability to rapid changes, making them suitable to different areas, contributing to these areas' success. Furthermore, agile methodologies support adaptive collaboration between domain experts and knowledge engineers [AH06]. Although agile methodologies may seem to have advantages over their traditional counterparts, they are not always the best choice for large projects. Therefore, [DD09; SP21] motivate to first analyze the characteristics of the project to be tackled, and then choose the best approach. Multiple studies evaluated methodologies for ontology development, such as [Alf22; CFG03; SDV22; Tud20]. Psaromantis et al. in [PDK22] adopted collaborative aspects from software development [All+21] for dealing with ontologies as source code.

Additional considerations include that there is no sound methodology in line with every domain and case and that it is possible to combine them, resulting in a derived data-driven methodology [Hip+21]. It is also important to mention that approaches bringing significant improvements to the whole ontology engineering process exist, such as ontology modularization and pattern-based ontology modeling [SHH19]. Moreover, there are also several techniques for ontology reuse as indicated in [RT20].

Considerations for the Manufacturing Domain

In this section, we extend our description of the Internet of Production and the manufacturing domain (see section 2.3) with specifics regarding ontology development, because we evaluate our results in this field.

Frameworks and models that facilitate the design and creation of ontologies include the *Reference Framework for Digitalisation 4.0* [Int16b; Pla21], which consists of business layers, life-cycle and value Streams, and hierarchy levels. It offers the participants involved in the manufacturing process a common understanding, and provides a digital (and virtual) representation of all the information in a manufacturing environment. In practice, one should represent the stored digital representation of the manufacturing environment as a formal model using W3C standards to provide semantics in the manufacturing domain [BCB21b]. As soon as ontologies are created, stakeholders need to first focus on *top-level* ontologies to support consistent ontology development across domains [OVP19], and adopt a *top-down* method to model assets, products, and relationships [Zho+21].

Another good practice for designing large ontologies in manufacturing is to do it in a modular form because modular ontologies are usually more reusable, extensible, and maintainable [BCB21b; SS15]. There exist techniques to support the modularization process [Le+22] and modeling principles tailored for the domain of production and manufacturing [BCB21b; PDK22; Zho+21]. Yahya et al. in [YBA21] encourage developers to consider different types of models in the manufacturing domain to describe products, processes, and resources. Challenges and limitations for the semantic models in manufacturing still include production line models not following *Linked Open Data* principles, application-specific limitations of semantic models, and a lack of real-time data availability [YBA21].

In summary, the state of the art of available vocabularies and best practices for vocabulary creation is broad in scope and application, and deep in technical solutions. However, the situation is still challenging because there are too many ontology development methodologies

to choose the right one, and the available methods are often not user-friendly or non-expert oriented enough. Existing tools are often not easily adaptable to use cases and usually lack aspects such as usability and performance. There are many domain-agnostic and domain-specific vocabularies available, but practical reasons often hinder their use, including unclear benefits, lack of tool support, complicated languages such as [OWL](#), or ontologies being too large or complex.

6.3.2 Approach

We in the following address [requirement Domain-R2](#) by providing best practices and guidelines for the handling of vocabularies, especially for non-experts. Our approach is illustrated in [figure 6.6](#) and consists of two data collection stages and three subsequent sections as follows.

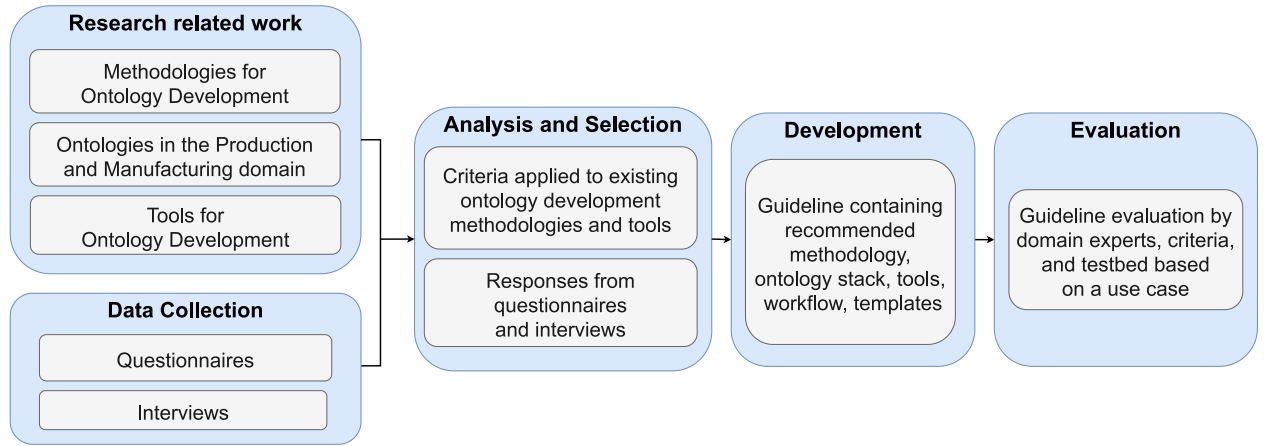


Figure 6.6: The methodology we use in this section to reach to the development and evaluation of best practices for vocabulary creation [\[Com22a\]](#).

The research on related work includes a snowballing research [\[Woh14\]](#) approach with initial keywords including *ontology development*, *agile methods*, *product ontology*, *i4.0 ontologies*, *manufacturing ontologies*, *ontology testing*, and *guideline best practices*. We complete our data collection with questionnaires on general knowledge of stakeholders, on specific problems of domain experts and knowledge engineers in knowledge modeling processes, and on domain-specific competence questions for an evaluation, respectively.

[Figure 6.7](#) presents our methodology for the first two steps in *research related work* depicted in [figure 6.6](#), which involves a literature review, classification of methodologies, criteria definition, and weighting. The process of ontology development consists of several steps, causing confusion among domain experts regarding the appropriate methodologies to use. We address this by conducting a comparative analysis of design patterns and differences in these methodologies (see [\[Sat+20\]](#)), defining criteria and weights to evaluate them. Based on this assessment, we propose a suitable ontology development methodology. We first establish evaluation criteria, drawing from literature, interviews, and questionnaires. We then create an evaluation matrix, assigning scores to each methodology per criterion. The top methodologies are selected, potentially combined, or the highest-scoring one is recommended. Agile methodologies are considered, given their adaptability to our dynamic domain, alongside non-agile approaches (waterfall, iterative).

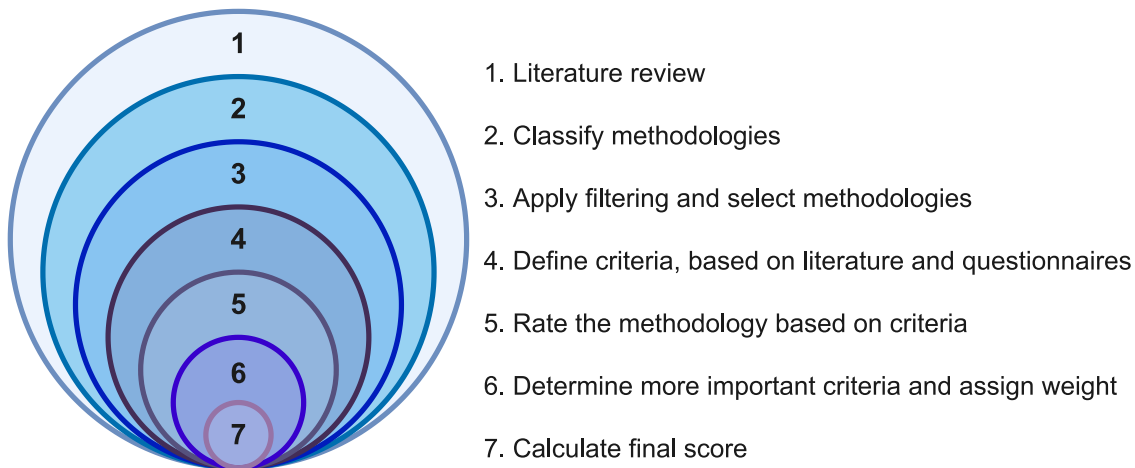


Figure 6.7: Our methodology to evaluate methodologies for ontology development starts with a literature review and incrementally evaluates the findings in multiple steps [Com22a].

The third step of *research related work* in figure 6.6 is on tool support for ontology development and includes a study of such tools and an evaluation based on identified indicators such as user-friendliness, feedback provided, license type (pricing), or collaboration. The approach is similar to the one depicted in figure 6.7 and consists of the steps: (i) Literature review, (ii) group tools based on functionality, (iii) apply filtering and select tools, (iv) define criteria, (v) rate the tools based on criteria, and (vi) calculate final scores.

The final two steps of our methodology (see figure 6.6) are the development of a guideline and a evaluation. The former includes guidance for ontology development in the form of recommended methodologies, ontology stacks, tool support, workflow, and templates. The latter involves domain experts and criteria in an evaluation in a use case from the manufacturing domain.

The outcome of this section is a guideline booklet for ontology development in the manufacturing domain, more specifically in the Internet of Production. Its structure is illustrated in figure 6.8 and the guideline aims to support domain experts in designing and creating ontologies more intuitively and independently.

6.3.3 Results

We in the following present the results of our approach (see figure 6.6). Finally, we apply the *ADDIE* model [Bra78; Cul22], which is short for analysis, design, development, implementation, and evaluation, to build a guideline booklet for best practices in vocabulary creation.

Methodologies for Ontology Development

There are methodologies for ontology development and such for ontology engineering per se, which we both present chronologically from the mid nineties up to today in tables 6.2 and 6.3. Practical requirements include methodologies to follow a consensus [NM+01], provide available resources, have non-expert users in scope, and be applicable to general domain use cases.

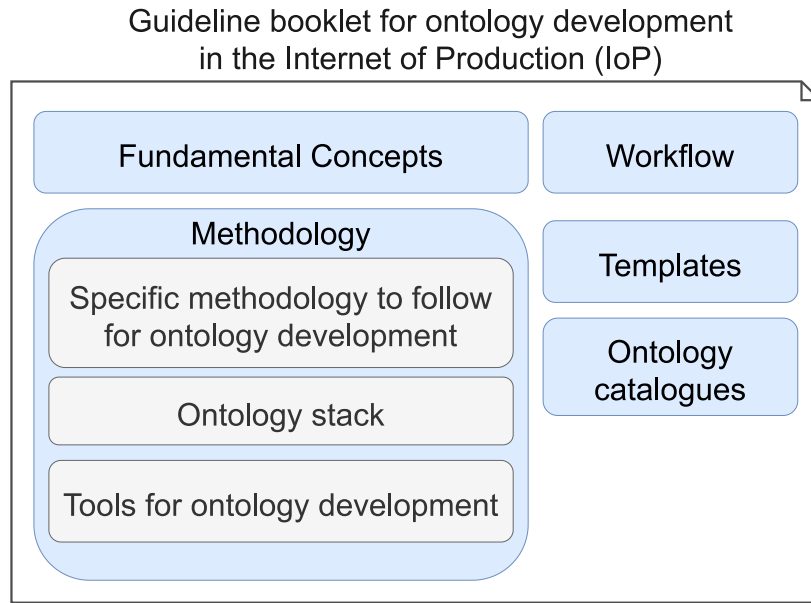


Figure 6.8: The outcome of this section is a guideline booklet for the process of ontology development, which contains fundamental concepts, methodologies, a suggested workflow, templates, and ontology catalogs [Com22a].

Therefore, table 6.3 distinguishes whether the presented citations meet these requirements. For a detailed investigation, see [Com22a].

Table 6.2: Guidelines and principles for ontology development [Com22a], distinguishing between defined principles (PRN) and other considerations (OTH).

Approach	Focus	Year	Type
MENELAS [Bou+94]	Taxonomic ontology development	1994	PRN
PLINIUS [JBV98]	General ontology development	1994	PRN
Ontolingua [JBV98]	Modular ontology development	1995	PRN
CommonKADS [JBV98]	Modular design, redesign and reuse	1995	OTH
Mikrokosmos [Mah96]	Design choices	1996	OTH
Physsys [JBV98]	Dynamic knowledge construction	1996	OTH
ODPs [GP09]	Recurrent ontology design problems	2009	OTH
BO [GR12]	Obtain reduced versions from larger ontologies	2012	OTH
BPOD [Rud+16]	Best practices in ontology development	2016	PRN
MIRO [Mat+18]	Change management and ontology description	2018	PRN

The *analysis and selection* stage of our approach shown in figure 6.6 contains the collection and application of evaluation criteria for ontology development methodologies. The gathered criteria are presented in figures 6.9 and 6.10, measured and accumulated, and the final scores are listed in table 6.4.

Table 6.3: Methodologies for ontology development and their primary focus over the years [Com22a]. The last column indicates whether the methodologies meet practical requirements.

Approach	Focus	Year	Meets Req.?
IDEF5 [JBV98]	Ontology development, modification, maintenance	1994	
TOVE [GF94]	Ontologies evaluation and maintenance	1994	
EM [Usc96]	Separate levels of formality	1996	
Methontology [FGJ97]	Ontologies maintenance. Evolving prototypes	1997	
ONIONS [GPS99]	Ontologies acquisition	1999	
OD101 [NM+01]	Ontology development steps and best practices	2001	✓
OTKM [SSS04]	Ontology development, maintenance, management	2004	✓
DILIGENT [Pin09]	Whole process of ontology development. Local vs global changes	2004	✓
HCOME [KV06]	Ontology evaluation in knowledge work	2005	✓
UPON [DMN05]	Use-case driven, based on Unified Process	2005	✓
RapidOWL [AH06]	Knowledge extraction, structure and processing	2006	✓
ODMPA [BLC08]	Ontology development in public administration domain	2008	✓
NeOn [PBV08]	Ontology development (Scenario-based)	2008	✓
DOGMA [STM08]	Whole process of ontology development	2008	✓
XD [Pre+09]	Problem space and solution space identification	2009	✓
OEM [SSS09]	Ontology development and maintenance (Process-oriented)	2009	✓
COLM [Luc09]	Ontologies maintenance	2009	✓
AMSACO [PMS10]	Ontology development in the business domain	2010	
KEADO [Yun+11]	Ontology development of domain ontologies	2011	✓
OntoClippy [Dah11]	Tool-supported ontology development	2011	
ANEMONE [ÖÖÜ11]	Modular ontology development	2011	
GMCAO [Zhe+12]	Ontology development in the agricultural domain	2012	
YAMO [DCM15]	Facet ontology development	2015	✓
MOM [KH16]	Modular approach with ontology design patterns	2016	✓
SAMOD [Per16]	Test-Driven ontology development	2016	✓
UPON Lite [DM16]	Rapid prototyping and ontology development	2016	✓
IIAM [JSS16]	Ontology development in the education domain	2016	
LOT [Cas17]	Industrial method for ontology development	2017	✓
YAMO+ [MCD17]	Ontology development of analytical core ontologies	2017	
MetROn [TKC18]	Ontology development of reusable ontologies	2018	✓
XOD [He+18]	Ontology development focusing on interoperability	2018	✓
AMOD [ADH19]	Agile and customizable ontology development	2019	✓
OntoDI [Yun+19]	Ontology development process on data integration	2019	✓
MOMo [SHH21]	Modular ontology development with design patterns	2021	✓
BCIO [Mic+21]	Iterative steps	2021	

C₁. Detailed Guidelines About OntoDevProcess

The methodology includes detailed information about the ontology development process, with a detailed explanation.

C₂. Use Of ODPs

The methodology includes information about the use of ODPs and recommends their use.

C₃. Supports Collaborative OntoDev

The methodology is suitable for collaborative ontology development and includes information on how to collaborate.

C₄. Supports Ontology Reuse

The methodology recommends reusing existing ontologies and building the new ontology by integrating existing ones.

C₅. Methodology Type

The possible methodology types are Agile, Waterfall, and Lifecycle Support.

C₆. Lifecycle Coverage

The methodology indicates the stages it covers in the ontology development process (ontology creation, ontology evaluation, etc.)

C₇. Defines CQs And Functional Requirements

The methodology indicates how to define competency questions and how to consider other functional requirements.

C₈. Supports Ontology Maintainability

The methodology indicates how it supports the maintainability of the ontology over time.

C₉. Supports Ontology Modifiability

The methodology indicates how it supports modifications in the ontology over time.

C₁₀. Supports Ontology Extensibility

The methodology indicates how it supports extending the ontology over time to cover new knowledge.

C₁₁. Recommends Tools

The methodology includes information on which tools and standards are more suitable for supporting the ontology development process.

C₁₂. Recommends Best Practices

The methodology includes information on the best practices to follow when creating and maintaining ontologies.

C₁₃. Supports Interoperability

The methodology supports the use of top-level ontologies and the integration of different ontologies.

C₁₄. Suitable For Different Users

The methodology considers different levels of knowledge and participation of actors.

C₁₅. Methodology Is Rooted In Previous Ones

The methodology builds on other approaches. It has taken into account previous methodologies and is based on them.

Figure 6.9: Description of criteria defined based on the literature review to evaluate methodologies for ontology development [Com22a].

C₁₆. Suitability To Our Domain

The methodology is suitable for our domain. It can be used for developing ontologies in the manufacturing and production domain.

C₁₇. Provides Templates

The methodology provides templates to use in the ontology development process.

C₁₈. Specifies When To Build Ontologies

The methodology provides information on when to create ontologies, explaining the need for ontologies.

C₁₉. Covers Existing Ontologies In The Domain

The methodology recommends existing ontologies in the manufacturing and production domain.

C₂₀. Provides Examples

The methodology includes examples explaining the ontology development process.

C₂₁. Technical Considerations

The methodology provides information about the more technical considerations, such as name conventions, domain and range, etc.

C₂₂. Result Oriented Approach

The methodology defines deliverables for different stages of the ontology development process.

C₂₃. Covers Problem Definition

The methodology provides guidelines to help in the problem definition.

C₂₄. Examples Of Complex Ontologies

The methodology provides examples of how to model complex ontologies.

C₂₅. Modelling Alternatives

The methodology includes information on alternatives to model the same problem.

C₂₆. Well-explained Workflow

The methodology provides a well-explained workflow covering the ontology development process.

C₂₇. Close Collaboration With Domain Experts

The methodology includes information on how to closely collaborate with domain experts.

Figure 6.10: Description of criteria defined based on our analysis of questionnaires results to evaluate methodologies for ontology development [Com22a].

Table 6.4: Evaluation of methodologies for ontology development. The final results are in descending order according to the final scores obtained by each methodology [Com22a]. See table 6.3 for details and citations for the approaches.

Approach	Final Score	Approach	Final Score
LOT	98.5	HCOME	71
NeOn	94	UPON	69
MOMo	91.5	OD101	68
AMOD	91	KEADO	67.5
XD	86.5	RapidOWL	66.5
YAMO	79.5	OntoDI	65.5
DILIGENT	78	COLM	65
MOM	77.5	OEM	64
SAMOD	77	MetROn	62
XOD	76.5	DOGMA	61
UPON Lite	74.5	OTKM	59
ODMPA	72.5		

In summary, we recommend using the best-scoring methodology *Linked Open Terms (LOT)* introduced in [Pov+22], which is available at <https://lot.linkeddata.es>. We recommend it because it is a reasonably comprehensive methodology, and because it is an evolving approach, it covers several aspects that need to be considered during the ontology development process. The methodology focuses on engineering aspects and provides valuable insights for engineers from different domains. It also recommends tools and best practices, and covers the entire ontology development process, including the later stages of ontology evaluation, documentation, and maintenance.

Ontology Development Tools

Following our approach shown in figure 6.6, we in the following present ontology development tools. We separate the illustrations into rather technical tools that fulfill particular tasks and tools that address multiple stages of the ontology development process. These lists have commonalities with those presented in section 2.2.1 and will be evaluated against requirements derived from the collaboration of domain experts and knowledge engineers. Among other requirements (see [Com22a]), the tools should (i) have an active development or community, (ii) not be limited to a single domain, (iii) be accessible for non-technical users, (iv) not be proprietary or closed source, and (v) provide a user interface and a documentation.

Table 6.5: Technical tools supporting the ontology development process [Com22a]. The *REF* column indicates online references of the tools.

Automation	REF	Reasoners	REF	Validation	REF
ROBOT	ROBOT	HermiT	HermiT	ShEx	ShEx
OnToology	OnToology	ELK	ELK	SWRL	SWRL
RDFLib	RDFLib	Pellet	Pellet	shaclex	shaclex
r2rml-parser	r2rml-parser	Fact++	Fact++	Turtle Validator	TurtleV
RDF2rml	RDF2rml	HyLAR	HyLAR	RDFUnit	RDFUnit
OntoMaven	OntoMaven	CEL	CEL		
OWL API	owlAPI	jcel	jcel		
Tawny-OWL	tawny-owl	openllet	openllet		
AR2DTool	AR2DTool	Racer	Racer		
owl2vcs	owl2vcs	kaon2	kaon2		
owlready2	owlready2	OWL-RL	OWL-RL		
dosdp-tools	dosdp-tools				
RDFSharp	RDFSharp				

Frameworks	REF	Serialization	REF	Triple Stores	REF
Apachen Jena	ApacheJena	Turtle	Turtle	Profium Sense	Profium
Jena	Jena	N-Quads	n-triples	Blazegraph	Blazegraph
ODK	ODK	RDF/XML	RDF/XML	RDFstore-js	RDFstore

Queries	REF	Others	REF
SPARQL-DL	SPARQL-DL	Corese	Corese
DL Query tab	DLQuery	sublime-text-turtle-sparql	sublime-text-turtle
Profium Sense	ProfiumSense	linked-data.syntaxes	SyntaxHighlight
Blazegraph	Blazegraph	OntoVerbal	OntoVerbal
RDFstore-js	RDFstore-js	CLaRO	CLaRO

Table 6.6: Tools supporting multiple stages of the ontology development process [Com22a].
The *REF* column indicates online references of the tools.

	Tool	Remarks	REF
Miscellaneous	How to diff RDF	Multiple implementations for comparing RDF files	HowToDiffRDF
	ChImp	Allows visualisation of changes in ontologies and their impact	ChImp
	MODL	Curated collection of well-documented ontology design patterns	MODL
	Graphol	Visual language for developing ontologies	Graphol
	Chowlk Converter	OWL implementation from UML diagrams	Chowlk
	FrODO	Drafting ontologies from competency questions automatically	FrODO
	Grafoo	Graphical framework for OWL ontologies,	Grafoo
	Semantic Turkey	Knowledge management and acquisition	SemanticTurkey
	Gra.fo	Visual tool for designing KGs	Gra.fo
	OWLAx	Generates axioms from diagrams	OWLAx
	CameoConcept	Hides the formal model behind its graphical user interface	CameoConcept
Languages	RDFS	Data-modeling vocabulary for RDF data	RDFS
	OWL	Knowledge representation about things	OWL2
	RDF	Represents a directed graph containing a triple	RDF
	SHACL	Description and validation of RDF graphs	SHACL
	Graphol	Graphical language for ontology modeling	Graphol
	SWRL	Enables combination of Horn-like rules with OWL ontologies	SWRL
	OTTR	Language for representing and instantiating RDF graph and OWL ontology modeling patterns	OTTR

(Continues on next page...)

(...Continued from previous page)

Tools supporting multiple stages of the ontology development process [Com22a]. The *REF* column indicates online references of the tools.

	Tool	Remarks	REF
Editors	Protégé	Knowledge management	Protégé
	WebProtégé	Web-based collaborative knowledge management	Protégé
	WebVOWLEditor	Visual modeling with guidance	WebVOWLEditor
	DogmaModeler	Ontology modeling tool designed for non-IT experts	DogmaModeler
	TopBraidComposer	Multi-purpose Semantic Web modeling and querying tool	TopBraidComposer
	OWLGrEd	Multi-purpose graphical ontology editor	OWLGrEd
	Fluent Editor	Multi-purpose editor for ontologies	FluentEditor
	Neologism 2.0	Graph editor for quick vocabulary creation	Neologism2.0
	CoModIDE	Modular ontology development	CoModIDE
	VocBench	Collaborative editor for ontologies	VocBench
	Unilexicon	Taxonomy manager	Unilexicon
	VoCol	Integrated environment for collaborative ontology development	VoCol
	VoCoReg	An extension of VoCol	VoCoReg
	Eddy	A graphical editor for ontologies in OWL 2	Eddy
Visualisations	VOWL	User-oriented ontologies visualisations	Vowl
	WebVOWL	Web application for interactive visualizations	WebVOWL
	ontology-visualization	Visualization	ontology-visualization
	ontodia	Data visualisation, navigation, and exploration	ontodia
	RDFDot	Drawing graphs from RDF files	RDFDot
	visualRDF	Graphical visualization of RDF graphs	visualRDF
	OWLGrEd	Multi-purpose graphical ontology editor	OWLGrEd
Docs	Grapholscape	Advanced web viewer for ontologies written in Graphol	Grapholscape
	Widoco	Customized documentation of ontologies by GUI and command line	Widoco
	LODE	Service for ontologies documentation	LODE

(Continues on next page...)

(...Continued from previous page)

Tools supporting multiple stages of the ontology development process [Com22a]. The *REF* column indicates online references of the tools.

	Tool	Remarks	REF
Validator	SHACL	Description and validation of RDF graphs	SHACL
	ShEx	Description and validation of RDF graphs	ShEx
	RDF Validator	Validate and visualise RDF/XML files	RDFValidator
	RDFUnit	Read and produce complying RDF	RDFUnit
	SHACL Play	Online RDF data validation with SHACL	SHACLPlay
	OOPS!	Allows common pitfalls detection in an ontology	OOPS!
	Themis	Executes tests and verify the ontology	Themis
	TDDOnto	Test-Driven ontology development	TDDOnto
	OntoCheck	Verifies ontology naming conventions and metadata completeness	OntoCheck
	OntoDebug	Supports the user in finding faulty axioms in ontologies	OntoDebug
Querying	SPARQL	Query languages and protocols to manipulate triples from RDF graphs	SPARQL
	SPARQL Playground	Examples of SPARQL queries	Playground
	Sparnatural	Visual SPARQL query builder	Sparnatural
	sparqlblocks	Build SPARQL queries with blocks	CameoConcept
	Visual SPARQL	Build SPARQL queries	VisualSPARQL
	BigCQ	A dataset of competency question templates mapping to SPARQL-OWL templates	BigCQ
Toolkits	NeOn Toolkit	Supports the whole ontology development process	NeOnToolkit
	Ontoanimal tools	Set of tools to help achieve Linked (Open) Data	OntoZoo
	PoolParty Suite	Knowledge management tool	PoolParty
	TopQuadrant Suite	Set of tools for semantic solutions	TopQuadrant
	Mondeca	Set of tools for semantic solutions	Mondeca
	Mobi	Collaborative KG platform to publish and discover data, data models, and analytics	Mobi
	AtomGraph	KG management	AtomGraph
	AllegroGraph	Set of tools for semantic solutions	AllegroGraph
	Ontotext	Set of tools for semantic solutions	Ontotext
	Stardog	Set of tools for semantic solutions	Stardog
	Semafora	Set of tools for semantic solutions	Semafora
	Data World	Set of tools for semantic solutions	DataWorld
	Refinitiv	Set of tools for semantic solutions	Refinitiv
	Ontop	Virtual KG system with several features	Ontop
	Anzo Suite	Set of tools for semantic solutions	AnzoSuite

The *analysis and selection* stage of our approach shown in [figure 6.6](#) contains the collection and application of evaluation criteria for ontology development tools. The gathered criteria are presented in [figure 6.11](#), measured and accumulated, and the resulting tools with the highest scores are listed in [table 6.6](#). Note that we in [section 6.4](#) propose an improvement to the *editor* type, which makes vocabulary creation even more accessible for non-experts.

C₁. Suitability The tool solves the intended task.	C₆. Interoperability The tool supports interoperability with functionalities such as importing, exporting and merging files.
C₂. Feedback The tool provides feedback to the user-based input.	C₇. Versioning The tool supports the versioning of the ontology files and related outputs.
C₃. User-friendliness The tool is easy to understand and use.	C₈. Usability The tool is easy to use, especially for non-technical users (DEs), i.e. no extensive technical or programmatically skills are required.
C₄. Collaboration The tool allows for the collaboration of different participants involved in OntoDevProcess.	C₉. Training There are available tutorials and documentation related to the tool.
C₅. Extensibility The tool is extensible, for example by plug-ins.	C₁₀. Software License The possible values are: free and open, and non-free

Figure 6.11: Description of criteria defined based on literature review and our analysis of questionnaires results to evaluate tools for ontology development [[Com22a](#)].

Table 6.6: The tool recommendations for the process of ontology development, grouped by their typical use [[Com22a](#)].

Type	Tools
Editor	Protégé Desktop, WebProtégé
Visualize	WebVOWL
Validate	OOPS!, Themis
Documentation	Widoco
Version Control	Git (e.g., GitHub or GitLab)

Existing Ontologies and Ontology Catalogs

Following our approach shown in [figure 6.6](#), we in the following present both domain and supporting ontologies and classify them by their coverage. We also investigate ontology catalogs, which are services that provide appropriate ontologies for users to (re)use in their applications. [Table 6.7](#) illustrates existing upper and middle ontologies, and [table 6.8](#) lists ontologies from the manufacturing domain (see [section 2.3](#)), including a matching of the ontologies to

areas such as *products*, *processes* or *sensors*. The acronyms in this table represent: Manufacturing, products, processes, resources, plant components, activities, scheduling, maintenance, sensors, robotics, engineering, batch processing, measurements, standards, and simulation.

Table 6.7: Upper and middle ontologies to consider in ontology development [Com22a].

Name	Focus
SUMO [NP01b]	Linguistics, search, reasoning
OntoCape [Muñ+09]	Process engineering
BFO [SKB05]	Information retrieval, analysis and integration in various domains
Cyc [RL+02]	Concepts and rules about the world
PROTON [Cas+05]	Semantic annotation and information retrieval
Sowa's [MCR08]	Logical, linguistics and philosophical concepts
UFO [Gui+22]	Concepts about Philosophy, Linguistics, Cognitive Science
UMBEL [Yab09]	Subject concepts (a distinct subset of the more broadly concepts)
WordNet [Pri22]	Similar to a Thesaurus. Represent concepts
DBpedia [DBp22]	Describes the concepts related to DBpedia
DOLCE [BM10]	The intuitive and cognitive bias underlying common-sense
GFO [Her10]	Categories such as objects, processes, time and space, relations, roles, etc.
KYOTO [Vos+08]	Encode Natural Language concepts
YAMATO [MT17]	Quality and quantity, objects, processes and events
COSMO [Gui21]	Broad semantic interoperability
gist [Sem22]	Maximum coverage of typical business ontology concepts
MASON [Sem22]	Automatic cost estimation and manufacturing simulation
SIMPM [SS19]	Model three fundamental constraints of manufacturing process planning: variety, time, and aggregation
MSDL [Kje+09]	Represents conventional manufacturing processes

Analyzed and recommended supporting ontologies (see section 2.2.1) include ORG [Rey12], Time [Cox+17], vCard [IM14], SKOS [MB09], SKOS-XL, FOAF [BM00], QUDT, DCMI-Terms [KB07], RDFDataCube [CRT14], QB4ST, OM [RAT13], Metadata4Ing [Arn+22], DCAT [Alb+19], PROV-O [Leb+13], VoID [Ale+11], and DQV. Ontology catalog services can additionally support domain experts when creating vocabularies. Practical requirements include that ontology catalogs need to be available, up-to-date, location-independent, and not limited to irrelevant domains. Therefore, table 6.9 distinguishes whether the presented citations meet these requirements. See [Com22a] for details, explanation, and further references.

All results and insights created above are accumulated into a guideline booklet entitled “*Guidelines for the Creation of Semantic Models in the Internet of Production (IoP)*” [Com22b], which incorporates the workflow proposed in figure 6.12.

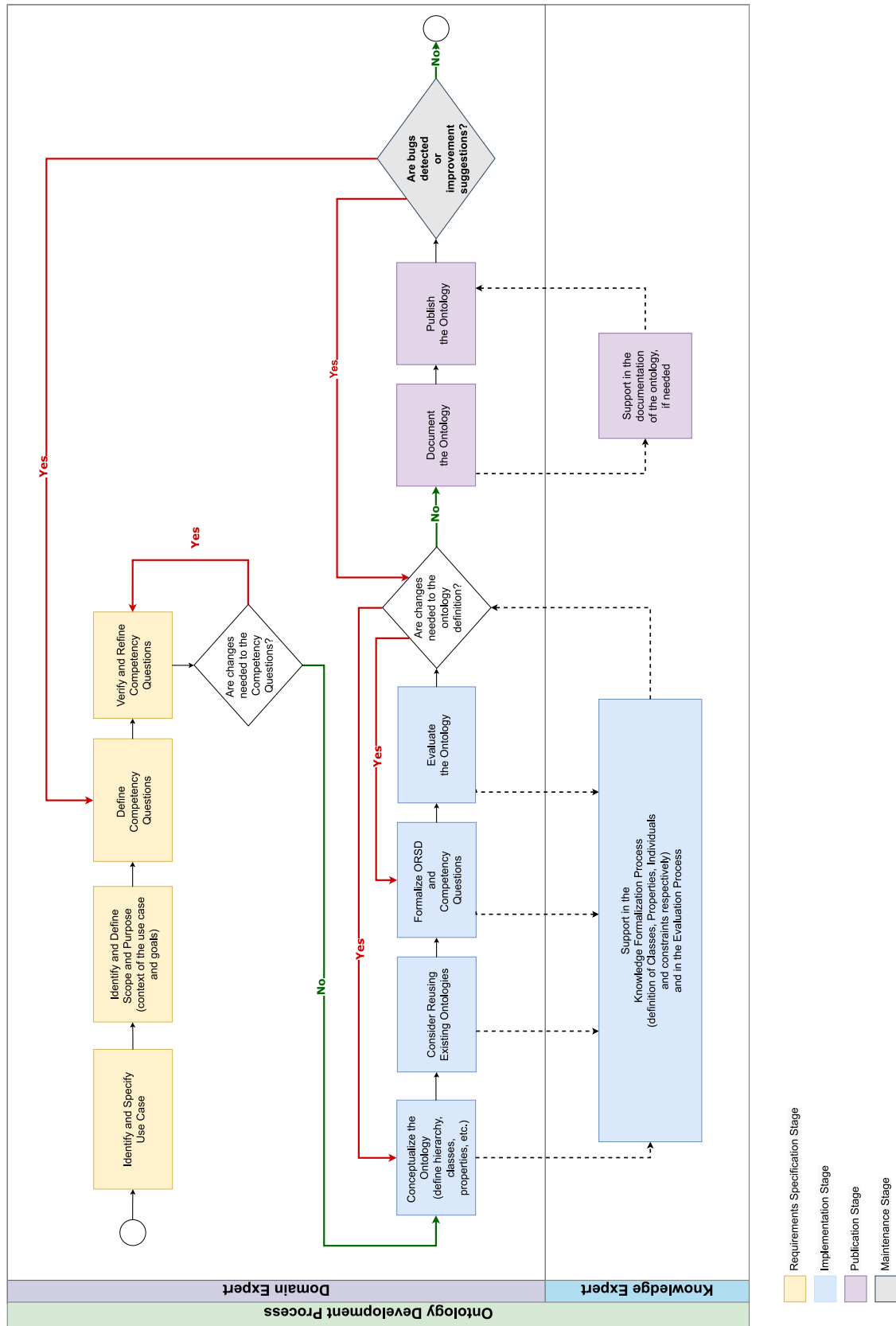


Figure 6.12: The proposal of a workflow to use during the process of ontology development considering close collaboration between domain experts and knowledge engineers [Com22a].

Table 6.8: Existing manufacturing domain ontologies and their addressed fields (see [section 2.3](#)) [[Com22a](#)]. Acronyms are explained in the text reference for the table.

	MFG	PROD	PROC	RES	P.COMP	ACTV	SHED	MAINT	SENS	ROBT	ENG	BATCH	MSMT	STD	SIM
SOIL [Göp+21]	✓	✓	✓	✓									✓		
iFAB [OVP19]	✓														
ADACOR [OVP19]	✓		✓		✓	✓									
MASON [BCB21b ; Lem+06]	✓	✓	✓	✓											
MaSDeM [OVP19]	✓	✓	✓	✓							✓				
PSL [BCB21b]	✓		✓			✓	✓								✓
OntoCAPE [OVP19]			✓								✓				
BaPrOn [Muñ+09]			✓			✓						✓			
FABMAS [MS03]	✓	✓		✓		✓	✓								
PrOnto [OVP19 ; YBA21]	✓	✓	✓		✓										
ARUM [HV15]	✓	✓	✓	✓			✓								
RGOM [YBA21]	✓	✓	✓	✓			✓	✓						✓	
ONTO-PDM [PDT12]	✓	✓	✓	✓			✓							✓	
MPMO [Cao+20]	✓		✓	✓							✓				
MCCO [OVP19]	✓	✓													
SCRO [BCB21a]	✓		✓	✓	✓	✓									
MSDL [OVP19]	✓	✓	✓	✓											
AMU [Kai+21]	✓	✓	✓	✓	✓										
MPO [Ram+14]	✓		✓	✓							✓				
IMAMO [Kai+21]	✓			✓	✓			✓	✓						
IEO [OVP19]	✓	✓	✓	✓	✓							✓			
COMP. [COM21]	✓	✓	✓	✓											
MaRCO [RBI20]	✓			✓											
ExtruOnt [RBI20]	✓		✓	✓						✓					
OntoProg [Cao+22]	✓			✓				✓							
Onto-ICMS [Cao+19]	✓		✓	✓	✓	✓		✓							
AMLO [Kov+18]	✓	✓	✓	✓	✓				✓		✓			✓	
WSSN [Ben+12]								✓							
M3 [Gyr+15]									✓	✓			✓	✓	
CSIRO-SO [NC09]			✓						✓				✓		
OPW [Ayi+21]	✓	✓	✓	✓		✓									
CDM-Core [Maz+16 ; Kai+21]	✓		✓				✓			✓			✓	✓	
ROMAIN [Kar+19 ; Kai+21]			✓	✓				✓							
STO [Gra+17]	✓													✓	
SSN [ECF20]				✓					✓						
newSSN [ECF20]				✓					✓						
AWARE				✓					✓	✓					
OntologySim [May+22]	✓	✓	✓	✓		✓									✓
CMSD [BLK18]	✓	✓	✓	✓											✓

Table 6.9: Ontology catalogs with their link and description [Com22a]. The last column indicates whether the catalogs meet practical requirements.

Name	Link	Description	Meets Req.?
VoCol Service	vocoreg.com	It provides access to ontologies by term search and a full list of ontologies	✓
NFDI4Ing Terminology Service	nfdi4ing.de	It has a search option by terms and a list of ontologies	✓
Linked Open Vocabularies	lov.linkeddata.es	It provides filtering options by tag, for example <i>industry</i> . It also has a full list of vocabularies and terms	✓
Ontohub	ontohub.org	It has different filters such as ontology type and formality level	✓
Bartoc	bartoc.org	It offers a list of vocabularies. It has a search option and also a filtering system	✓
Finnish Ontology Library Service	onki.fi	It provides access to Finnish and international ontologies, vocabularies and thesauri	
Vocabularies for Open Data for Cities	vocab.linkeddata.es	It provides a list of ontologies. It is also possible to filter by domain or ontology title	
OntoCommons	ontocommons	A list of ontologies in the manufacturing domain and other industries	✓
BioPortal	bioportal	A repository of biomedical ontologies	
vocab.org	vocab.org	It provides a small list of vocabularies	
GitHub	github.com	A search by keyword <i>ontology</i>	✓
ontologi.es	ontologi.es	It provides a small list of ontologies and vocabularies	
AberOWL	aber-owl.net	It provides search by terms, but only related to biological data	
Ontology Repository	ontologies/lib/	It provides a folder system with ontology files and related statistical information	
OBO Foundry	obofoundry.org	It provides a folder system with ontology files and related statistical information	
ODP - Exemplary ontologies	ODP ontologies	A catalog of exemplary ontologies	✓
Ontology Lookup Service	ebi.ac.uk/ols	A repository for biomedical ontologies	

6.3.4 Case Study: Experiences from the Internet of Production

In the following, we evaluate the results of this section, which have been summarized in a guidelines booklet entitled “*Guidelines for the Creation of Semantic Models in the Internet of Production (IoP)*” [Com22b] (see figure 6.12 for its workflow). We evaluate this guidelines booklet in the Internet of Production as representative for the manufacturing domain (see section 2.3). The evaluation consists of two parts. First, we examine an ontology created by a domain expert using the proposed guideline booklet and evaluate its quality, and thus the usefulness and effectiveness of the booklet. The second part is an aggregated evaluation of the guideline itself, considering its structure and content.

We leverage the manufacturing use case “*optimization of the injection phase*” for the first part of this evaluation. This use case aims to improve the quality of manufactured parts and to increase the productivity of the injection molding process. Its scope includes multiple processes (e.g., *holding pressure*), attributes (e.g., *geometry*), objects (e.g., *cylinder tip*), and constraints (e.g., *maximum pressure*). Our evaluation methodology is as follows. A domain expert from the Internet of Production project creates an ontology to formalize knowledge using the proposed guideline booklet as support. We will then assess the created ontology using *OOPS!* (see sections 2.2.1 and 6.3.3) and additionally compare the actual result with the initial goals, which we collected with questionnaires. This comparison includes test cases with positive and negative expected results, and follows the ontology testing methodology presented in [BSP12]. The results of the *OOPS!* evaluation are shown in figure 6.13 and consist of a list of possible pitfalls listed as *P04*, *P08*, and so on.

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🛑 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	2 cases Minor 🟡
Results for P08: Missing annotations.	36 cases Minor 🟡
Results for P10: Missing disjointness.	ontology* Important ⚠️
Results for P11: Missing domain or range in properties.	1 case Important ⚠️
Results for P13: Inverse relationships not explicitly declared.	10 cases Minor 🟡
Results for P19: Defining multiple domains or ranges in properties.	1 case Critical 🛑
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P41: No license declared.	ontology* Important ⚠️

Figure 6.13: Results of the *OOPS!* evaluation of the initial version of the ontology created by a domain expert in the use case of our evaluation [Com22a].

While [Com22a] analyzes these pitfalls in more detail, we can summarize that most of the problems found by *OOPS!* relate to a more advanced formalization of knowledge. On the other hand, this means that most of the relevant tasks in the process of ontology development were successfully completed with the help of the proposed guideline. The domain expert was able to successfully create an ontology as a knowledge formalization in Protégé.

The second part of this evaluation examines whether the created ontology adequately rep-

resents the intended knowledge on a semantic level. We therefore use the *Themis*⁵ tool (see [section 6.3.3](#) and [table 6.3](#)) to verify the consistency in the ontology with defined unit tests. These test cases, in the form of competency questions, cover expected answers to ontology queries such as “*Is the injection mold part of an injection molding machine?*” → *yes* or “*what influences a product?*” → *a process*. See [Com22a] for more details and the associated *Themis* test suite, which encodes these test cases into machine-readable queries using ontology terms such as `subClassOf` or `disjointWith`. A reasoner⁶ (see [section 2.2.1](#)) shows that, despite some unexpected inferences, the results were mostly as expected and even improved the domain expert’s understanding.

Additional questionnaires on the overall content of the proposed guide were sent to researchers and domain experts from the *Internet of Production* project. Results include that most participants (strongly) agree that the document meets the previously defined requirements, which include for example *clear examples*, *helpful templates*, *validation checks*, and *modularity*. This general satisfaction also applies to the overall structure and organization of the document. See [Com22a] for the complete results of this part of the evaluation.

To summarize this evaluation: The created ontology, while of good quality, has undefined elements identified in its evaluation against the defined goals. Given its ongoing evolution, future iterations (see [section 6.3.1](#)) aim to address missing elements and conflicting definitions. This involves formalizing knowledge, particularly about object property characteristics, not fully understood by domain experts. Progress is noted as the domain expert successfully created the ontology using templates and tools, marking progress in their participation in ontology development for manufacturing and production. Positive feedback on the guideline booklet for domain experts reinforces its usefulness. To improve its quality, examples suggested by evaluators, including definitions of fundamental concepts and the advantages of building on existing ontologies, will be incorporated. In the context of dataspace, the proposed guideline booklet in this section adequately addresses [requirement Domain-R2](#) by providing best practices and guidelines for vocabulary development and handling, especially for non-experts.

Contributions

The main contribution of this section is a best practices guidebook for vocabulary development, especially for non-experts. It covers all relevant parts of ontology development, including the process of ontology development, different levels of ontologies, methodologies, best practices, design principles, state of the art analysis, tool support for different types of applications, ontology catalogs, and vocabularies for the manufacturing domain. An evaluation in the *Internet of Production* proves both its general applicability and its concrete applicability in the manufacturing domain. Additional contributions are the intermediate analyses of the above mentioned aspects, which have been incorporated in an accumulated version in the guideline booklet.

⁵<https://themis.linkeddata.es/index.html>

⁶<http://owl.cs.manchester.ac.uk/tools/fact/>

6.4 Vocabulary Creation for Non-experts

In this section, we address [requirement Domain-R3](#) and make the creation of ontologies, or at least drafts of them, more accessible to non-experts. This mainly concerns the *describe* and *integrate* phases of the domain layer depicted in [figure 6.1](#), but partially also the other phases highlighted there.

Common vocabularies and ontologies are crucial for many applications. Even though standards and recommendations already cover many areas, customization is usually required to properly represent concrete use cases. Domain experts are usually not familiar with ontology engineering, which creates special requirements for the necessary tool support. Basic drafting tools are usually too imprecise, while comprehensive ontology editors are often too complex for non-experts. This section presents a tool for rapid vocabulary creation by domain experts. Its assisted vocabulary creation and collaborative graph editing allow even non-experts to quickly create vocabularies, and reduce the time and effort required for collaborative vocabulary creation. An [RDF](#) export enables support for Semantic Web tools.

The benefits of good ontologies and vocabularies are undisputed in the dataspace community and the need for semantics is imposed by core features such as distributed data sharing, which requires common languages for communication. General domains have received a lot of attention, resulting in many recommendations and standards (e.g. *SKOS*, *DCAT*, and *FOAF*), and general information models for dataspaces (see [chapter 4](#)). Niche domains, on the other hand, often face the problem that no suitable vocabularies exist. At the same time, ontology creation is a complex, time-consuming task, and rarely something that domain experts are accustomed to doing (see the previous [section 6.3](#)). Reducing both the effort and the involvement of ontology experts is crucial, especially for small-scale application contexts [[BS05](#)]. This can be done with graphical solutions, which often do not offer the same flexibility as command line interfaces, but are considered to be more user-friendly environments for new users who are substantially used to graphical user interfaces [[FW12](#); [The04](#)]. We have identified the following key requirements in [[Lip+21b](#)]:

- ease of use (e.g., graphical editor) [[FW12](#); [The04](#); [LGD20](#); [LGD21](#)],
- fast development,
- [RDF](#) support,
- vocabulary reuse,
- free and open source, and
- collaboration.

In this section, we present a remake of *Neologism* [[Bas+09](#)], which has been used to define and publish vocabularies on the US Government Open Data portal⁷. It is available as open source (MIT) at <https://github.com/Semantic-Society/Neologism>, where we also offer a live demonstration page, a video, and further development towards a full-featured extension of the original solution. Neologism 2.0 assists users in designing vocabularies in a graph editor and is intentionally not a full-featured ontology editing tool like Protégé [[Noy+01](#)]. It produces vocabulary sketches for bootstrapping other Semantic Web applications, making domain vocabulary creation accessible to non-experts within dataspaces.

⁷see: <https://joinup.ec.europa.eu/node/45149>



Figure 6.14: The proposed vocabulary drafting process is embedded into a typical user workflow [Lip+21b].

For domain experts unfamiliar with the Semantic Web, most existing solutions are feature-rich and complex. Neologism 2.0 specifically addresses the need for ease of use in the first steps of vocabulary creation in specific domains: It offers simple tools for quick and easy prototyping of initial vocabulary sketches, while encouraging best practices and the reuse of existing terminology from customizable origins. This allows non-experts to quickly draft vocabularies in a user-friendly graphical editor and export them for further development. Each vocabulary draft is assigned a unique, shareable [URL](#) that allows for easy publishing and collaborative work. [Figure 6.14](#) shows a recommended workflow that uses Neologism 2.0 to create initial vocabulary drafts and imports their [RDF](#) exports into Protégé for later refinement by Semantic Web experts, which can ultimately lead to fully functional ontologies.

6.4.1 Related Work

There are several tools for vocabulary creation, which are shown in [table 6.10](#), but none of them meet all our above requirements and fill the gap of easy creation of decent but simple vocabularies. While many tools are graphical editors all tools support [RDF](#) import or export, only half of the tools supports fast ontology drafts for non-experts. Only a few tools support the reuse of external vocabularies (e.g., with recommendation systems) or real-time collaboration features. Only *Mobi* [iNo17] allows easy, instant publication of drafted vocabularies, but most tools are free and open source.

Research on related work includes comparing *Neologism 2.0* and *Neologism 1.0* [Bas+09], both web-based systems simplifying [RDF](#) vocabulary creation. Neologism 2.0 prioritizes user-friendliness, featuring a graphical editor, simplified UI, and autocompleting fields. It distinguishes itself from Neologism 1.0 by eliminating complex elements. While Neologism 1.0 used *Vapour* for validation, consistency checking is not in scope for either version. Neologism 2.0 aims at user-friendly creation, allowing Semantic Web experts to refine drafts.

Fully-featured ontology editors like Protégé can be overwhelming for non-experts. We advocate for graphical editors, as illustrated in the Neologism 1.0 comparison. Tools like WebVOWL (see [Loh+16; WLA18] or [table 6.10](#)) support graph-based editing but lack real-time collaboration. Neologism 2.0 simplifies ontology management, allowing creators to easily manage access rights and publish independently. It integrates multiple features, offering a middle ground between basic sketches and comprehensive ontology editors, aiming for seamless integration with cutting-edge tools in the scope of dataspace.

Table 6.10: Comparison of ontology drafting tools based on [Lip+21a]. Note that this comparison adds to sections 2.2.1 and 6.3.3.

Tool	Description	Graphical Editor	Fast Development	RDF Support	Vocabulary Reuse	Free & Open Source	Collaboration
NeOn Toolkit [Haa+08]	Java-based editor maintained by the NeOn Foundation until 2011	✓		✓		✓	
AceWiki [Kuh08]	Collaborative ontology management through ACE sentences (translatable to first-order logic)			✓	✓	✓	✓
SADL [CM13]	Semantic model and rules definition using English-like DSL		✓	✓		✓	
Mobi [iNo17]	Free KG platform on RDF and OWL	✓		✓		✓	✓
Protégé [Noy+01]	Fully-featured ontology editor, web version available [Tud+13]	✓		✓		✓	
TopBraid Composer [Top20]	Comprehensive support in ontology creation (commercial)	✓		✓	✓		✓
Grafo [dat20]	Commercial tool for collaborative, visual KG design	✓	✓	✓			✓
ExcelRDF [Ham20]	RDF vocabulary creation from Excel		✓	✓		✓	
CoModIDE [SH19]	Modular ontology engineering with graphical modeling (Protégé plugin)	✓	✓	✓	✓	✓	
WebVOWL [Loh+16; WLA18]	Interactive web application for ontology visualization	✓	✓	✓		✓	
Graphviz [Ell+01]	Graph drawing tool (Protégé plugin)	✓		✓		✓	

6.4.2 Approach

The methodology to address [requirement Domain-R3](#) with easy ontology drafts for non-experts is based on [figure 6.14](#) and includes the design and development of a graphical editor that fulfills all requirements posted in [section 6.4](#) [Lip+21b]. Our approach to these requirements is as follows:

- Ease of use: Graphical editor for simple use, web-based tool for low entry barrier
- Fast development: Simple actions and auto-complete features
- [RDF](#) support: [RDF](#) exports in common serialization formats
- Vocabulary reuse: A recommendation engine for existing ontology [URIs](#) (see [section 6.5](#) for details)
- Free and open source: Source code on GitHub with an open license
- Collaboration: Real-time collaboration feature for users

To put this approach into action, [figure 6.15](#) shows the proposed high-level architecture of Neologism 2.0, which consists of the three core components: the frontend, the backend, and the recommender. The web frontend is implemented in *TypeScript* using *Angular 9.0* and provides authorization and authentication. A simple dashboard allows to manage, create, edit, share, search and export vocabularies. Data persistence is managed by a *Meteor* backend that provides programmatic endpoints for editing operations on classes and properties. As a key feature, Neologism 2.0 instantly synchronizes changes between all clients and the server to enable real-time collaboration, eliminating user confusion caused by asynchronous edits.

6.4.3 Results

In the graphical editor of Neologism 2.0, users create vocabularies, and screenshots of this can be found in [figure 6.16](#). This figure shows how users can add new classes by entering a name and description (left), or add properties by selecting a range and entering a name and description in a pop-up window (center). The graph editing (right) is interactive. All [URIs](#) are automatically generated for easy access. Creating new classes requires only a name and description, with the [URI](#) automatically filled from a generated ontology base [URI](#) and the class name. The same process applies to properties, with an additional requirement to define the scope. User input on the web frontend contributes to the ontology

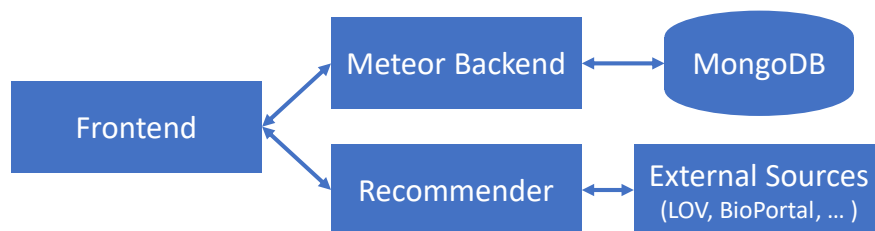


Figure 6.15: Neologism 2.0’s frontend communicates with a backend to persist information, and can interact with a recommender to improve the quality of drafted vocabularies by reusing existing vocabularies [Lip+21b].

6 Creating a Common Domain Understanding in Dataspaces

export. We use `rdf:type` to specify `rdfs:Class`, `owl:Class`, and `owl:ObjectProperty`. The class and property names and descriptions in the graph editor are later serialized as `rdfs:label/comment`. Finally, `rdfs:domain` and `rdfs:range` are used to represent each property's nodes in the resulting export [RDFS/OWL](#) file in turtle format.

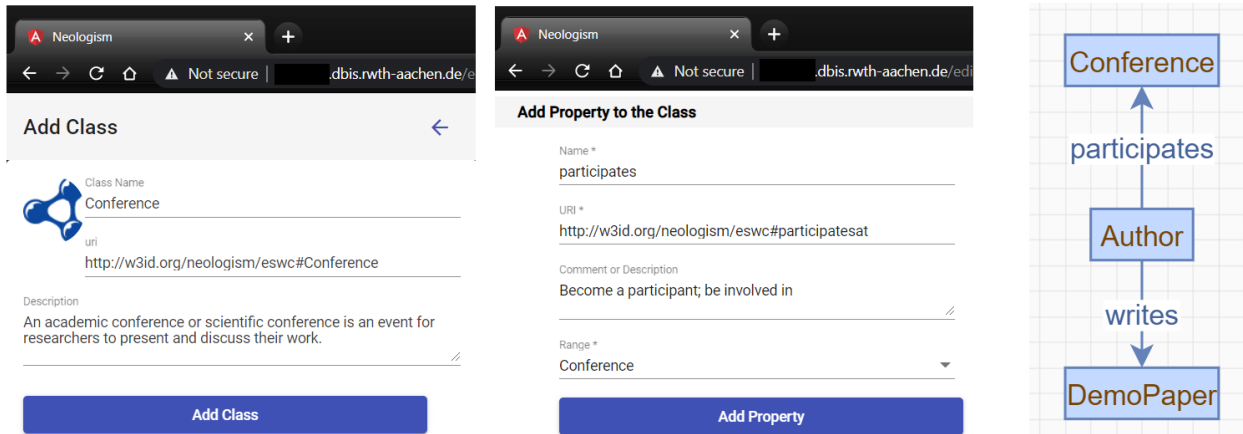


Figure 6.16: Screenshots of class and property creation in Neologism 2.0 [Lip+21b; Lip+21a].

The graph editor displays the current state and allows interaction with both classes and properties, including editing and deleting. It supports real-time collaboration by continuously synchronizing its state. A recommendation system suggests existing terms from configurable sources, depending on the application, see [section 6.5](#) for details. It can also be used as a proxy to other recommendation systems such as *Linked Open Vocabularies* or BioPortal (for both, see [table 6.9](#)). The codebase of both Neologism 2.0 and its recommendation system is published as open source on GitHub with an open license [Lip+21a]. In summary, these features address all of the above requirements.

Limitations of our approach include the following. Neologism 2.0 intentionally avoids support for more complex [OWL](#) semantics, such as cardinality restrictions, motivated by a focus on simplicity for rapid ontology creation. This limits its applicability as a general purpose ontology editor compared to related work. However, this limitation by design is consistent with the design process outlined in [figure 6.14](#). Furthermore, Neologism 2.0 does not currently support importing existing [RDF](#) vocabularies for further editing.

In summary, the free vocabulary prototyping tool Neologism 2.0 fits well into dataspace workflows where domain experts create first drafts that are later revised by Semantic Web experts in more comprehensive tools. Its web-based graphical editor allows all types of users to quickly create vocabularies, using a recommendation system (see [section 6.5](#)), a vocabulary publishing service, real-time synchronization for collaboration, and [RDF](#) export for further processing in other semantic tools. With these features, Neologism 2.0 covers all major requirements. As future research, both usability and integration into real-world applications could be evaluated through a user study in dataspace use cases from the manufacturing and mobility sectors. Our open source GitHub repository [Lip+21a] allows collaboration and discussion with the community to bridge the gap between simple sketches and comprehensive ontology editors, and finally to improve the collaboration between domain experts and the Semantic Web in the context of dataspace.

Contributions

In this section, we have addressed [requirement Domain-R3](#) with the goal of improving the accessibility of ontology creation for non-experts, with a particular focus on the *describe* and *integrate* phases of the domain layer as depicted in [figure 6.1](#).

Recognizing the crucial role of shared vocabularies and ontologies in applications, we introduced Neologism 2.0 as a tool tailored for rapid vocabulary creation by domain experts. This tool, with its assisted vocabulary creation, collaborative graph editing, and recommendation system, significantly reduces the time and effort required for collaborative vocabulary creation. Its [RDF](#) export functionality facilitates rapid integration with other semantic tools.

In summary, Neologism 2.0 proved to be a valuable asset within dataspace workflows. It enables domain experts to efficiently create initial drafts that are later refined by Semantic Web experts. The tool’s web-based graphical editor, recommendation systems, collaborative features, and [RDF](#) export capability catered to diverse user expertise. Finally, the potential of the tool for real-world applications, particularly in the manufacturing and mobility sectors, was highlighted. Future research opportunities include usability evaluations and integration into real-world applications. The open source nature of Neologism 2.0 on GitHub encourages collaboration to improve the interaction between domain experts and Semantic Web professionals in the context of dataspace.

6.5 Recommendations for Vocabulary Use

In this section, we address [requirement Domain-R4](#) and propose recommendation systems for vocabulary use. Metadata can be created and shared using *semantic annotations* (see [section 2.2.1](#)) based on common languages or common vocabularies. Since dataspace are distributed systems, their participants are also distributed, and thus they create ontologies largely independently of each other. The use of recommender systems helps to ensure that these different vocabularies do not drift apart, but rather share more common content. The reuse of common knowledge or concepts makes it easier for others to understand and is a good compromise between using only (quasi-)standards (which limit the expressiveness) and using arbitrary, uncontrolled language (which are hard or impossible to understand later). It also is a widely-used good practice, see [section 6.3](#). Due to the large number of vocabularies available, selection is often difficult and time-consuming, which is where recommender systems help by automating the process with recommendations. This is especially true for the *describe*, *analyze*, and *publish* phases of the domain layer illustrated in [figure 6.1](#), but partly also for the other phases highlighted there.

The generic approach of a recommendation system in this section and in this thesis in general is as follows. The *domain layer* of the [DLCM](#) in dataspace is shown in [figure 6.1](#) and highlights phases like *describe*, *integrate*, and *analyze*. These phases mainly rely on proper metadata, which is shared along with data in dataspace. Data (re)users can leverage these metadata to improve the understanding of the described data. As soon as the metadata becomes largely self-explanatory and uses common language, no additional communication with the provider of given (meta-)data is needed anymore. This reduces the time loss, costs, and complexity of data (re)use in dataspace. Recommendation systems in this context concretely aim to suggest things for reuse to a user. One goal of this is to make a created semantic description, vocabulary, ontology, or set of metadata entries more understandable for others such as possible data consumers. Technically spoken, recommendation systems

add knowledge from sources, or references to such knowledge, to a local knowledge base such as a vocabulary or an instance of a Gaia-X self description (see [section 2.1.1](#)).

This section covers two aspects of recommendations for vocabulary use. First, general principles of combining knowledge from multiple sources and integrating it into a local knowledge base. This can be done by reference, where links from local to external identifiers are built, or by value, where external knowledge is copied to a local knowledge base, eventually with a specified scope⁸. The second aspect of this section is about a thorough recommendation system with practical impact in vocabulary creation, which we demonstrate in the vocabulary drafting solution *Neologism 2.0* presented in [section 6.4](#).

6.5.1 Use Case Requirements

This section consists of two parts on the principles of recommender systems for ontology concepts. These are directly derived from the scope of recommendation systems in dataspace: For ontologies designed by distributed participants in dataspace, local definitions (labels) are transformed into global definitions (ontology URIs) to make implicit overlaps (labels) of ontologies explicit (URIs). We first present a discussion of two alternatives for recommendation systems. This section then discusses principles of ontology ranking that are needed to make decisions about which (external) ontology element best fits a given task.

Recommendation systems for ontologies in dataspace play a crucial role in providing users with valuable insights and relevant information. We distinguish between two approaches, namely *batch recommender* and *normal recommender*, both of which can have respective advantages based on processing modes and timing of recommendation generation:

- **Batch Recommender:**

- **Processing Mode:** Processes recommendations in predefined batches or sets of data.
- **Timing:** Recommendations are typically generated offline or in scheduled intervals, independent of real-time user interactions.
- **Advantages:** Efficient resource utilization, comprehensive analysis of the entire dataset, suitability for scenarios where recommendations can be planned and executed offline.

- **Normal Recommender:**

- **Processing Mode:** Generates recommendations in real-time or on-the-fly, responding to user queries or interactions.
- **Timing:** Recommendations are provided immediately or interactively based on user input or actions.
- **Advantages:** Responsiveness to user needs, adaptability to dynamic changes in the dataset, suitability for scenarios requiring timely and interactive recommendation delivery.

⁸Approaches such as *call by value* or *call by reference* are also used in functions of programming languages, see https://en.wikipedia.org/wiki/Evaluation_strategy#Strict_binding_strategies.

In the context of ontology recommendation in dataspace, the choice between a batch recommender and a normal recommender depends on the specific requirements of the application. Batch recommenders are appropriate for scenarios where comprehensive analysis of the entire dataset is possible and where real-time responsiveness for each individual term is not a critical factor. On the other hand, normal recommenders are suitable for dynamic environments where individual recommendations need to be provided in real time, adapting to changes in the dataset or user interactions. Following the [DLCM](#) in dataspace (see [chapter 3](#) and especially [section 3.3](#)), a batch recommender is often more beneficial because it can take into account the complete user input to rank and provide the best recommendations.

6.5.2 Related Work

This section extends the general background information from [section 2.2.2](#) and puts an emphasis on recommendation systems from two different perspectives: Methods to extract knowledge from available knowledge sources and a discussion of these knowledge sources.

Research Areas for Recommendation There are multiple research areas, which cover the task of finding a good semantic concept for a given label or search term, including ontology matching and alignment, schema matching, semantic tagging, semantic annotation, and semantic modeling [[Pau+19](#); [Lip17](#)].

In the research areas of schema matching and ontology matching and alignment, a task is to align a given ontology A with another ontology B that may differ in size and content. Approaches such as structure-based, instance-based, or lexical-based methods are commonly used to address the problem of finding an appropriate match for a given entity in one ontology in another [[M H+15](#)]. For lexical and structure-based methods, the use of higher-level ontologies (cf. [[MLR10](#)]) or external sources such as Wikipedia, DBPedia or WordNet is widespread [[M H+15](#); [Smi+11](#); [Mae+02](#)].

In contrast to ontology matching, semantic annotation deals with the task of finding an appropriate semantic concept for a given label of a data item (e.g., the column title in a table) [[GKL12](#)], [[Ram+15](#)]. The area of semantic modeling continues by additionally building meaningful semantic relations and more abstract concepts (e.g., combining *name*, *age*, and *hair color* into the abstract concept *person*) [[Tah+16](#); [Tah+15](#)]. To perform the task of semantic annotation, Goel et al. [[GKL12](#)] utilize training data from prior datasets to train a machine learning model for the representation of the data, which enables it to propose semantic concepts without considering labels. On the other hand, approaches such as [[Sye+10](#)] or [[Wan+12](#)] use knowledge bases such as WordNet, Wikipedia, or Probase to suggest concepts based on the labels of the current data. To build complete semantic models, Taheriyani et al. in [[Tah+16](#); [Tah+15](#)] additionally use domain-specific ontologies and frequent relationships in Linked Data sources. In addition to these research areas, the field of semantic tagging focuses on the automatic generation or suggestion of accurate tags for unstructured and binary data, such as documents, images, and audio or video files [[Du+12](#)]. Approaches use tag ontologies such as *MOAT* and *SCOT* [[Kim+08](#)], search engines such as *Google* [[SS14](#)], or knowledge bases such as *WordNet*, *DBPedia*, *OpenCyc*, or *Wikipedia* [[Du+12](#); [MJS10](#); [HPL15](#)]. See [section 6.5.2](#) for more details on these knowledge bases.

Principles of Ontology Rankings Ontology ranking is the process of ranking different ontologies or terms of these based on a set of metrics or analytical measurements [SAS11]. In the following, we compare state-of-the-art recommendation and ranking concepts and discuss the different approaches. Ontology recommendation uses ontology ranking but focuses on providing terms or ontologies that may fit a given context [Mar+17]. In the following, we give an overview of different ontology ranking algorithms.

Subhashini et al. in [SAS11] reviewed several ranking algorithms, which are outlined below:

- **AktiveRank** [ABS06] is based on several analytical measures that use the graph structure of ontologies.
- **The content-based ontology ranking algorithm** [JA06] receives a set of ontologies from a search engine. The received ontologies are ranked by the number of concept labels that match a set of terms extracted from WordNet (for WordNet, see [Mil95] or table 6.7).
- **The Ontology Structure Rank Algorithm** [WC09] ranks ontologies using class names, semantic relationship, and ontology structure. The weights of the measures can be adjusted according to the user's needs and the focus of an application.
- **The Semantic-aware importance flooding algorithm** [CYY10] also converts the ontology into a directed graph and executes an iterative fixpoint computation to compute the relevance of nodes.

In the work presented by the authors in [TA07], ontologies are evaluated based on their content relevance to a set of keywords and user preferences. The evaluation considers both instances and schemas of ontologies. The process starts with a keyword and the corresponding populated ontology, which is enriched using *WordNet* for related keywords. The enriched set of keywords is then submitted to *Swoogle*, a former search engine for Semantic Web ontologies and terms. The related ontologies are then evaluated and ranked using specific metrics.

In the study conducted by the authors in [Ala+07], query expansion is performed through search engines such as *Wikipedia* and *Google*. The goal is to identify the most relevant terms related to a specific topic and then query the ontology repository. The ranking of results is based on the frequency of each query term in class labels, property labels, and property values for datatype properties (e.g., string values properties). Normalization is applied to prioritize ontologies that cover a wide variety of terms over those that cover the same terms more frequently.

The work presented in [LPA11] focuses on the architecture of recommenders for semantic datasets. Datasets are transformed into a graph structure, with nodes and edges assigned weights for clustering. Two recommender approaches are used: A memory-based recommender and a model-based recommender. The former computes paths from a set of given input entities, ordering reachable entities based on a similarity score derived from edge weights. The latter builds a model from the data set, clustering similar entities. The choice between the two approaches depends on the scenario and query, with the results aggregated into a unified result list.

The authors of [GW09] focus on the biomedical domain and need a large amount of user specifications considering this domain. The concept of [Zie04] focuses on similarity and neighborhood measurements, computing specific trust and ranking functions.

ESKAPE [Pau+19], now PLASMA [Pau+21], focuses on providing semantic recommendations for dataset input. They added structural recommendations for terms, for example suggesting a graph of terms with specific relationships to a given keyword. The goal is to adequately describe the keyword with its surrounding relationships [PLM19b]. In ESKAPE, the recommended concepts are lists that are not tightly bound to ontologies.

BioSS [Mar+14] and BioPortal [Mar+17] are both approaches that provide multiple keyword searches that return ontology sets that could cover a certain amount of the input keywords.

In summary, the explored ontology recommendation approaches with our dataspace scope offer diverse strategies with individual advantages and limitations. The content-based evaluation in [TA07] provides a holistic evaluation using WordNet for keyword enrichment, but may face limitations due to its reliance on specific search engines. Query expansion approaches, as seen in [Ala+07], prioritize diverse term coverage, but are influenced by the choice of search engines and normalization techniques. The architecture-focused work in [LPA11] allows flexibility in approach selection, but effectiveness depends on appropriate method choices. In the biomedical domain, [GW09] emphasizes user specifications, and [Zie04] focuses on trust and ranking functions. PLASMA (formerly ESKAPE) [Pau+19] integrates semantic and structural recommendations, but may lack tight ontology binding. BioSS [Mar+14] and BioPortal [Mar+17] offer versatile keyword searches that address different input keywords, but the coverage depends on the available ontologies. Overall, the presented ontology recommendation approaches in data spaces share limitations. Scalability issues in processing large datasets, graph transformations, and complex similarity computations, especially in dynamic dataspace, are among the challenges. The lack of standardized methods hinders effective comparison and benchmarking, and prevents the establishment of common evaluation frameworks in the field. Despite advances, some approaches still struggle to capture nuanced semantic relationships between terms, which affects the accuracy of recommendations. In addition, the dynamic nature of ontologies poses challenges, as changes in ontology structures and the introduction of new terms can affect the relevance and accuracy of recommendations over time. To address these limitations, research is needed to improve the effectiveness and adaptability of ontology recommendation methods in dataspace.

Available Knowledge Bases

We classify knowledge bases into two categories: *Structured domain knowledge bases*, which are characterized by a structured framework that emphasizes querying for specific relationships and semantics within a defined domain and *general knowledge bases*, which cover a wide range of topics with a more flexible structure.

Structured Knowledge Bases

- **Ontology-Based Knowledge Bases:** Use formal ontologies to represent entities, concepts, and relationships in a shared understanding of a domain. Example: OpenCyc, based on the Cyc ontology, models general human knowledge using formal semantics (see [LG89] or table 6.11).
- **Linked Open Data:** Publishes structured data on the Web, interlinking datasets to create a global web of data, using URIs and standard data formats. Example: DBpedia extracts structured information from Wikipedia, contributing to the Linked Open Data cloud (see table 6.11).

- **Ontology-Based Search Engines:** Use ontologies to improve search results by understanding user queries and content semantics for more contextually relevant results. Example: Watson⁹ retrieves information from the Semantic Web using ontology-based approaches.
- **Semantic Web Standards:** In general, standardized technologies such as [RDF](#) and [SPARQL](#) are fundamental to representing and querying structured data on the Semantic Web. Example: [RDF](#) enables the expression of relationships between entities, while [SPARQL](#) enables the querying of [RDF](#) datasets. See [section 2.2.1](#) for details.
- **Linked Open Vocabularies:** Serves as a hub for finding, sharing, and reusing Semantic Web vocabularies to promote interoperability and consistency. Example: Linked Open Vocabularies [[Van+17a](#)] provides a curated collection of ontologies, including FOAF, Dublin Core, and SKOS.

Structured knowledge bases are a good foundation for recommender systems because they provide a basis for representing, linking, and querying data in a meaningful and contextually rich way. They play a critical role in realizing the vision of the Semantic Web, where information is not only represented as text, but also understood and connected by machines, and are therefore beneficial for use in dataspace.

General Knowledge Bases An overview of general knowledge bases is given in [table 6.11](#) and includes, in addition to the name, reference and homepage, statements whether the knowledge bases cover specific or general domains and whether their facts are collected manually or automatically.

Table 6.11: Overview of general knowledge bases. Based on contents from [[Fär+15](#); [Pau+19](#)]. Note that the *name* column includes links to the respective homepages.

Name	Covered Domains	Provenance of Facts
WordNet [Mil95]	Linguistics	Manually curated
Freebase [Bol+08]	Various	User-generated, closed in 2016
Wikidata [VK14]	General knowledge	User-edited, automated
OpenCyc [LG89]	General knowledge	Manually curated
DBpedia [Leh+15]	Linked Data	Automated (Wikipedia info boxes)
BabelNet [NP12]	General knowledge	Automated, integrates WordNet, Wikipedia
Wolfram Alpha	Various	Manual curation and automated processes
YAGO [SKW07]	General knowledge	Automated (Wikipedia, GeoNames)
Google Knowledge Graph [Sin+12]	General knowledge (instances)	Manual curation and automated processes

WordNet [[Mil95](#)] is a form of taxonomy and contains nouns, verbs, adjectives, and adverbs grouped into sets of synonyms called *synsets*. Its details are as follows [[Lip17](#)]. WordNet is available online or can be downloaded and used as an offline database, and is free to use as of the WordNet 3.0 license. It covers mostly linguistics and is manually curated. Relations

⁹<https://kmi.open.ac.uk/technologies/name/watson/>

in WordNet are limited, and the main relations are synonymy and hypernymy (*isA*), and synsets usually have a description called *gloss*. Since synonymy is covered in synsets, the core relation is hypernymy/hyponymy, which links more general synsets to more specific ones and vice versa. The *isA* relation is transitive and hierarchical, resulting in a tree structure with an entity root that can be reached from any synset. WordNet distinguishes between *Types* and *Instances*. Verbs are also arranged hierarchically, explaining more and more specific manners like *communicate*, *talk*, *whisper*. Note that manners vary among verbs, depending on semantic range, and that WordNet contains few adverbs. Adjectives are organized in terms of antonyms, including antonym pairs, semantically close, and also adjectives derived from nouns (“criminal” from “crime”). It should be noted that WordNet is no longer funded, so the current version 3.1 is expected to be the last.

Freebase, established in 2007, served as a collaborative knowledge base and open database, designed to organize and interlink information across diverse subjects [Bol+08]. It addressed various domains and its content was generated by users. Acquired by Google in 2010, Freebase allowed users to contribute and edit data in a structured format, fostering a collective effort to build an extensive repository of interconnected knowledge. In 2014, following the announcement of Freebase’s discontinuation¹⁰, its data was migrated to *Wikidata*, a free and collaborative knowledge base hosted by the Wikimedia Foundation [Pel+16]. This migration aimed to preserve the structured data created by the Freebase community, ensuring its continued availability within the broader Wikidata ecosystem. The integration of Freebase data into Wikidata not only enriched the latter’s content but also highlighted the collaborative nature of knowledge-sharing initiatives on the web. Despite Freebase’s closure in 2016, its legacy lives on through the contributions integrated into Wikidata, emphasizing the significance of collaborative efforts in maintaining open knowledge resources.

Launched in 2012 by the Wikimedia Foundation, *Wikidata* is a free and collaborative knowledge base that serves as a central hub for structured data across all Wikimedia projects [VK14]. Designed to be edited and maintained by a global community of volunteers, Wikidata contains information on a diverse array of topics, including people, places, and concepts. With a focus on multilingualism, Wikidata aims to provide a comprehensive and interconnected understanding of the world’s knowledge by linking data items to each other. Its structured and open nature facilitates the creation of a vast KG, supporting not only Wikipedia but also external applications and services. Wikidata plays a crucial role in enhancing the accessibility and interconnectedness of information on the internet, fostering a collaborative approach to knowledge-sharing on a global scale.

OpenCyc developed by Cycorp, is an open-source version of the Cyc knowledge base, which is a large-scale, formal ontology intended to represent general human knowledge [LG89]. Launched with the goal of making a substantial portion of Cyc’s extensive knowledge available to the public, OpenCyc provides a comprehensive framework for representing entities, concepts, and the relationships between them in various domains. OpenCyc’s ontology encompasses a wide range of topics, including common-sense reasoning, linguistics, and general knowledge. Users can leverage OpenCyc’s structured and formal representation to enhance applications requiring semantic understanding, such as natural language processing and artificial intelligence. The open-source nature of OpenCyc encourages collaborative contributions and usage, fostering the development of applications and systems that benefit from a rich and well-structured knowledge base.

¹⁰<https://web.archive.org/web/20190320195748/https://plus.google.com/109936836907132434202/posts/bu3z2wVqcQc>

BabelNet, established in 2009, is a multilingual lexicalized [KG](#) that seamlessly integrates and links information from diverse language resources, including WordNet, Wikipedia, and other lexical databases [\[NP12\]](#). Notably, BabelNet employs automatic integration mechanisms to compile its extensive knowledge base, encompassing a wide range of concepts across languages. This integration process involves the aggregation of information from sources like WordNet for lexical data, Wikipedia for encyclopedic content, and GeoNames for geographical information. By leveraging automatic integration, BabelNet achieves a comprehensive and up-to-date representation of concepts, making it a valuable resource for tasks such as natural language processing, multilingual information retrieval, and cross-lingual semantic analysis.

Wolfram Alpha, launched in 2009 by Wolfram Research, is a computational knowledge engine that excels not only in solving mathematical formulas but also in transforming search terms into knowledge, aligning with its slogan “*Making the world’s knowledge computable*”. This unique approach sets it apart from traditional search engines, as Wolfram Alpha focuses on providing specific and computed results rather than mere links to web pages. In addition to its extensive coverage across diverse domains, including mathematics, science, linguistics, and general knowledge, Wolfram Alpha shares a similarity with knowledge bases like BabelNet and WordNet – it uses synsets. It primarily relies on a combination of manual curation and automated processes to populate its knowledge base. By leveraging synsets, Wolfram Alpha enhances the understanding of word meanings and relationships, contributing to its ability to provide precise and contextually relevant information. Employing curated databases and advanced algorithms, Wolfram Alpha remains a valuable tool for users seeking computationally derived answers across various disciplines.

YAGO, short for *Yet Another Great Ontology*, is a comprehensive knowledge base developed by the Max Planck Institute for Informatics [\[SKW07\]](#). It adopts an ontology-driven approach, utilizing a formal and structured representation of knowledge. YAGO integrates information from various sources, including Wikipedia, WordNet, and GeoNames, to create an extensive and interconnected [KG](#). Unlike purely manual curation, YAGO employs automated methods for knowledge extraction, leveraging natural language processing and machine learning techniques. This allows YAGO to capture factual information, entities, and relationships across diverse domains. Emphasizing consistency and coherence within its ontology, YAGO serves as a valuable resource for applications requiring a structured and interconnected understanding of knowledge. It contributes to research in areas such as the semantic web, information retrieval, and knowledge representation.

The *Google Knowledge Graph* is a knowledge base created by Google to enhance the search experience by providing more relevant and contextual information directly in search results [\[Sin+12\]](#). Launched in 2012, the Knowledge Graph relies on a diverse set of data sources, including Wikipedia, CIA World Factbook, and Freebase (now Wikidata). It is designed to understand the relationships between entities, enabling the presentation of information in a structured and interconnected manner. The Knowledge Graph utilizes a combination of human-curated data and machine-generated algorithms to compile a vast repository of factual information about people, places, and things. Google’s Knowledge Graph is instrumental in enriching search results by offering instant answers, related topics, and contextual details, demonstrating the integration of human-curated knowledge with automated algorithms to enhance the search experience for users.

While these knowledge bases offer substantial contributions, certain limitations should be recognized. WordNet, despite its richness in lexical relationships, has a narrower focus on

linguistics. Freebase, now integrated into Wikidata, faced discontinuation, and its migration presents potential challenges in maintaining the collaborative spirit of the original platform. Wikidata, while multilingual and collaborative, may face issues of data accuracy and vandalism due to its open editing model. OpenCyc’s comprehensive ontology may present challenges in terms of complexity and resource intensity. BabelNet’s automatic integration, while efficient, may introduce noise or inaccuracies into its extensive knowledge base. Wolfram Alpha’s computational approach may limit its coverage in certain domains because it relies on curated databases. YAGO’s automated extraction methods may struggle with complex and nuanced knowledge. Google’s Knowledge Graph, while enriching search results, may be susceptible to biases inherent in its data sources. Understanding these limitations is critical for informed use of these knowledge bases in applications such as dataspace recommendation systems.

6.5.3 Approach

This section addresses two common application areas, in which recommendations for vocabulary use play a role: Semantic modeling (see [section 2.2.1](#)) and development of vocabularies or ontologies.

Application Areas for Recommendations

The concept of *semantic modeling* can be illustrated with a motivating example from [Pau+19]. It shows the need for semantic annotation in general, and in particular the need to support the modeling process with recommendations for semantic concepts from multiple knowledge bases. A data provider planning to share a dataset within a dataspace would follow our proposed [DLCM](#) for dataspaces (see [section 3.3](#)) and enrich the dataset with valuable metadata so that any data consumer can understand and interpret the data. This metadata can be represented as annotations for each label of the dataset, putting it into a semantic context and creating domain-specific meaning. Furthermore, this approach allows the expression of relationships such as hierarchies, which some data formats such as [CSV](#) files do not support naturally. [Figure 6.17](#) shows an example dataset in raw format and a corresponding semantic model. During the process of semantic modeling, each attribute is linked to an ontology concept, which fosters common understanding. Additional relations can be added to further clarify the context. The relation `isA` indicates superclasses, similar to `rdfs:subClassOf`. The semantic model is not limited in size or level of detail and represents the data provider’s expression of the information contained in the dataset. The use of a common language in the form of well-known vocabularies enables a common understanding in dataspace of both data providers and data consumers. This is where recommendation systems come in, as they can recommend existing concepts in the form of identifiers for reuse. This ensures that data providers do not reinvent the wheel for parts of their semantic models, but instead (re)use common languages whenever possible.

In the second application area, namely the *development of vocabularies*, actors such as data providers develop vocabularies, which are later used to express and understand metadata for datasets. For more details, see phases such as *describe*, *integrate*, and *analyze* in the [DLCM](#) for dataspaces in [figure 3.12](#)). The process of ontology development is described in [section 6.3](#). Recommendation systems come in a decision needs to be made on whether a new concepts is introduced or an existing one from is reused instead. Similar to the task

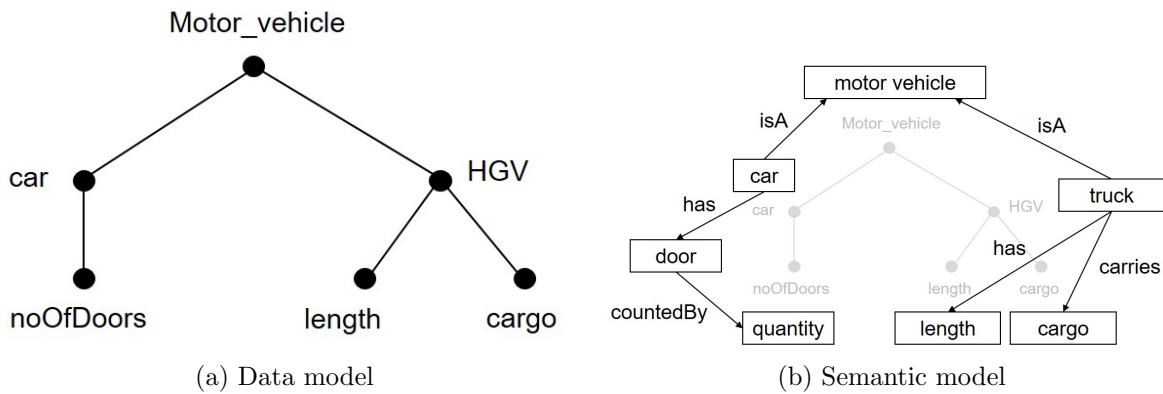


Figure 6.17: Labels (attributes) of an exemplary dataset and a corresponding semantic model [Pau+18].

of semantic modeling, the role of recommendation systems includes suggesting concept from knowledge bases for reuse. This can be done detached for individual terms (see *normal recommender* in section 6.5.1) or for multiple terms at once (see *batch recommender*).

Approach for Recommendations in these Application Areas

Our approach to enabling recommendations for the two application areas is shown in figure 6.18 and is as follows. The workflow designs recommendations in the realms of *semantic modeling* and *vocabulary development*. In the initial steps, tailored to each application area, we engage in the automated extraction of metadata labels combined with either automated or manual modeling of concepts and relations. Following this, the subsequent uniform process begins with the homogenization of the obtained data, ensuring consistency and coherence. The context is then determined to establish a clear understanding of the surrounding information. Our approach proceeds to collect recommendations based on the contextualized data, which are subsequently ranked. Finally, the ranked recommendations are applied, contributing to the ongoing enhancement of semantic modeling and vocabulary development in a coherent and unified manner.

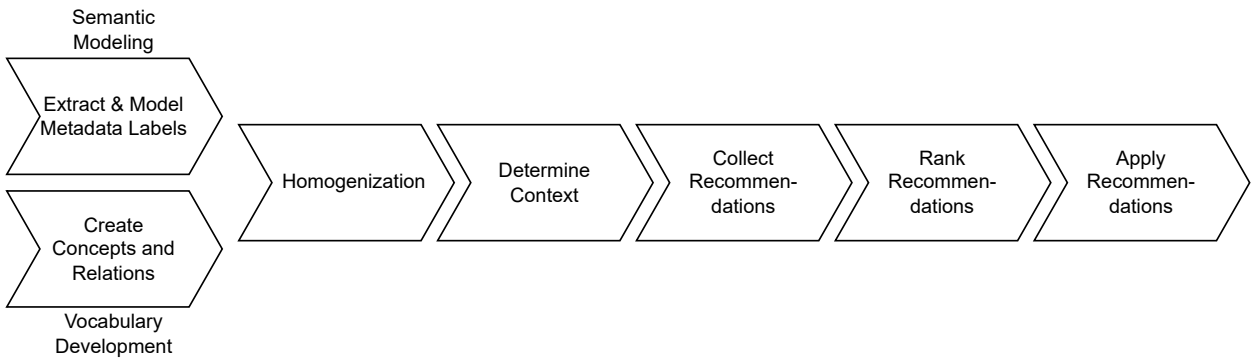


Figure 6.18: Our approach to enable recommendations for the application areas *semantic modeling* and *vocabulary development*.

The specific data formats processed in this proposed workflow are as follows. The first two “parallel” steps, of which only one is selected, output one or more so-called labels, which are a plain text (string). Homogenization does not change this structure. The *Determine Context*

step optionally adds a context description, which can also be a plain text label or a more complex description. The next step collects recommendations by collecting a set of **PIDs** for each label, e.g. in the form of **URIs**. The ranking of these recommendations converts these sets into sorted lists using metrics, so that the best possible match is eventually in the first position. In the last step, the best matching suggestions are selected by the target application or by the user via a user interface. In summary, a textual label is replaced by a semantically suitable **PID** (e.g. a **URI** in our experiments).

The contributions of this section will include methods for homogenization of semantic models and ontologies, a framework for extracting recommendations from knowledge bases, methods and algorithms for combining and ranking such recommendations, and a demonstration of a complete solution. Replacing plain text with **PIDs** makes shared content more understandable to others, which supports the goal of creating a common domain understanding in dataspace.

6.5.4 Results

This presentation of our results for recommendations for vocabulary use follows our approach illustrated in [figure 6.18](#). It covers the two application areas introduced in [section 6.5.3](#), *semantic modeling* and *vocabulary development*. These two areas are first addressed individually, followed by a presentation of the remaining results in a uniform manner. [Figure 6.19](#) illustrates an example execution of our approach. Note that we use text representations, such as *seminar teacher*, instead of **PIDs** or **URIs** for better readability. The data types for each single entity throughout the steps are (from left to right): Text, text, text, unordered set of texts (identified by **URIs**), ordered list of strings (identified by **URIs**), string (identifier by a **URI**) In summary, this execution includes the used data types such as unordered sets for collected recommendations but ordered lists for ranked recommendations.

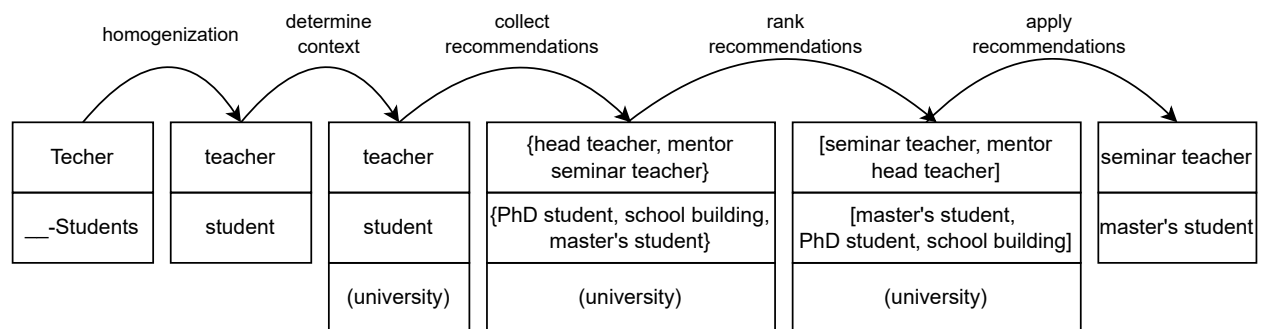


Figure 6.19: An example execution of our approach presented in [figure 6.18](#).

Extract and Model Metadata Labels

The results for *extraction and modeling of metadata labels* in the *semantic modeling* application area are as follows. Our developed platform ESKAPE [[Pom+18a](#)] allows data providers to annotate data sources with semantic models providing additional knowledge, which enables later consumers to find, understand, and use this data. It uses the approach shown in [figure 6.17](#). A data model (left) is automatically generated when loading (semi)structured data into the platform. Supported formats include structured ones such as [CSV](#), Excel, or

relational databases, and semi-structured ones such as [JSON](#) or [XML](#). The generated data model depends on the structure and content of the input data, and may contain unclear terms, since they are taken directly from the input data. The graphical interface of ESKAPE allows users to manually develop a corresponding semantic model (see [Figure 6.17](#)), which adds semantic concepts to the previously generated plain labels and allows additional structure through relations. The output of this step in the ESKAPE platform is a semantic model and can be considered as metadata description for phases like *describe* or *integrate* in our proposed [DLCM](#) for dataspace shown in [figure 3.12](#).

Create Concepts and Relations

In the application area of *vocabulary development*, we enable users to *create concepts and relations* with our vocabulary drafting tool Neologism 2.0 (see [section 6.4](#)). With its graphical vocabulary editor shown in [figure 6.16](#), users can incrementally add classes and properties to build an ontology. URIs are automatically generated and users can augment concepts with descriptions using fields such as `rdfs:comment` or `skos:definition`. The output of this step is an ontology draft, which can be exported to [RDF](#) and, similar to the output of the above *semantic modeling* step, can be considered as source for metadata descriptions for the *describe* or *integrate* phases of the [DLCM](#) for dataspace.

Homogenization

Homogenization can ensure the consistency of generated or created metadata. We conduct a study of how people create labels, which affects both attributes for input data from which metadata labels are extracted, and the direct specification of such labels in the process of semantic modeling or vocabulary development. Our manual review of 272 datasets from open data platforms leads to the following sources of unwanted heterogeneity [[Pau+19](#)]:

- **Abbreviations:** People tend to abbreviate labels instead of writing the full label. *Lat*, *lon*, or *www* are common examples we encountered.
- **Different languages:** Labels may be in different languages depending on the source of the data. It is even likely that the languages are mixed within the same dataset.
- **Natural language:** People tend to describe data attributes in more detail rather than labeling data with a precise concept or a single word. An example is '*Number of accidents in which people were injured*'.
- **Time labels:** Depending on the data, users tend to label columns or data attributes with points in time or time ranges (for example, *Monday, 1-2 pm*).
- **Misspelling:** The person who labeled the data made a simple mistake. Examples are *Acess Point* or *Telephone Number*.
- **Splitting characters:** In addition to spaces in labels, some labels contain special characters like underscores (`_`) or hyphens (`-`) to split words. An example we discovered is *telephone_number*.
- **Camel case text:** Similar to splitting characters, some users will split words using the camel case syntax (e.g., *StreetNumber*).

- **Additional information:** In a few data sets, we identified labels that contained additional valuable semantic knowledge that could be used to specify the semantic concept in more detail (e.g., *temperature in °C*).
- **Plural:** In several datasets, columns were marked with the plural instead of the singular (e.g., *streets* vs. *street*).
- **Random labels:** Labels that have no meaning or are just randomly generated strings, such as ascending numbers or hash codes (e.g., *8321b319b1781*).

We overcome these sources of unwanted heterogeneity with a processing technique [Pau+19] that, for example, converts labels to lowercase, translates labels to English, substitutes special characters, or splits camel case labels into multiple words. We also integrate the real-time spell-checking service *ngx-spellchecker* to correct any spelling mistakes during the process of creating and editing labels [Zej21]. As the user types text in Neologism 2.0, a drop-down list of suggestions is presented, see figure 6.20. Choosing the correct spelling homogenizes the data for further processing. We use version 1.0.5 with the `normalized_en-US.dic` dictionary for a demonstration of the English language in our vocabulary creation tool Neologism 2.0.

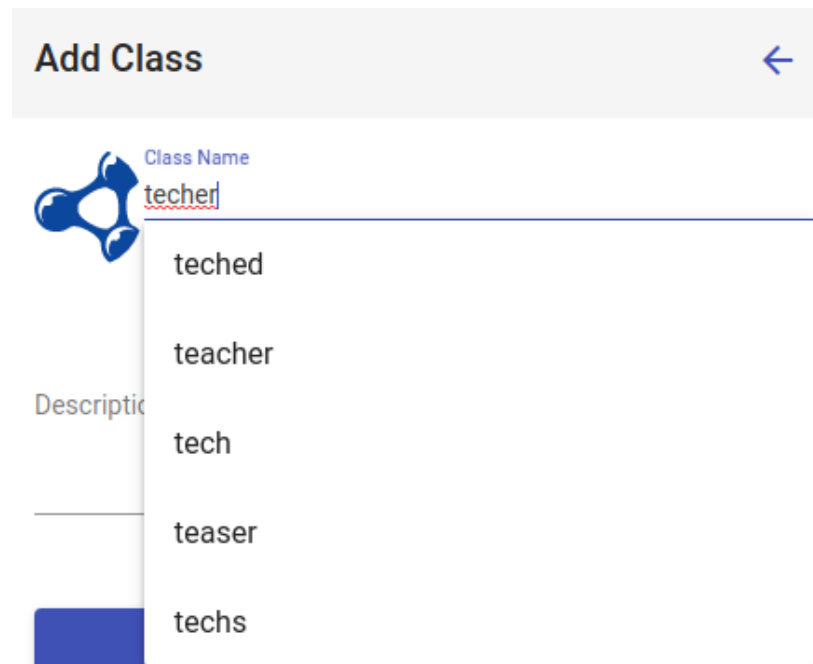


Figure 6.20: A screenshot of the suggestions generated by the `ngx-spellchecker` in Neologism 2.0 [Zej21] for a misspelling of *teacher* as *techer*.

Determine Context

Determining context in recommendation systems ensures that recommendations are not only relevant and accurate, but also tailored to the unique requirements of the user's current domain, ultimately contributing to the effectiveness and quality of the results. A simple example for the input *address* is the choice between an *IP address* and a *postal address*, depending on the application context. Understanding the context...

6 Creating a Common Domain Understanding in Dataspaces

- ...enables the recommender to provide suggestions of higher **relevance** aligned with the user's domain, scenario, or task,
- ...enhances **precision** by considering nuanced requirements, contributing to the overall quality of recommendations,
- ...may require varied approaches or structures, allowing the **customization** of suggestions to specific user needs,
- ...helps **disambiguation** of terms, ensuring clarity and preventing misunderstandings in semantic modeling and vocabulary development,
- ...contributes to the **consistency** of semantic models by aligning suggestions with existing structures, and
- ...improves **efficiency** by streamlining the modeling process, reducing the need for extensive manual adjustments, particularly in complex domains.

The context of a recommendation system can be automatically recognized [Pau+19] or explicitly entered by the user, for example, as a domain keyword [Zej21] in figure 6.21. This figure shows how the user can enter a domain label (marked in red) to enable better recommendations based on that context. The other red marks trigger the recommender or limit the number of results.

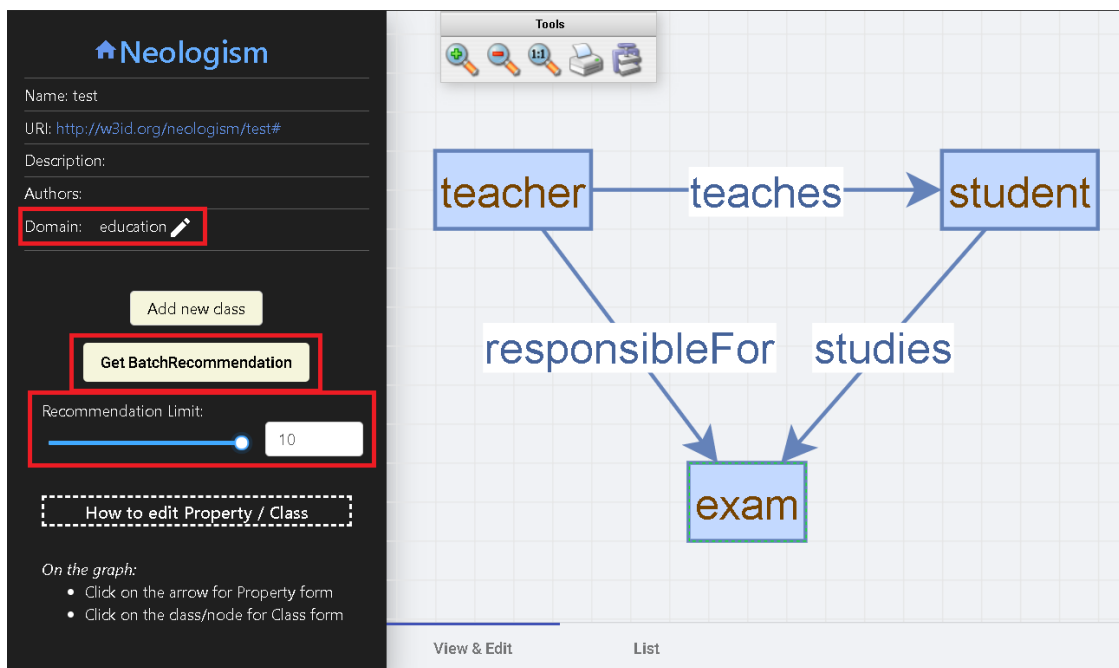


Figure 6.21: Screenshot of a vocabulary created in Neologism 2.0 on the right and controls on the left [Zej21].

Collect Recommendations

Building on the previous steps of our approach shown in figure 6.18, the task of collecting recommendations can use homogenized models or vocabularies as input and optionally leverage context information [Pau+19]. Recommendations can be gathered from knowledge bases.

Connections to knowledge bases can be established using connectors that are individually designed for each knowledge base to match its interfaces and to be generally interoperable with it. The input for a collection task is a label or keyword for which recommendations are requested. Preprocessing steps such as homogenization or providing context information can improve the quality of the results. The output of recommendation collection in our context is an unsorted set of selected concepts, which can be identified by [URIs](#) only, or by accumulated information such as labels or descriptions.

A query for recommendations for a single label to a single knowledge base connector would return an unordered set of concepts for that label. Note that implementations may even return sorted results, but for consistency and clarity we will not consider this in this section, but in the following section on *ranking*. See [listing 6.1](#) for an example query response. The raw query [URL](#)¹¹ of this listing was used on January 3, 2024. The listing shows an extract from a total number of 44 concept recommendations returned by the *Linked Open Vocabularies API*.

```

1 ["http://www.aktors.org/ontology/portal#PhD-Student",
2  "http://sw-portal.deri.org/ontologies/swportal#Student",
3  "http://swrc.ontoware.org/ontology#Student",
4  "http://vivoweb.org/ontology/core#Student",
5  "https://w3id.org/AIRO#Student",
6  "http://semanticscience.org/resource/SIO_000405",
7  "https://w3id.org/se0#Student",
8  "http://vivoweb.org/ontology/core#StudentOrganization",
9  "http://linkedscience.org/teach/ns#Student",
10 ... ]

```

Listing 6.1: Concept recommendations returned by the *Linked Open Vocabularies API* for a sample query for the single label *student*.

This can be represented as the following formula. Let L be the set of possible input labels, and C be the set of all available concepts or terms in a knowledge base. φ_{single} then returns an unordered set of n terms for a given keyword k [Zej21; Pau+19]. Note that the number of actual results may be less than n if a knowledge base does not provide a sufficient number of results for a query.

$$\begin{aligned}\varphi_{single} &: L \rightarrow C^n, \\ \varphi_{single}(l) &= \{c_1, \dots, c_n\}\end{aligned}$$

Furthermore, multiple labels can be queried at once, returning n recommended concepts for each of the m input labels [Zej21; Pau+19]. Note that if a knowledge base itself does not support such queries, its connector can simply implement them by merging the results for each label.

$$\begin{aligned}\varphi_{multiple} &: L^m \rightarrow C^{m \times n}, \\ \varphi_{multiple}(l_1, \dots, l_m) &= (\overbrace{\{c_{1,1}, \dots, c_{1,n}\}}^{\text{label 1}}, \dots, \overbrace{\{c_{m,1}, \dots, c_{m,n}\}}^{\text{label m}})\end{aligned}$$

¹¹<https://lov.linkeddata.es/dataset/lov/api/v2/term/search?q=student&type=class>

Note that we can use optional context information, represented by Ξ , to possibly create better results:

$$\varphi_{multiple, context} : L^m \times \Xi \rightarrow C^{m \times n},$$

$$\varphi_{multiple, context}((l_1, \dots, l_m), \xi) = (\overbrace{\{c_{1,1}, \dots, c_{1,n}\}}^{\text{label 1}}, \dots, \overbrace{\{c_{m,1}, \dots, c_{m,n}\}}^{\text{label } m})$$

See figure 6.22 for the proposed architecture to solve these tasks. It uses preprocessed and homogenized labels and optional context information (see section 6.5.4) to retrieve, process and return query results. The ranking steps used to optimize recommendations from each knowledge base are covered in the next section. From a technical perspective, a connector collects recommendations from a knowledge base as follows. It establishes a connection to the desired knowledge base, which can be done via a local or network connection. Authentication and clearing, such as payments, are usually handled automatically, for example by providing a token to a (remote) knowledge base service. The connector typically constructs a query specifically designed for the targeted knowledge base, including the label with additional information. After the query is executed, the connector processes the response and transforms it into a unified schema that enables consistent processing (e.g., selection and ranking) of the results.

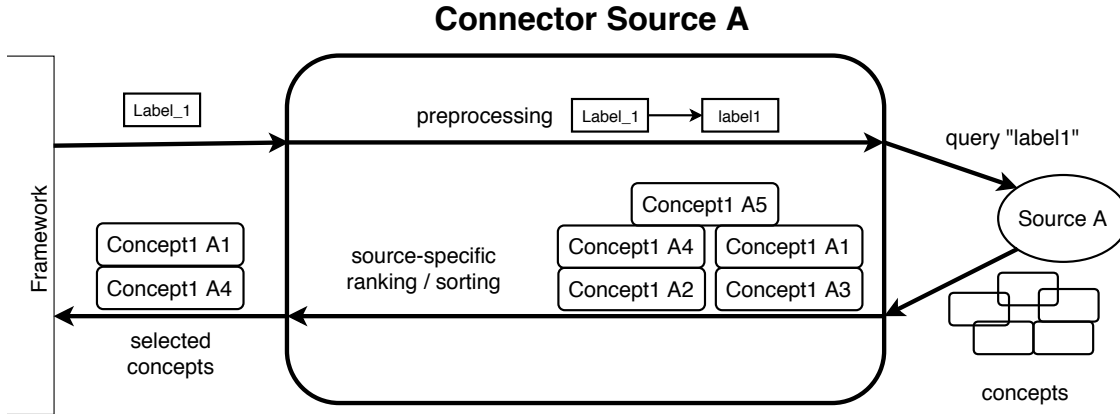


Figure 6.22: A connector designed for each knowledge base is used to collect recommendations from it [Pau+19].

As examples, we have implemented connectors for multiple knowledge bases. In [Pau+18], we demonstrate experimental implementations of an internal smart city KG, the Google Knowledge Graph [Sin+12], WordNet [Mil95; Pri22], and BabelNet [NP12]. In [Zej21], we demonstrate an implementation of a connector for the *Linked Open Vocabularies API* [Van+17b] and *BioPortal* (see table 6.9 and section 6.5.2 for both), and integrate a whole recommender solution into *Neologism 2.0* (see section 6.4). This solution is implemented in Java SDK version 1.8 and is available as open source on GitHub¹². It provides a REST controller for convenient remote access. A sample call in JSON format is shown in listing 6.2, and the representation of a portion of the response is shown in a user interface screenshot in figure 6.23. This visual representation in *Neologism 2.0* includes the URIs, labels, descriptions (comments), and ranking scores (see next section) of the collected recommendations.

¹²<https://github.com/Semantic-Society/Recommender>

```

1  {
2      "domain": "education",
3      "classes": ["teacher", "exam", "student"],
4      "properties": ["responsibleFor", "teaches", "studies"],
5      "limit": 10
6  }

```

Listing 6.2: Example content including a domain, three class names, and three property names of a vocabulary under construction submitted to the recommender [REST](#) interface [\[Zej21\]](#).

Keyword: student			
☐	URI: http://sw-portal.deri.org/ontologies/swportal#Student	Label: Student	Comment: Student Score: 1.00
☐	URI: http://linkedscience.org/teach/ns#Student	Label: Student	Comment: Student of the course. Score: 0.98
☐	URI: http://semanticscience.org/resource/SIO_000405	Label: student	Comment: A student is an individual who is attends an educational institution. Score: 0.98
☐	URI: http://vivoweb.org/ontology/core#Student	Label: Student	Comment: Score: 0.89
☐	URI: http://swrc.ontoware.org/ontology#Student	Label: Student	Comment: Score: 0.87
☐	URI: http://dev.poderopedia.com/vocab/StudentEducationalConnection	Label: Student	Comment: Score: 0.86
☐	URI: http://www.aktors.org/ontology/portal#Student	Label: Student	Comment: Score: 0.86
☐	URI: http://vivoweb.org/ontology/core#StudentOrganization	Label: Student Organization	Comment: A student organization is an organization Score: 0.82
☐	URI: http://courseware.rkbexplorer.com/ontologies/courseware#Student-Interaction-Type	Label: Student-Interaction-Type	Comment: A type of interaction in which a student engages Score: 0.79
☐	URI: http://semanticscience.org/resource/SIO_000876	Label: student advisor role	Comment: A student advisor role is the role of an Score: 0.73
☐	Keep current node or property.		

Figure 6.23: Extract of a response to the example query shown in [listing 6.2](#) for the keyword *student* [\[Zej21\]](#).

The main contribution of this section is a solution for collecting terms or concepts from arbitrary knowledge bases, such as ontologies or ontology search services, for further use in recommendations in dataspace.

Rank Recommendations

Following our approach described in [figure 6.18](#), the *rank recommendation* step processes the collected recommendations from each knowledge base and applies ranking metrics to create an overall recommendation result for the user. Note that we distinguish between intra source ranking, which selects and sorts results from individual knowledge bases (see [figure 6.22](#)), and inter source ranking, which combines and ranks recommendations from multiple knowledge bases (see [figure 6.24](#)). The inter source ranking builds on the initial individual intra source rankings. A high-level overview of this framework is shown in [figure 6.24](#), where ranking metrics are applied to the collected recommendations from multiple connectors (inter source scoring).

The input for a recommendation ranking task is a set of input labels and a set of recommendations collected from one or more knowledge bases via connectors. The output of this task in our context is as sorted list of selected and ranked concept recommendations for each of the given keyword. This includes first an intra source ranking for each connector, resulting in a sorted list of recommendations for each connector. This feature is supported

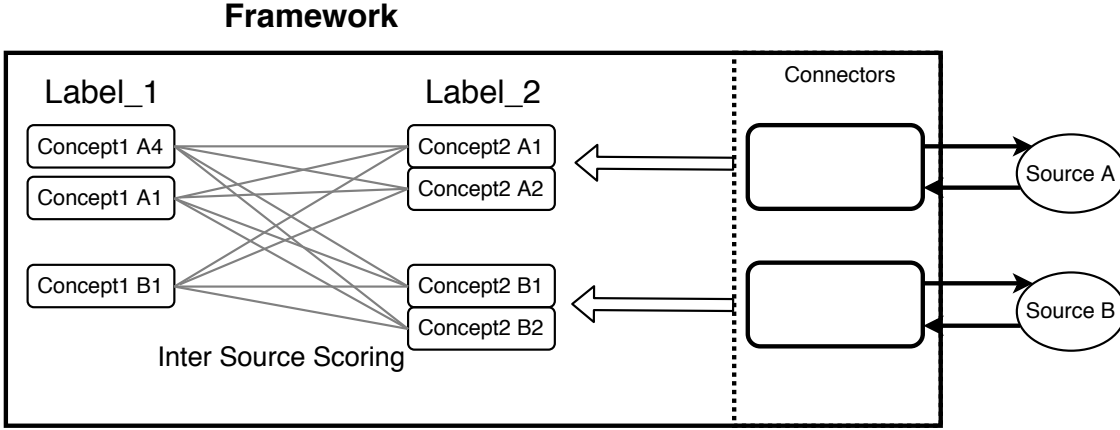


Figure 6.24: The recommendation ranking task involves processing recommendations from multiple connectors and applying metrics to calculate scores for an overall recommendation score [Pau+19].

by most knowledge bases, but can otherwise also be added to the connector. Combining both subsequent ranking methods, intra and inter source ranking, this can be represented by the following formula, which extends the formula φ from the previous section on *collect recommendations*: Let L be the set of possible input labels, and C be the joint set of all available concepts or terms in all connected knowledge bases, and $(c_{l1}, \dots, c_{ln})$ the collected n recommended concepts from a knowledge base for label l . $\varphi_{overall}$ then returns a single ordered list of n terms for each given label l , which are selected and ranked from these collected recommendations. These ordered lists for each keyword represents the complete, overall recommendations based on a combination of all connected o knowledge bases [Zej21; Pau+19]:

$$\begin{aligned}
 \varphi_{overall} : L^m \times C^{n \times o} &\rightarrow C^{m \times n}, \\
 \varphi_{overall}((l_1, \dots, l_m), &(\overbrace{(c_{1,1,1}, \dots, c_{1,n,1})}^{\text{label 1 from connector 1}}, \dots, \overbrace{(c_{1,1,o}, \dots, c_{1,n,o})}^{\text{label 1 from connector } o}), \dots) \\
 = (&\underbrace{(c_{1,1}, \dots, c_{1,n})}_{\text{label 1 joined from all connectors}}, \dots, \underbrace{(c_{m,1}, \dots, c_{m,n})}_{\text{label } m \text{ joined from all connectors}})
 \end{aligned}$$

Note that optional context information can be added similarly to the collection task in the previous section by integrating the context (Ξ) into the input of the ranking method:

$$\varphi_{overall} : L^m \times C^{n \times o} \times \Xi \rightarrow C^{m \times n}$$

In the following, we will first cover intra source ranking, which is done per knowledge base connector, and then inter source ranking, which combines all these individual results into one overall ranking.

Intra Source Ranking Rankings for individual knowledge base connectors can be achieved in two ways. Either the knowledge base itself supports ranking or this feature needs to be added to the respective connector. In any case, this steps takes m labels and sorts the collected recommendations for each of these. Technically, from each label's set of collected

unsorted recommendations C_m , it produces a list of sorted recommendations. Assuming the ranking metrics to be optimal for a given query task, the result is an optimally sorted list of recommendations for each input label, from a single knowledge base only.

Most APIs and other query interfaces of widely used knowledge bases actually provide intra source rankings. This holds for all knowledge bases used in our experiments. In cases where the intra source ranking must be done by the connector, the heuristics and other solutions for inter source rankings (see next paragraph) can simply be applied for intra source ranking, too.

Inter Source Ranking In this final step of recommendation ranking (see figure 6.18), the results of all connected knowledge bases are accumulated and ranked into a final result. The inter source ranking transforms the previously independent lists of recommendations from each connector into related lists that take into account the similarities and commonalities of their entries. This closes gaps when one knowledge base returns few or no results for some labels (e.g., because it only covers a certain domain), and at the same time improves the overall quality of the recommendations (e.g., because links or similarities between different knowledge bases can be used).

We propose an approach based on heuristics as follows [Pau+19]. In order to select the most appropriate concepts for each label, we compare the set of concepts C_{l_i} for each label l_i with those concepts C_{l_j} for each label l_j , where $j \in \{1...n\}$, $j \neq i$. For this reason, all concepts will be evaluated according to how they relate to other concepts. Under the assumption that multiple sources return results from different domains, we aim to identify the most prominent domain or combination of labels. The data sources, the knowledge bases, provide for concepts their name(s) and description, as well as possible related concepts (from the same knowledge base). Names are referred to as *main labels* in the following, and textual descriptions are referred to as *glossaries*. For these three properties, the total number of possible pairwise combinations is six. However, we do not compare the sets of related concepts with each other, as this would introduce too much inaccuracy. The remaining five possible combinations are used in the next paragraph to define similarity scores between two concepts.

The scoring is done by a comparison of keywords from the concept labels, the glossary, and the labels of related concepts. For two concepts c_a and c_b , a coherence score is calculated using five scores s_1 to s_5 . Each concept c_x is assumed to consist of a set of main labels ml_x , a glossary g_x , and related concept labels r_x . Note that in the RDF world, the term *concept label* is usually represented as `rdfs:label` and the term *glossary* is usually represented as `rdfs:comment`¹³ or `skos:definition`¹⁴. A keyword extraction algorithm KE is used to extract keywords from the glossaries, since they mostly contain natural language¹⁵.

¹³“Used to provide a human-readable description of a resource”, see <https://www.w3.org/TR/rdf-schema>.

¹⁴“Supplies a complete explanation of the intended meaning of a concept”, see <https://www.w3.org/TR/skos-primer/>.

¹⁵We use <https://stanfordnlp.github.io/CoreNLP/>

$$\begin{aligned}
c_x &:= (ml_x, g_x, r_x) \\
s_1(c_a, c_b) &:= \frac{|KE(g_a) \cap KE(g_b)|}{\min\{|KE(g_a)|, |KE(g_b)|\}} \\
s_2(c_a, c_b) &:= |ml_a \cap KE(g_b)| \\
s_3(c_a, c_b) &:= |ml_a \cap r_b| \\
s_4(c_a, c_b) &:= |r_a \cap r_b| \\
s_5(c_a, c_b) &:= |ml_a \cap ml_b|
\end{aligned} \tag{6.1}$$

The equation (6.1) shows the definition of a semantic concept c_x for the algorithm and the five sub-scores mentioned above [Pau+18].

- s_1 weighs the glossary keyword matches by comparing all found keywords for both concepts and scoring matches to the total number of possible matches, where the total number of possible matches is equal to the number of items in the smaller keyword set.
- s_2 yields scores if labels from one concept match keywords from the second concept's glossary.
- s_3 scores the number of concept c_a labels matching labels in c_b 's related concepts in the same knowledge base (e.g., `subClassOf` relations). From related concepts, only the labels are checked, not the glossary or even more related concepts.
- s_4 increases if matches of both concepts' related entities exist (e.g., sibling relation indicated by same parent element).
- s_5 compares the labels of both concepts and increases if there is a direct match.

The coherence score for a concept c_k is calculated using the set of concepts C_{l_i} for each label l_i , where C_l is the set of concepts for all labels. The total score S for the concept c_k is then defined as:

$$S(c_k) = \sum_{c_o \in C_l \setminus C_{l_i}} \sum_{p \in 1, \dots, 5} \omega_p \times s_p(c_k, c_o) \tag{6.2}$$

Each score is weighted by ω to allow fine tuning of the results for each comparison. An example demonstration of this inter source ranking can be seen in figure 6.25. In this sample, both evaluated concepts have two labels, namely *train* and *railroad train*, and *railway* and *railroad*. String matches at *railroad* boost score ω_5 . The *isA* links are similar to `rdfs:subClassOf` and for example can boost sub-score ω_4 . The circled terms at the bottom represent keywords extracted from both concept's glossaries, which contribute to the sub-scores ω_1 and ω_2 . In summary, each match on one score yields the mentioned ω value, and thus increases the score. By performing this calculation for each concept $c_k \in C_{l_i}$ and each $C_{l_i} \in C_l$ we receive a coherence score S for each concept of each label.

We proposed two example implementations of these intra and inter source ranking approaches: An integrated solution in the ESKAPE platform uses the knowledge bases *WordNet*, *BabelNet*, and a local smart city KG [Pau+19], and a standalone recommender solution leverages the *Linked Open Vocabulary API* [Van+17a] as knowledge base to recommend ontology concepts [Zej21].

In summary, the recommendation ranking step processes the recommendations, which were previously collected from one or more knowledge bases, and ranks these in two steps with different metrics in order provide best-possibly sorted recommendations for each label.

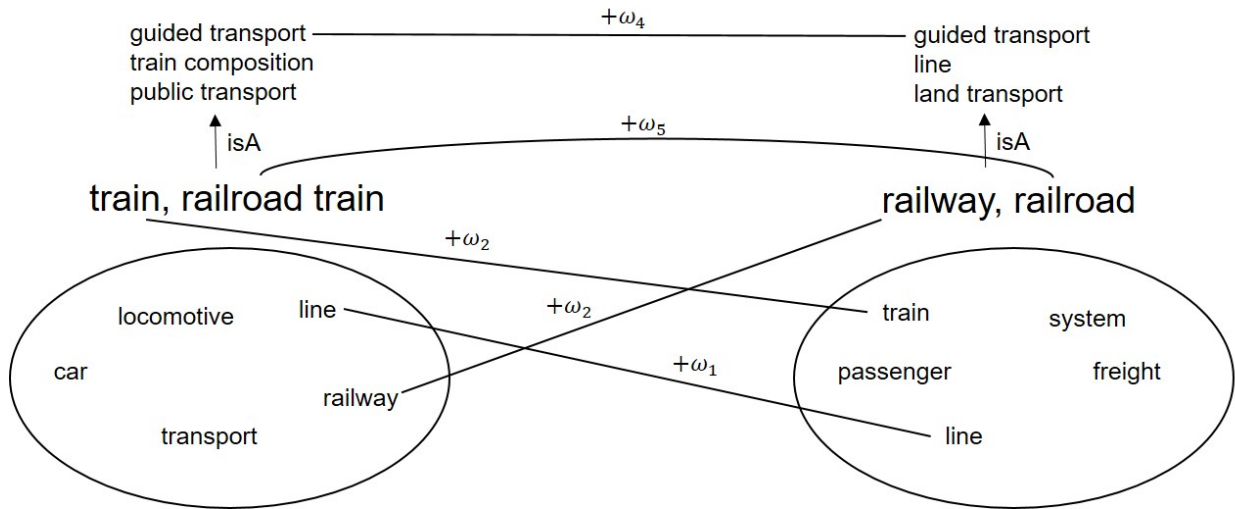


Figure 6.25: Example calculation of the sub-scores ω_1 to ω_5 for inter source ranking [Pau+18].

Apply Recommendations

The final step of the approach shown in figure 6.18 is to apply the collected and ranked recommendations to the target system that needs them. Depending on whether the labels for which recommendations are needed come from a semantic modeling context or a vocabulary development context (both see figure 6.18), the target system differs. Our results cover both options and include implementations for both the ESKAPE platform and Neologism.

We demonstrate the application of recommendations to semantic modeling tasks by an implementation of such in the ESKAPE platform. ESKAPE automatically extracts data models and users create corresponding semantic models to express a domain understanding (see figure 6.17). During creation, these semantic models only consist of plain text labels with directed connections between them. With the help of collected and ranked recommendations, the platform can replace ambiguous text labels with unambiguous ontology URIs. A screenshot of ESKAPE’s extended user interface is shown in figure 6.26. In this example, the user can choose from two recommendations for the text label *temp*, which are highlighted in red. Only for simplicity of this screenshot, the ontology URIs are shown as the labels *temperature* and *room temperature*.

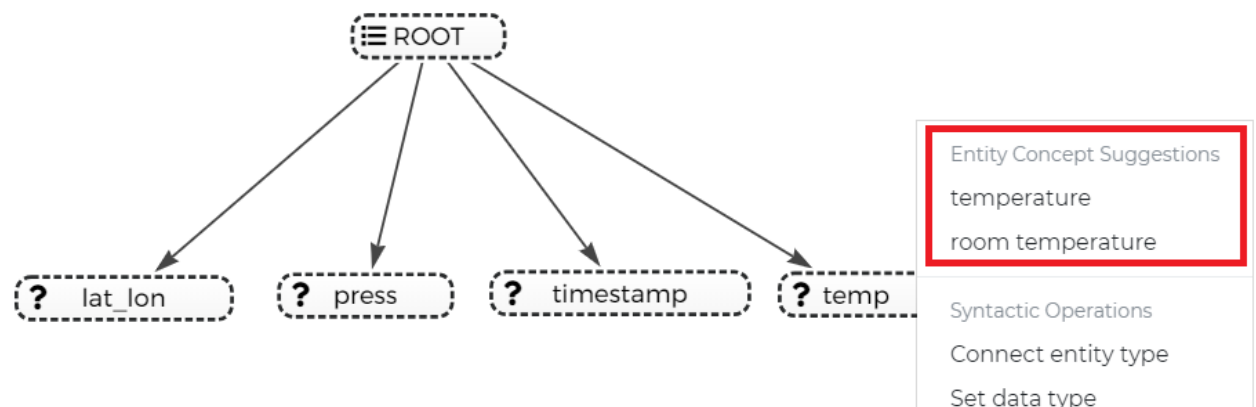


Figure 6.26: In the ESKAPE platform [Pom+18a], recommendations can be applied to a semantic model via a pop-up menu [Pau+19].

The overall workflow in ESKAPE is for the user to select recommendations for all entities in their semantic model (and only introduce a new concept if no existing one fits). This leads to a resulting semantic model in which ambiguous labels are replaced by unambiguous PIDs such as ontology URIs. When a data provider shares this semantic model with other participants in a dataspace, they are more likely to have the same domain understanding as the data provider. This improved common understanding of shared data and other resources in dataspace greatly increases their (re)usability.

Likewise, we demonstrate the application of recommendations to vocabulary development tasks by implementing them in Neologism. We develop a standalone recommender API, available as open source on GitHub¹⁶, as a REST endpoint to serve requests for recommendations. Users first create their ontology drafts in Neologism’s graph-based user interface (see figure 6.21 on the right). Similar to the implementation approach above in ESKAPE, applying recommendations means replacing plain text labels (in this example *teacher*, *student*, etc.) with ontology URIs. A sample call in JSON format is shown in listing 6.2. For each label, Neologism provides a list of recommendations for the user to choose from (see figure 6.23). Besides the URI, the user is shown a text label (`rdfs:label`), a textual description (`rdfs:comment` or `skos:definition`), and a ranking score to facilitate understanding (see figure 6.27). In this figure, details such as ontology URI, label, and description are shown to the user to help them evaluate whether this recommendation fits their intended content. If no existing ontology concept fits, the user can choose to keep the current draft and introduce a new ontology concept including a URI. Repeating this step for all labels in the vocabulary design results in a vocabulary design where all entities are replaced by unique PIDs, such as ontology URIs. Whenever this ontology draft or further developments of it are used or shared in dataspace, other participants can use the linked PIDs and will have a better common understanding of the domain. This greatly increases the (re)usability of resources such as datasets and their descriptions in dataspace.

Class Name
Teacher
.....
Class URI
http://linkedscience.org/teach/ns#Teacher
.....
Description
Teacher of the course.
.....

Figure 6.27: Our extension for Neologism [Zej21] includes showing additional information for each offered recommendation from the list displayed in figure 6.23.

6.5.5 Evaluation

When evaluating the two recommender systems for vocabulary use in dataspace, it is crucial to consider their proposed recommendations in the context of the target applications. The scenarios discussed in this section, *semantic modeling* and *vocabulary development*, both

¹⁶<https://github.com/Semantic-Society/Recommender>

involve a human in the loop. The role of users is integral in determining the effectiveness of our developed recommender systems. The main objective of these evaluations is to assess the match between the recommendations provided by our systems and the expectations of the users acting in dataspace. By including the human element, we gain insights from quantitative metrics of how well recommendations match intended meaning and from qualitative aspects of user experience and satisfaction.

In both cases, users will actively utilize the recommendation systems for *ESKAPE* or *Neologism*, respectively. This will enable us to assess the significance and practical usefulness of our recommendations in real-world scenarios. The assessments will not only evaluate the precision of concept recommendations, but also explore user perceptions of semantic clarity and the usability of the system. Our evaluation aims to collect valuable feedback for vocabulary recommendations to ensure that they are not only technically sound, but also effectively address the diverse expectations and preferences of dataspace participants.

We conduct **the first evaluation scenario (*semantic modeling*)** in the *ESKAPE* platform as follows [Pau+19]. The task for users is to create a semantic model for given datasets with the help of our recommender system. More precisely, users select the concept from lists of recommendations, which they judge as best fitting for their domain-specific semantic model. An automatic tracking of chosen list positions yields quantitative results. For example, a recommendation system’s decision is considered successful if it has placed a recommendation high on the recommendation list and the user selects it as the best match. In contrast, the decision is considered less successful if the user has to scroll far down the recommendation list to select a suitable concept.

Table 6.12: The name, source, number of labels, and description of the publicly available datasets used to evaluate our proposed recommender system for the task of *semantic modeling*.

Name	Source	Labels	Description
Crime Data	Vancouver Open Data ¹⁷	10	Crime statistics from the public domain
Cultural Spaces	Vancouver Open Data ¹⁸	11	Locations and attributes of cultural spaces
Recycling	European Open Data ¹⁹	7	Waste and recycling sites
Clearance	San Francisco Open Data ²⁰	6	Clearance heights for large vehicle circulation
Parking	San Francisco Open Data ²¹	12	Parking meter operating schedules

¹⁷Used <http://data.vancouver.ca/datacatalogue/crime-data.htm> is offline; refer to <https://vpd.ca/crime-statistics/>

¹⁸Used <http://data.vancouver.ca/datacatalogue/culturalSpaces.htm> is offline; refer to <https://opendata.vancouver.ca/explore/dataset/cultural-spaces>

¹⁹Used URL <https://www.europeandataportal.eu/data/en/dataset/east-sussex-county-council-recycling-sites> is offline; refer to <https://data.europa.eu/data/datasets/east-sussex-county-council-recycling-sites>

²⁰Used <https://data.sfgov.org/Transportation/Clearance-Heights-for-Large-Vehicle-Circulation/sccd-iwvp> is offline; refer to <https://data.sfgov.org/Transportation/Clearance-Heights-Bridges-and-Underpasses/eb2x-6eay>

²¹<https://data.sfgov.org/Transportation/Meter-Operating-Schedules/6cqq-dxku>

Table 6.13: Heat map of how often, on average, recommendations from the top three list positions were chosen in different scenarios.

	Knowledge Base	WordNet			BabelNet			Local KG			Combined		
	List Position	1	2	3	1	2	3	1	2	3	1	2	3
Dataset	Crime Data	3	1	2	6		1	3			6		2
	Cultural Spaces	4	2		5	1	1				6	1	
	Recycling	4		1	4		1				4		1
	Clearance	1	2	1	2	1		3			2	2	
	Parking	1	1		1		1	5			5		1

Arbitrary, publicly available datasets with an appropriate number of labels were used for this evaluation. See [table 6.12](#) for more information. A summary of the frequency with which recommendations from the top three list positions were selected, on average, across various scenarios is presented in [table 6.13](#). The evaluation shown in this table includes various knowledge bases (referred to as *sources*) and datasets referenced in [table 6.12](#) in the recommender evaluation section. Empty cells in the table indicate instances where no selections were made from the top three recommendations, for the sake of readability. For example, when building a semantic model for the *crime data* dataset using a recommender system based solely on *WordNet*, the top recommendation was selected three times, the second recommendation once, and the third recommendation twice. In contrast, the local **KG** performed suboptimally for the *recycling* dataset, as users never selected any of the top three recommendations. The best results were achieved when a general-purpose knowledge base was combined with a domain-specific **KG**, as shown in the *combined* column.

The quality of our recommender system’s top 3 recommendations is demonstrated by the fact that users selected a concept from the top 3 recommendations frequently in all data sets. Furthermore, a detailed analysis of all individual semantic models yielded the following qualitative results. As with the crime data set, missing concepts were primarily caused by too specific labels, such as **applied color rule** or **cultural_space_name**. However, using multiple knowledge bases simultaneously has a positive effect, as it allows for a wider range and combination of domains.

In an additional, qualitative evaluation, we compared our approach to *Babelfy*²², an online algorithm for context-aware semantic annotation. Babelfy considers relations in the underlying BabelNet network when annotating sentences, keywords, or labels simultaneously. Our evaluation (see [Pau+19] for details) uses on an exemplary set of labels related to IT infrastructure. Both systems, Babelfy and our recommender system operate on the same single knowledge base BabelNet to ensure that the results are comparable. The results are presented in [table 6.14](#). In this table, text in parentheses indicates domains for ambiguous labels. Good recommendations, as rated by the user, are highlighted in bold. Both systems successfully identify unambiguous labels such as *computer*, while the interpretation of the acronym *lan* or *network* differs between the two, lacking a clear superior concept. In the case of ambiguous labels like *subnet*, *address*, and *port*, our system outperforms Babelfy. Babelfy returns common concepts without context and, for *subnet*, even selects a subpar second concept related to mathematical topology. While we acknowledge that this random sample of labels may not represent general input, it demonstrates strengths of our ranking algorithms over this Babelfy baseline in recommending semantic concepts for labels.

²²<http://babelfy.org/>

Table 6.14: Detailed qualitative comparison between the results of Babelfy and our proposed recommender system for semantic modeling.

label	Babelfy result	Framework result
computer	computer	computer
lan	local area network	local area network
network	[interconnection of things or people]	Internet
subnet	subnet [mathematic topology]	subnetwork [IP]
address	address [postal]	IP address
port	port [sea port]	interface [computing]

We conduct **the second evaluation scenario (vocabulary development)** in the *Neologism* vocabulary drafting tool as follows [Zej21]. The evaluation aims to measure the quality of our proposed recommendation system, the complexity of the task, and the usability of our recommender integration in Neologism. As in the first evaluation scenario, we evaluate the quality of recommendations by automatically tracking which recommendations users select as match for their tasks. From the ontology algorithms presented in section 6.5.2, we choose the baseline for our experiments based on the number of citations when this thesis was written. According to Google Scholar, AktiveRank [ABS06] was cited 251 times, Linked Open Vocabularies [Van+17a] was cited 416 times, and the other approaches presented were cited less than 50 times each. Thus, we use the Linked Open Vocabularies API as the baseline for our evaluation results, see section 6.5.2 and table 6.9. To measure the usability of the implementation, we use the System Usability Scale questionnaire [Bro13], which can be seen in figure 6.28. It is a commonly used measure of system usability. The individual scenario complexity for each of the users is measured using the After-Scenario Questionnaire [Lew91], which can be seen in Figure 6.29. Both of them are short and simple and cover the complexity of the evaluation tasks as well as the overall usability of our integration into Neologism.

The task for users is to create vocabulary drafts for given use cases with the help of Neologism and our attached recommender system. More precisely, users receive tabular datasets for which they create classes and relations between them in a graph editor (see figures 6.16 and 6.21) with the help of our recommender system. A handout document with straightforward explanations ensures that everyone understands the task, regardless of their prior knowledge of vocabulary development. The participants in this evaluation come from diverse backgrounds, including economy, engineering, research, and mobility. For instance, some of them are part of the *Mobility Data Space* project, as seen in section 2.1.1 or [Dre+21; Fra19]. The three datasets used in the second evaluation scenario cover the domains of weather, Covid-19 test stations, and Open Data datasets, as shown in excerpts in figure 6.30.

The results of the second evaluation scenario (vocabulary development) are as follows. Figures 6.31 to 6.33 prove that our recommender system produces recommendations of high quality for both classes and properties and are discussed in detail below.

Figure 6.31 compares the performance of our proposed recommender system (blue line) with the baseline *Linked Open Vocabularies* (red line), for classes on the left plot and properties on the right plot [Zej21]. The x-axes represent the *selected recommendation rank*, which ranges from 1 to 10, and represents the position of a recommendation in a list presented to a user, with 1 being the top recommendation. The y-axes represent the cumulative quantity, which ranges from 0 to 1 (0 % to 100 %), and represents the number of recommendations at or below one rank (cumulative) selected by users on average. The red line represents

6 Creating a Common Domain Understanding in Dataspaces

For each of the following statements, please mark one box that best describes your reactions to Ontology Prototyping today.

	Strongly disagree						Strongly agree			
1. I think that I would like to use Ontology Prototyping frequently.	☐	1	☐	2	☐	3	☐	4	☐	5
2. I found Ontology Prototyping unnecessarily complex.	☐	1	☐	2	☐	3	☐	4	☐	5
3. I thought Ontology Prototyping was easy to use.	☐	1	☐	2	☐	3	☐	4	☐	5
4. I think that I would need the support of a technical person to be able to use Ontology Prototyping.	☐	1	☐	2	☐	3	☐	4	☐	5
5. I found the various functions in Ontology Prototyping were well integrated.	☐	1	☐	2	☐	3	☐	4	☐	5
6. I thought there was too much inconsistency in Ontology Prototyping.	☐	1	☐	2	☐	3	☐	4	☐	5
7. I would imagine that most people would learn to use Ontology Prototyping very quickly.	☐	1	☐	2	☐	3	☐	4	☐	5
8. I found Ontology Prototyping very cumbersome (awkward) to use.	☐	1	☐	2	☐	3	☐	4	☐	5
9. I felt very confident using Ontology Prototyping.	☐	1	☐	2	☐	3	☐	4	☐	5
10. I needed to learn a lot of things before I could get going with Ontology Prototyping.	☐	1	☐	2	☐	3	☐	4	☐	5

Figure 6.28: Excerpt of the System Usability Scale [Bro13] questionnaire adapted and used for the second evaluation scenario *vocabulary development* [Zej21] in Neologism.

1. Overall, I am satisfied with the ease of completing this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

2. Overall, I am satisfied with the amount of time it took to complete this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

3. Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

Figure 6.29: Excerpt of the After-Scenario Questionnaire [Lew91] used for the second evaluation scenario *vocabulary development* [Zej21] in Neologism.

the baseline system, and the blue line represents our proposed recommender system. For example, the 0.8 at rank 4 on the blue line in the left plot means that 80 % of the time the desired class is between ranks 1 and 4 in the recommendation list presented by our system. The blue line is consistently above the red line for both classes (left plot) and properties (right plot). This shows that our system consistently presents recommendations that users select as most appropriate higher than the baseline. In other words, our suggestion system is providing better recommendations than the *Linked Open Vocabulary* baseline system.

Figure 6.32 shows a violin plot that visualizes the effectiveness of your proposed recommender system [Zej21], which re-ranks recommendations from the *Linked Open Vocabularies API*. The y-axis represents the initial ranks of concepts in the *Linked Open Vocabularies* (baseline), while the x-axis shows their final ranks after your system has applied intra and inter source rankings (see *rank recommendations* in section 6.5.4). The violin plots indicate the distribution of final ranks for each initial rank, over all experiments. A key observation

event	severity	certainty	headline	instruction	parameter	value
Schneeverwe	Moderate	Likely	Amtliche WARNUNG vor SCHN	Stellenweise	None	
Starkwind	Minor	Likely	Warnhinweis vor STARKWIND	Null	None	
Leichter Schn	Minor	Likely	Amtliche WARNUNG vor LEICH	Null	<5 [cm]	

(a)

Teststellen	Name	Testste	Standort	Test	Standort	Standort	T	Email	Internet	Stadtteil	Latitude	Longitude
11-001	Mitsubishi-El	Siegburgerstr.	40591	DÄ	%sseldorf			https://coron	Oberbilk	51.206.234	6.808.547	
11-002	MedCo Testz	Vogelsanger \	40470	DÄ	%sseldo	info@mec	www.medco-	MÄ	rsen	5.125.725	6.799.792	
11-003	Corona-Test-f	Umlandstra	Ä	40237	DÄ	%sseldo	info@sma	www.smartm	DÄ	%sselt	51.231.518	6.801.278

(b)

keyword	mbox	name	identifier	title	description	homepage	language	format	issued
Korruptio	umbo@	Auswärtiges	f3b57b84	Maßnahmen	Liste mit M	https://ww	deutsch/	XML	2014-09-25T00:00:00
hotel	bast@rv	Regionalverla	1302c5b	POI - Hotels	l Daten-Dow	https://dat	DEU	CSV	2019-08-01T00:00:00
Entwickl	RLGS21	undesminist	fc997483	BMZ Projekt	Daten und I	https://ww	ENG	XML	2014-09-25T00:00:00

(c)

Figure 6.30: Excerpts of the three publicly available datasets used for the second evaluation scenario [Zej21], including the domains *weather*, *Covid-19 test stations*, and *Open Data datasets*. For more details, see [Zej21].

from this plot is that many concepts initially ranked up to over 40 in the baseline end up in the top 10 ranks after our algorithms are executed. This, together with the results of figure 6.31 that our system performs better, proves that the additional steps of intra and inter source ranking significantly improve the overall quality of recommendations. The red dashed line in the figure represents a boundary related to the user interface, which typically displays only the top 10 recommendations. This context further emphasizes the effectiveness of your recommender system, as it often promotes recommendations to ranks within this top 10 limit, even when their initial ranks in the baseline are much lower. This ensures that users are presented with higher quality recommendations when interacting with Neologism’s user interface. This effectively demonstrates the improvement our system brings to the recommendation process. In summary, this plot provides a clear visual evidence that your recommender system outperforms the baseline Linked Open Vocabularies in terms of ranking quality, ensuring that users are consistently presented with the most relevant and highest quality recommendations.

The aggregated results of the *System Usability Scale* questionnaire completed during our experiments are shown in figure 6.33 and demonstrate the good usability of the integration of our proposed recommender system into the user interface of Neologism. The complete questionnaire is depicted in figure 6.28. All user feedback is accumulated into one final score between 0 and 100, represented by the x-axis. Our system achieves a score of 88, which is between *excellent* and *best imaginable* in the adjective ratings, or a B+, close to A-, in the grade scale, or well-positioned in the highest score of the *acceptability ranges*. This score proves the high usability of the integration of our proposed recommender system into the graph-based editor of the Neologism vocabulary drafting tool.

The aggregated results of the *After-Scenario Questionnaire*, as presented in figure 6.29, are shown in figure 6.34 and describe qualitative aspects of user experience and satisfaction. The x-axis represents the three questions about the ease of completing the task, the amount of time needed to complete the task, and the support information when completing the task. The y-axis represents the responses in a violin plot, ranging from 1 (very satisfied) to 7 (not

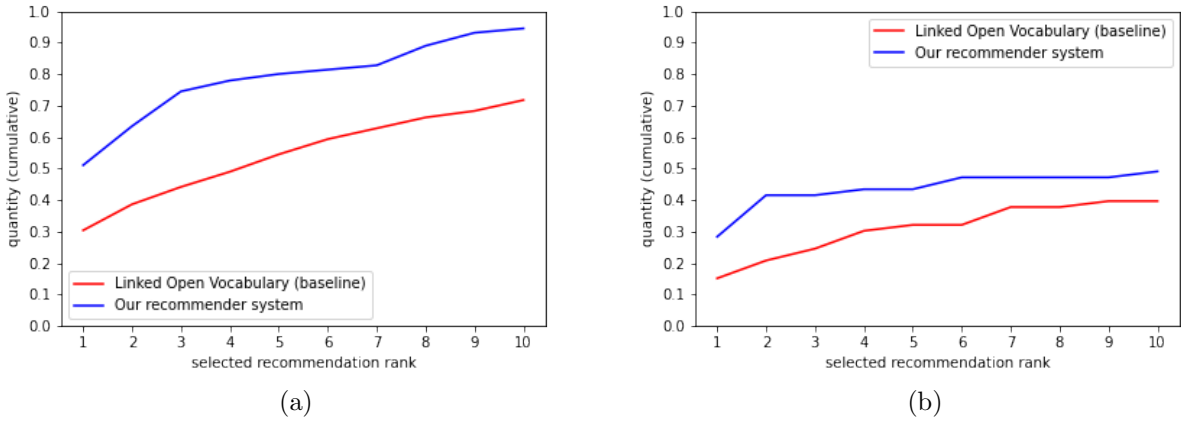


Figure 6.31: Performance comparison of our recommender system and the baseline *Linked Open Vocabularies* for (a) classes and (b) properties.

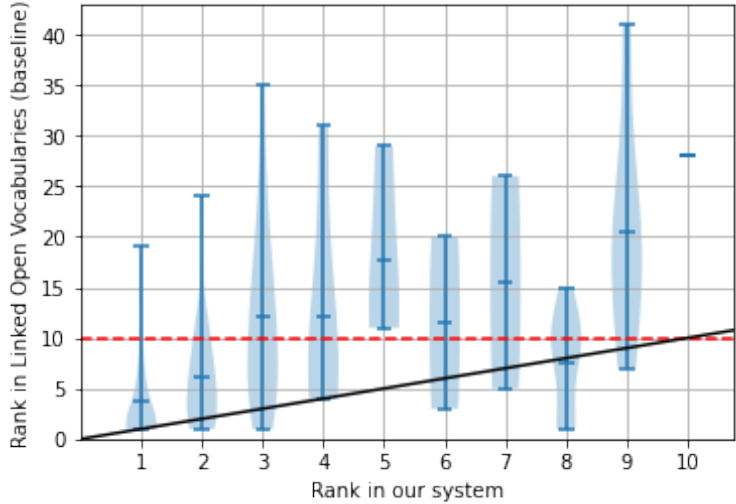


Figure 6.32: Effectiveness of the inter and intra source rankings of our proposed recommender system.

satisfied). On average, users were moderately satisfied with the ease of completing the tasks, while some users were very satisfied and others were not very satisfied. This is probably due to the diversity of participants in our evaluation, including users with and without prior knowledge of vocabulary development. The same is true for question 2 about the time required. The responses to question 3 show that the participants in our experiments were very satisfied with the introduction and example material provided. In summary, the results of the *After-Scenario Questionnaire* show that the tasks and support information for this evaluation were more than adequate.

In conclusion, the evaluations of the recommender systems for vocabulary use in dataspace, specifically within the contexts of semantic modeling and vocabulary development, have been successful. The evaluation systems, ESKAPE and Neologism, have proven their practical usefulness in real-world scenarios. The recommendations provided by our proposed systems effectively aligned with the users' expectations and the intended meaning. The precision of concept recommendations was high, and user perceptions of semantic clarity and system

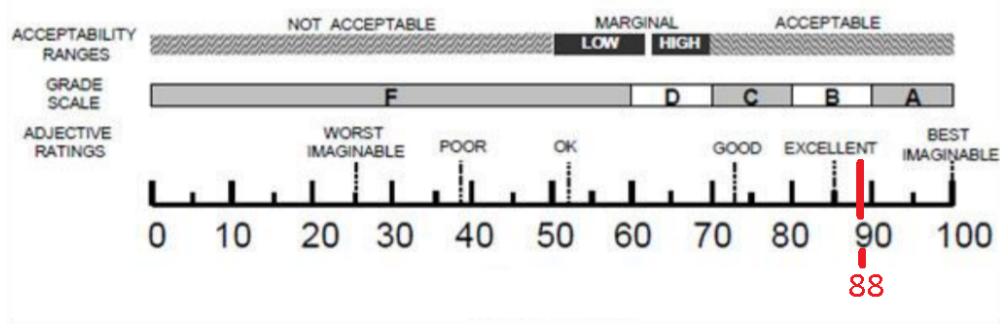


Figure 6.33: The results of the *System Usability Scale* questionnaire (see figure 6.28) indicate an excellent usability of the proposed recommender system.

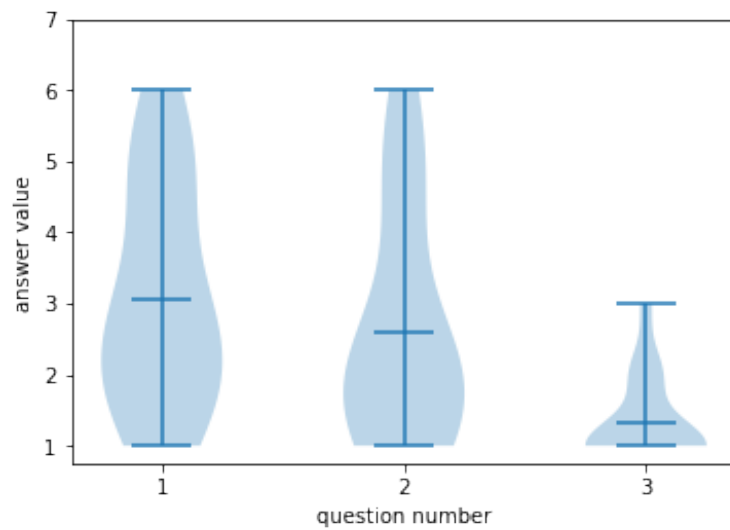


Figure 6.34: User experience and satisfaction results from the *After-Scenario Questionnaire* (see figure 6.29).

usability were positive. The feedback collected has confirmed that our vocabulary recommendations are not only technically sound but also cater to the diverse expectations and preferences of dataspace participants. This successful evaluation is a testament to the effectiveness of our recommender systems and their potential to significantly support creating a common domain understanding in dataspace.

Contributions

In conclusion, this section has successfully addressed [requirement Domain-R4](#) by proposing effective recommendation systems for vocabulary use in dataspace. The use of semantic annotations, as discussed in [section 2.2.1](#), has been instrumental in creating and sharing metadata based on common languages or vocabularies.

The contributions of this section include:

- **Recommendation Systems:** Effective recommendation systems for vocabulary use in dataspace, successfully addressing [requirement Domain-R4](#). These systems ensure that different vocabularies share more common contents, facilitating the reuse of common knowledge or concepts.

- **Automation of Vocabulary Selection:** We automated the process of vocabulary selection, which is often difficult and time-consuming due to the large number of vocabularies available. This has significantly reduced the time loss, costs, and complexity of data (re)use in dataspace.
- **Practical Impact Demonstrations:** We demonstrated the practical impact of these recommendations through two key applications:
 - **Neologism 2.0:** The vocabulary drafting solution, Neologism 2.0, presented in [section 6.4](#), serves as a testament to the practical impact of these recommendations.
 - **ESKAPE Platform:** The integration of these recommendations into the ESKAPE platform further demonstrates their practicality and effectiveness in real-world applications.

These contributions have advanced the field of vocabulary use in dataspace, and the positive evaluation of the proposed systems and methodologies attests to their effectiveness and practicality. Through successful integration into applications with graphical user interfaces, the contributions enable recommendations for vocabulary use to a broad range of users and thus contribute to a better common domain understanding in dataspace.

6.6 Evolving Semantic Annotations

In this section, we address [requirement Domain-R4](#) and propose methodologies to evolve semantic annotations of resources. Semantic annotations or descriptions are created either manually or (partially) automatically and usually represent only a snapshot in time. In a constantly changing and complex world, we need methods for continuously extending, improving, and combining semantic annotations of data and other resources in dataspace. Examples of dataspace assets that may require updates are shown in [table 6.15](#). The level (*data* or *vocabulary*) at which updates must be applied is specified in the *type* column. *Data* indicates assets where updates are typically made at the content data level, using shared vocabularies. *Vocabulary* marks assets where updating typically means changing the shared vocabularies themselves.

These examples highlight the dynamic nature of dataspace and the need for methodologies to continuously extend, improve, and combine semantic annotations of data and other resources. This is especially true for the *assure*, *describe*, *discover*, and *publish* phases of the domain layer illustrated in [figure 6.1](#), but partly also for the other phases highlighted there. Methods for continuously evolving semantic annotations are crucial to ensure that the descriptions of these artifacts remain accurate and up-to-date, facilitating effective data (re)use.

While *data* changes can usually be handled by program logic and (semi)automatic integration, *vocabulary* changes are more complex and require more sophisticated methods. This section therefore focuses on this second group of changes and proposes methodologies for evolving knowledge representations such as data models or ontologies.

A potential use case for this task is as follows (see [\[PLM19b\]](#) for details): A multinational manufacturing enterprise utilizes dataspace to organize and share its diverse data, and adds semantic annotation to the data. However, several months after creating the semantic annotations for the first time, additional attributes are necessary to properly describe the

Table 6.15: Examples of dataset assets that may require updates over time.

Asset	Description	Type
Datasets	As datasets get new data, their semantic annotations may need updates to account for new variables or data distribution changes.	Data
APIs	APIs changes, like new endpoints or altered data structures, require updates to their semantic annotations.	Data
Ontologies	Ontologies may need semantic annotation updates as they evolve with our changing understanding of a domain.	Vocabulary
Data Standards	Updates to data standards, reflecting new data types or usage changes, require corresponding semantic annotation updates.	Vocabulary
Data Models	Changes in data models, due to underlying data shifts or usage alterations, necessitate semantic annotation updates.	Vocabulary
Software Artifacts	Changes in software artifacts like services or frameworks require semantic annotation updates to reflect their altered functionality or usage.	Data
Participants	Evolving roles and responsibilities of dataspace’s participants may require updates to their semantic annotations.	Data

data. Describing evolving data using static annotations can be challenging, often requiring external expertise and resulting in high costs and delays in providing data in the dataspace. Therefore, there is a crucial need for an approach that enables the continuous evolution of annotations, even for non-experts in dynamic dataspace contexts.

The objective of this section is to provide methodologies to facilitate such tasks. Given a knowledge representation such as an ontology at a certain point in time, evolve it, possibly with the help of external knowledge bases, to reach an intended, updated state. More specifically, we propose an approach that consists of an evolving **KG** containing an internal ontology. As dataspace participants add new data and annotate it with concepts from the ontology, the ontology is continuously evolved on-demand. To accomplish this, we propose a system that uses external knowledge bases to assist users in maintaining the ontology by providing evolution strategies. Methods to ensure high quality of the ontology during its development are also investigated. We evaluate the accuracy and usability of our approach through a user study. We measure how the evolved ontology, as part of a **KG**, contributes to a common domain language in dataspaces.

6.6.1 Related Work

Related work in this context includes approaches that maintain a knowledge representation, including one or more ontologies, and are focused on semantic annotation or modeling. Please refer to [table 6.16](#) for an overview of these related works, which are presented in the following. The approaches use zero to three knowledge bases and some even allow to extend

them with new ones. Knoblock et al. support only one ontology at a time, but it can be changed for another dataset. All approaches support classes and most of them also support relations (properties). The supported semantic tagging techniques range from manual (M) to semi-automatic (SA) to automatic (A). The evolution of KGs can be context-aware (CA) or context-unaware (CU). Although Wikipedia as the basis of some approaches (*) is constantly growing, their KGs do not evolve based on user input processed for semantic tagging. While most approaches fully trust input from arbitrary users, Du et al. use peer review for quality assurance.

Table 6.16: Related approaches in the area of semantic annotation that maintain a knowledge representation including one or more ontologies [PLM19b; Lip17].

Approach	Knowledge Bases	Extendable	Relations Support	Tagging	KG Evolution	Trusting User Input
ESKAPE [Pau+19]	1		✓	SA	CU	fully trust
UNIpedia [MJS10; KD12]	3			A	(*)	
Bashar et al. [BLG16]	2		✓	A	CU	fully trust
Mining Wikipedia [MW13]	1		✓	A	(*)	
Knoblock et al. [Gup+12; Tah+14]	1	(✓)	✓	SA	no	fully trust
Di Caro and Boella [DB+16]	2		✓	A	(*)	
Liu et al. [Liu+12]	1			A	(*)	
Du et al. [Du+12]	0			M		peer review
FuhSen [Col+16; Col+17]	3	✓	✓(3)	A		
Cluster Tagging [HPL15]	1			SA		
Delicious Concept Augmentation [Tes+08; LEC+07]	1		✓	A		
MAFRA [Mae+02]	2	✓	✓		CA	fully trust
Hughes et al. [HCM09]	2+	✓	✓	M		
Giunchiglia et al. [GAP12]	2+	✓				
Temporal Concepts [Kuz16]	1	✓	✓		CU	
Semdrops [Tor+11]	0		✓	M	CA	fully trust
Ontology learning [Xia+16]	1		✓		CA	

The ESKAPE [Pau+19] information platform uses semantic models in addition to data models to manage heterogeneous data. See also section 6.5.1. Users do not have to consider formats or models of data sources, but can process and query data on an information-centered level. ESKAPE does not rely solely on a pre-defined ontology, but on a combination of such an ontology with user-defined semantic models that represent the user's particular understanding of their data. This combined information is stored in a KG. The process of semantic modeling in ESKAPE annotates datasets and entities within them with concepts from ontologies and creates links between them. Data sources are thus replaced by models describing them, and users no longer need to inspect data, but rather use semantic models

with precisely explained meanings. ESKAPE will serve as a use case for this section, and our contributions will be integrated and demonstrated in this solution.

UNIPedia, introduced by Kalender et al. [MJS10] and extended in [KD12], is a semantic content tagging and search platform enhanced with management functionality. The authors criticize user-generated tags for their ambiguity and lack of contextual relevance, often describing topics tangential to the content. Their solution integrates various knowledge bases, including *WordNet*, *Wikipedia*, and *OpenCyc* [Cyc01] (although OpenCyc was discontinued in early 2017), to automatically apply semantic tags to documents. While their approach is consistent with this section’s goal of leveraging multiple knowledge bases, their fully automated method, which employs heuristics such as eliminating blacklisted words and searching for similar terms when concepts are not found, distinguishes their work. Although experiments demonstrate high mapping accuracy, the lack of publicly available datasets prevents the reproducibility of results. While fully automated approaches require less effort than semi-manual ones, they may yield less accurate results. Furthermore, Kalender et al. do not maintain their KG, but rely on external knowledge bases, both online and offline.

Bashar et al. propose a framework for automatic personalized ontology learning [BLG16] that incorporates a standard ontology and local information provided by the user. This facilitates labeling with concepts based on user-specific documents complemented by general semantic knowledge. Extracted relations include **isA** (subClassOf), **relatedTo**, and **partOf**, although limited by the attachment of only a single ontology. Their distinction between general and user-specific knowledge is relevant. In contrast to the evolving KG discussed in this section, Bashar et al. assume a sufficiently comprehensive ontology that covers all concepts except those introduced by the document author. These new terms introduce new concepts associated with the current knowledge domain, but lack contextual enrichment.

Milne and Witten in [MW13] propose an open source toolkit that mines Wikipedia to build a knowledge database. Their toolkit uses an XML dump of Wikipedia to build a database and a model that identifies relationships between concepts (i.e., Wikipedia pages), while supporting parallel processing to minimize execution time. A common application is semantic annotation of newspaper documents, where nouns are detected and linked to Wikipedia pages. Notably, concepts retain their original Wikipedia structure rather than being converted to an internal format as in other approaches. Several researchers, such as Zesch, Müller, and Gurevych [ZMG08], Ponzetto and Strube [PS07], and Gabrilovich [Gab10], use approaches similar to that of Milne and Witten. However, they rely on a single knowledge base and perform concept tagging by linking nouns directly to Wikipedia pages, neglecting user-driven KG expansion, concept validation, and user-specific knowledge incorporation.

Knoblock et al. present across several publications [Gup+12; Tah+14; KS15; Ram+15; Kno+12; Tah+16] *KARMA*, an information integration tool that allows users to integrate data from different sources. The semantic modeling is based on a predefined user ontology that provides concepts for data mapping. Their focus is on automating the creation of semantic models and learning techniques, although users retain the ability to modify models after creation. Taheriyani et al. in [Tah+15] contribute to the discovery of semantic relations by leveraging linked data; uncertain matches require manual expert validation. Each semantic model creation requires a specific dataset and ontology, resulting in a knowledge base per use case that increases with different datasets. *KARMA* lacks the ability to incorporate learned knowledge into the user ontology, rendering its KG static with no evolutionary potential. In addition, Pham et al. in [Pha+16] describe automatic semantic labeling using machine learning techniques.

Di Caro and Boella in [DB+16] propose automatic enrichment with common sense knowledge to overcome the limitations of WordNet, in particular its limited relationships such as synonymy (cf. `owl:sameAs` or better `skos:closeMatch`) and hypernymy (cf. `rdfs:subClassOf`). They enrich WordNet synsets with common sense data automatically extracted from ConceptNet 5 [SH+12], drawing from six knowledge sources. YAGO [SKW07] (also see table 6.11) uses a similar strategy, merging WordNet with semantic knowledge from Wikipedia. ConceptNet is derived from the *Open Mind Common Sense* project [Sin02], as illustrated in figure 6.35. Di Caro and Boella enrich WordNet synsets by mapping them to represented nodes, incorporating extensive semantic knowledge. Many studies take similar approaches to ontology enrichment, using diverse sources such as the World Wide Web [Agi+00], machine learning [HTC08; Mon+01], or domain-specific knowledge [BBP04]. Despite different semantic background sources, all methods enrich concepts or synsets with additional semantic relations. Typically, two knowledge bases are used, although additional sources can extract data from multiple knowledge bases. Automatic tagging matches labels to WordNet synsets, Wikipedia pages, or any concept within the respective database. Because the knowledge bases used are highly trusted, no internal validation of semantic relationships is performed. Furthermore, user expertise is not taken into account, as these systems focus on fully automated semantic enrichment rather than concept tagging.

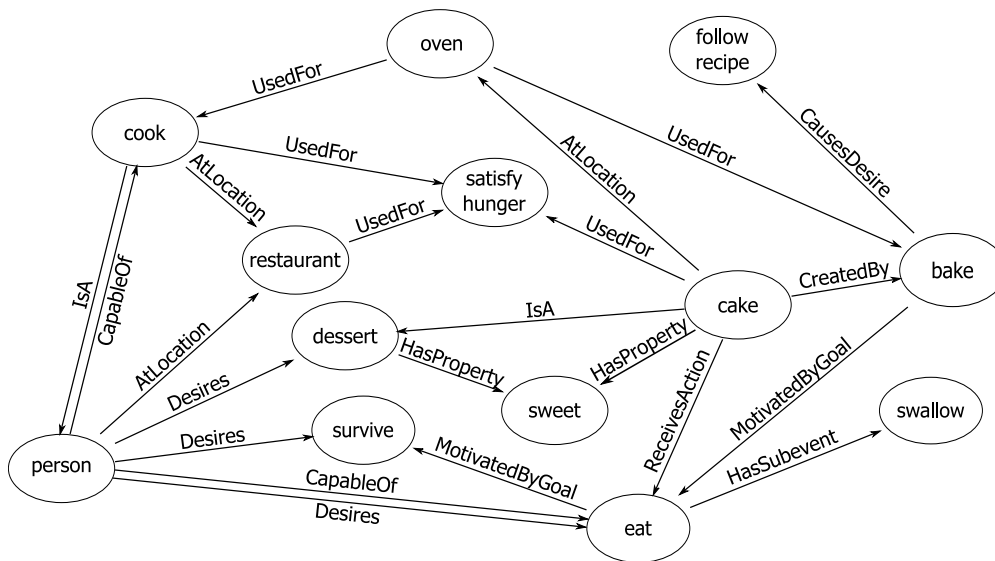


Figure 6.35: A cluster of related concepts in *ConceptNet 5* [SH+12].

Liu, Wu, Jiang, and Tung in [Liu+12] introduced a cross-domain search framework that is applicable to concept tagging similar to the scope of this section. Resources are automatically associated with concepts (pages) from Wikipedia based on their distinct meanings, robust descriptions, links between concepts, and up-to-dateness. An illustrative scenario involves linking images on *Flickr* to videos on *YouTube* or documents on *Delicious*²³, across domains. Within a domain or website, concept similarity helps identify optimal matches. For example, on a site with many photos of Singaporean locations, a new photo tagged “Marina Bay” is more likely to be about Marina Bay in Singapore than anywhere else. However, the static nature of Wikipedia as the underlying knowledge base inhibits evolution during the tagging process; if no concept is found for a resource, no further evolution occurs. Since the framework primarily supports concept tagging, relations are overlooked, making validation

²³<https://del.icio.us>, was online in 2017 but has since been discontinued.

infeasible. Singhal and Srivastava in [SS14] propose a similar approach by first enriching given tags through web search engines and then refining them through a clustering algorithm to remove noise. Unlike mapping resources to Wikipedia pages or ontology concepts, they describe resources semantically using a set of tags.

Du, Rau, Huang, and Chen in [Du+12] aim to improve the quality of tags manually annotated by users, as an alternative to a fully automated process. Their focus is on images on publicly available platforms such as *Flickr*, where users are typically non-experts. They improve tag quality using their *Waking And Sleeping* algorithm, which categorizes each tag as important or unimportant, since non-expert users may produce both low-quality and high-quality tags, necessitating filtering. Figure 6.36 shows an example where user-tagged images are classified into good (waking) and bad (sleeping) tags. The data points (i.e., images) are not inspected by the algorithm, relying solely on user input. One difference to this thesis section is their lack of ontology support and semantic links to concepts.

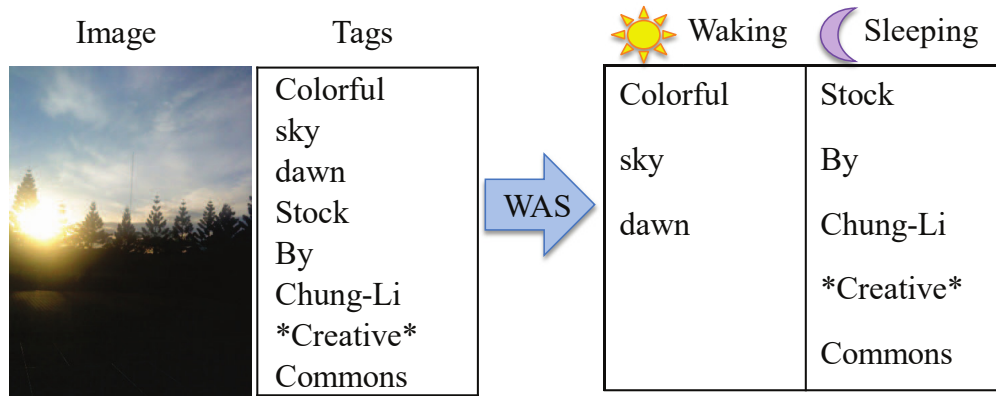


Figure 6.36: The Waking and Sleeping algorithm classifies user tags into good tags (waking) and bad tags (sleeping) [Du+12; Lip17].

Collarana et al. in [Col+16; Col+17] introduce *FuhSen*, a semantic data integration method that constructs a context-rich graph at query time based on a given search term and threshold. This approach gathers information from different data sources (knowledge bases) using wrappers in response to keyword queries. After similarity evaluation, comparison, and result aggregation, an output graph is generated around the search result with connected entities and their respective relations. Although they emphasize entities, their approach could also accommodate ontology concepts and relations. Despite calling the output a **KG**, it differs from the definition of this thesis section in that it merely enriches search results with relevant context, forming a neighborhood graph that effectively represents aggregated results by providing semantic connections to related entities. While querying multiple databases is consistent with this work, *FuhSen* does not maintain a **KG**, nor does it provide user interactions beyond specifying a threshold. Without internally maintaining or recognizing relationships, validation of such concepts is not feasible.

Hong, Park, and Lee in [HPL15] present a tagging technique that addresses problems with user tags in social media, such as ambiguity or the prevalence of tags that are unrelated to content (e.g., 50% of *Flickr* tags do not correspond to the image [Iva+12]). Their approach replaces traditional tagging, in which users directly enter words, with a semi-automatic process that is divided into two parts. Consider a user who uploads a video to a social media platform and provides only a video title. In the first phase, keywords are extracted and page information is parsed from a Wikipedia dump, forming word clusters for each

meaning from disambiguation pages. Users select the word cluster (meaning) that best fits their video, and its top n words are annotated as tags. Notably, entire clusters representing concepts are selected rather than individual tags. This process is similar to this work in that ambiguous words are enriched with additional information for user discrimination, with suggested meanings provided to users. However, Hong et al. rely on Wikipedia for query-based operations and do not maintain a **KG** to learn from user choices, incorporate user knowledge, or add tags not present in Wikipedia.

Folksonomies allow users to freely assign tags to resources, but the lack of structure between these tags limits search capabilities. Laniado, Eynard, and Colombetti in [LEC+07] propose a *Delicious* extension that maps user-annotated keywords to a WordNet synset subgraph. Zablith et al. in [Zab+15] adopt a similar strategy, using WordNet to uncover taxonomical relations between classes, while Tesconi et al. in [Tes+08] map to Wikipedia pages, which are less explicitly structured than WordNet. Laniado et al. derive a hierarchical context from WordNet for tags from Delicious, distinguishing them accurately by considering tags used together. The returned WordNet subtree is pruned to remove unnecessary information near the root synset (concept), such as $X \xrightarrow{isA} \dots \xrightarrow{isA} matter \xrightarrow{isA} physical\ entity \xrightarrow{isA} entity$. Note that *isA* is similar to `rdfs:subClassOf`. Their approach assigns a concept with context to each selected tag, which is consistent with the goals of this thesis section. However, the classification is fully automatic, based on other tags used alongside a tag, and a static taxonomy (WordNet) is used, making learning impossible. Dal Mas follows a similar approach, incorporating a model that evolves with use, but lacks the handling of synonyms or ambiguous terms and relies on a fixed set of user tags, precluding the creation of new tags (concepts).

Maedche, Motik, Silva, and Volz in [Mae+02] present the *Mapping Framework for Distributed Ontologies*, or *MAFRA* for short, which facilitates the concurrent use of distributed ontologies. Since ontologies typically specialize in specific domains, it becomes practical to use multiple ontologies to effectively cover more domains. The authors outline a five-step process for ontology mapping, starting with normalization and proceeding to similarity detection based on lexical features (e.g., WordNet) or attributes and semantic links (see [Doa+02; Sil02]). The crucial third step is to establish a semantic bridge from a concept in one ontology to the closest one in another, followed by the execution of this mapping in the fourth step. Although the authors mention post-processing to check for duplicates or erroneous links, this has not been done in detail. Notable similarities to this work include interactivity and an incremental process that matches concepts one at a time. However, the framework does not adequately detect and handle collisions, and it does not perform actual tagging, as its primary focus is on ontology mapping. While a **KG** is not explicitly used, its presence is implicit when the result is stored in a new ontology.

Hughes, Chrichton, and Mattman in [HCM09] propose the use of a common main ontology alongside domain-specific ontologies to capture both common sense and expert knowledge, following the approach of Uschold and Gruninger [UG04]. They target expert users, integrate multiple ontologies, and acknowledge a growing process, but the specifics of ontology creation or evolution remain unspecified. Semantic modeling is done manually, likely requiring human ontology administrators to make changes to the underlying **KG**. While verification and validation are considered critical in the final phase of ontology engineering, the methodology for achieving this remains undisclosed.

Giunchiglia, Autayeu, and Pane in [GAP12] provide a detailed description of ontology matching algorithms and present an open source tool for identifying semantic relations such as

equivalence, less general, more general, and disjoint between concepts (classes) from two ontologies. The basic algorithm consists of four steps, starting with concept mapping for each label, using external knowledge bases such as WordNet for concept detection. Subsequent steps compute semantic relations between labels and nodes from one ontology to those from the other, using external sources to detect syntactic and semantic relations (e.g., synonyms). While their minimal semantic matching algorithm removes redundancy and provides clearer matching results, it lacks concept support and does not maintain a **KG** to learn from user input. The ontology matching problem is formally defined by Shvaiko and Euzenat in [SE11], who use relations between two ontology classes similar to those used by Giunchiglia.

Kuzey in [Kuz16] maintains a **KG** and continuously enriches it with temporal information. Newspaper articles serve as the knowledge base, which is scraped for events to augment the **KG** or update known facts, such as finding new name aliases or updating time periods. Similarities to this work include support for classes and relations, the ability for the **KG** to evolve based on input, and the potential to add new knowledge bases in the future. However, Kuzey's approach uses automatic tagging without the ability to get advice from the user. It also lacks validation of newly inserted information due to the fixed set of relations that are added with each new event and the lack of additional automatic relation insertion.

Torres et al. in [Tor+11] present a manual tagging system called *Semdrops*, which was further developed by Paiva, Ramos-Cabrer, and Gil-Solla in [PRG12]. Their system, implemented as a Firefox plugin²⁴, allows users to tag web resources in three categories: category (including hierarchies), property (relation), and attribute (e.g., height of a mountain). The tagging process allows users to enter either a category, relationships, or classes. Users are fully trusted, so no inconsistencies or conflicts are detected or handled.

Xiao et al. use DBpedia in [Xia+16] to supervise a learning process of a domain ontology from unstructured text, instead of defining the structure of the ontology as a post-processing step. They rely solely on a single knowledge base (DBpedia) to verify relations through supervising methods. There is no tagging of the instances involved, and users are not involved, resulting in a lack of leverage of user knowledge. The ontology is constantly evolving as more unstructured text is used as input for classes and relations, both of which are evaluated using an appropriate weighting scheme.

The related works discussed above represent promising approaches to various aspects of ontology and **KG** evolution in the context of semantic annotation. However, they are not without limitations. Some of the common limitations observed across these related works include:

1. Lack of user involvement: Many approaches do not actively engage users in the ontology creation process, leading to potential gaps in leveraging user knowledge and preferences.
2. Limited validation mechanisms: Some methods lack robust validation mechanisms for newly inserted information or relationships, which may result in inaccuracies or inconsistencies within the **KG**.
3. Dependency on single knowledge bases: Several approaches rely heavily on a single knowledge base for ontology verification or relation validation, which may restrict the diversity and accuracy of information incorporated into the **KG**.
4. Absence of dynamic evolution: While some methodologies allow for the continuous

²⁴<http://sourceforge.net/projects/semdrops/>, last update in 2013 but has since been discontinued.

evolution of ontologies based on new data inputs, others lack mechanisms for dynamic adaptation and refinement over time.

5. Inadequate handling of conflicts or inconsistencies: Many approaches do not effectively address conflicts or inconsistencies that may arise during the ontology creation or **KG** development process, potentially leading to data integrity issues.

Recognizing these limitations is crucial for understanding the state of the art in semantic annotation in connection with evolving ontologies and **KGs**. This is taken into account in our approach presented below.

6.6.2 Approach

Figure 6.37 depicts a structured workflow that illustrates the iterative process of evolving a **KG** in response to user input during semantic annotation, as detailed in [PLM19b]. In this figure, steps that require user interaction are highlighted in gray. The process begins with a user entering a label that identifies a concept to be used in semantic annotation. The system then evaluates whether a matching concept already exists in the **KG**; if so, it is seamlessly used. In cases where no matching concept is found, the system uses external knowledge bases to search for potential matches, as discussed in section 6.5. These are integrated into the local **KG** as an evolutionary step. If these searches are unsuccessful, users are prompted to create new concepts, which are then integrated into the evolving structure of the **KG**, ensuring its continuous enrichment and refinement in alignment with user needs.

Of these three actions, two are basically covered by the state of the art in Semantic Web technologies. That is, the action of using an existing concept from a **KG** and the action of creating a new concept in a **KG** (see, for example, section 6.3.1). The main challenge that remains for research in this context is to propose a methodology to integrate external knowledge into a **KG** that satisfies a trade-off between information overload and lack of semantic context. While simply importing an ontology class without linking it to any concepts in the **KG** would withhold contextual information, importing too many classes and properties could lead to information overload. This is the main goal of our approach in this section, which we address by proposing an approach that enables users to choose precisely which external knowledge is relevant to their local domain and should therefore be integrated into the **KG**. See figure 6.38 for an example of integrating a concept from the **IDS** information model into a local **KG**. We improve the connectivity of the **KG** by importing valuable context of a class or property. Since knowledge bases such as ontologies can be very large (see the example excerpt from the **IDS** information model on the right), these contexts must be selected carefully. Our approach treats users as experts in their domains and allows them to specify exactly what to import. Two example options on the right are highlighted with a solid and a dotted red line, which differ in the extent of additional context to be imported. In addition, subsequent validation methods are needed to avoid conflicts or inconsistencies between statements as the **KG** evolves.

6.6.3 Results

This section presents our results for evolving semantic annotations, and demonstrates them with implementations and evaluations in the *ESKAPE* platform (see sections 6.5.1 and 6.5.4 and table 6.16), which was later renamed to *PLASMA*. To build semantic models, *ESKAPE*

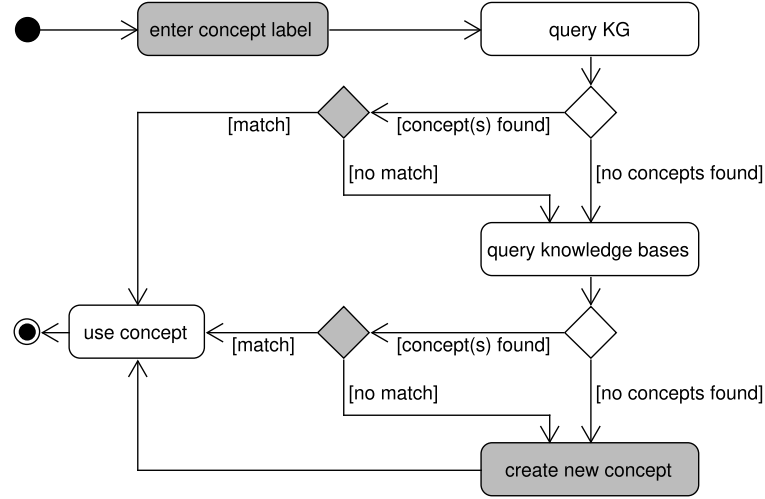


Figure 6.37: Workflow of our approach to continuously evolve a **KG** based on user needs during semantic annotation [PLM19a; Lip17].

provides users with a graphical interface [Pom+18b] where users can select concepts and relationships from ESKAPE’s underlying **KG**, encouraging them to build decisions on this shared terminology.

We mainly present two results, namely a methodology to continuously monitor the growing **KG** for inconsistencies and a user wizard for integrating new concepts from external knowledge bases.

Continuous **KG Monitoring** We propose a two-part approach for monitoring and ensuring the consistency of the continuously evolving **KG**.

The first part of the approach uses strict definitions as well as insights learned from user choices to improve the semantic consistency of the **KG**. We are extending ESKAPE’s **KG** to include logical axioms for relations, which can then be automatically validated during future evolution of the **KG**. The following possible properties for relations E in graphs $G = (V, E)$ can either be imported from external knowledge bases or explicitly stated by users [Lip17]:

- Reflexive: $\forall x \in V : xEx$
- Irreflexive: $\forall x \in V : \neg xEx$
- Symmetric: $\forall x, y \in V : xEy \Rightarrow yEx$
- Antisymmetric: $\forall x, y \in V : xEy \wedge yEx \Rightarrow x = y$
- Asymmetric: $\forall x, y \in V : xEy \Rightarrow \neg yEx$ (irreflexive and antisymmetric)
- Transitive: $\forall x, y, z \in V : xEy \wedge yEz \Rightarrow xEz$

These are validated whenever the respective relations are used in semantic models and during periodic checks of the **KG**, e.g. once a day. For example, if users use an irreflexive relation in their semantic model but add a self-loop, they will be prompted to check for conflicts via a pop-up in the user interface. This could mean adding, removing, or changing relations, or using different ones.

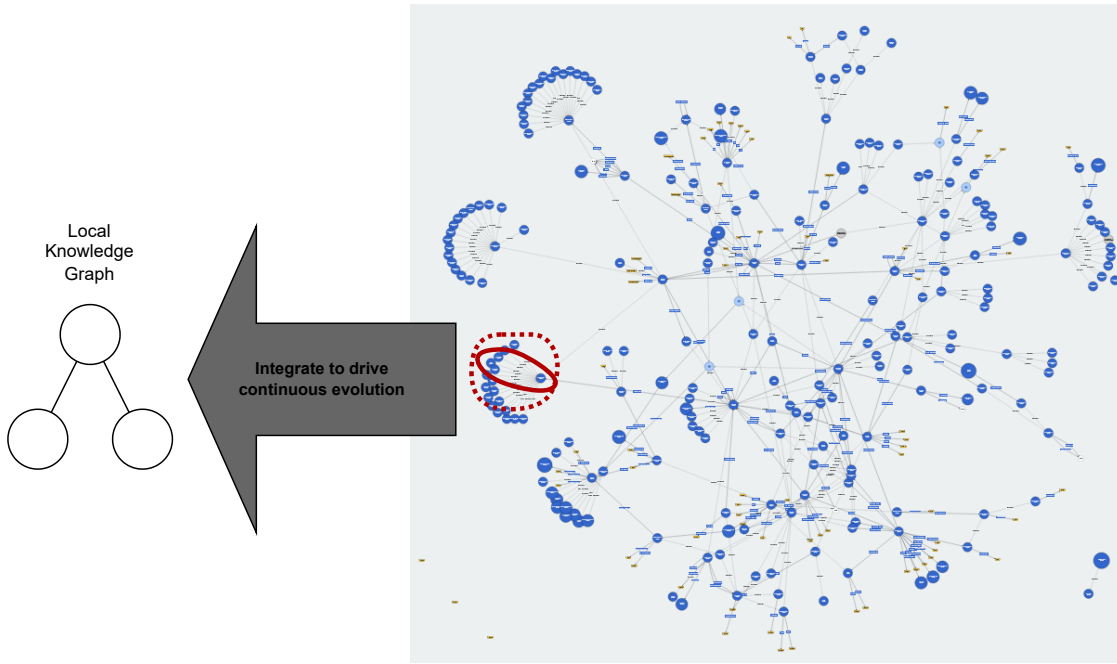


Figure 6.38: Our approach focuses on continuously integrating information from external knowledge bases into a local **KG** during semantic annotation.

The second part of the approach targets disagreements between semantic models from different users. We propose a **KG**, which learns insights from finished semantic models [PLM19b]. Like the human learning process, it assimilates and monitors knowledge provided by users via semantic models, and subsequently learns insights that then expand **KG**. We realize this feature by adding local weights to each relation, and updating the weight of a relation whenever its adjacent concepts are used in semantic models. Thus, we enable a threshold learning process in order to strengthen or weaken semantic relations depending on their usage. The weights allow us to distinguish between frequently used knowledge, i.e. common knowledge agreed upon by multiple users, and specialized knowledge or inconsistencies that are valid only in a small number of cases. As an example demonstration, new knowledge could be added to the ontology of the **KG** after being specified in at least three semantic models provided by different users, and suggested to other users if the majority (50%) of the semantic models used it similarly.

User Wizard to Import Knowledge We in the following extended this user interface with a user wizard to provide functionalities for addressing the requirements and approach mentioned above in figures 6.37 and 6.38. These features include to preview concepts with their valuable context and to avoid duplication and ambiguity by guiding users. When the user imports knowledge from an external knowledge base, we follow a semi-supervised evolution strategy in which the user can decide what additional knowledge from the external knowledge base they want to import.

The workflow of our proposed user wizard is shown in figure 6.37 and starts with the user entering a search term. The wizard first searches the local **KG** for a matching concept. For example, when the input label is *hotspot* and the **KG** already contains the concept *access point*, the wizard would suggest *access point* because it is a synonym for *hotspot*. The wizard provides a rich preview for each queried concept, along with contextual information, as shown

in figure 6.39 on the left. It displays a concept not only with its name and description, but also with additional features such as synonym labels and a neighborhood, which can be built by including one-hop neighbors of certain specified relationships such as `rdfs:subClassOf` or `partOf`. These links to other concepts can improve expressiveness and thus understanding. If a concept is not present in the local KG, it must be added. An approach for recommending and using concepts from external knowledge bases was introduced in section 6.5.4. In this section, we propose an extension to that approach that not only adds single concepts from external knowledge bases, but also includes their context to improve their expressiveness. See figure 6.39 for an illustration of (a) the concept visualization generated by our proposed user wizard [PLM19a] from the knowledge base *BabelNet*. The concept is placed in the center (a), with its main label and related concepts, including their relationships, arranged around it. Next to it, the wizard lists synonym labels and a description. In a final step (b), the user selects which additional contextual information to import into the local KG.

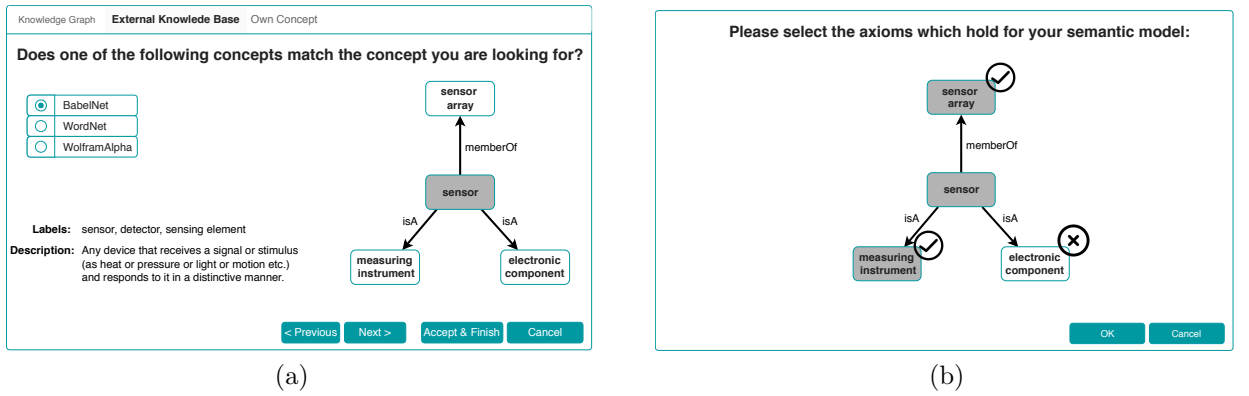


Figure 6.39: Screenshot of a sample concept generated by our proposed user wizard from a knowledge base [Lip17].

This wizard is a solution to the step of importing concepts from external knowledge bases shown in the approach in figure 6.37. The other two paths, namely using a concept from the local KG or creating a new one, are taken care of by the state of the art (see section 6.6.2). Our proposed approach thereby covers all workflow steps from figure 6.37. It also meets the requirements of not only importing single concepts from external knowledge bases but also valuable contextual information (see figure 6.38) by enabling users to include such in the import (see figure 6.39 on the right).

In summary, we have presented a solution for continuously monitoring an evolving KG and a user wizard that enables the user to import concept together with valuable contextual information from external sources to evolve a local KG during semantic annotation.

6.6.4 Evaluation

In this section, we evaluate our proposed approach, focusing on how effectively our user wizard improves the quality of semantic models and, consequently, the growing KG. To do this, we conducted a user study in which participants used our user wizard to create semantic models. Our goal was to determine whether the wizard improves the quality of semantic models compared to manual creation.

We initiated a user study in which participants created two semantic models for each dataset: one using our wizard and one without, allowing for an A/B testing approach for comparison. We describe our participant recruitment, setup, dataset selection, and quality definition for semantic models tailored to our evaluation, and then present the evaluation results.

For the evaluation, we recruited seven individuals from an interdisciplinary research institute, spanning humanities, engineering, and computer science, with varying levels of experience in semantic modeling. Three participants had prior familiarity with semantic modeling, while four were inexperienced, ensuring diverse perspectives and approaches to semantic modeling.

To assess the quality of the semantic models, we used a qualitative evaluation method that involved pairwise comparison of the participants' semantic models, a time-consuming process that required a small group of participants.

All participants followed a standardized procedure in a dedicated room equipped with a laptop running ESKAPE, with external knowledge bases limited to *BabelNet* to avoid potential inconsistencies with the merge and rank algorithms (see [section 6.5.4](#)). An evaluation expert accompanied the participants to observe their actions, the challenges they faced, and the solutions they applied, as well as to address non-data-related questions.

Similarly to the evaluation presented in [section 6.5.5](#), each participant received an introduction to the functionalities of the used tools before proceeding with the evaluation using the same datasets as detailed in [table 6.12](#), accompanied by comprehensive dataset descriptions.

Participants sequentially created semantic models for each dataset with and without the assistance of our user wizard. The wizard facilitated the addition of selected knowledge to the **KG**, enabling precise growth of semantic information based on user preferences. Snapshots of the **KG** and the generated semantic models were captured after each step for subsequent analysis.

The evaluation aimed to validate our hypothesis regarding the objectivity and consistency of semantic models generated by our user wizard and its impact on improving the quality of the growing **KG**.

Analysis and Discussion

Various metrics for evaluating the quality of semantic models have been proposed in the field of semantic modeling. According to Paulheim, there is no universally accepted gold standard for evaluating a semantic model [Pau17]. Metrics that rely on precision and recall face limitations in semantic modeling, since determining recall requires knowing the number of positive results that should have been included in the semantic model. This task is often difficult or even impossible to perform, given the innumerable potential candidates, without knowing which additional concepts needed to satisfactorily describe a given concept.

Therefore, in this evaluation, we consider a user's semantic model to be effective if it shows significant overlap with other semantic models created for the same dataset. Our goal is to promote objective and coherent semantic models to facilitate stable **KG** growth, incorporating relevant expert knowledge while excluding personal perspectives unrelated to the content. Optimal **KG** evolution occurs through the addition of semantic models that share commonalities and foster robust interconnectivity, while overly personalized models may lead to isolated subgraphs.

As a measure of similarity between semantic models, we use the Jaccard index [Jac08] for analysis. This metric quantifies the number of shared concepts between two semantic models. In this scope, we define two concepts, c_1 and c_2 as identical if they have exactly the same name. In addition, we introduce refined criteria for identifying identical concepts that are informed by prior observations in semantic modeling. These criteria include (i) capitalization (e.g., `flight` vs. `Flight`) (ii) word spacing (e.g., `flight number` vs. `flightnumber`), (iii) camel case as a combination of the former (e.g., `FlightNumber` vs. `flight number`), and (iv) underscores (e.g., `flight_number` vs. `flight number`). Consequently, we identify two concepts from different participants as equivalent if they satisfy one of the above rules. See the homogenization step described in section 6.5.4 for more details.

The Jaccard index is defined as the ratio of the number of elements shared by two sets to the number of elements in their union. The formula for computing the Jaccard index of two sets is $J(A, B) = \frac{A \cap B}{A \cup B}$. We refer to A and B as the concepts present in two semantic models, respectively. The Jaccard index is equal to zero if two semantic models do not share any concepts, and equal to one if they share all concepts. For this evaluation, we compute the average Jaccard index (similarity) between each semantic model and all others within the same setting (same dataset and same un/assisted state). For one dataset in the unassisted setting, we compute the Jaccard index between the semantic model of participant 1 and those of others, followed by that of participant 2, and so on.

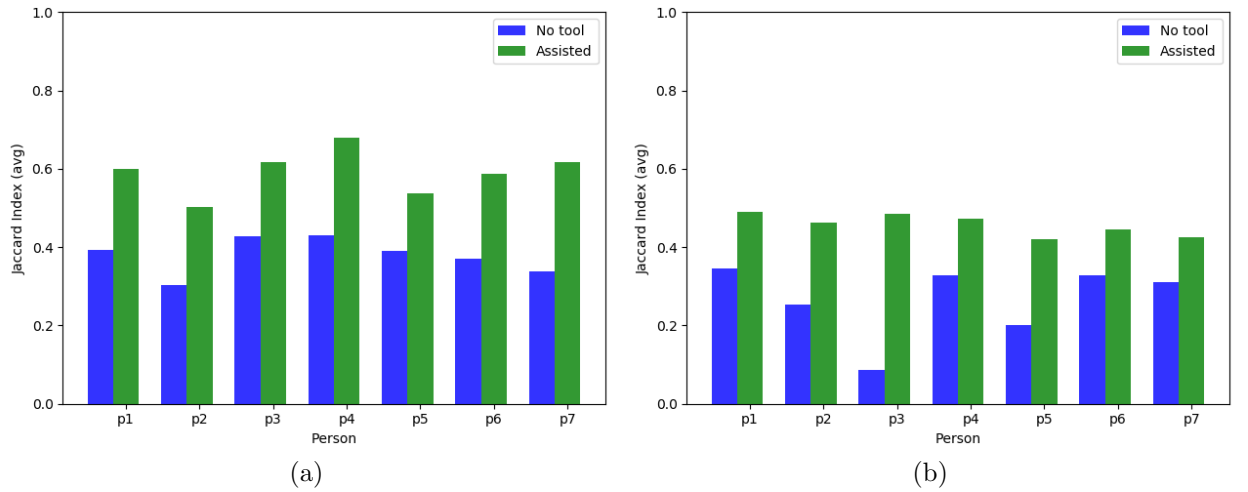


Figure 6.40: The average Jaccard index of each attendee’s semantic model with those of others for the (a) *crime* and (b) *parking* datasets introduced in table 6.12, comparing our assisted wizard with a no-tool baseline [PLM19a; Lip17].

Figure 6.40 shows the average Jaccard indexes for the (a) *crimes* and (b) *parking* datasets, comparing the no-tool setup with the assisted run. The average Jaccard index for our proposed user wizard is consistently higher. The results indicate that the number of shared concepts between different semantic models increases when participants use our user wizard. In other words, the semantic models become more similar, contributing to a common domain understanding for participants in dataspace. This means that semantic models become more consistent and objective when users have access to additional knowledge during model creation. Furthermore, since the use of our user wizard reduces the diversity of concepts used, the stability of the KG’s ontology will subsequently increase, fostering agreement within the domain community on the shared vocabulary in this KG. The increased similarity of

semantic models supports a common domain understanding, for example, among participants in dataspace.

Additional (Qualitative) Observations

In addition to our primary findings, we gained additional insights during this evaluation process. As participants used our proposed wizard to identify matching concepts for annotation, and when they encountered situations where neither the [KG](#) nor external knowledge bases offered a concept that matched their understanding, several users went back several times to search for alternative terms. This observation suggests a hypothesis: over time, the community may converge on a common vocabulary, fostering the development of more consistent and objective semantic models.

When introducing their own concepts – where they differed from the suggested terms – users typically provided names, but often did not provide adequate descriptions. For example, descriptions such as *airline: “airline of a flight”* were common. Improving the specificity of descriptions would facilitate the reusability of concepts among users in subsequent instances. In addition, participants showed variation in the directionality of relations when connecting multiple properties to a central node representing the core of a dataset. While some semantic models represented relations from a general concept to its properties using descriptors such as *has*, others took the opposite approach with relations such as *partOf*.

Five participants were confident that they would be able to build much more complex semantic models, while two participants believed that the complexity limit had already been reached.

In conjunction with the above findings, we investigated whether people with more experience in semantic modeling tend to create more complex semantic models. This investigation encompasses two aspects: first, experienced users invest time in selecting concepts that are precisely aligned rather than rushing to complete the semantic model; second, semantic modeling experts prioritize reusing existing concepts from the [KG](#) or external ontologies over manually introducing new ones whenever possible. This tendency likely stems from an understanding of the benefits associated with shared concepts, as opposed to concepts derived solely from individual intuition. We evaluated this hypothesis using two metrics during our user studies: precise time measurements for each step in the user wizard, and tracking the selection of suggested concepts by users.

[Figure 6.41](#) illustrates the time spent by semantic expert participants compared to those with no expertise in semantic modeling, referred to as nonspecialists. Significant differences are observed in determining the appropriate concept label (page 1 of our wizard), where nonspecialists typically made decisions within about 7 seconds, with no decision exceeding 27 seconds. Conversely, experts took an average of 14 seconds, with the longest decision taking up to 125 seconds, which exceeds the graph limits for this figure due to readability constraints. There is no significant difference in the selection of concepts from the local [KG](#) (page 2 of our wizard), since all experiments started with an empty [KG](#), rendering this page of the wizard ineffective. The selection of suggested concepts from external knowledge bases (page 3 of our wizard) again resulted in notable differences between the two user groups. The nonspecialists took an average of 7 seconds and a maximum of 32 seconds to select the best matching concepts from the provided suggestions, with one user completing the selection in just 0.8 seconds. In contrast, experts took an average of 15.9 seconds, with the longest selection taking up to 64 seconds. Nonspecialists tend to create new concepts (page

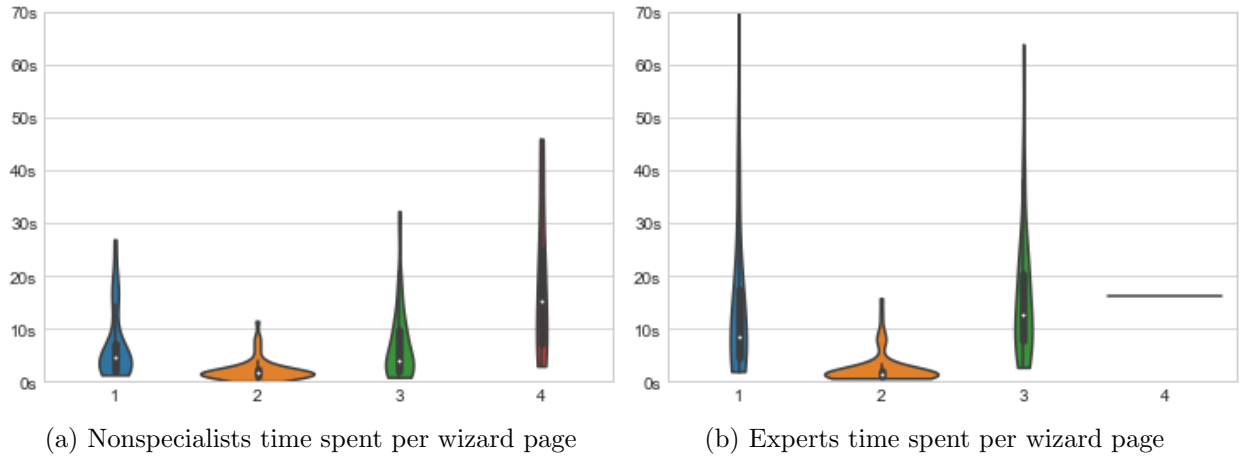


Figure 6.41: Violin plots [Lip17] of time spent for (a) nonspecialists and (b) experts.

4), while experts tend to reuse existing concepts. To generate a new concept, nonspecialists took between 3 and 46 seconds, with an average of about 18 seconds, while experts generated only one new concept within 16 seconds.

In addition to the quantitative measure of time spent on the wizard pages, we considered a second measure. In this qualitative assessment, we look at how users formulate their search terms when searching for matching concepts. A subset of the results is presented in Table 6.17, where nonspecialists typically adopt the provided label from the dataset as their search term. However, this approach yields suboptimal results in terms of expressiveness, since the search terms do not adequately represent the concepts they are trying to find. For example, for the labels in table 6.17, the search term “X” returned nine concepts from external sources, none of which matched, so a new concept had to be created manually. Similarly, the remaining nonspecialist search terms couldn’t be found in any external knowledge base, necessitating manual creation. Conversely, expert users replaced given labels with their corresponding concepts, such as `color` instead of `cap color`, enabling them to discover and reuse existing concepts from external knowledge bases.

Our evaluation has two limitations. First, we simulated the situation where a user has a dataset at hand and wants to annotate it in order to publish it to a dataspace. Unlike most real-world use cases, our participants were not familiar with the dataset and its meaning. We provided metadata and additional information along with the datasets to address this, but at the cost of additional time required by the participants to properly understand the meaning of each dataset. Second, we focused on determining the average Jaccard index for semantic models and compare the scenarios supported by our user wizard with those where no wizard was used. This evaluation requires resetting the KG between test runs in order to

Table 6.17: Qualitative evaluation [Lip17] of search terms entered for given labels. Nonspecialists tend to adopt the given label, while experts formulate a more detailed search term.

Label	X	Post ID	Cap Color	Quantity_Units	Flight status
Nonspecialist	X	Post ID	Cap Color	Quantity_Units	Flight status
Expert	Coordinate	Identifier	Color	Unit	Status

make them comparable. As a result, we were not able to monitor the overall growth of the [KG](#) over a long period of time, with different users working on the same [KG](#).

In summary, our quantitative and qualitative evaluations demonstrate that our approach for evolving semantic annotations meets the requirements of dataspace participants as participants add new content over time. The proposed solutions, such as the user wizard, make semantic models more similar and thus more interoperable, even when created by different dataspace participants. This leads to more common domain languages and understanding in dataspace participants. Our comparison between semantic modeling experts and nonspecialists revealed the following insights. Nonspecialists typically show less care in performing tasks within the semantic modeling process, resulting in less accurate and objective semantic models. They often do not hit results from external knowledge bases, as they tend to simply use the given dataset column labels as search terms. In contrast, experts carefully formulate queries to find the correct concepts, and spend considerably more time selecting the appropriate concept from suggestions in external knowledge bases (see [figure 6.41](#)). Despite our initial goal of enabling domain experts without experience with semantic models to independently describe data sets, our evaluation confirmed that experienced users construct more complex models. Consequently, there is an urgent need for assistance solutions to improve support for nonspecialists in this area, which our proposed user wizard adequately addresses.

Contributions

In conclusion, this section has successfully addressed [requirement Domain-R5](#) by proposing a user wizard and a monitoring mechanism to continuously evolve a [KG](#). This allows semantic annotations and common vocabularies to not only represent a snapshot in time, but to continuously improve in an ever-changing and complex world. With our contributions, the semantic description of data and other resources in dataspace participants will become more similar, interoperable, and up-to-date. These contributions include:

- **Manage Domain Knowledge Over Time:** A knowledge management system consisting of a local knowledge graph with user-created semantic models and an ontology that can evolve over time. This resolves [requirement Domain-R5](#) by allowing external knowledge to be imported or new knowledge to be created by dataspace participants.
- **Context-aware Knowledge Evolution:** A solution with a user wizard that not only imports single concepts but also adds valuable contextual information and links to other concepts to the local [KG](#). As a side effect, the recommendations provided by the user wizard make semantic models more common and homogeneous, promoting common domain languages.
- **Continuous Quality Assurance:** A two-part approach ensures the consistency of the continuously evolving [KG](#) by validating defined rules for relations (e.g., symmetry) and learning insights from semantic models created by different users.

These contributions have advanced the field of semantic annotation in dataspace participants, making semantic models more similar and to adapt changes at any time. The positive evaluation of the proposed system and its user wizard proves its effectiveness and practicality. Through a successful demonstration in a graphical user interface application, the contributions enable a continuous evolution of shared semantic annotation for a wide range of users, thus contributing to a better common domain understanding in dataspace participants.

6.7 Enhanced Validation Techniques

In this section, we address [requirement Domain-R6](#) and develop validation approaches to facilitate common information design in the decentralized environment of dataspace. In dataspace, both the descriptive vocabularies and the descriptions of the resources themselves are created and exchanged between participants in a decentralized manner. An integrated use of such data or services in applications requires reliable guarantees for them, not only on the data layer, but also on the domain layer (see [figure 6.1](#)). Validation techniques can ensure conformance to certain specifications at different stages of the data lifecycle (e.g., assure, submit, integrate in said lifecycle figure) or between interfaces of dataspace participants or services (e.g., APIs such as [section 5.4](#)). Addressing [requirement Domain-R6](#) means proposing good but convenient methods for validation in dataspace. *Good* in this context refers to precise and scalable solutions, including state-of-the-art solutions such as knowledge inference, while *convenient* refers to ease of use and inclusion of a wide range of users of a dataspace, including non-experts.

More specifically, Semantic Web technologies such as [SHACL](#) are used in dataspace to realize the required validation techniques. The correct implementation of so-called *validation shapes*, which define the expected form of data, can be a non-trivial task for dataspace users who are not Semantic Web experts. This is especially true when more complex constraints are required that are not natively supported by validation languages and require the use of more advanced features such as [SHACL-SPARQL](#) [KK17]. We address the above need for *good* and *convenient* methods by simplifying the design of complex constraints and making validation steps that require inference more scalable. Our key contributions include:

- **Shortcuts for Commonly Used Constraints:** We provided shortcuts in the form of template-based properties for commonly used constraints that are not yet natively supported by validation languages, simplifying and standardizing their application for users.
- **A Scaling Inference Solution for Validation:** Our modification of the current common practice of [RDF](#) validation with [SHACL](#) applies inference only once to the shape graph instead of on each instance.

This section first studies the state of the art in languages and techniques for validation, and then presents our approach, results, and evaluation to address both of the above aspects of [requirement Domain-R6](#). This section first studies the state of the art in languages and techniques for validation, and then presents our approach, results, and evaluation to address both of the above aspects of [requirement Domain-R6](#). Note that these exact contents were also submitted for publication [Moo+24] during the completion of this thesis.

6.7.1 Related Work

The state of the art in this area includes validation languages that allow dataspace participants to design and evaluate constraints to ensure the conformance of resources in dataspace. These resources, ranging from data to services to dataspace assets such as participant descriptions, can be validated against arbitrary specifications, including local agreements or global standards. The state of the art in validation solutions and techniques includes software applications as well as common practices for their application.

Validation Languages

This section extends the background on validation languages (see [section 2.2.1](#)) with a systematic evaluation of these based on their support for commonly required constraints. Several languages have been developed to provide extensive validation capabilities for RDF data. In 2017, Dominik Tomaszuk conducted a survey that offers an overview and comparison of the most relevant languages available for RDF validation [Tom17]. In his study, Tomaszuk identified five key validation tools: ShEx [PLS14], SHACL [KK17], ReSH [Rym14], DSP [Nil08], and SPIN [KHI11]. He compared these tools, along with the built-in validation features of OWL [Bao+12], in terms of their ability to express 17 commonly used constraints. Given that SHACL was significantly influenced by SPIN and is considered its legitimate successor [Knu17a], the references to SHACL in our following comparison implicitly also include SPIN. Table 6.18 presents a summary of our comparison, which uses the following metrics. Expressiveness is determined by how many of the 17 commonly used constraints [Tom17] can be expressed, resulting in a score ranging from zero to 17. In addition, we evaluate whether the validation languages are standardized (e.g., W3C standards), they are used in the dataspace initiatives Gaia-X or IDS, and whether they support inference during validation. These are boolean measures.

Table 6.18: Comparison of validation languages based on four requirements.

Validation Language	Supported Constraints	Web Standard	Used in IDS and Gaia-X	Inference Support
SHACL [KK17]	16	✓	✓	✓
ShEx [PLS14]	16	✗	✗	✗
ReSH [Rym14]	9	✗	✗	✗
DSP [Nil08]	10	✗	✗	✗
OWL [Bao+12]	13	✓	✓	✓

SHACL [KK17] is a constraint language for RDF. It can express 16 of the 17 commonly used constraints. It is a W3C recommendation and is used in dataspace initiatives such as Gaia-X and the IDS. Existing SHACL validators provide support for RDFS inference. ShEx [PLS14] is a language for describing RDF graph structures that can specify conditions that RDF data graphs must satisfy to be considered *conformant*. It can express 16 of the 17 commonly used constraints. Any inference must be done separately on the RDF graph, the ShEx processor itself does not interact with any inference mechanism [Gay+18]. ReSH [Rym14] is a high-level RDF vocabulary for specifying the form of RDF resources. It is only capable of expressing 9 of 17 commonly used constraints. There is no additional inference support beyond the general ability to perform inference directly on the RDF graph. DSP [Nil08] is used to define structural constraints on data. It is only capable of expressing 10 of the 17 commonly used constraints. There is no additional inference support beyond the general ability to perform inference directly on the RDF graph. OWL [Bao+12] is a language for describing RDF graph structures and is used in dataspace initiatives such as Gaia-X and the IDS. It is part of the W3C's Semantic Web technology stack. While it has built-in validation capabilities, it can only express 13 of 17 commonly used constraints. It is possible to apply inference mechanisms to graphs described using OWL.

Based on the results shown in Table 6.18, we will focus on SHACL for the rest of this section because it best meets our requirements.

Solutions and Techniques for Validation

Since we have identified [SHACL](#) as the most appropriate validation language for our requirements, we will focus in this section on solutions that implement SHACL. *Apache Jena SHACL* [The11b] implements both [SHACL](#) core and [SHACL-SPARQL](#) constraints and provides a reader and writer for a compact [SHACL](#) syntax. The *TopBraid SHACL API* [Top15] is an open source implementation of [SHACL](#) based on Apache Jena that performs [SHACL](#) constraint checking and rule inference. There is also a data governance tool that uses this API, the *TopBraid EDG* [Top17]. A GitHub issue opened in February 2021 [Pal21] shows that the topic of inference for [RDF](#) validation can be further improved. The issue raises the question of subclass inference through [SHACL](#). The answers suggest that today the only way to perform this kind of inference is to import the whole ontology (or even multiple ontologies) into the data graph, which is probably not a high-performance solution. In addition, there is no easily accessible documentation that shows what kind of inference is being performed.

Web-based validators are available for simple validation tasks. For example, the SHACL Playground [Knu17b] is implemented by TopQuadrant, who are also responsible for the TopBraid EDG mentioned above. There is an updated implementation of the SHACL Playground that can be found under the name *Zazuko SHACL Playground* [Zaz21]. Another web-based validator is provided as part of the European Commission’s *DG DIGIT Interoperability Test Bed* [DG 20b]. The interface provides an easy way to upload a shape and data graph to perform the validation. In addition to the data validator, they also offer a shape validator [DG 20a], which checks if the provided [SHACL](#) shape conforms to certain rules, namely the core [W3C](#) syntax rules, the extended [W3C](#) syntax rules, and the extended [W3C](#) syntax rules and best practices. These web-based validators have in common that they are designed for simple validation tasks and do not support more advanced tasks. For example, it is not possible to import ontologies into the data graph for inference or use advanced features such as [SHACL-SPARQL](#) for the SHACL Playground.

6.7.2 Approach

Our approach to enhancing validation techniques for dataspace consists of two steps: The introduction of shortcut templates and the optimization of inference methods during validation. Our approach finally includes a combination of these two steps into one solution.

Introducing Shortcut Templates for Constraints

We commence by addressing the challenge of dealing with commonly used constraints that are not natively supported by validation languages (see [figure 6.42](#)). The current state-of-the-art approach for [SHACL](#) is to manually formulate [SHACL-SPARQL](#) constraints, which can be complex and ambiguous (since one constraint can be modeled using varying [SHACL-SPARQL](#) constructs). To streamline and standardize this process, we propose the introduction of shortcut templates in the form of new properties. Our proposed methodology is illustrated in [figure 6.43](#). The approach entails four stages: (1) the analysis of the feasibility of common yet unsupported constraints, as presented by Hartmann [Har16], (2) the design of templates for them, (3) the aggregation of these templates for ease of use, and (4) the implementation of a demonstrator for them.

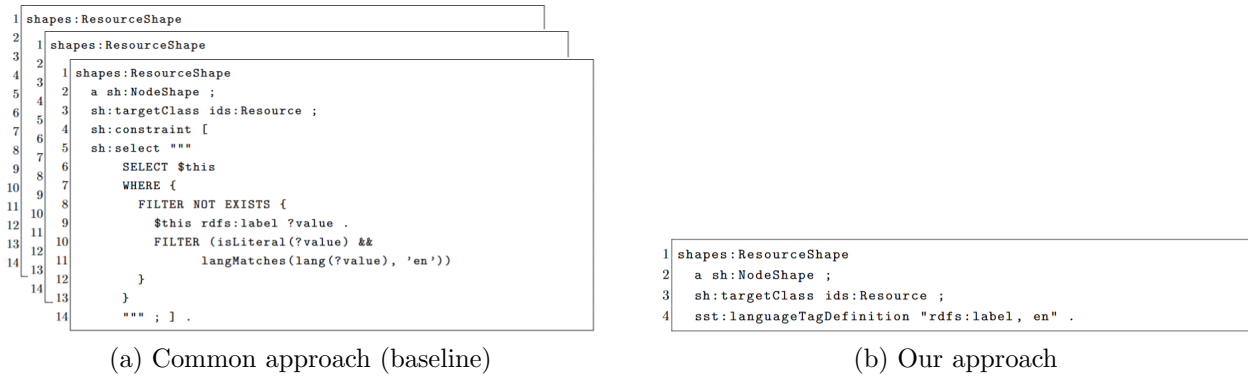


Figure 6.42: The first part of our two-part approach [Moo22; Moo+24] is to introduce shortcuts for commonly used constraints that are not yet natively supported by validation languages.

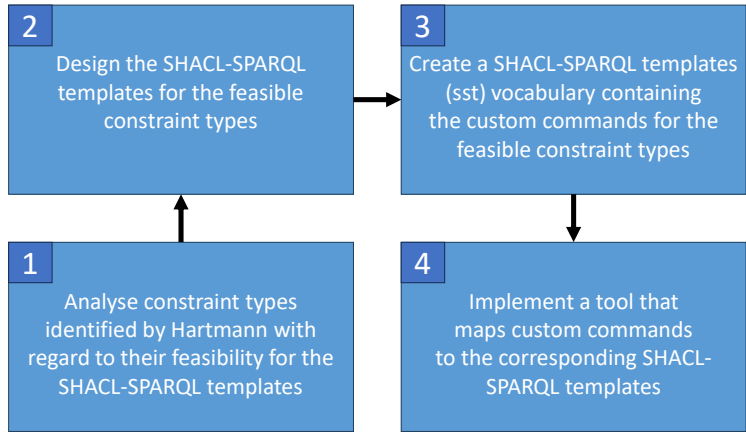


Figure 6.43: The methodology of the first part of our two-part approach [Moo22; Moo+24], which introduces **SHACL-SPARQL** templates as shortcuts for commonly used constraints not natively supported by **SHACL**.

These shortcut templates serve as intuitive bridges between unsupported constraints and native **SHACL** validators, significantly reducing the cognitive overhead for users. By encapsulating complex constraints into reusable templates, users are enabled to express validation rules more concisely and effectively. Furthermore, the adoption of standardized templates improves interoperability and promotes best practices across different validation scenarios.

Validation Enhancement through Inference Optimization

In the second part of our approach, we address the inefficiencies inherent in current **RDF** validation processes (see figure 6.44). Inference is commonly applied as part of the **RDF** validation with **SHACL**, whereby the entire ontology file needed for inference is appended to each data instance. This significantly increases the size of each instance to be validated. Our proposed solution modifies this practice by applying the inference only once to the shapes graph. This optimization results in a shapes graph of slightly higher complexity, but eliminates the need for expensive inference on each data graph during validation. With this approach, we aim for a more scalable solution for recurrent validation against the same shapes graph.

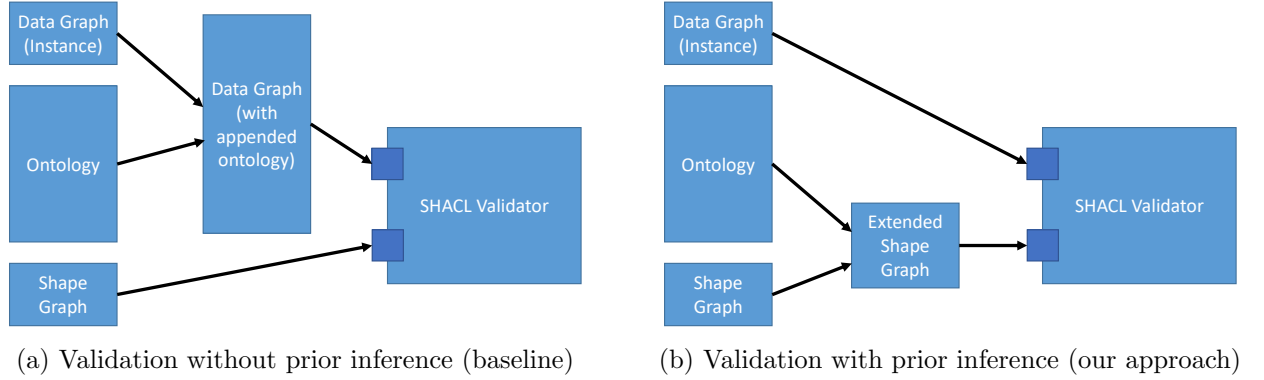


Figure 6.44: The second part of our two-part approach [Moo22; Moo+24] is to modify a current common practice for RDF validation with SHACL.

Our approach not only reduces the computational overhead but also increases the agility of the validation process. By decoupling the inference step from individual data instances, we aim to create a more streamlined and resource-efficient workflow. This optimization could prove particularly advantageous for large-scale dataspace, since they require frequently performed validation tasks.

Combining the Solutions

The final step of our approach is to combine these two solutions (see figure 6.45). The input data is pre-processed by our proposed scaling inference solution, which optimizes the shapes graph for subsequent validation. Subsequently, our constraint template mapping solution outputs native SHACL shape graphs using SHACL-SPARQL. This ensures compatibility with any native SHACL validator supporting SHACL-SPARQL, thereby facilitating seamless validation of data against predefined constraints.

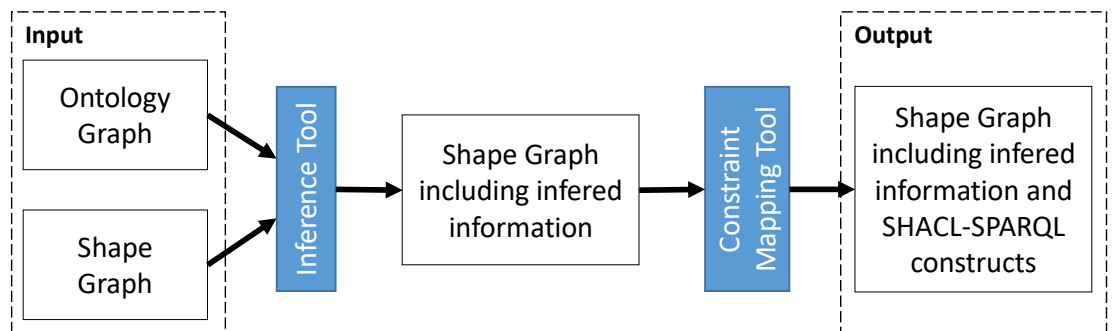


Figure 6.45: The complete solution of our two-step approach [Moo22; Moo+24], including scaling inference (see figure 6.44) and constraint template mapping (see figure 6.42).

6.7.3 Results

This section presents the results and implementation of the previously described approach. Our work enhances the validation process of existing validation tools, such as the command-line tool pySHACL [SC18], by implementing easy-to-use commands for complex constraints and performing an inference step prior to validation. As this is a two-step approach, the results will be presented for each step separately.

Resulting Shortcut Templates for Constraints

The results of this work include 16 constraint templates implementing commonly used constraints [Tom17] using SHACL-SPARQL. This approach allows for the utilization of the built-in SPARQL support of the SHACL validation language. Using these templates, a potential user desiring to validate a data instance is not required to implement a complex SHACL-SPARQL solution. Instead, the user is required to execute a simple command, which is then replaced by the corresponding SHACL-SPARQL template during the mapping process. Listing 6.3 illustrates one of the created constraint templates. This template implements a constraint that a certain language tag (German) must be present for a data property. A valid example instance (line 12) has a German language tag (@de) and an invalid instance (line 13) has only an English language tag (@en).

```

1 :CountryShape a sh:Shape ;
2 sh:scopeClass :Country ;
3 sh:constraint [
4 sh:message "Values of [...] must have a German language tag." ;
5 sh:sparql """
6 SELECT $this ($this AS ?subject) (:germanLabel AS ?predicate)
   (?value AS ?object)
7 WHERE { $this :germanLabel ?value .
8         FILTER(isLiteral(?value) ||
9               !langMatches(lang(?value), "de"))}""" ;].
10
11 :ValidCountry a :Country ; :germanLabel "Deutschland"@de .
12 :InvalidCountry a :Country ; :germanLabel "Germany"@en .

```

Listing 6.3: Example SHACL-SPARQL template to validate that a certain language tag is present for a data property.

Figure 6.46 (a) shows an example of a possible input graph for the constraint mapping tool. This SHACL shapes graph defines a constraint on the cardinality of English language tags on the `rdfs:label` property using one of the custom commands defined in this work. The constraint mapping tool then maps this command to the corresponding SHACL-SPARQL construct as defined in the constraint template we designed. Figure 6.46 (b) shows the result of this mapping process. To achieve the mapping from figure 6.46 (a) to (b), a simple command with two input parameters is sufficient. This figure shows how the proposed solution transforms an introduced shortcut (see line 6 in (a)) into corresponding SHACL-SPARQL expressions (see lines 5 – 17 in (b)), which can be used natively in any SHACL validator.

For a command line execution, the first parameter is the path to the SHACL shapes graph that contains our custom `sst` commands. The second parameter is the output location for

the created [SHACL](#) shapes graph containing the [SHACL-SPARQL](#) templates. A possible command line is:

```
python Mapping.py pathTo/input.shacl.ttl pathTo/output.shacl.ttl
```

This mapping is implemented using a graph-based approach. The graph is searched for triples using one of our custom commands as an edge between two nodes. If such a triple is found, it is replaced by a new triple using the `sh:sparql` command as a predicate and the corresponding [SHACL-SPARQL](#) template as an object. The input file is imported as a graph data structure using the Python package *RDFlib* [Kre02].

```

1 @prefix gax: <http://w3id.org/gaia-x> .
2 @prefix sh: <http://www.w3.org/ns/shacl> .
3 @prefix sst: <http://example.org/sst-vocab> .
4
5 sh:ProviderShape a sh:NodeShape ;
6 sst:languageTagCardinalityMin "rdfs:label,rdfs:comment,en,1";
7 sh:targetClass gax:Provider .

```

(a) Input

```

1 @prefix gax: <http://w3id.org/gaia-x> .
2 @prefix sh: <http://www.w3.org/ns/shacl> .
3
4 sh:ProviderShape a sh:NodeShape ;
5 sh:sparql [ sh:message "Values of the constraint [...]" ;
6 sh:select """
7 SELECT $this
8 WHERE {
9     SELECT (COUNT(?value) as ?count)
10    WHERE {
11        { $this rdfs:label ?value } UNION { $this rdfs:comment
12          ?value }
13        FILTER(isLiteral(?value) && langMatches(lang(?value), '
14          en'))
15    }
16    GROUP BY $this
17 }
18 HAVING (SUM(?count) < 1)
19 """ ] ;
20 sh:targetClass gax:Provider .

```

(b) Output

Figure 6.46: Example execution for a constraint that all labels and comments have at least one English value, using our proposed constraint mapping tool.

Resulting Inference Optimization

Our solution for performing inference as a preprocessing step is implemented as a tool similar to the mapping tool above. Again, we use a graph-based approach consisting of two tasks. The first task is to search the ontology graph for all subclass and subproperty relations. The second task is to search the shapes graph for occurrences of parent classes and properties and add more links to cover all related subclasses and subproperties. For the second task, the tool identifies certain types of links in the shape graph. Figure 6.47 shows a minimal example of what the input and output graphs of the inference tool look like when performing a simple subclass inference of the `sh:targetClass`. This figure shows how our tool updates the SHACL graph using the information that `ex:hasMother` and `ex:hasFather` are subproperties of `hasParent`. This is implemented by using `sh:and` (cf. line 5 in (b)).

```

1  @prefix ex: <http://example.org/> .
2  @prefix sh: <http://www.w3.org/ns/shacl#> .
3
4  ex:PersonShape
5  a sh:NodeShape ;
6  sh:targetClass ex:Person ;
7  sh:property [
8  sh:path [ sh:inversePath ex:hasParent ] ;
9  sh:nodeKind sh:IRI ; ] .

```

(a) Input

```

1  @prefix ex: <http://example.org/> .
2  @prefix sh: <http://www.w3.org/ns/shacl#> .
3
4  ex:PersonShape
5  a sh:NodeShape ;
6  sh:targetClass ex:Person;
7  sh:property [
8  sh:and ( [
9  sh:path [sh:inversePath ex:hasParent] ;
10 sh:nodeKind sh:IRI ;
11 ] [
12 sh:path [sh:inversePath ex:hasMother] ;
13 sh:nodeKind sh:IRI ;
14 ] [
15 sh:path [sh:inversePath ex:hasFather] ;
16 sh:nodeKind sh:IRI ; ] ) ] .

```

(b) Output

Figure 6.47: Example execution of a subproperty inference.

Analogous to the realization of the constraint templates, our transitive inference tool is realized by a command-line tool implemented in Python 3. More specifically, we use the Python library *RDFlib* [Kre02], which allows an implementation of our graph-based approach. To perform the inference step, a simple command-line command with three input parameters is

sufficient. The first is the path to the ontology graph containing additional information, the second is the path to the [SHACL](#) shapes graph, and the third is the output location for the extended [SHACL](#) shapes graph. A possible command line execution is:

```
python mapping.py ontology.ttl shape.ttl newShape.ttl
```

6.7.4 Evaluation

To assess the effectiveness and scalability of our proposed enhanced validation techniques, we perform an evaluation using example instances from Gaia-X. This approach was chosen to demonstrate the feasibility of our approach in real-world application scenarios. The instances, as shown in [figure 6.48](#), represent typical data providers within the Gaia-X ecosystem. They include properties related to data attributes and links to related objects. In these examples, each provider instance contains properties that define the type, name, tax ID, and label. Other properties describe the relationship between the different providers, such as subsidiary or project partner relationships, and whether a provider trusts the other providers. Understanding the details of these instances is important for evaluating the performance of our validation techniques. It should be noted that the names and data of these vendors have been anonymized for this work.

Properties \ Instances	Provider 1	Provider 2	Provider 3	Provider 4
ID	gax:Provider1	gax:Provider2	gax:Provider3	gax:Provider4
rdfs:type	gax:Provider	gax:Provider	gax:Provider	gax:Provider
gax:hasName	Provider 1	Provider 2	Provider 3	Provider 4
gax:hasSalesTaxID	101	102	103	104
ex:hasProjectPartner	gax:Provider2	gax:Provider1		
ex:trusts	gax:Provider1	gax:Provider2	gax:Provider3	gax:Provider4
ex:isSubsidiaryOf			gax:Provider1	gax:Provider1, gax:Provider3
rdfs:label	Provider 1	Provider 2	Provider 3	Provider 4

Figure 6.48: Overview of the four example instances of providers from the Gaia-X dataspace used in this evaluation.

Evaluation of the Impact of Shortcut Templates for Constraints

To evaluate the impact of the constraint template enhancement on the time required for the validation process and the error proneness of this process, we conduct a hands-on experiment that simulated a real-world work situation. For this purpose, we ask a Semantic Web expert to implement a defined set of constraints in [SHACL](#) once without the constraint mapping tool and once with the constraint mapping tool. The developer has expertise in working with [SHACL](#) on a weekly basis and has a fundamental understanding of [SPARQL](#).

The experiment is structured as follows. A data graph is provided, which describes the four different provider instances shown in [figure 6.48](#). The data graph is required to satisfy four constraints: (C1) Each provider trusts itself, (C2) the legally binding name of a provider must be unique, (C3) the property "ex:hasProjectPartner" is symmetric, and (C4) at least

one label with an English language tag must be defined. An example data graph contains one error for each of these constraints, meaning a validation with a correct shapes graph should lead to four validation errors. The developer is tasked with creating a [SHACL](#) shapes graph without using the constraint mapping tool, given the data graph and the constraint requirements. After 30 minutes, a validation of the data graph is conducted. For the second part of the experiment, the developer uses the constraint mapping tool in the form of a Python command line tool. In addition, written documentation on how to use the tool is provided. Again, the allotted time is 30 minutes. The results of the experiment are presented in [table 6.19](#), which shows that our approach led to improvements in both the time needed to build the shapes graph and the correctness of the validation process.

	Without tool (baseline)	With tool
Implementation of C1	incorrect implementation	successfully implemented
Implementation of C2	not implemented	successfully implemented
Implementation of C3	not implemented	successfully implemented
Implementation of C4	incorrect implementation	successfully implemented
Time needed	Experiment stopped after 30 minutes	Done after 15 minutes

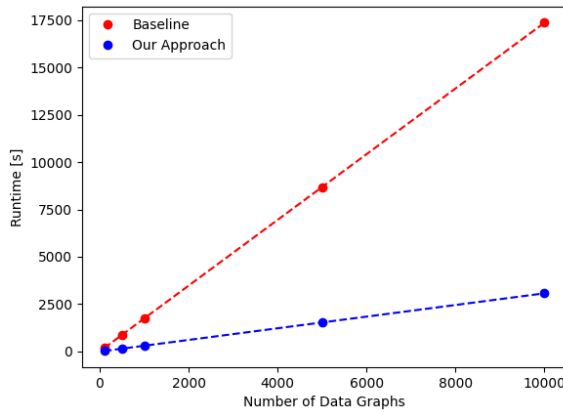
Table 6.19: Overview of the experiment results, determining the utility of our constraint mapping tool.

Evaluation of the Inference Optimization

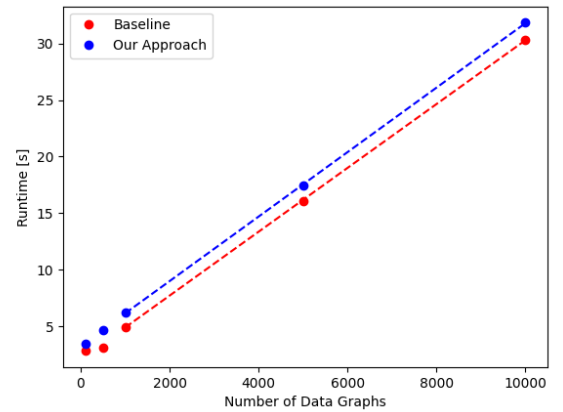
In the second part of our evaluation, the inference optimization, we focus on measuring the execution times of our proposed validation techniques under different scenarios. [Table 6.20](#) summarizes the results of several runs of our experiment, comparing the baseline common approach with our proposed method. In both scenarios, *individual data graphs* and *merged data graph*, we compared the state-of-the-art common approach as a baseline with our approach. Scalability is measured by a growing number of 100 – 10000 input data graph files. Each cell on the right contains three measurements representing repeated experiments. Notably, our approach achieves a significant reduction in execution times for individual data graphs by about 80 %, demonstrating its efficiency and scalability. However, the execution times for a merged data graph remain relatively unchanged between the baseline and our approach. [Figure 6.49](#) visualizes the average execution times of all runs for a comprehensive analysis. These results validate the practicality and scalability of our proposed validation approaches for application in real-world scenarios.

Table 6.20: Execution times measured in our experiments.

		Number of Files				
		100	500	1000	5000	10000
individual data graphs	Baseline	171.85	882.09	1737.95	8600	17200
		177.36	884.06	1808.8	8850	17700
		172	866.37	1658.9	8600	17200
	Our Approach	31.54	142.45	289.93	1600	3200
		30.41	144.34	298.7	1500	3000
		30.2	145.57	258.76	1500	3000
merged data graph	Baseline	2.82	3.02	5.12	16.24	30.68
		2.76	3.17	4.73	15.83	30.21
		2.84	3.19	5.02	16.08	30.15
	Our Approach	3.34	4.47	5.99	17.26	31.48
		3.44	4.67	6.09	18.73	32.08
		3.55	4.82	6.48	17.41	31.96



(a) Validating individual data graphs



(b) Validating merged data graph

Figure 6.49: Visualization of the average execution times of our experiments (see [table 6.20](#)).

Contributions

In conclusion, this section has successfully addressed [requirement Domain-R6](#) by proposing a scaling inference solution and a constraint template mapping solution. This facilitates common information design in the decentralized environment of dataspace by making validation techniques scalable and usable for dataspace participants. An evaluation in a real-world scenario with anonymized participants from the Gaia-X dataspace demonstrated feasibility, scalability, and practical impact. These contributions include:

- **A Scaling Solution for Validation:** Our modification of the current common practice of [RDF](#) validation with [SHACL](#) applies inference only once on the shape graph instead of on each instance. The shape graph becomes slightly more complex, but expensive inference on each data graph is avoided. This contributes to [requirement Domain-R6](#) by making recurrent validation against the same shape graph more scalable.

- **Shortcuts for Commonly Used Constraints:** We provided shortcuts in the form of template-based properties for commonly used constraints that are not yet natively supported by validation languages, simplifying and unifying their application for users. We have introduced [SHACL-SPARQL](#) templates, which are used by our proposed mapping tool to enable native support by any [SHACL](#) validator. This contributes to [requirement Domain-R6](#) by allowing a wide range of dataspace users, including non-experts, to formulate even complex constraints for validation.

The contributions provide reliable guarantees for an integrated use data or services in applications on the domain layer (see [figure 6.1](#)), such as [APIs](#) (see [section 5.4](#)). The proposed scaling but convenient solutions for validation in dataspace allow the diverse participants of dataspace, including non-experts, to use shortcuts for expressing common validation constraints and to validate these much faster with standard validators. The proposed enhanced validation techniques improve interoperability by providing guarantees for all kinds of interfaces between participants or services in dataspace, which contributes to a better common understanding in dataspace.

6.8 Discussion and Conclusion

This chapter has undertaken a comprehensive exploration of the complexities inherent in dataspace, with a particular focus on cultivating a shared domain understanding among dataspace participants. This is to facilitate effective data reuse and knowledge dissemination (see [RQ 4](#)). Exploring the layers of the [DLCM](#) (see [figure 6.1](#)), with particular emphasis on the domain layer, has allowed an in-depth examination of how domain-specific knowledge interlinks with data and metadata to enrich understanding and use within dataspace. From [RQ 4](#) we derived the need to identify methods to improve metadata quality and promote valuable data and knowledge reuse within dataspace through common domain understanding.

Our investigation began with an examination of the foundational layers of dataspace, building on previous contributions from this thesis that solved challenges related to heterogeneous data access and metadata modeling (see [chapters 4 and 5](#)). Through this exploration, we laid the groundwork for proposing solutions that enable seamless data exchange and discovery within dataspace. However, we identified a critical gap: the need for a common understanding among participants to leverage the full potential of data and metadata. To address this gap, we outlined a set of requirements and a methodological approach (see [section 6.1](#)). These requirements underscored the need for common vocabularies, best practices, and accessible tools tailored for both experts and non-experts. Our methodological approach aimed to systematically address these requirements, resulting in a set of contributions.

Each section of this chapter contributed distinctively to our overarching goal of advancing common domain understanding within dataspace. We addressed challenges ranging from identifier semantics to knowledge representation for non-experts, and presented solutions tailored to different participant profiles and domains. The proposed solutions include:

- **Revised Layer Model:** Separation of identifiers, structure, and reasoning to reduce false assertions and support identification techniques (see [section 6.2](#)).
- **Best Practices Guidebook:** Facilitate best practices for vocabulary development and relevant tool support, especially for non-experts (see [section 6.3](#)).

- **Neologism 2.0:** A non-expert tool for the rapid, collaborative drafting of vocabularies that can later be refined by Semantic Web experts (see [section 6.4](#)).
- **Recommender System:** Effective recommendation systems ensure that different vocabularies share more common content, facilitating the reuse of common knowledge or concepts for all dataspace participants (see [section 6.5](#)).
- **Continuous Knowledge Evolution:** Enabling semantic annotations to evolve over time, ensuring context-awareness, relevance and interoperability (see [section 6.6](#)).
- **Enhanced Validation Techniques:** Scaling validation processes and simplifying constraint formulation for diverse dataspace participants (see [section 6.7](#)).

These contributions collectively address the identified requirements and pave the way for improved collaboration and knowledge sharing within dataspace. By fostering a common domain understanding, we enable participants to unlock the full potential of data and metadata, overcoming disciplinary boundaries and technical complexities. In summary, our efforts represent a significant step toward realizing the vision of dataspace as dynamic ecosystems of shared knowledge and resources. However, our journey is not complete. Future efforts may include usability evaluations, more real-world applications, and further refinement of our proposed solutions. Through an improved common domain understanding among dataspace participants, we aim to complete the other layers of the dataspace [DLCM](#) and enable seamless (re)use of information within dataspace.

7 Conclusion and Outlook

Digital transformation is accelerating at a rapid pace, driven by the exponential growth of data. In this era where data has become the heartbeat of industries, organizations, and societies, the demand for control over personal and proprietary data throughout its lifecycle has never been more pressing. Dataspaces have emerged as a promising solution to this challenge, offering the potential of data sovereignty by enabling individuals and organizations to maintain greater autonomy and control over their information. However, despite their potential, dataspaces still have significant research gaps, particularly in the areas of interoperability and establishing a common understanding among diverse stakeholders. These gaps present significant barriers to the seamless sharing and utilization of data within dataspaces. Fortunately, semantic technologies offer promising methods for overcoming these challenges. By providing a framework for structuring and linking data in machine-readable formats, semantic technologies hold the key to improving data interoperability and fostering common understanding across heterogeneous dataspace environments.

This dissertation has undertaken a comprehensive exploration of the role of semantic technologies in dataspaces. The approach taken is to integrate semantic technologies into dataspaces to address the challenges of interoperability and common understanding. The research journey began with the development of a [DLCM](#) tailored for dataspaces, followed by an investigation of information models aimed at facilitating a common understanding of data within dataspaces. Subsequent chapters addressed the complexities of heterogeneous data access and cultivating a shared domain understanding among dataspace participants. Through empirical research and theoretical frameworks, this dissertation has demonstrated the efficacy of semantic technologies in improving data management practices within dataspaces, paving the way for a more interconnected and collaborative global digital ecosystem.

Chapter Outline

This chapter concludes this thesis and is structured as follows:

- [Section 7.1](#) first summarizes the overall contribution of this thesis. It details in the context of dataspaces the solutions provided for developing a data lifecycle model, for exploring information models for common general concepts, for accessing data despite various forms of heterogeneity, and for creating a common domain understanding among participants.
- [Section 7.2](#) then discusses the limitations and impact of the presented solutions.
- [Section 7.3](#) finally outlines the remaining open challenges and directions for future work related to dataspaces.

7.1 Key Contributions

To address the challenge of successfully managing data in dataspace by establishing interoperability and a common understanding in dataspace, we identified and addressed four research questions (see [section 1.2](#)):

RQ 1. *What are the key stages and challenges in the data lifecycle within dataspace, and how can a comprehensive data lifecycle model address these challenges?*

RQ 2. *How can both human and machine understanding be achieved in dataspace and what are useful information models to agree on general concepts within dataspace?*

RQ 3. *How can data access from heterogeneous sources, such as manufacturing systems, be effectively facilitated within dataspace to support seamless integration and utilization of diverse data?*

RQ 4. *How can reuse of data and knowledge within dataspace be achieved, and how can recommendations, best practices, and validation mechanisms enhance data and metadata quality to support this reuse?*

We have addressed these research questions with a series of key contributions. These contributions include new concepts and approaches, as well as their respective implementations, which enable rapid evaluation and adoption in dataspace. Below, we summarize the key contributions and describe how they address the respective research questions.

We addressed [RQ 1](#) in [chapter 3](#) by developing a [DLCM](#) for dataspace:

- Investigated existing [DLCMs](#) and evaluated their suitability for dataspace before proposing a model best suited to address the needs of dataspace.
- Proposed a framework for managing data throughout its lifecycle within dataspace, serving as a foundational framework for data governance, sovereignty, and interoperability.
- Developed a tailored [DLCM](#) specifically designed for dataspace, addressing the unique challenges and requirements of these environments, and fine-tuned the proposed [DLCM](#) to maximize its impact within dataspace, ensuring alignment with the dynamic nature of these environments.

We addressed [RQ 2](#) in [chapter 4](#) by investigating the principles of information models for defining a common core in dataspace:

- Identified the key role of information models in enabling effective communication, collaboration, and trust among participants in decentralized dataspace ecosystems.
- Provided structured representations of data, services, participants, and interactions, serving as a foundation for robust mechanisms for data governance, sovereignty, and interoperability.
- Presented two illustrative examples of information models designed to meet the needs and requirements of dataspace environments.

7 Conclusion and Outlook

- Conducted a comparative evaluation of the two information models to assess their strengths and areas for improvement, highlighting their potential impact on enhancing common understanding within dataspaces.

We addressed [RQ 3](#) in [chapter 5](#) by advancing the principles of data access for dataspaces, covering various types of heterogeneity:

- Proposed solutions to improve data accessibility from heterogeneous systems within dataspaces, bridging the gap between low-level query interfaces and higher-level query languages.
- Addressed the challenge of managing resource constraints by proposing an adaptive data load management solution for dataspaces, transforming static data collection into adaptive data collection.
- Proposed a solution for managing dynamic data changes within dataspaces, emphasizing the need for robust strategies to proactively monitor and manage data and metadata changes.
- Addressed the challenge of enriching metadata within dataspaces by proposing a methodology for generating rich metadata through data mappings.

We addressed [RQ 4](#) in [chapter 6](#) by contributing to a common domain understanding in dataspaces:

- Separated identifiers, structure, and reasoning in a revised Semantic Web layer model to reduce false assertions and support identification techniques.
- Facilitated best practices for vocabulary development and relevant tool support, especially for non-experts, through the Best Practices Guidebook.
- Developed Neologism 2.0, a non-expert tool for rapid, collaborative vocabulary drafting, to be refined by Semantic Web experts.
- Designed an effective recommendation system to ensure that different vocabularies share more common content, facilitating the reuse of common knowledge or concepts for all dataspace participants.
- Enabled semantic annotations to evolve over time, ensuring context-awareness, relevance, and interoperability.
- Improved validation techniques to scale validation processes and simplify constraint formulation for diverse dataspace participants.

The combination of the presented approaches and solutions enables the realization of the semantic foundations of dataspaces, thus facilitating interoperability and common understanding in dataspaces. This, in turn, ensures the proper management of data in dataspaces.

7.2 Discussion and Limitations

There are several limitations to the research presented in this thesis, although it makes a significant contribution to the understanding and advancement of semantic technologies

in dataspace. First, the scope of the research may not have encompassed all aspects of dataspace, and certain nuances or complexities may have been overlooked. In addition, the implementation of semantic solutions in real dataspace may face practical challenges that are not fully addressed in this research. Furthermore, the evaluation of the proposed solutions may be limited in scope or scale, and the generalizability of the findings may vary across different contexts.

In addition, the adoption and acceptance of semantic technologies within organizations and industries may face resistance or barriers due to factors such as cost, complexity, and cultural norms. Comprehensive change management strategies and stakeholder engagement efforts are required to overcome these challenges.

Furthermore, while semantic technologies have the potential to enhance data interoperability and comprehension, they are not a solution per se and may not fully address all data management challenges in dataspace. Consequently, it may be necessary to adopt a combination of approaches and technologies to achieve comprehensive data governance and management. It is also important to consider the evolving nature of technology and data ecosystems. As new technologies and standards emerge, the applicability and effectiveness of semantic solutions in dataspace may change. Therefore, ongoing monitoring and adaptation of semantic approaches is essential to ensure their continued relevance and effectiveness.

A final limitation and potential risk of dataspace is the possibility of not achieving the anticipated impact and widespread reach that they promise nowadays. Despite their potential to revolutionise data management and drive innovation, dataspace face a number of challenges that could hinder their effectiveness. Technical obstacles, including data interoperability, scalability, and security, as well as organizational barriers, such as resistance to change and siloed data practices, present significant challenges to the adoption and impact of dataspace. In addition, socio-economic factors such as unequal access to technology and regulatory constraints may limit the inclusivity and reach of dataspace. This is particularly evident in the European Union, which is currently the primary driving force in this research area, with a focus on privacy, security, and sovereignty. Addressing these challenges will require collaborative efforts in technical innovation, organizational change, and socioeconomic policy to ensure that dataspace fulfill their potential in today's digital landscape.

7.3 Directions for Future Work

Despite the contributions made by this research, there remain several avenues for future work to further enhance the understanding and application of semantic technologies in dataspace.

One area for future exploration is the development of advanced semantic models and ontologies tailored specifically for dataspace. These models could incorporate domain-specific knowledge and semantics to improve data interoperability and understanding across heterogeneous environments.

Another promising direction is the integration of semantic technologies with emerging technologies such as artificial intelligence and blockchain. By leveraging synergies between these technologies, novel approaches for data management, governance, and trust could be developed, paving the way for more intelligent and secure dataspace.

Furthermore, conducting extensive empirical studies and case studies in real-world dataspace could provide valuable insights into the practical challenges and benefits of imple-

7 Conclusion and Outlook

menting semantic solutions. These studies could help validate the effectiveness of semantic approaches and inform best practices for their deployment.

Another emerging trend that warrants exploration in future research is the utilization of *large language models* such as GPT-4 [Ope23] (Generative Pre-trained Transformer) or Mixtral 8x7B [Mis23] for enhancing data management and understanding in dataspace [Arn+24]. Large language models have demonstrated remarkable capabilities in natural language processing, knowledge extraction, and data synthesis [Lew+20]. This research area has been identified by the *German Research Foundation DFG* in a recent press release [DFG23], formulating a guideline for dealing with generative models. Integrating large language models with semantic technologies could enable advanced capabilities such as automated data annotation, context-aware data retrieval, and natural language-based querying of dataspace. Furthermore, leveraging large language models for data summarization and knowledge extraction could facilitate rapid insights generation from vast and heterogeneous datasets within dataspace.

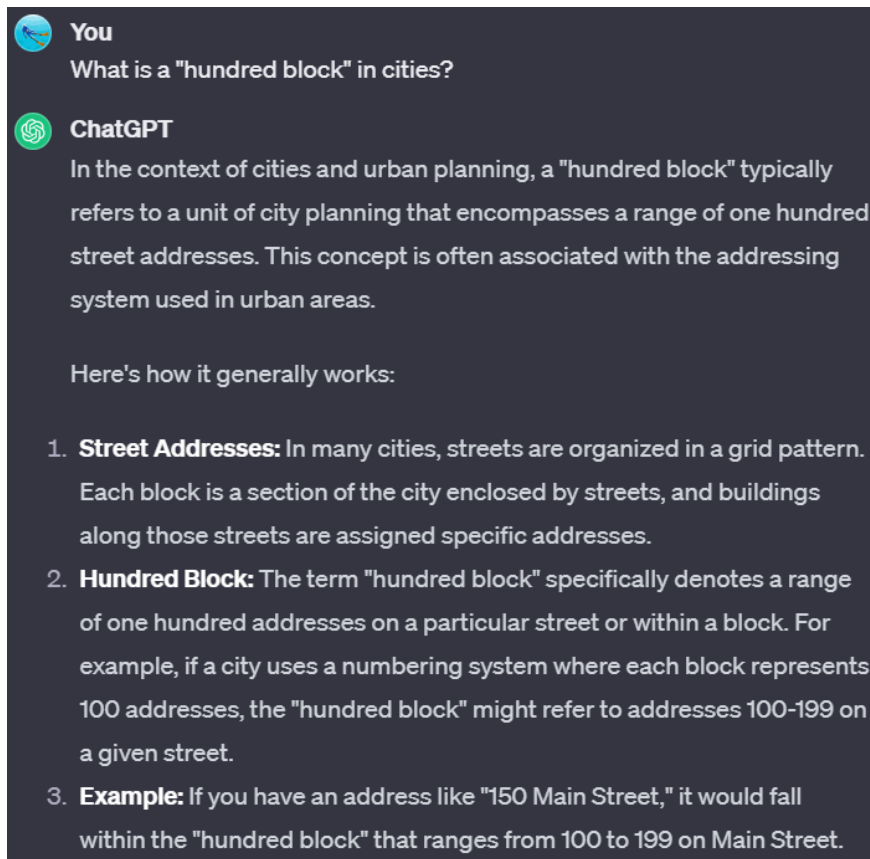


Figure 7.1: While conventional recommender systems (see [section 6.5](#)) could not find a concept for a specialized term, ChatGPT could.

An example is shown in [figure 7.1](#), which demonstrates that, in contrast to conventional recommender systems (see [section 6.5](#)), large language models can provide broad knowledge even for niche domains or vague terms and can significantly blur the boundaries of interoperability, thereby reducing barriers and facilitating the creation of a common understanding more easily. This result was generated with ChatGPT 3.5 on January 16, 2024. However, the ethical implications, biases, and risks associated with the use of Large language models in dataspace must be carefully considered and addressed. Future research could explore meth-

ods for effectively harnessing the power of large language models while mitigating potential drawbacks, thereby unlocking new opportunities for data-driven innovation in dataspace.

A crucial aspect for future work involves transitioning from the current research-driven "push" stage of dataspace initiatives to a market-driven "pull" stage characterized by promising use cases and industry needs. While many dataspace initiatives are currently spearheaded by research institutions and governmental bodies like the European Commission, there's a pressing need to engage industry stakeholders more extensively in the development and deployment of dataspace. This transition requires fostering greater collaboration and partnership between academia, industry, and policymakers to align research efforts with market demands and real-world applications. Initiatives aimed at incentivizing industry participation, such as funding mechanisms, industry-academia collaborations, and regulatory frameworks that encourage data sharing and innovation, are essential for creating a conducive environment for market-driven dataspace. Moreover, promoting awareness and understanding of the potential benefits of dataspace among industry stakeholders and facilitating knowledge transfer from research to practice can help catalyze industry adoption and uptake of dataspace. By bridging the gap between research-driven initiatives and industry needs, future efforts can propel dataspace into a new phase of market-driven innovation and impact.

Finally, fostering interdisciplinary collaboration between researchers, practitioners, and stakeholders from various domains could facilitate the co-creation of innovative solutions and standards for semantic-driven dataspace. By bringing together diverse expertise and perspectives, collaborative efforts could accelerate the development and adoption of semantic technologies in dataspace.

Bibliography

- [ABS06] Harith Alani, Christopher Brewster, and Nigel Shadbolt. “Ranking Ontologies with AKTiveRank.” In: *The Semantic Web - ISWC 2006*. Ed. by Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora M. Aroyo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–15. ISBN: 978-3-540-49055-5. DOI: [10.1007/11926078_1](https://doi.org/10.1007/11926078_1) (cit. on pp. 166, 187).
- [ADH19] Abdelghany Salah Abdelghany, Nagy Ramadan Darwish, and Hesham Ahmed Hefni. “An Agile Methodology for Ontology Development.” In: *International Journal of Intelligent Engineering and Systems* 12.2 (2019), pp. 170–181 (cit. on pp. 140, 144).
- [Agi+00] Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martínez. “Enriching very large ontologies using the WWW.” In: *arXiv preprint cs/0010026* (2000) (cit. on p. 196).
- [AH06] S. Auer and Heinrich Herre. “RapidOWL - An Agile Knowledge Engineering Methodology.” In: *Ershov Memorial Conference*. 2006 (cit. on pp. 140, 144).
- [Ala+07] Harith Alani, Natasha F Noy, Nigam Shah, Nigel Shadbolt, and Mark A Musen. “Searching ontologies based on content: experiments in the biomedical domain.” In: *Proceedings of the 4th international conference on Knowledge capture*. 2007, pp. 55–62 (cit. on pp. 166, 167).
- [Alb+19] R Albertoni, D Browning, S Cox, A Gonzalez Beltran, A Perego, and P Winstanley. “Data Catalog Vocabulary (DCAT)-Version 2.” In: *W3C Recommendation* 3 (2019). Accessed on 2024-05-23. URL: <https://www.w3.org/TR/vocab-dcat-2/> (cit. on pp. 56, 152).
- [Ale+11] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. *Describing Linked Datasets with the VoID Vocabulary*. W3C Interest Group Note. Accessed on 2024-05-23. The Semantic Web Interest Group, 2011. URL: <https://www.w3.org/TR/void/> (cit. on pp. 56, 152).
- [Alf22] Yousef Alfaifi. “Ontology Development Methodology: A Systematic Review and Case Study.” In: *2022 2nd International Conference on Computing and Information Technology (ICCIT)*. 2022, pp. 446–450. DOI: [10.1109/ICCIT52419.2022.9711664](https://doi.org/10.1109/ICCIT52419.2022.9711664) (cit. on p. 140).
- [All+21] Dean Allemang, Pawel Garbacz, Przemyslaw Gradzki, Elisa F. Kendall, and Robert Trypuz. “An Infrastructure for Collaborative Ontology Development.” In: *Formal Ontology in Information Systems - Proceedings of the Twelfth International Conference, FOIS 2021, Bozen-Bolzano, Italy, September 11-18, 2021*. Ed. by Fabian Neuhaus and Boyan Brodaric. Vol. 344. Frontiers in Artificial Intelligence and Applications. IOS Press, 2021, pp. 112–126. DOI: [10.3233/FAIA210375](https://doi.org/10.3233/FAIA210375) (cit. on p. 140).

- [ALN11] Sören Auer, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. “Introduction to Linked Data and Its Lifecycle on the Web.” In: *Reasoning Web. Semantic Technologies for the Web of Data*. Ed. by Axel Polleres, Claudia d’Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter Patel-Schneider. Vol. 6848. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, Jan. 2011, pp. 1–75. ISBN: 978-3-642-23031-8. DOI: [10.1007/978-3-642-23032-5_1](https://doi.org/10.1007/978-3-642-23032-5_1) (cit. on pp. 33, 35).
- [Ara+13] Carlos Buil Aranda, Olivier Corby, Souripriya Das, Lee Feigenbaum, Paula Gearon, Birte Glimm, Steve Harris, Sandro Hawke, Ivan Herman, Nicholas Humfrey, Nico Michaelis, Chimezie Ogbuji, Matthew Perry, Alexandre Pas-sant, Axel Polleres, Eric Prud’hommeaux, Andy Seaborne, and Gregory Todd Williams. *SPARQL 1.1 Query Language for RDF*. W3C Recommendation. Accessed on 2024-05-23. The W3C SPARQL Working Group, 2013. URL: <https://www.w3.org/TR/sparql11-overview/> (cit. on pp. 14, 23, 67).
- [Arn+22] Susanne Arndt, Benjamin Farnbacher, Marc Fuhrmans, Stephan Hachinger, Johanna Hickmann, Nils Hoppe, Martin Thomas Horsch, Dorothea Iglezakis, Ashish Karmacharya, Giacomo Lanza, Sophia Leimer, Johannes Munke, Džulia Terzijska, Johannes Theissen-Lipp, Cord Wiljes, and Jürgen Windeck. *Meta-data4Ing: An ontology for describing the generation of research data within a scientific activity*. Tech. rep. Contributions: SA, MF, DI, and GL identified the research gap and designed and conceived the work. All authors performed, analyzed, and interpreted the work. All authors drafted and revised the paper. NFDI, 2022. DOI: [10.5281/zenodo.5957104](https://doi.org/10.5281/zenodo.5957104) (cit. on pp. 32, 47, 152).
- [Arn+24] Benedikt T. Arnold, Johannes Theissen-Lipp, Diego Collarana, Christoph Lange, Sandra Geisler, Edward Curry, and Stefan Decker. “Towards Enabling FAIR Dataspaces Using Large Language Models.” In: *The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, May 26–27, 2024, Hersonissos, Greece*. Contributions: BA and SD identified the research gap. BA, JTL, and DC designed and conceived the work. BA performed, analyzed, and interpreted the work. BA drafted the paper. BA, JTL, DC, CL, SG, and EC reviewed and revised the paper. SD and CL provided guidance and funding. CEUR-WS. June 2024. URL: <https://ceur-ws.org/Vol-3705/short09.pdf> (cit. on p. 226).
- [ASS15] Robert Arp, Barry Smith, and Andrew D Spear. *Building Ontologies with Basic Formal Ontology*. MIT Press, 2015 (cit. on p. 138).
- [Aue22] S. Auer. “Semantic Integration and Interoperability.” In: ed. by B. Otto, M. ten Hompel, and S. Wrobel. Cham: Springer International Publishing, 2022, pp. 195–210. ISBN: 978-3-030-93975-5. DOI: [10.1007/978-3-030-93975-5_12](https://doi.org/10.1007/978-3-030-93975-5_12) (cit. on p. 3).
- [Ayi+21] Kudirat Ayinla, Edlira Vakaj, Franco Cheung, and Abdel-Rahman H. Tawil. “A Semantic Offsite Construction Digital Twin- Offsite Manufacturing Production Workflow (OPW) Ontology.” In: 2021 (cit. on p. 154).
- [Bad+20a] Sebastian Bader, Fabian Bruckner, Böge Böge, Dennis O. Kubitza, Jörg Langkau, Dileep Murthy, and Ralf Nagel. *Specification: IDS Meta Data Broker*. Tech. rep. Accessed on 2024-05-23. Anna-Louisa-Karsch-Str. 2, 10178 Berlin: International Data Spaces Association, May 2020. URL:

- https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-White-Paper-Specification-IDS-Meta-Data-Broker.pdf (cit. on p. 32).
- [Bad+20b] Sebastian Bader, Jaroslav Pullmann, Christian Mader, Sebastian Tramp, Christoph Quix, Andreas Müller, Haydar Akyürek, Matthias Böckmann, Benedikt Imbusch, Johannes Theissen-Lipp, Sandra Geisler, and Christoph Lange. “The International Data Spaces Information Model - An Ontology for Sovereign Exchange of Digital Content.” In: *Proceedings of the 19th International Semantic Web Conference*. Contributions: SB identified the research gap. SB, JP, CM, ST, and CL designed, conceived, and interpreted the work. JP, CM, ST, CL, JL, HA, MB, BI, and SB performed, analyzed, and evaluated the work. CQ, AM, and SG provided feedback and reviewed the paper. CL and SB provided funding and guidance. Cham: Springer International Publishing, 2020, pp. 176–192. DOI: [10.1007/978-3-030-62466-8_12](https://doi.org/10.1007/978-3-030-62466-8_12) (cit. on pp. 7, 9, 28, 32, 45, 53–57).
- [Bai+05] James Bailey, François Bry, Tim Furche, and Sebastian Schaffert. “Web and semantic web query languages: A survey.” In: *Reasoning Web: First International Summer School 2005, Msida, Malta, July 25-29, 2005, Revised Lectures* (2005), pp. 35–133 (cit. on p. 14).
- [Bai+09] James Bailey, François Bry, Tim Furche, and Sebastian Schaffert. “Semantic Web Query Languages.” In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 2583–2586. ISBN: 978-0-387-35544-3. DOI: [10.1007/978-0-387-39940-9_1317](https://doi.org/10.1007/978-0-387-39940-9_1317) (cit. on p. 14).
- [Bao+12] Jie Bao, Diego Calvanese, Bernardo Cuenca Grau, Martin Džbor, Achille Fokoue, Christine Golbreich, Sandro Hawke, Ivan Herman, Rinke Hoekstra, Ian Horrocks, Elisa Kendall, Markus Krötzsch, Carsten Lutz, Deborah L. McGuinness, Boris Motik, Jeff Pan, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph, Alan Ruttenberg, Uli Sattler, Michael Schneider, Mike Smith, Evan Wallace, Zhe Wu, Antoine Zimmermann, Jeremy Carroll, Jim Hendler, and Vipul Kashyap. *OWL 2 Web Ontology Language*. W3C Recommendation. Accessed on 2024-05-23. W3C OWL Working Group, 2012. URL: <https://www.w3.org/TR/owl2-overview/> (cit. on pp. 18, 210).
- [Bār+10] Jānis Bārzdīņš, Guntis Bārzdīņš, Kārlis Čerāns, Renārs Liepiņš, and Artūrs Sprogis. “OWLGrEd: a UML style graphical notation and editor for OWL 2.” In: *Proceedings of 7th International Workshop “OWL: Experience and Directions*. Vol. 34. 2010 (cit. on p. 20).
- [Bas+09] Cosmin Basca, Stéphane Corlosquet, Richard Cyganiak, Sergio Fernández, and Thomas Schandl. “Neologism: Easy Vocabulary Publishing.” In: *Scripting and Development for the Semantic Web (SFSW)*. 2009. URL: <http://ceur-ws.org/Vol-368/paper10.pdf> (cit. on pp. 158, 159).
- [BBP04] Luisa Bentivogli, Andrea Bocco, and Emanuele Pianta. “ArchiWordnet: integrating Wordnet with domain-specific knowledge.” In: *Proceedings of the 2nd International Global Wordnet Conference*. 2004, pp. 39–47 (cit. on p. 196).

- [BCB21a] Sadeer Beden, Qiushi Cao, and Arnold Beckmann. “SCRO: A Domain Ontology for Describing Steel Cold Rolling Processes towards Industry 4.0.” In: *Information* 12.8 (2021). ISSN: 2078-2489. DOI: [10.3390/info12080304](https://doi.org/10.3390/info12080304). URL: <https://www.mdpi.com/2078-2489/12/8/304> (cit. on p. 154).
- [BCB21b] Sadeer Beden, Qiushi Cao, and Arnold Beckmann. “Semantic Asset Administration Shells in Industry 4.0: A Survey.” In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021, pp. 31–38. DOI: [10.1109/ICPS49255.2021.9468266](https://doi.org/10.1109/ICPS49255.2021.9468266) (cit. on pp. 140, 154).
- [Bec+14] David Beckett, Tim Berners-Lee, Eric Prud’hommeaux, and Gavin Carothers. “RDF 1.1 Turtle.” In: *World Wide Web Consortium* (2014), pp. 18–31. URL: <https://www.w3.org/TR/turtle/> (cit. on p. 13).
- [Bec+21] Fabian Becker, Pascal Bibow, Manuela Dalibor, Aymen Gannouni, Viviane Hahn, Christian Hopmann, Matthias Jarke, István Koren, Moritz Kröger, Johannes Lipp, Judith Maibaum, Judith Michael, Bernhard Rumpe, Patrick Sapel, Niklas Schäfer, Georg J. Schmitz, Günther Schuh, and Andreas Wortmann. “A Conceptual Model for Digital Shadows in Industry and Its Application.” In: *Conceptual Modeling*. Ed. by Aditya Ghose, Jennifer Horkoff, Vítor E. Silva Souza, Jeffrey Parsons, and Joerg Evermann. Vol. 13011. Lecture Notes in Computer Science. Contributions: JMi and MD identified the research gap and designed and conceived the work. MD, JMi, PB, FB, MK, JL, PS, NS, JMa, AG, VH, and GS performed, analyzed, and interpreted the work. IK and AW reviewed the paper and provided feedback. AW, GS, BR, CH, and MJ provided guidance and funding. Cham: Springer International Publishing, 2021, pp. 271–281. ISBN: 978-3-030-89021-6. DOI: [10.1007/978-3-030-89022-3_22](https://doi.org/10.1007/978-3-030-89022-3_22) (cit. on pp. 58, 60).
- [Bee+10] Jakob Beetz, Léon van Berlo, Ruben de Laat, and Pim van den Helm. “BIM-server.org – An open source IFC model server.” In: *Proceedings of the CIP W78 conference*. 2010, p. 8 (cit. on pp. 117, 121).
- [Ben+12] Rimel Bendadouche, Catherine Roussey, Gil De Sousa, Jean-Pierre Chanet, and Kun Mean Hou. “Extension of the Semantic Sensor Network Ontology for Wireless Sensor Networks: The stimulus-WSNnode-communication Pattern.” In: *5th International Workshop on Semantic Sensor Networks in conjunction with the 11th International Semantic Web Conference (ISWC)*. 2012, 16–p (cit. on p. 154).
- [Ben19] Mahdi Bennara. “Linked Service Integration on the Semantic Web.” PhD thesis. Université de Lyon, 2019. URL: <https://theses.hal.science/tel-02484830v2> (cit. on p. 100).
- [Ber00] Tim Berners-lee. *Semantic Web on XML - The Semantic Web Stack*. Accessed on 2024-05-23. 2000. URL: <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html> (cit. on pp. 11, 12).
- [BGM14] Dan Brickley, Ramanathan V. Guha, and Brian McBride. *RDF Schema 1.1*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2014. URL: <https://www.w3.org/TR/rdf-schema/> (cit. on p. 18).
- [BHB09] Christian Bizer, Tom Heath, and Tim Berners-Lee. “Linked Data - The Story So Far.” In: *Int. J. Semantic Web Inf. Syst.* 5 (2009), pp. 1–22 (cit. on p. 21).

Bibliography

- [BHB11] Christian Bizer, Tom Heath, and Tim Berners-Lee. “Linked data: The story so far.” In: *Semantic services, interoperability and web applications: emerging concepts*. IGI global, 2011, pp. 205–227 (cit. on p. 24).
- [BHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. “The Semantic Web.” In: *Scientific American* 284.5 (2001), pp. 34–43. URL: <https://www.jstor.org/stable/26059207> (cit. on pp. 3, 95, 133).
- [Bis+12] Stefan Bischof, Stefan Decker, Thomas Krennwallner, Nuno Lopes, and Axel Polleres. “Mapping between RDF and XML with XSPARQL.” In: *Journal on Data Semantics* 1.3 (2012), pp. 147–185. DOI: [10.1007/s13740-012-0008-7](https://doi.org/10.1007/s13740-012-0008-7) (cit. on pp. 18, 67).
- [Bis22] Mike Bishop. *HTTP/3 (RFC 9114)*. June 2022. DOI: [10.17487/RFC9114](https://doi.org/10.17487/RFC9114). URL: <https://www.rfc-editor.org/info/rfc9114> (cit. on p. 14).
- [BK03] Jeen Broekstra and Arjohn Kampman. “SeRQL: A second generation RDF query language.” In: *Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval*. Citeseer. 2003, pp. 13–14 (cit. on p. 14).
- [BLC08] Graciela Brusa, M Laura Caliusco, and Omar Chiotti. “Towards Ontological Engineering: A Process for Building a Domain Ontology from Scratch in Public Administration.” In: *Expert Systems* 25.5 (2008), pp. 484–503 (cit. on p. 144).
- [BLG16] Md Abul Bashar, Yuefeng Li, and Yang Gao. “A framework for automatic personalised ontology learning.” In: *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2016, pp. 105–112 (cit. on pp. 194, 195).
- [BLK18] Christian Block, Dominik Lins, and Bernd Kuhlenkötter. “Approach for a Simulation-based and Event-driven Production Planning and Control in Decentralized Manufacturing Execution Systems.” In: *Procedia CIRP* 72 (2018). 51st CIRP Conference on Manufacturing Systems, pp. 1351–1356. ISSN: 2212-8271. DOI: [10.1016/j.procir.2018.03.204](https://doi.org/10.1016/j.procir.2018.03.204). URL: <https://www.sciencedirect.com/science/article/pii/S2212827118303639> (cit. on p. 154).
- [BM00] Dan Brickley and Libby Miller. *FOAF Vocabulary Specification 0.99*. Namespace Document. Accessed on 2024-05-23. Semantic Web Interest Group, 2000. URL: <http://xmlns.com/foaf/spec/> (cit. on p. 152).
- [BM07] Philip A. Bernstein and Sergey Melnik. “Model management 2.0: manipulating richer mappings.” In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’07. Beijing, China: Association for Computing Machinery, 2007, pp. 1–12. ISBN: 9781595936868. DOI: [10.1145/1247480.1247482](https://doi.org/10.1145/1247480.1247482) (cit. on p. 16).
- [BM10] Stefano Borgo and Claudio Masolo. “Ontological Foundations of dolce.” In: *Theory and Applications of Ontology: Computer Applications*. Dordrecht: Springer Netherlands, 2010, pp. 279–295. ISBN: 978-90-481-8847-5. DOI: [10.1007/978-90-481-8847-5_13](https://doi.org/10.1007/978-90-481-8847-5_13) (cit. on p. 152).
- [BN84] Andrew D. Birrell and Bruce Jay Nelson. “Implementing Remote Procedure Calls.” In: *ACM Trans. Comput. Syst.* 2.1 (Feb. 1984), pp. 39–59. ISSN: 0734-2071. DOI: [10.1145/2080.357392](https://doi.org/10.1145/2080.357392) (cit. on p. 100).

- [Bol+08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. “Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge.” In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’08. Vancouver, Canada: Association for Computing Machinery, 2008, pp. 1247–1250. ISBN: 9781605581026. DOI: [10.1145/1376616.1376746](https://doi.org/10.1145/1376616.1376746) (cit. on pp. 168, 169).
- [Bon21] Francesco Bonfiglio. *Gaia-X Vision & Strategy*. Tech. rep. Avenue des Arts 6-9, 1210 Brussels: Gaia-X European Association for Data and Cloud AISBL, Dec. 2021. URL: <https://gaia-x.eu/wp-content/uploads/2021/12/Vision-Strategy.pdf> (cit. on pp. 7, 9).
- [Bou+94] Jacques Bouaud, Bruno Bachimont, Jean Charlet, and Pierre Zweigenbaum. “Acquisition and structuring of an ontology within conceptual graphs.” In: *Proceedings of ICCS*. Vol. 94. 1994, pp. 1–25 (cit. on p. 143).
- [BP08a] Diego Berrueta and Jon Phipps. *Best Practice Recipes for Publishing RDF Vocabularies*. W3C Working Group Note. The W3C Semantic Web Deployment Working Group, Aug. 2008. URL: <https://www.w3.org/TR/swbp-vocab-pub/> (cit. on p. 18).
- [BP08b] Diego Berrueta and Jon Phipps. *Best Practice Recipes for Publishing RDF Vocabularies*. W3C Working Group Note. 2008. URL: <https://www.w3.org/TR/swbp-vocab-pub/> (cit. on p. 56).
- [Bra+98] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, François Yergeau, et al. *Extensible Markup Language (XML) 1.0*. 1998. URL: <https://www.w3.org/TR/1998/REC-xml-19980210.html> (cit. on p. 13).
- [Bra78] Robert K. Branson. “The Interservice Procedures for Instructional Systems Development.” In: *Educational Technology* 18.3 (1978), pp. 11–14. ISSN: 00131962. URL: <http://www.jstor.org/stable/44418942> (visited on 10/27/2023) (cit. on p. 142).
- [Bre+18] Christian Brecher, Evgeny Kusmenko, Achim Lindt, Bernhard Rumpe, Simon Storms, Stephan Wein, Michael von Wenckstern, and Andreas Wortmann. “Multi-Level Modeling Framework for Machine as a Service Applications Based on Product Process Resource Models.” In: *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*. ISCSIC ’18. Stockholm, Sweden: Association for Computing Machinery, 2018. ISBN: 9781450366281. DOI: [10.1145/3284557.3284714](https://doi.org/10.1145/3284557.3284714) (cit. on p. 26).
- [Bri04] Dan Brickley. *WGS84 Geo Positioning: An RDF Vocabulary*. W3C Candidate Recommendation Draft. Accessed on 2024-05-23. Semantic Web Interest Group, 2004. URL: <https://www.w3.org/2003/01/geo/> (cit. on p. 56).
- [Bro13] John Brooke. “SUS: A Retrospective.” In: *Journal of usability studies* 8.2 (2013), pp. 29–40 (cit. on pp. 187, 188).
- [BS05] Eva Blomqvist and Kurt Sandkuhl. “Patterns in Ontology Engineering: Classification of Ontology Patterns.” In: *ICEIS (3)*. 2005, pp. 413–416 (cit. on p. 158).

Bibliography

- [BSP12] Eva Blomqvist, Azam Seil Sepour, and Valentina Presutti. “Ontology Testing - Methodology and Tool.” In: *Knowledge Engineering and Knowledge Management*. Ed. by Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d’Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 216–226. ISBN: 978-3-642-33876-2 (cit. on p. 156).
- [BT09] Adrian Burton and Andrew Treloar. “Designing for Discovery and Re-Use: The ANDS Data Sharing Verbs Approach to Service Decomposition.” In: *International Journal of Digital Curation* 4.3 (Dec. 2009), pp. 44–56. DOI: [10.2218/ijdc.v4i3.124](https://doi.org/10.2218/ijdc.v4i3.124) (cit. on pp. 32–34).
- [Bys+10] Jeppe Byskov-Nielsen, Juha-Matti Savolainen, Martin Snogdahl Christensen, and Peter Balling. “Ultra-short pulse laser ablation of metals: threshold fluence, incubation coefficient and ablation rates.” In: *Applied Physics A* 101.1 (May 2010), pp. 97–101 (cit. on p. 98).
- [Cam+22] David B. Cameron, Arnljot Skogvang, Mihaly Fekete, Henrik Martinsson, Morten Strand, Marcel Castro, Kirsten Helle, Anders Gjerver, Christian Mahesh Hansen, and Kristin Høines Johnsen. “The Digital Design Basis. Demonstrating a framework to reduce costs and improve quality in early-phase design.” In: *Digital Chemical Engineering* 2 (2022), p. 100015. ISSN: 2772-5081. DOI: [10.1016/j.dche.2022.100015](https://doi.org/10.1016/j.dche.2022.100015). URL: <https://www.sciencedirect.com/science/article/pii/S2772508122000060> (cit. on p. 139).
- [Cao+19] Qiushi Cao, Franco Giustozzi, Cecilia Zanni-Merk, François de Bertrand de Beuvron, and Christoph Reich. “Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach.” In: *Cybernetics and Systems* 50.2 (2019), pp. 82–96. DOI: [10.1080/01969722.2019.1565118](https://doi.org/10.1080/01969722.2019.1565118) (cit. on p. 154).
- [Cao+20] Qiushi Cao, Ahmed Samet, Cecilia Zanni-Merk, François de Bertrand de Beuvron, and Christoph Reich. “Combining Chronicle Mining and Semantics for Predictive Maintenance in Manufacturing Processes.” In: *Semantic Web* 11.6 (2020), pp. 927–948 (cit. on p. 154).
- [Cao+22] Qiushi Cao, Cecilia Zanni-Merk, Ahmed Samet, Christoph Reich, de Bertrand de Beuvron. François, Arnold Beckmann, and Giannetti. Cinzia. “KSPMI: A Knowledge-based System for Predictive Maintenance in Industry 4.0.” In: *Robotics and Computer-Integrated Manufacturing* 74 (2022), p. 102281. ISSN: 0736-5845. DOI: [10.1016/j.rcim.2021.102281](https://doi.org/10.1016/j.rcim.2021.102281). URL: <https://www.sciencedirect.com/science/article/pii/S0736584521001617> (cit. on p. 154).
- [Cas+05] Núria Casellas, Mercedes Blázquez, Atanas Kiryakov, Pompeu Casanovas, Marta Poblet, and Richard Benjamins. “OPJK into PROTON: Legal Domain Ontology Integration into an Upper-Level Ontology.” In: *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*. Ed. by Robert Meersman, Zahir Tari, and Pilar Herrero. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 846–855. ISBN: 978-3-540-32132-3 (cit. on p. 152).

- [Cas17] García Castro. “D2.2 Detailed Specification of the Semantic Model - Vicinity Project.” In: *Tech. Rep. Universidad Politécnica de Madrid (UPM)* (2017) (cit. on p. 144).
- [CFG03] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. “Methodologies, Tools and Languages for Building Ontologies. Where Is Their Meeting Point?” In: *Data & Knowledge Engineering* 46.1 (2003), pp. 41–64. ISSN: 0169-023X. DOI: [10.1016/S0169-023X\(02\)00195-7](https://doi.org/10.1016/S0169-023X(02)00195-7). URL: <https://www.sciencedirect.com/science/article/pii/S0169023X02001957> (cit. on p. 140).
- [Clo14] Cloud Native Computing Foundation. *Kubernetes (K8s) - Production-Grade Container Scheduling and Management*. Accessed on 2024-05-23. 2014. URL: <https://kubernetes.io/> (cit. on p. 87).
- [CM13] Andrew Crapo and Abha Moitra. “Toward a Unified English-like Representation of Semantic Models, Data, and Graph Patterns for Subject Matter Experts.” In: *International Journal of Semantic Computing* 7.03 (2013), pp. 215–236. DOI: [10.1142/S1793351X135000250](https://doi.org/10.1142/S1793351X135000250) (cit. on p. 160).
- [CMT11] European Commission, Directorate-General for Mobility, and Transport. *Intelligent transport systems in action : Action plan and legal framework for the deployment of intelligent transport systems (ITS) in Europe*. Publications Office, 2011. DOI: [doi/10.2832/44199](https://doi.org/10.2832/44199). URL: https://transport.ec.europa.eu/transport-themes/intelligent-transport-systems/road/action-plan-and-directive_en (cit. on p. 28).
- [Col+16] Diego Collarana, Mikhail Galkin, Christoph Lange, Irlán Grangel-González, Maria-Esther Vidal, and Sören Auer. “Fuhsen: A federated hybrid search engine for building a knowledge graph on-demand (short paper).” In: *On the Move to Meaningful Internet Systems: OTM 2016 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2016, Rhodes, Greece, October 24-28, 2016, Proceedings*. Springer. 2016, pp. 752–761 (cit. on pp. 194, 197).
- [Col+17] Diego Collarana, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, and Sören Auer. “Semantic data integration for knowledge graph construction at query time.” In: *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. IEEE. 2017, pp. 109–116 (cit. on pp. 194, 197).
- [Com+12] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. “The SSN Ontology of the W3C Semantic Sensor Network Incubator Group.” In: *Journal of Web Semantics* 17 (2012), pp. 25–32. DOI: [10.1016/j.websem.2012.05.003](https://doi.org/10.1016/j.websem.2012.05.003) (cit. on p. 99).
- [COM21] COMPOSITION Consortium. *D6.8 Collaborative Manufacturing Services Ontology and Language II*. Accessed on 2024-05-23. 2021. URL: https://www.composition-project.eu/wp-content/uploads/2019/03/D6.8_Collaborative_Manufacturing_Services_Ontology_and_Language_II.pdf (cit. on p. 154).

Bibliography

- [Com22a] Lina Teresa Molinas Comet. “Establishment of a Guideline for the Intuitive Creation of Semantic Models in the Internet of Production (IoP).” Advisors: Iraklis Dimitriadis, Patrick Sapel, and Johannes Theissen-Lipp. Master’s thesis. RWTH Aachen University, 2022 (cit. on pp. 138, 139, 141–157).
- [Com22b] Lina Teresa Molinas Comet. *Guidelines for the Creation of Semantic Models in the Internet of Production (IoP)*. Tech. rep. Appendix of the master thesis entitled “Establishment of a Guideline for the Intuitive Creation of Semantic Models in the Internet of Production (IoP)”, to be published. Aachen, Germany: RWTH Aachen University, 2022 (cit. on pp. 152, 156).
- [Cou+21] H. Cousijn, R. Braukmann, M. Fenner, C. Ferguson, R. van Horik, R. Lammey, A. Meadows, and S. Lambert. “Connected Research: The Potential of the PID Graph.” In: *Patterns (New York, N.Y.)* 2.1 (2021), p. 100180. DOI: [10.1016/j.patter.2020.100180](https://doi.org/10.1016/j.patter.2020.100180) (cit. on p. 23).
- [Cox+17] Simon Cox, Chris Little, Jerry Hobbs, and Feng Pan. *Time Ontology in OWL*. W3C Candidate Recommendation Draft. Accessed on 2024-05-23. Spatial Data on the Web Working Group, 2017. URL: <https://www.w3.org/TR/owl-time/> (cit. on pp. 56, 152).
- [CQ11] Kevin Crowston and Jian Qin. “A capability maturity model for scientific data management: Evidence from the literature.” In: *Proceedings of the American Society for Information Science and Technology* 48.1 (2011), pp. 1–9. DOI: [10.1002/meet.2011.14504801036](https://doi.org/10.1002/meet.2011.14504801036) (cit. on pp. 33, 36).
- [CRI16] European Commission, Directorate-General for Research, and Innovation. *Realising the European open science cloud : first report and recommendations of the Commission high level expert group on the European open science cloud*. Brussels, Belgium: Publications Office, 2016. DOI: [10.2777/940154](https://doi.org/10.2777/940154) (cit. on pp. 8, 9).
- [CRT14] Richard Cyganiak, Dave Reynolds, and Jeni Tennison. *The RDF Data Cube Vocabulary*. W3C Recommendation. Accessed on 2024-05-23. Government Linked Data Working Group, 2014. URL: <https://www.w3.org/TR/vocab-data-cube/> (cit. on pp. 56, 152).
- [CS14] Gavin Carothers and Andy Seaborne. *RDF 1.1 N-Triples*. W3C Working Group Note. Accessed on 2024-05-23. W3C Working Group, 2014. URL: <https://www.w3.org/TR/n-triples/> (cit. on p. 13).
- [Cul22] Richard Culatta. *ADDIE Model*. Accessed on 2024-05-23. 2022. URL: <https://www.instructionaldesign.org/models/addie/> (cit. on p. 142).
- [Cur20] E. Curry. “Future Research Directions for Dataspaces, Data Ecosystems, and Intelligent Systems.” In: *Real-time Linked Dataspaces: Enabling Data Ecosystems for Intelligent Systems*. Cham: Springer International Publishing, 2020, pp. 297–304. ISBN: 978-3-030-29665-0. DOI: [10.1007/978-3-030-29665-0_18](https://doi.org/10.1007/978-3-030-29665-0_18) (cit. on pp. 3, 10).
- [Cyc01] Cycorp, Inc. *OpenCyc*. Accessed on 2017-10-26, now discontinued. 2001. URL: <http://www.opencyc.org> (cit. on p. 195).
- [CYY10] Yixi Chen, Keting Yin, and Xiaohu Yang. “A semantic-aware algorithm to rank concepts in an OWL ontology graph.” In: *Information Technology Journal* 9.4 (2010), pp. 825–831 (cit. on p. 166).

- [Dah11] Nikolai Dahlem. “OntoClippy: A User-Friendly Ontology Design and Creation Methodology.” In: *Int. J. Intell. Inf. Technol.* 7.1 (Jan. 2011), pp. 15–32. ISSN: 1548-3657. DOI: [10.4018/jiit.2011010102](https://doi.org/10.4018/jiit.2011010102) (cit. on p. 144).
- [Dai+02] Leslie Daigle, D van Gulik, Renato Iannella, and Patrik Faltstrom. *Uniform Resource Names (URN) Namespace Definition Mechanisms*. Tech. rep. Accessed on 2024-05-23. IETF Network Working Group, 2002. URL: <https://www.ietf.org/rfc/rfc3406.txt> (cit. on p. 18).
- [Dal+20] Manuela Dalibor, Judith Michael, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. “Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits.” In: *Conceptual Modeling*. Springer, 2020, pp. 377–387. DOI: [10.1007/978-3-030-62522-1_280](https://doi.org/10.1007/978-3-030-62522-1_280) (cit. on p. 59).
- [dat20] data.world, Inc. *Grafo Toolkit*. 2020. URL: <https://grafo.io/> (cit. on p. 160).
- [Dat24] Data Space Support Centre. *Data Spaces Blueprint v1.0*. Blueprint. Accessed on 2024-05-23. 2024. URL: <https://dssc.eu/space/BVE/357073006/Data+Spaces+Blueprint+v1.0> (cit. on p. 8).
- [DB+16] Luigi Di Caro, Guido Boella, et al. “Automatic enrichment of wordnet with common-sense knowledge.” In: *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*. European Language Resources Association (ELRA). 2016, pp. 819–822 (cit. on pp. 194, 196).
- [DBp22] DBpedia Association. *DBpedia Ontology (DBO)*. Accessed on 2024-05-23. 2022. URL: <https://www.dbpedia.org/resources/ontology/> (cit. on p. 152).
- [DCM15] Biswanath Dutta, Usashi Chatterjee, and Devika P Madalli. “YAMO: Yet Another Methodology for Large-scale Faceted Ontology Construction.” In: *Journal of Knowledge Management* (2015) (cit. on p. 144).
- [DD09] Tore Dyba and Torgeir Dingsoyr. “What Do We Know about Agile Software Development?” In: *IEEE Software* 26.5 (2009), pp. 6–9. DOI: [10.1109/MS.2009.145](https://doi.org/10.1109/MS.2009.145) (cit. on p. 140).
- [De +20] Ben De Meester, Tom Seymoens, Anastasia Dimou, and Ruben Verborgh. “Implementation-independent function reuse.” In: *Future Generation Computer Systems* 110 (2020), pp. 946–959. ISSN: 0167-739X. DOI: [10.1016/j.future.2019.10.006](https://doi.org/10.1016/j.future.2019.10.006) (cit. on p. 120).
- [Dec+00] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. “The Semantic Web: the roles of XML and RDF.” In: *IEEE Internet Computing* 4.5 (2000), pp. 63–73. DOI: [10.1109/4236.877487](https://doi.org/10.1109/4236.877487) (cit. on p. 13).
- [Dec+04] Stefan Decker, Michael Sintek, Andreas Billig, Nicola Henze, Andreas Harth, Andreas Leicher, Miklos Zoltan, Jose-Luis Ambite, and Gustaf Neumann. “Triple - an RDF Rule Language with Context and Use Cases.” In: *W3C Workshop on Rule Languages for Interoperability, 27-28 April 2005, Washington, DC, USA*. W3C, 2004. URL: <https://www.w3.org/2004/12/rules-ws/paper/98/> (cit. on p. 14).
- [Deu+99] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. “A query language for XML.” In: *Computer networks* 31.11-16 (1999), pp. 1155–1169 (cit. on p. 14).

- [DFG23] DFG, German Research Foundation. *AI and ChatGPT in Science and the Humanities – DFG Formulates Guidelines for Dealing with Generative Models for Text and Image Creation*. Accessed: 13.05.2024. Sept. 2023. URL: <https://www.dfg.de/en/service/press/press-releases/2023/press-release-no-39> (cit. on p. 226).
- [DFH11] John Domingue, Dieter Fensel, and James A. Hendler. *Handbook of Semantic Web Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN: 978-3-540-92912-3. DOI: [10.1007/978-3-540-92913-0](https://doi.org/10.1007/978-3-540-92913-0) (cit. on p. 12).
- [DG 20a] DG DIGIT. *SHACL shape validator*. Accessed: 2024-04-23. 2020. URL: <https://www.itb.ec.europa.eu/shacl/shacl/upload> (cit. on p. 211).
- [DG 20b] DG DIGIT. *SHACL validator*. Accessed: 2024-04-23. 2020. URL: <https://www.itb.ec.europa.eu/shacl/any/upload> (cit. on p. 211).
- [DHH11] Stefan Decker, Siegfried Handschuh, and Manfred Hauswirth. “Towards Networked Knowledge.” In: *Foundations for the Web of Information and Services: A Review of 20 Years of Semantic Web Research*. Ed. by Dieter Fensel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 155–174. ISBN: 978-3-642-19797-0. DOI: [10.1007/978-3-642-19797-0_9](https://doi.org/10.1007/978-3-642-19797-0_9) (cit. on p. 50).
- [Dim+13] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Erik Mannens, and Rik Van de Walle. “Extending R2RML to a Source-independent Mapping Language for RDF.” In: *International Semantic Web Conference (Posters & Demos)*. Vol. 1035. 2013, pp. 237–240 (cit. on pp. 17, 115).
- [Dim+14] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. “RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data.” In: *Ldow*. 2014 (cit. on pp. 17, 23, 115).
- [Dim+20] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. “RDF Mapping Language (RML).” In: *W3C, Unofficial Draft 15* (2020) (cit. on pp. 17, 115).
- [DM16] Antonio De Nicola and Michele Missikoff. “A lightweight methodology for rapid ontology engineering.” In: *Communications of the ACM* 59.3 (2016), pp. 79–86 (cit. on p. 144).
- [DMN05] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. “A Proposal for a Unified Process for Ontology Building: UPON.” In: *Database and Expert Systems Applications*. Ed. by Kim Viborg Andersen, John Debenham, and Roland Wagner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 655–664. ISBN: 978-3-540-31729-6 (cit. on p. 144).
- [Doa+02] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. “Learning to map between ontologies on the semantic web.” In: *Proceedings of the 11th international conference on World Wide Web*. 2002, pp. 662–673 (cit. on p. 198).
- [Doc20] Simulink Documentation. *Simulation and Model-Based Design*. 2020. URL: <https://www.mathworks.com/products/simulink.html> (cit. on p. 59).

- [Dre+21] Holger Drees, Dennis Oliver Kubitza, Johannes Lipp, Sebastian Pretzsch, and Christoph Schlueter Langdon. “Mobility Data Space—First Implementation and Business Opportunities.” In: *29th ITS World Congress*. Contributions: HD and SP designed and conceived the basic work. JL, DK, and SP performed, analyzed, and interpreted the work. HD, DK, JL, SP, and CL drafted and revised the paper. Hamburg, Germany: ERTICO – ITS Europe, Oct. 2021, pp. 1–12 (cit. on pp. 7, 27, 28, 187).
- [DSC12] Souripriya Das, Seema Sundara, and Richard Cyganiak. *R2RML: RDB to RDF Mapping Language*. Tech. rep. Accessed on 2024-05-23. W3C, 2012. URL: <https://www.w3.org/TR/r2rml/> (cit. on pp. 17, 23, 115, 118).
- [DSE21] Michael Diepenbroek, Sonja Schimmler, and Barbara Ebert. *Sektionskonzept Common Infrastructures zur Einrichtung einer Sektion im Verein Nationale Forschungsdateninfrastruktur (NFDI) e.V.* Oct. 2021. DOI: [10.5281/zenodo.5607490](https://doi.org/10.5281/zenodo.5607490) (cit. on p. 47).
- [Du+12] Wen-Hau Du, Jer-Wei Rau, Jen-Wei Huang, and Yung-Sheng Chen. “Improving the quality of tags using state transition on progressive image search and recommendation system.” In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2012, pp. 3233–3238. DOI: [10.1109/ICSMC.2012.6378289](https://doi.org/10.1109/ICSMC.2012.6378289) (cit. on pp. 165, 194, 197).
- [ECF20] Boulos El Asmar, Syrine Chelly, and Michael Färber. *AWARE: An Ontology for Situational Awareness of Autonomous Vehicles in Manufacturing*. 2020 (cit. on p. 154).
- [Ecl16] Eclipse Foundation, Inc. *The Eclipse RDF4J Framework*. 2016. URL: <https://rdf4j.org/> (cit. on p. 20).
- [ELE07] Mohamad Eid, Ramiro Liscano, and Abdulmotaleb El Saddik. “A Universal Ontology for Sensor Networks Data.” In: *2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. IEEE. 2007, pp. 59–62. DOI: [10.1109/CIMSA.2007.43625390](https://doi.org/10.1109/CIMSA.2007.43625390) (cit. on pp. 99, 100).
- [Ell+01] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. “Graphviz — Open Source Graph Drawing Tools.” In: *International Symposium on Graph Drawing*. Springer. 2001, pp. 483–484. DOI: [10.1007/3-540-45848-4_570](https://doi.org/10.1007/3-540-45848-4_570) (cit. on p. 160).
- [EOS22] EOSC Association AISBL. *Strategic Research and Innovation Agenda of the European Open Science Cloud*. Tech. rep. Rue de Luxembourg 3, BE-1000 Brussels, Belgium, Nov. 2022. DOI: [10.2777/935288](https://doi.org/10.2777/935288). URL: <https://eosc.eu/sites/default/files/SRIA%201.1%20final.pdf> (cit. on pp. 8, 9, 31).
- [EU 20] EU member states. *Towards a next generation cloud for Europe - Joint Declaration on Cloud*. Tech. rep. Brussels, Belgium: The European Commission and the Council of the European Union, Oct. 2020. URL: <https://digital-strategy.ec.europa.eu/en/news/towards-next-generation-cloud-europe> (cit. on pp. 7, 10).
- [Eur20a] European Commission. *A European Strategy for Data*. Tech. rep. Feb. 2020. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0066> (cit. on pp. 3, 8–10, 23).

Bibliography

- [Eur20b] European Commission. *A European Strategy for Data* COM/2020/66. Tech. rep. 2020 (cit. on p. 27).
- [Eur20c] European Commission (EC). *Sustainable and Smart Mobility Strategy—Putting European Transport on Track for the Future*. 2020 (cit. on p. 28).
- [Eur22] European Commission. *Staff Working Document on Common European Data Spaces*. Tech. rep. Brussels, Belgium, Mar. 2022 (cit. on pp. 8–10).
- [Eyn13] Veerle van den Eynden. *Data Life Cycle & Data Management Planning*. Tech. rep. Accessed on 2024-05-23. Colchester CO4 3SQ, UK: UK Data Archive, University Of Essex, Apr. 2013. URL: <https://dam.ukdataservice.ac.uk/media/187718/dmplanningdm24apr2013.pdf> (cit. on pp. 33, 43).
- [Fär+15] Michael Färber, Basil Ell, Carsten Menne, and Achim Rettinger. “A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago.” In: *Semantic Web Journal* 1.1 (2015), pp. 1–5 (cit. on p. 168).
- [Fen+10] Dieter Fensel, Florian Fischer, Jacek Kopecký, Reto Krummenacher, Dave Lambert, and Tomas Vitvar. *WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web*. 2010 (cit. on p. 100).
- [Fen+11] Dieter Fensel, Federico Michele Facca, Elena Simperl, and Ioan Toma. “Lightweight Semantic Web Service Descriptions.” In: *Semantic Web Services*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 279–295. ISBN: 978-3-642-19193-0. DOI: [10.1007/978-3-642-19193-0_12](https://doi.org/10.1007/978-3-642-19193-0_12) (cit. on p. 100).
- [FGJ97] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. “Methontology: From Ontological Art Towards Ontological Engineering.” In: *AAAI-97 Spring Symposium Series* (1997) (cit. on p. 144).
- [FGP21] Serge Chávez Feria, Raúl García-Castro, and María Poveda-Villalón. “Converting UML-based ontology conceptualizations to OWL with Chowlk.” In: *ESWC2021 Poster and Demo Track*. 2021. URL: <https://openreview.net/forum?id=u1Vp2y-QE1> (cit. on p. 117).
- [FHH04] Richard Fikes, Patrick Hayes, and Ian Horrocks. “OWL-QL—a language for deductive query answering on the Semantic Web.” In: *Journal of Web Semantics* 2.1 (2004), pp. 19–29 (cit. on p. 14).
- [FHM05] M. Franklin, A. Halevy, and D. Maier. “From databases to dataspace: a new abstraction for information management.” In: *ACM Sigmod Record* 34.4 (2005), pp. 27–33. DOI: [10.1145/1107499.1107502](https://doi.org/10.1145/1107499.1107502) (cit. on pp. 3, 10).
- [Föd13] Föderale IT-Kooperation. *GOVData*. Accessed: 2024-05-23. 2013. URL: <https://govdata.de> (cit. on p. 26).
- [Fra19] Fraunhofer Institute for Transportation and Infrastructure Systems IVI. *Mobility Data Space: Sovereign Mobility Data Ecosystem*. Accessed on 2024-05-23. Dec. 2019. URL: <https://www.mobility-data-space.de/en.html> (cit. on pp. 28, 187).
- [Fra20] Thomas Francart. *SHACL Play! Online RDF data validation with SHACL*. Tech. rep. <https://shacl-play.sparna.fr/play/>. Sparna, 2020. URL: <https://shacl-play.sparna.fr/play/> (cit. on p. 20).
- [Fra22] Fraunhofer-Gesellschaft. *Data Spaces Support Centre*. Accessed on 2024-05-23. 2022. URL: <https://dssc.eu/> (cit. on p. 8).

- [FW12] Andisheh Feizi and Chui Yin Wong. “Usability of user interface styles for learning a graphical software application.” In: *2012 International Conference on Computer & Information Science (ICCIS)*. Vol. 2. IEEE. 2012, pp. 1089–1094 (cit. on p. 158).
- [Gab10] Evgeniy Gabrilovich. *Wikipedia preprocessor (wikiprep)*. Accessed on 2017-10-27, now offline. 2010. URL: <http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/> (cit. on p. 195).
- [GAI22] GAIA-X European Association for Data and Cloud. *Gaia-X Architecture Document*. Tech. rep. Accessed on 2024-05-23. Avenue des Arts 6-9, 1210 Brussels: Gaia-X European Association for Data and Cloud AISBL, Oct. 2022. URL: <https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/> (cit. on pp. 7, 9, 31, 32, 46).
- [Gao+12] Shudi Sandy Gao, C. Michael Sperberg-McQueen, Henry S. Thompson, Noah Mendelsohn, David Beech, and Murray Maloney. *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2012. URL: <https://www.w3.org/TR/xmlschema11-1/> (cit. on p. 13).
- [GAP12] Fausto Giunchiglia, Aliaksandr Autayeu, and Juan Pane. “S-Match: an open source framework for matching lightweight ontologies.” In: *Semantic Web 3.3* (2012), pp. 307–317 (cit. on pp. 194, 198).
- [Gar17] Daniel Garijo. “WIDOCO: a wizard for documenting ontologies.” In: *International Semantic Web Conference*. Springer, Cham. 2017, pp. 94–102. DOI: [10.5281/zenodo.591294](https://doi.org/10.5281/zenodo.591294) (cit. on p. 20).
- [GAT95] GATE research team. *General Architecture for Text Engineering (GATE)*. 1995. URL: <https://gate.ac.uk/> (cit. on p. 20).
- [Gay+18] Jose Emilio Labra Gayo, Eric Prud’hommeaux, Iovka Boneva, and Dimitris Kontokostas. *Validating RDF Data*. Springer International Publishing, 2018. ISBN: 9783031794780. DOI: [10.1007/978-3-031-79478-0](https://doi.org/10.1007/978-3-031-79478-0) (cit. on p. 210).
- [GBM15] R. V. Guha, Dan Brickley, and Steve MacBeth. “Schema.Org: Evolution of Structured Data on the Web: Big Data Makes Common Schemas Even More Necessary.” In: *Queue* 13.9 (Nov. 2015), pp. 10–37. ISSN: 1542-7730. DOI: [10.1145/2857274.2857276](https://doi.org/10.1145/2857274.2857276) (cit. on p. 22).
- [GF94] Michael Gruninger and Mark S Fox. “The Design and Evaluation of Ontologies for Enterprise Engineering.” In: *Workshop on Implemented Ontologies, European Conference on Artificial Intelligence (ECAI)*. 1994 (cit. on p. 144).
- [GFB22] GFBio - Gesellschaft für Biologische Daten e.V. *GFBio Data Life Cycle*. Accessed on 2024-05-23. 2022. URL: <https://www.gfbio.org/training/material/data-life-cycle/> (cit. on pp. 33, 40, 49).
- [GKL12] Aman Goel, Craig A Knoblock, and Kristina Lerman. “Exploiting structure within data for accurate labeling using conditional random fields.” In: *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer ... 2012, p. 1 (cit. on pp. 19, 165).

Bibliography

- [GML21] Sara Garavelli, Anu Märkälä, and Iris Liinamaa. *EOSC National Structures: an overview of the national EOSC coordination and engagement mechanisms in Europe*. Nov. 2021. DOI: [10.5281/zenodo.5668275](https://doi.org/10.5281/zenodo.5668275) (cit. on p. 47).
- [Gön+19] Philipp Gönnheimer, Jonas Hillenbrand, Thomas Betz-Mors, Philip Bischof, Lorenz Mohr, and Jürgen Fleischer. “Auto-configuration of a Digital Twin for Machine Tools by Intelligent Crawling.” In: *Production at the Leading Edge of Technology*. Springer, 2019, pp. 543–552. ISBN: 978-3-662-60416-8. DOI: [10.1007/978-3-662-60417-5_54](https://doi.org/10.1007/978-3-662-60417-5_54) (cit. on p. 66).
- [Goo15a] Google. *Bazel*. 2015. URL: <https://bazel.build/> (cit. on p. 96).
- [Goo15b] Google. *Protocol Buffers*. 2015. URL: <https://protobuf.dev/> (cit. on pp. 96, 100).
- [Goo16] Google. *gRPC*. 2016. URL: <https://grpc.io/> (cit. on pp. 96, 100).
- [Göp+21] Amon Göppert, Lea Grahn, Jonas Rachner, Dennis Grunert, Simon Hort, and Robert H Schmitt. “Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems.” In: *Journal of Intelligent Manufacturing* (2021), pp. 1–20 (cit. on p. 154).
- [GP09] Aldo Gangemi and Valentina Presutti. “Ontology Design Patterns.” In: *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 221–243. ISBN: 978-3-540-92673-3. DOI: [10.1007/978-3-540-92673-3_10](https://doi.org/10.1007/978-3-540-92673-3_10) (cit. on pp. 137, 139, 140, 143).
- [GPS99] Aldo Gangemi, Domenico M Pisanelli, and Geri Steve. “An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies.” In: *Data & Knowledge Engineering* 31.2 (1999), pp. 183–220 (cit. on p. 144).
- [GR12] Julián Garrido and Ignacio Requena. “Towards Summarizing Knowledge: Brief Ontologies.” In: *Expert Systems with Applications* 39.3 (2012), pp. 3213–3222 (cit. on p. 143).
- [Gra+17] Irlán Grangel-González, Paul Baptista, Lavdim Halilaj, Steffen Lohmann, Maria-Esther Vidal, Christian Mader, and Sören Auer. “The industry 4.0 Standards Landscape from a Semantic Integration Perspective.” In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2017, pp. 1–8. DOI: [10.1109/ETFA.2017.8247584](https://doi.org/10.1109/ETFA.2017.8247584) (cit. on p. 154).
- [Gro04] Structural Reform Group. *DDI VERSION 3.0 CONCEPTUAL MODEL*. Accessed on 2024-05-23. June 2004. URL: <https://ddialliance.org/sites/default/files/Concept-Model-WD.pdf> (cit. on pp. 33–35).
- [Gru+93] Thomas R Gruber et al. “A translation approach to portable ontology specifications.” In: *Knowledge acquisition* 5.2 (1993), pp. 199–220 (cit. on pp. 18, 129, 131).
- [GSA15] A. Gyrard, M. Serrano, and G. A. Atemezing. “Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things.” In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 2015, pp. 412–417. DOI: [10.1109/WF-IoT.2015.73890900](https://doi.org/10.1109/WF-IoT.2015.73890900) (cit. on p. 98).

- [Gui+22] Giancarlo Guizzardi, Alessander Botti Benevides, Claudenir M Fonseca, Daniele Porello, João Paulo A Almeida, and Tiago Prince Sales. “UFO: Unified foundational ontology.” In: *Applied ontology* Preprint (2022), pp. 1–44 (cit. on p. 152).
- [Gui21] Guinard, Anne. *A survey of Top-Level Ontologies 1.0.0*. Accessed on 2022-06-21. 2021. URL: <https://digitaltwinhub.co.uk/files/file/90-a-survey-of-top-level-ontologies> (cit. on p. 152).
- [Gup+12] Shubham Gupta, Pedro Szekely, Craig A Knoblock, Aman Goel, Mohsen Taheriyani, and Maria Muslea. “Karma: A system for mapping structured sources into the semantic web.” In: *Extended Semantic Web Conference*. Springer. 2012, pp. 430–434 (cit. on pp. 194, 195).
- [GW09] Maiga Gilbert and Ddembe Williams. “A flexible biomedical ontology selection tool.” In: *Strengthening the Role of ICT in Development* (Jan. 2009) (cit. on pp. 166, 167).
- [Gyr+15] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. “Integrating Machine-to-Machine Measurement Framework into oneM2M Architecture.” In: *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE. 2015, pp. 364–367 (cit. on p. 154).
- [Haa+08] Peter Haase, Holger Lewen, Rudi Studer, Duc Thanh Tran, Michael Erdmann, Mathieu d’Aquin, and Enrico Motta. “The NeOn Ontology Engineering Toolkit.” In: *WWW 2008 Developers Track*. 17th International World Wide Web Conference. WWW 2008 (Peking, China, Apr. 21–25, 2008). 2008 (cit. on p. 160).
- [Hal+16] Lavdim Halilaj, Niklas Petersen, Irlán Grangel-González, Christoph Lange, Sören Auer, Gökhan Coskun, and Steffen Lohmann. “Vocol: An integrated environment to support version-controlled vocabulary development.” In: *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*. Springer. 2016, pp. 303–319 (cit. on p. 54).
- [Ham20] Karl Hammar. “Linked Data Creation with ExcelRDF.” In: *ESWC*. Springer. 2020, pp. 104–109. DOI: [10.1007/978-3-030-62327-2_18](https://doi.org/10.1007/978-3-030-62327-2_18) (cit. on p. 160).
- [Har16] Thomas Hartmann. “Validation framework for rdf-based constraint languages.” PhD thesis. Karlsruhe Institute of Technology, Germany, 2016 (cit. on p. 211).
- [HB11a] Tom Heath and Christian Bizer. “Linked Data: Evolving the Web into a Global Data Space.” In: *Synthesis Lectures on the Semantic Web: Theory and Technology* 1.1 (2011), pp. 1–136. ISSN: 2160-4711. DOI: [10.2200/S00334ED1V01Y201102WBE001](https://doi.org/10.2200/S00334ED1V01Y201102WBE001) (cit. on pp. 10, 22).
- [HB11b] Matthew Horridge and Sean Bechhofer. “The OWL API: A Java API for OWL Ontologies.” In: *Semant. Web* 2.1 (Jan. 2011), pp. 11–21. ISSN: 1570-0844 (cit. on p. 20).
- [HCM09] J Steven Hughes, Daniel J Crichton, and Chris A Mattmann. “Ontology-based information model development for science information reuse and integration.” In: *2009 IEEE International Conference on Information Reuse & Integration*. IEEE. 2009, pp. 79–84 (cit. on pp. 194, 198).

Bibliography

- [He+18] Yongqun He, Zuoshuang Xiang, Jie Zheng, Yu Lin, James A Overton, and Edison Ong. “The eXtensible ontology development (XOD) principles and tool implementation to support ontology interoperability.” In: *Journal of biomedical semantics* 9.1 (2018), pp. 1–10 (cit. on p. 144).
- [Her10] Heinrich Herre. “General Formal Ontology (GFO): A foundational ontology for conceptual modelling.” In: *Theory and applications of ontology: computer applications*. Springer, 2010, pp. 297–345 (cit. on p. 152).
- [HFM06] A. Halevy, M. Franklin, and D. Maier. “Principles of Dataspace Systems.” In: *Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS ’06. Chicago, IL, USA: Association for Computing Machinery, 2006, pp. 1–9. ISBN: 1595933182. DOI: [10.1145/1142351.1142352](https://doi.org/10.1145/1142351.1142352) (cit. on pp. 3, 10, 31).
- [Hig08] Sarah Higgins. “The DCC Curation Lifecycle Model.” In: *International Journal of Digital Curation* 3.1 (Dec. 2008), pp. 134–140. DOI: [10.2218/ijdc.v3i1.48](https://doi.org/10.2218/ijdc.v3i1.48) (cit. on pp. 33, 38).
- [Hip+21] Jean-Laurent Hippolyte, Michael Chrubasik, Frédéric Brochu, and Maurizio Bevilacqua. “A Domain-agnostic Ontology for Unified Metrology Data Management.” In: *Measurement: Sensors* 18 (2021), p. 100263. ISSN: 2665-9174. DOI: [10.1016/j.measen.2021.100263](https://doi.org/10.1016/j.measen.2021.100263). URL: <https://www.sciencedirect.com/science/article/pii/S2665917421002269> (cit. on p. 140).
- [Hit+08] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, and York Sure. *Semantic Web: Grundlagen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-33993-9. DOI: [10.1007/978-3-540-33994-6](https://doi.org/10.1007/978-3-540-33994-6) (cit. on p. 12).
- [Hit21] Pascal Hitzler. “A Review of the Semantic Web Field.” In: *Communications of the ACM* 64.2 (2021), pp. 76–83. ISSN: 0001-0782. DOI: [10.1145/3397512](https://doi.org/10.1145/3397512) (cit. on pp. 11, 18, 137).
- [Hoc22] Philipp Hochmann. “Erweiterung von semantischen RML Mappings am Beispiel von heterogenen Quelldaten.” Advisors: Johannes Theissen-Lipp and Niklas Schäfer. Master’s thesis. RWTH Aachen University, 2022 (cit. on pp. 118, 120, 122, 124).
- [Hog+21] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. De Melo, C. Gutierrez, S. Kirrane, J. E. Labra Gayo, R Navigli, S. Neumaier, A.-C. Ngonga Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. “Knowledge Graphs.” In: *ACM Computing Surveys* 54.4 (July 2021), pp. 1–37. DOI: [10.1145/3447772](https://doi.org/10.1145/3447772) (cit. on pp. 24, 140).
- [Hog20] Aidan Hogan. “The Semantic Web: Two Decades On.” In: *Semantic Web* 11.1 (2020), pp. 169–185 (cit. on p. 3).
- [Hor+04] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. 2004. URL: <https://www.w3.org/Submission/SWRL/> (cit. on p. 15).

- [Hor+21] Martin Horsch, Taras Petrenko, Volodymyr Kushnarenko, Bjoern Schembera, Bianca Wentzel, Alexander Behr, Norbert Kockmann, Sonja Schimmler, and Thomas Bönisch. “Interoperability and architecture requirements analysis and metadata standardization for a research data infrastructure in catalysis.” In: *International Conference on Data Analytics and Management in Data Intensive Domains*. Springer. 2021, pp. 166–177 (cit. on p. 47).
- [HPL15] H. K. Hong, K. W. Park, and D. H. Lee. “A Novel Semantic Tagging Technique Exploiting Wikipedia-Based Associated Words.” In: *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*. Vol. 3. 2015, pp. 648–649. DOI: [10.1109/COMPSAC.2015.87](https://doi.org/10.1109/COMPSAC.2015.87) (cit. on pp. 165, 194, 197).
- [HSK08] Tom Hannelius, Mikko Salmenpera, and Seppo Kuikka. “Roadmap to adopting OPC UA.” In: *2008 6th IEEE International Conference on Industrial Informatics*. IEEE. 2008, pp. 756–761. DOI: [10.1109/INDIN.2008.4618203](https://doi.org/10.1109/INDIN.2008.4618203) (cit. on pp. 26, 66, 79, 82).
- [HTC08] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. “Combining WordNet and ConceptNet for automatic query expansion: a learning approach.” In: *Information Retrieval Technology: 4th Asia Infomation Retrieval Symposium, AIRS 2008, Harbin, China, January 15-18, 2008 Revised Selected Papers 4*. Springer. 2008, pp. 213–224 (cit. on p. 196).
- [Hum05] Albert Humphrey. “SWOT analysis for management consulting.” In: *SRI alumni Newsletter* 1 (2005), pp. 7–8. URL: <https://alumni.sri.com/newsletters/2005/AlumNews-Dec-2005.pdf> (cit. on p. 135).
- [HV15] Ondřej Hrcuba and Pavel Vrba. “Ontologies for flexible production systems.” In: *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2015, pp. 1–8 (cit. on p. 154).
- [HWS21] N. Hartl, E. Wössner, and Y. Sure-Vetter. “Nationale Forschungsdateninfrastruktur (NFDI).” In: *Informatik Spektrum* 44.5 (2021), pp. 370–373. DOI: [10.1007/s00287-021-01392-6](https://doi.org/10.1007/s00287-021-01392-6) (cit. on pp. 8, 9).
- [Ian+18] R Iannella, M Steidl, S Myles, and V Rodriguez-Doncel. *ODRL Vocabulary & Expression 2.2: W3C Recommendation, 15 February 2018*. 2018. URL: <https://www.w3.org/TR/odrl-vocab/> (cit. on pp. 28, 55, 56).
- [Ide17] Kingsley Uyi Idehen. *Semantic Web Layer Cake Tweak, Explained*. Tech. rep. OpenLink Software, 2017. URL: <https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab> (cit. on pp. 11, 12).
- [IEC20] IEC TR 62541:2020. *OPC Unified Architecture*. Standard. Geneva, CH: International Electrotechnical Commission, 2020 (cit. on pp. 66, 67, 72).
- [Igl+20] Enrique Iglesias, Samaneh Jozashoori, David Chaves-Fraga, Diego Collarana, and Maria-Esther Vidal. “SDM-RDFizer: An RML interpreter for the efficient creation of rdf knowledge graphs.” In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 3039–3046 (cit. on p. 120).

Bibliography

- [Igl+22] Ana Iglesias-Molina, Julián Arenas-Guerrero, Thomas Delva, Anastasia Dimou, and David Chaves-Fraga. *RML-star*. Tech. rep. Accessed on 2024-05-23. W3C Knowledge Graph Construction Community Group, 2022. URL: <https://w3id.org/rml/star/spec> (cit. on pp. 17, 115).
- [IM14] Renato Iannella and James McKinney. *vCard Ontology – for describing People and Organizations*. W3C Interest Group Note. Accessed on 2024-05-23. Semantic Web Interest Group, 2014. URL: <https://www.w3.org/TR/vcard-rdf/> (cit. on p. 152).
- [Inf94] Information Technology Task Force. *ISO/IEC 7498-1:1994, Basic Reference Model: The Basic Model*. Standard. Genève, Switzerland: International Organization for Standardization, 1994 (cit. on p. 50).
- [iNo17] iNovex. *Mobi: Distributed Knowledge Graph Platform*. Accessed on 2024-05-23. 2017. URL: <https://mobi.solutions/> (cit. on pp. 159, 160).
- [inr03] inray Industriesoftware GmbH. *OPC Router*. Accessed on 2024-05-23. 2003. URL: <https://opcfoundation.org/products/view/opc-router> (cit. on p. 83).
- [INS17] Ali Ismail, Ahmed Nahar, and Raimar Scherer. *Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard*. July 2017 (cit. on p. 117).
- [Ins20] Institute of Data Science - Maastricht University. *Using Ontologies*. Accessed on 2024-05-23. 2020. URL: <https://maastrichtu-ids.github.io/best-practices/docs/using-ontologies/> (cit. on p. 139).
- [Int16a] International Data Spaces e. V. *International Data Spaces*. Accessed on 2024-05-23. 2016. URL: <https://www.internationaldataspaces.org/> (cit. on pp. 28, 46).
- [Int16b] International Organization for Standardization. *DIN SPEC 91345:2016-04, Reference Architecture Model Industrie 4.0 (RAMI4.0)*. Berlin, 2016 (cit. on pp. 59, 140).
- [Int17] International Organization for Standardization. *DIN ISO 55000:2017-05, Asset management - Overview, principles and terminology*. Standard. Berlin: Beuth Verlag GmbH, 2017. DOI: [10.31030/2647200](https://doi.org/10.31030/2647200) (cit. on p. 59).
- [Int18] International Organization for Standardization. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*. Standard. Geneva, CH: International Organization for Standardization, Nov. 2018 (cit. on p. 116).
- [Int23] International Data Spaces Association. *Reference Architecture Model for the Industrial Data Space v4.2.0*. Accessed on 2024-05-23. Mar. 2023. URL: <https://docs.internationaldataspaces.org/ids-ram-4/> (cit. on pp. 7, 9, 28, 31, 45).
- [Iva+12] Ivan Ivanov, Peter Vajda, Jong-Seok Lee, and Touradj Ebrahimi. “In tags we trust: Trust modeling in social tagging of multimedia content.” In: *IEEE Signal Processing Magazine* 29.2 (2012), pp. 98–107 (cit. on p. 197).
- [JA06] Matthew Jones and Harith Alani. “Content-based ontology ranking.” In: *9th International Protégé Conference* (2006) (cit. on p. 166).
- [Jac08] Paul Jaccard. “Nouvelles recherches sur la distribution florale.” In: *Bull. Soc. Vaud. Sci. Nat.* 44 (1908), pp. 223–270 (cit. on p. 205).

- [JBV98] Dean Jones, Trevor Bench-Capon, and Pepijn Visser. “Methodologies for Ontology Development.” In: *IT & KNOWS information technologies and knowledge systems (Vienna, Budapest, 31 August-4 September 1998)*. 1998, pp. 62–75 (cit. on pp. 139, 140, 143, 144).
- [JHM04] Wesley M Johnston, JR Paul Hanna, and Richard J Millar. “Advances in dataflow programming languages.” In: *ACM computing surveys (CSUR)* 36.1 (2004), pp. 1–34. DOI: [10.1145/1013208.1013209](https://doi.org/10.1145/1013208.1013209) (cit. on p. 83).
- [JMK15] Jason Jaskolka, Wendy MacCaull, and Ridha Khedri. “Towards an Ontology Design Architecture.” In: *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2015, pp. 132–135 (cit. on p. 139).
- [JSO22] JSON Schema. *JSON Schema*. 2022. URL: <https://json-schema.org/> (cit. on p. 15).
- [JSS16] Santhosh John, Nazaraf Shah, and Leonid Smalov. “Incremental and iterative agile methodology (IIAM): hybrid approach for ontology design towards semantic web based educational systems development.” In: *International Journal of Knowledge Engineering* 2.1 (2016), pp. 13–19 (cit. on p. 144).
- [Käf+13] Tobias Käfer, Ahmed Abdelrahman, Jürgen Umbrich, Patrick O’Byrne, and Aidan Hogan. “Observing Linked Data Dynamics.” In: *The Semantic Web: Semantics and Big Data*. Ed. by David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Philipp Ciminiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph. Vol. 7882. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 213–227. ISBN: 978-3-642-38287-1. DOI: [10.1007/978-3-642-38288-8_15](https://doi.org/10.1007/978-3-642-38288-8_15) (cit. on pp. 18, 22).
- [Kai+21] Marco Kainzner, Christoph Klösch, Dominik Filipiak, Tek Raj Chhetri, and Anna Fensel. “Poster: Towards Reusable Ontology Alignment for Manufacturing Maintenance.” In: *Workshop on Ontology-Driven Conceptual Modelling of Digital Twins*. Amsterdam, the Netherlands: SEMANTiCS 2021 EU, Sept. 2021. URL: <http://ceur-ws.org/Vol-2941/paper9.pdf> (cit. on p. 154).
- [Kar+03] Gregory Karvounarakis, Aimilia Magganaraki, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, Michel Scholl, and Karsten Tolle. “Querying the semantic web with RQL.” In: *Computer networks* 42.5 (2003), pp. 617–640 (cit. on p. 14).
- [Kar+19] Mohamed Hedi Karray, Farhad Ameri, Melinda Hodkiewicz, and Thierry Louge. “ROMAIN: Towards a BFO Compliant Reference Ontology for Industrial Maintenance.” In: *Applied Ontology* 14.2 (2019), pp. 155–177 (cit. on p. 154).
- [Kay07] Michael Kay. *XSL Transformations (XSLT) Version 2.0*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2007. URL: <https://www.w3.org/TR/xslt20/> (cit. on pp. 13, 14).

Bibliography

- [KB07] J. Kunze and T. Baker. *The Dublin Core Metadata Element Set*. Tech. rep. Accessed on 2024-05-23. IETF Network Working Group, 2007. URL: <https://www.ietf.org/rfc/rfc5013.txt> (cit. on pp. 56, 133, 152).
- [KB13] Michael Kifer and Harold Boley. *RIF Overview (Second Edition)*. W3C Working Group Note. Accessed on 2024-05-23. 2013. URL: <https://www.w3.org/TR/rif-overview/> (cit. on p. 15).
- [KB22] J. A. Kunze and E. Bermès. *The ARK Identifier Scheme*. Internet-Draft draft-kunze-ark-36. Work in Progress. Internet Engineering Task Force, Oct. 2022. 45 pp. URL: <https://datatracker.ietf.org/doc/draft-kunze-ark/36/> (cit. on pp. 18, 23).
- [KD12] Murat Kalender and Jiangbo Dang. “Skmt: A semantic knowledge management tool for content tagging, search and management.” In: *2012 Eighth International Conference on Semantics, Knowledge and Grids*. IEEE. 2012, pp. 112–119 (cit. on pp. 194, 195).
- [KH16] Adila Krisnadhi and Pascal Hitzler. “Modeling With Ontology Design Patterns: Chess Games As a Worked Example.” In: *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*. Vol. 25. IOS Press, 2016. Chap. 1, pp. 3–21. DOI: [10.3233/978-1-61499-676-7-3](https://doi.org/10.3233/978-1-61499-676-7-3) (cit. on p. 144).
- [KHI11] Holger Knublauch, James A. Hendler, and Kingsley Idehen. *SPIN - Overview and Motivation*. 2011. URL: <https://www.w3.org/Submission/spin-overview/> (cit. on pp. 15, 210).
- [Kim+08] H. L. Kim, A. Passant, J. G. Breslin, S. Scerri, and S. Decker. “Review and Alignment of Tag Ontologies for Semantically-Linked Data in Collaborative Tagging Spaces.” In: *Semantic Computing, 2008 IEEE International Conference on*. 2008, pp. 315–322. DOI: [10.1109/ICSC.2008.79](https://doi.org/10.1109/ICSC.2008.79) (cit. on p. 165).
- [Kje+09] T. Kjellberg, A. von Euler-Chelpin, M. Hedlind, M. Lundgren, G. Sivard, and D. Chen. “The Machine Tool Model — A Core Part of the Digital Factory.” In: *CIRP Annals* 58.1 (2009), pp. 425–428. ISSN: 0007-8506. DOI: [10.1016/j.cirp.2009.03.035](https://doi.org/10.1016/j.cirp.2009.03.035). URL: <https://www.sciencedirect.com/science/article/pii/S0007850609001073> (cit. on p. 152).
- [KK17] Holger Knublauch and Dimitris Kontokostas. *Shapes constraint language (SHACL)*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2017. URL: <https://www.w3.org/TR/shacl/> (cit. on pp. 15, 56, 133, 134, 209, 210).
- [KLW11] H. Kagermann, W.-D. Lukas, and W. Wahlster. “Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur vierten industriellen Revolution.” In: *VDI Nachrichten* 13 (2011) (cit. on p. 66).
- [Kno+12] Craig A Knoblock, Pedro Szekely, José Luis Ambite, Aman Goel, Shubham Gupta, Kristina Lerman, Maria Muslea, Mohsen Taheriyani, and Parag Mallick. “Semi-automatically mapping structured sources into the semantic web.” In: *Extended semantic web conference*. Springer. 2012, pp. 375–390 (cit. on p. 195).
- [Knu17a] Holger Knublauch. *From SPIN to SHACL*. Tech. rep. Aug. 2017. URL: <https://spinrdf.org/spin-shacl.html> (cit. on p. 210).

- [Knu17b] Holger Knublauch. *SHACL Playground*. Accessed: 2024-04-23. 2017. URL: <https://shacl.org/playground/> (cit. on p. 211).
- [Koc+21] Max Kocher, Johannes Theissen-Lipp, Patrick Sapel, and Mauritius Schmitz. *Transforming XPath Expressions to OPC UA (X2OPC)*. 2021. URL: <https://github.com/JohannesLipp/Transforming-XPath-to-OPC-UA> (cit. on pp. 69, 78).
- [Koc21] Max Kocher. “Verbesserter Datenzugriff in OPC UA durch Transformation von XQuery-Ausdrücken.” Advisors: Johannes Lipp and Patrick Sapel. Bachelor’s thesis. RWTH Aachen University, 2021 (cit. on pp. 66, 67, 69–71, 74–76).
- [Kon05] Hans Peter de Koning. *Library for Quantity Kinds and Units: Schema, Based on QUDV Model OMG SysML(TM), Version 1.2*. 2005. URL: <https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu> (cit. on p. 99).
- [Kop+07] Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. “SAWSDL: Semantic Annotations for WSDL and XML Schema.” In: *IEEE Internet Computing* 11.6 (2007), pp. 60–67. DOI: [10.1109/MIC.2007.134](https://doi.org/10.1109/MIC.2007.134) (cit. on p. 100).
- [Kov+18] Olga Kovalenko, Irlán Grangel-González, Marta Sabou, Arndt Lüder, Stefan Biffl, Sören Auer, and Maria-Esther Vidal. “AutomationML Ontology: Modeling Cyber-Physical Systems for Industry 4.0.” In: *IOS Press Journal* 1 (2018) (cit. on p. 154).
- [Kre02] Daniel Krech. *RDFLib*. 2002. URL: <https://rdflib.dev/> (cit. on pp. 105, 215, 216).
- [Krö+24] Moritz Kröger, Corvin Lasogga, Martin Kratz, Christian Hinke, and Carlo Holly. “Platform for Adaptive Integration of Data-Driven Models And Simulations Into Ultra Short Pulse Manufacturing Systems.” In: *Journal of Laser Micro/Nanoengineering* (Feb. 2024). ISSN: 1880-0688. DOI: [10.2961/jlmn.2024.01.2002](https://doi.org/10.2961/jlmn.2024.01.2002) (cit. on p. 97).
- [KS15] Craig A Knoblock and Pedro Szekely. “Exploiting semantics for big data integration.” In: *AI Magazine* 36.1 (2015), pp. 25–38 (cit. on p. 195).
- [Kuh08] Tobias Kuhn. “Acewiki: Collaborative Ontology Management in Controlled Natural Language.” In: *arXiv preprint arXiv:0807.4623* (2008). DOI: [10.48550/arXiv.0807.4623](https://doi.org/10.48550/arXiv.0807.4623) (cit. on p. 160).
- [Kuz16] Erdal Kuzey. “Populating knowledge bases with temporal information.” PhD thesis. 2016 (cit. on pp. 194, 199).
- [KV06] Konstantinos Kotis and George A Vouros. “Human-Centered Ontology Engineering: The HCOME Methodology.” In: *Knowledge and Information Systems* 10.1 (2006), pp. 109–131 (cit. on p. 144).
- [LDU10] Jose L Martinez Lastra, Ivan M Delamer, and Fernando Ubis. *Domain Ontologies for Reasoning Machines in Factory Automation*. ISA, 2010 (cit. on p. 138).
- [Le +22] Andrew Le Clair, Alicia Marinache, Haya El Ghalayini, Wendy Maccaull, and Ridha Khedri. “A Review on Ontology Modularization Techniques - A Multi-Dimensional Perspective.” In: *IEEE Transactions on Knowledge and Data Engineering* (2022), pp. 1–1. DOI: [10.1109/TKDE.2022.3152928](https://doi.org/10.1109/TKDE.2022.3152928) (cit. on p. 140).

Bibliography

- [Leb+13] Timothy Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhaos. *PROV-O: The PROV Ontology*. W3C Recommendation. Accessed on 2024-05-23. The W3C SPARQL Working Group, 2013. URL: <https://www.w3.org/TR/prov-o/> (cit. on pp. 56, 57, 152).
- [LEC+07] David Laniado, Davide Eynard, Marco Colombetti, et al. “Using WordNet to turn a Folksonomy into a Hierarchy of Concepts.” In: *SWAP*. Vol. 7. 2007, pp. 192–201 (cit. on pp. 194, 198).
- [Leh+15] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. “Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia.” In: *Semantic web* 6.2 (2015), pp. 167–195 (cit. on pp. 22, 168).
- [Lem+06] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko. “MASON: A Proposal For An Ontology Of Manufacturing Domain.” In: *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS’06)*. 2006, pp. 195–200. DOI: [10.1109/DIS.2006.48](https://doi.org/10.1109/DIS.2006.48) (cit. on p. 154).
- [Lew+20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” In: *NeurIPS* 33 (2020), pp. 9459–9474 (cit. on p. 226).
- [Lew91] James R Lewis. “Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ.” In: *ACM Sigchi Bulletin* 23.1 (1991), pp. 78–81 (cit. on pp. 187, 188).
- [LG89] Douglas B Lenat and Ramanathan V Guha. *Building Large Knowledge-based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN: 978-0-201-51752-1 (cit. on pp. 167–169).
- [LGD20] Johannes Lipp, Lars Gleim, and Stefan Decker. “Towards Reusability in the Semantic Web: Decoupling Naming, Validation, and Reasoning.” In: *Proceedings of the 11th Workshop on Ontology Design and Patterns(WOP2020) co-located with 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, November 01-06, 2020*. Contributions: JL and LG designed, conceived, performed, and interpreted the work. SD provided feedback and reviewed the paper. SD provided funding and guidance. CEUR Workshop Proceedings. 2020. DOI: [10.24406/publica-fhg-409120](https://doi.org/10.24406/publica-fhg-409120). URL: <http://ontologydesignpatterns.org/wiki/WOP:2020> (cit. on pp. 67, 133, 134, 158).
- [LGD21] Johannes Lipp, Lars Christoph Gleim, and Stefan Decker. “Chapter 20. Towards Easy Reusability in the Semantic Web.” In: *Advances in Pattern-Based Ontology Engineering*. Ed. by Eva Blomqvist. Vol. 51. Studies on the Semantic Web. Contributions: JL and LG designed, conceived, performed, and interpreted the work. SD provided feedback and reviewed the paper. SD provided funding and guidance. IOS Press, 2021, pp. 331–336. DOI: [10.3233/SSW210023](https://doi.org/10.3233/SSW210023). URL: <https://publications.rwth-aachen.de/record/818920> (cit. on p. 158).

- [LIB14] Jürgen Lange, Frank Iwanitz, and Thomas J Burke. *OPC: Von Data Access bis Unified Architecture*. VDE, 2014. ISBN: 978-3-8007-3506-8 (cit. on p. 67).
- [Lip+20] Johannes Lipp, Maximilian Rudack, Uwe Vroomen, and Andreas Bührig-Polaczek. “When to Collect What? Optimizing Data Load via Process-driven Data Collection.” In: *Proceedings of the 22nd International Conference on Enterprise Information Systems*. Contributions: JL and MR designed, conceived, and interpreted the work. JL performed, analyzed, and interpreted the work. JL and MR drafted and revised the paper. UV and ABP provided funding and guidance. SCITEPRESS - Science and Technology Publications, 2020, pp. 220–225. ISBN: 978-989-758-423-7. DOI: [10.5220/0009439502200225](https://doi.org/10.5220/0009439502200225) (cit. on pp. 67, 79, 81, 82, 84–87, 93, 94).
- [Lip+21a] J Lipp, L Gleim, M Cochez, I Dimitriadis, H Ali, C Lange, and S Decker. *Quick Ontology Prototyping With Neologism*. 2021. URL: <https://github.com/Semantic-Society/Neologism/> (cit. on pp. 160, 162).
- [Lip+21b] Johannes Lipp, Lars Gleim, Michael Cochez, Iraklis Dimitriadis, Hussain Ali, Daniel Hoppe Alvarez, Christoph Lange, and Stefan Decker. “Towards Easy Vocabulary Drafts with Neologism 2.0.” In: *The Semantic Web: ESWC 2021 Satellite Events*. Ed. by Ruben Verborgh, Anastasia Dimou, Aidan Hogan, Claudia d’Amato, Ilaria Tiddi, Arne Bröring, Simon Mayer, Femke Ongenae, Riccardo Tommasini, and Mehwish Alam. Vol. 12739. Lecture Notes in Computer Science. Contributions: JL, ID, LG, and MC designed, conceived, and interpreted the work. HA, JL, DHA, and ID performed, analyzed, and evaluated the work. JL, ID, LG, and MC drafted and revised the paper. CL and SD reviewed the paper and provided guidance and funding. Cham: Springer International Publishing, 2021, pp. 21–26. ISBN: 978-3-030-80417-6. DOI: [10.1007/978-3-030-80418-3_4](https://doi.org/10.1007/978-3-030-80418-3_4) (cit. on pp. 158, 159, 161, 162).
- [Lip+21c] Johannes Lipp, Michael Rath, Maximilian Rudack, Uwe Vroomen, and Andreas Bührig-Polaczek. “Flexible OPC UA Data Load Optimizations on the Edge of Production.” In: *Enterprise Information Systems*. Ed. by Joaquim Filipe, Michał Śmiałek, Alexander Brodsky, and Slimane Hammoudi. Vol. 417. Lecture Notes in Business Information Processing. Contributions: JL and MRu designed, conceived, and interpreted the work. JL and MRa performed, analyzed, and interpreted the work. JL, MRa, and MRu drafted and revised the paper. UV and ABP provided funding and guidance. Cham: Springer International Publishing, 2021, pp. 43–61. ISBN: 978-3-030-75417-4. DOI: [10.1007/978-3-030-75418-1_3](https://doi.org/10.1007/978-3-030-75418-1_3) (cit. on pp. 67, 83, 88–92).
- [Lip+21d] Johannes Lipp, Siyabend Sakik, Moritz Kröger, and Stefan Decker. “LISSU: Integrating Semantic Web Concepts into SOA Frameworks.” In: *Proceedings of the 23rd International Conference on Enterprise Information Systems*. Won an award for the **Best Student Paper**. Contributions: JL and MK designed, conceived, and interpreted the work. SS, JL, and MK performed, analyzed, and evaluated the work. JL, MK, and SS drafted and revised the paper. SD provided funding, feedback, and guidance. SCITEPRESS - Science and Technology Publications, 2021, pp. 855–865. ISBN: 978-989-758-509-8. DOI: [10.5220/0010481408550865](https://doi.org/10.5220/0010481408550865) (cit. on pp. 98, 99, 103, 106, 112).

Bibliography

- [Lip17] Johannes Lipp. “Applying External Knowledge Bases and User-assisted Semantic Modelling to Evolve a Semantic Knowledge Graph.” Advisor: André Pomp. Master’s thesis. RWTH Aachen University, 2017. DOI: [10.18154/RWTH-2022-02399](https://doi.org/10.18154/RWTH-2022-02399) (cit. on pp. 18, 165, 168, 194, 197, 201, 203, 205, 207).
- [Liu+12] Chen Liu, Sai Wu, Shouxu Jiang, and Anthony KH Tung. “Cross domain search by exploiting wikipedia.” In: *2012 IEEE 28th International Conference on Data Engineering*. IEEE. 2012, pp. 546–557 (cit. on pp. 194, 196).
- [Loh+16] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. “Visualizing Ontologies with VOWL.” In: *SW* (2016), pp. 399–419. DOI: [10.3233/SW-150200](https://doi.org/10.3233/SW-150200) (cit. on pp. 20, 159, 160).
- [Los+11] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber, and B. Pîrvu. “Semantic Service Discovery and Orchestration for Manufacturing Processes.” In: *ETFA2011*. 2011, pp. 1–8. DOI: [10.1109/ETFA.2011.6058988](https://doi.org/10.1109/ETFA.2011.6058988) (cit. on p. 26).
- [LPA11] Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak. “An architecture for smart semantic recommender applications.” In: *11th International Conference on Innovative Internet Community Systems (I2CS 2011)*. Gesellschaft für Informatik eV. 2011 (cit. on pp. 166, 167).
- [LS06] Benedikt Linse and Andreas Schroeder. “Beyond xml and rdf: The versatile web query language xcerpt.” In: *Proceedings of the 15th international conference on World Wide Web*. 2006, pp. 1053–1054 (cit. on p. 14).
- [LS20] Johannes Lipp and Katrin Schilling. “The Semantic Web in the Internet of Production: A Strategic Approach with Use-Case Examples.” In: *2020 IARIA 14th International Conference on Advances in Semantic Processing*. Contributions: JL and KS designed, conceived, performed, and interpreted the work. JL and KS drafted and revised the paper. IARIA, 2020, pp. 68–72. ISBN: 978-1-61208-813-6. URL: http://www.thinkmind.org/index.php?view=article&articleid=semapro_2020_2_80_30040 (cit. on pp. 26, 65, 134).
- [LSK21] J Lipp, Siyabend Sakik, and Moritz Kröger. *LISSU: Integrating Semantic Web Concepts Into SOA Frameworks*. 2021. URL: <https://github.com/JohannesLipp/lissu> (cit. on p. 104).
- [LSV11] Christoph Legat, Christian Seitz, and Birgit Vogel-Heuser. “Unified Sensor Data Provisioning with Semantic Technologies.” In: *ETFA2011*. IEEE. 2011, pp. 1–8. DOI: [10.1109/ETFA.2011.6058989](https://doi.org/10.1109/ETFA.2011.6058989) (cit. on p. 67).
- [Luc09] Markus Luczak-Rösch. “Towards Agile Ontology Maintenance.” In: *International Semantic Web Conference*. Springer. 2009, pp. 965–972 (cit. on p. 144).
- [M H+15] M. H. Khan, S. Jan, I. Khan, and I. A. Shah. “Evaluation of linguistic similarity measurement techniques for ontology alignment.” In: *Emerging Technologies (ICET), 2015 International Conference on*. 2015, pp. 1–6. DOI: [10.1109/ICET.2015.7389207](https://doi.org/10.1109/ICET.2015.7389207) (cit. on p. 165).
- [Mad+19] Christian Mader, Jaroslav Pullmann, Niklas Petersen, Steffen Lohmann, Christoph Lange-Bever, Anna Kasprzik, Andreas Müller, Eva Corsi, Sandra Geisler, Michael Theß, Haydar Akyürek, Matthias Böckmann, Johannes Lipp, and Lina Molinas Comet. *International Data Spaces Information Model*. 2019. URL: <https://w3id.org/idsa/core> (cit. on p. 28).

- [Mae+02] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. “MaFra—a mapping framework for distributed ontologies.” In: *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web: 13th International Conference, EKAW 2002 Sigüenza, Spain, October 1–4, 2002 Proceedings 13*. Springer, 2002, pp. 235–250 (cit. on pp. 165, 194, 198).
- [MAG19] MAGMA Gießereitechnologie GmbH. *MAGMASOFT Autonomous Engineering*. Accessed on 2024-05-23. 2019. URL: <https://www.magmasoft.com/en/solutions/magmasoft/> (cit. on p. 86).
- [Mah96] Kavi Mahesh. “Ontology Development for Machine Translation: Ideology and Methodology.” In: 1996 (cit. on p. 143).
- [Mar+14] Marcos Martínez-Romero, José M Vázquez-Naya, Javier Pereira, and Alejandro Pazos. “BiOSS: A system for biomedical ontology selection.” In: *Computer methods and programs in biomedicine* 114.1 (2014), pp. 125–140 (cit. on p. 167).
- [Mar+17] Marcos Martínez-Romero, Clement Jonquet, Martin J. O’Connor, John Graybeal, Alejandro Pazos, and Mark A. Musen. “NCBO Ontology Recommender 2.0: an enhanced approach for biomedical ontology recommendation.” In: *Journal of Biomedical Semantics* 8.1 (June 2017). DOI: [10.1186/s13326-017-0128-y](https://doi.org/10.1186/s13326-017-0128-y) (cit. on pp. 166, 167).
- [Mar+20] Francisco Martín-Recuerda, Dirk Walther, Siegfried Eisinger, Graham Moore, Petter Andersen, Per-Olav Opdahl, and Lillian Hella. “Revisiting Ontologies of Units of Measure for Harmonising Quantity Values—A Use Case.” In: *International Semantic Web Conference*. Springer. 2020, pp. 551–567. DOI: [10.1007/978-3-030-62466-8_340](https://doi.org/10.1007/978-3-030-62466-8_340) (cit. on p. 100).
- [Mat+18] Nicolas Matentzoglou, James Malone, Chris J. Mungall, and Robert Stevens. “MIRO: Guidelines for Minimum Information for the Reporting of an Ontology.” In: *Journal of Biomedical Semantics* 9 (2018) (cit. on p. 143).
- [May+22] M. C. May, L. Kiefer, A. Kuhnle, and G. Lanza. “Ontology-Based Production Simulation with OntologySim.” In: *Applied Sciences (Switzerland)* 12.3 (2022), Art.–Nr.: 1608. ISSN: 2076-3417. DOI: [10.3390/app12031608](https://doi.org/10.3390/app12031608) (cit. on p. 154).
- [Maz+16] Luca Mazzola, Patrick Kapahnke, Marko Vujic, and Matthias Klusch. “CDM-Core: A Manufacturing Domain Ontology in OWL2 for Production and Maintenance.” In: *KEOD*. 2016 (cit. on p. 154).
- [MB09] Alistair Miles and Sean Bechhofer. “SKOS simple knowledge organization system reference.” In: *W3C recommendation* (2009) (cit. on pp. 22, 56, 152).
- [MB13] Wiet Mazairac and Jakob Beetz. “BIMQL – An open query language for building information models.” In: *Advanced Engineering Informatics* 27 (Oct. 2013), pp. 444–456. DOI: [10.1016/j.aei.2013.06.001](https://doi.org/10.1016/j.aei.2013.06.001) (cit. on p. 117).
- [MCD17] Devika P Madalli, Usashi Chatterjee, and Biswanath Dutta. “An analytical approach to building a core ontology for food.” In: *Journal of Documentation* (2017) (cit. on p. 144).
- [McK+12] Katy McKen, Catherine Pink, Liz Lyon, and Matthew Davidson. *Research360: Data in the Research Lifecycle*. English. Accessed on 2024-05-23. June 2012. URL: <http://opus.bath.ac.uk/32292/> (cit. on pp. 33, 39).

Bibliography

- [MCR08] Viviana Mascardi, Valentina Cordi, and Paolo Rosso. “A comparison of upper ontologies (technical report disi-tr-06-21).” In: *Dipartimento di Informatica e Scienze dell'Informazione (DISI), Università degli Studi di Genova, Via Dodecaneso* 35 (2008), p. 16146 (cit. on p. 152).
- [Mer23] C. Mertens. *The Data Spaces Radar - Version 2.0*. Tech. rep. International Data Spaces Association, 2023. DOI: [10.5281/zenodo.8059216](https://doi.org/10.5281/zenodo.8059216) (cit. on pp. 7, 24, 46).
- [Mic+12] William K. Michener, Suzie Allard, Amber Budden, Robert B. Cook, Kimberly Douglass, Mike Frame, Steve Kelling, Rebecca Koskela, Carol Tenopir, and David A. Vieglais. “Participatory design of DataONE—Enabling cyberinfrastructure for the biological and environmental sciences.” In: *Ecological Informatics* 11 (2012), pp. 5–15. ISSN: 15749541. DOI: [10.1016/j.ecoinf.2011.08.007](https://doi.org/10.1016/j.ecoinf.2011.08.007) (cit. on p. 37).
- [Mic+21] S. Michie, R. West, A. N. Finnerty, E. Norris, A. J. Wright, M. M. Marques, M. Johnston, M. P. Kelly, J. Thomas, and J. Hastings. “Representation of Behaviour Change Interventions and Their Evaluation: Development of the Upper Level of the Behaviour Change Intervention Ontology.” In: *PubMed* (2021). DOI: [10.12688/wellcomeopenres.15902.2](https://doi.org/10.12688/wellcomeopenres.15902.2) (cit. on p. 144).
- [Mic+23] Judith Michael, István Koren, Iraklis Dimitriadis, Judith Fulterer, Aymen Gannouni, Malte Heithoff, Annkristin Hermann, Katharina Hornberg, Moritz Kröger, Patrick Sapel, Niklas Schäfer, Johannes Theissen-Lipp, Stefan Decker, Christian Hopmann, Matthias Jarke, Bernhard Rumpe, Robert H. Schmitt, and Günther Schuh. “A Digital Shadow Reference Model for Worldwide Production Labs.” In: *Internet of Production*. Ed. by Christian Brecher, Günther Schuh, Wil van der Aalst, Matthias Jarke, Frank T. Piller, and Melanie Padberg. Interdisciplinary Excellence Accelerator Series. Cham: Springer International Publishing, 2023, pp. 1–29. ISBN: 978-3-030-98062-7. DOI: [10.1007/978-3-030-98062-7_3-3](https://doi.org/10.1007/978-3-030-98062-7_3-3) (cit. on pp. 58, 60, 97).
- [Mic16] Microsoft. *Microsoft Power Automate*. Accessed on 2024-05-23. 2016. URL: <https://www.microsoft.com/en-us/power-platform/products/power-automate> (cit. on p. 83).
- [Mil95] George A Miller. “WordNet: a lexical database for English.” In: *Communications of the ACM* 38.11 (1995), pp. 39–41. DOI: [10.1145/219717.2197480](https://doi.org/10.1145/219717.2197480) (cit. on pp. 133, 166, 168, 178).
- [Mis23] Mistral AI. *Mixtral 8x7B - A 7B sparse Mixture-of-Experts Model*. Accessed on 2024-05-23. Dec. 2023. URL: <https://mistral.ai/news/mixtral-of-experts/> (cit. on p. 226).
- [MJ12] William K. Michener and Matthew B. Jones. “Ecoinformatics: Supporting Ecology as a Data-intensive Science.” In: *Trends in ecology & evolution* 27.2 (Feb. 2012), pp. 85–93. DOI: [10.1016/j.tree.2011.11.016](https://doi.org/10.1016/j.tree.2011.11.016) (cit. on pp. 15, 18, 19, 33, 37).
- [MJS10] M. Kalender, J. Dang, and S. Uskudarli. “UNIpedia: A Unified Ontological Knowledge Platform for Semantic Content Tagging and Search.” In: *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*. 2010, pp. 293–298. DOI: [10.1109/ICSC.2010.82](https://doi.org/10.1109/ICSC.2010.82) (cit. on pp. 165, 194, 195).

- [MLR10] V. Mascardi, A. Locoro, and P. Rosso. “Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation.” In: *IEEE Transactions on Knowledge and Data Engineering* 22.5 (2010), pp. 609–623. ISSN: 1041-4347. DOI: [10.1109/TKDE.2009.154](https://doi.org/10.1109/TKDE.2009.154) (cit. on p. 165).
- [Mon+01] Andrés Montoyo, Manuel Palomar, German Rigau, and Pau Gargalló. “Wordnet enrichment with classification systems.” In: *Proceedings of WordNet and Other Lexical Resources: Applications, Extensions and Customisations Workshop.(NAACL-01) The Second Meeting of the North American Chapter of the Association for Computational Linguistics*. Citeseer. 2001, pp. 101–106 (cit. on p. 196).
- [Moo+24] Paul Moosmann, Johannes Theissen-Lipp, Christina Gillmann, and Christoph Lange. “Enhanced RDF Validation Techniques for Dataspaces.” In: *Proceedings of the 20th International Conference on Semantic Systems, 17–19 September 2024*. Status: Accepted (currently being published). Contributions: JTL identified the research gap. PM and JTL designed and conceived the work. PM performed, analyzed, and evaluated the work. JTL and PM drafted the paper. PM, JTL, CG, and CL reviewed and revised the paper. CL provided funding. Amsterdam, the Netherlands: SEMANTiCS 2024 EU, Sept. 2024 (cit. on pp. 209, 212, 213).
- [Moo22] Paul Moosmann. “Enhancing SHACL Validation Through Constraint Templates and Inference.” Advisor: Johannes Theissen-Lipp. Master’s thesis. RWTH Aachen University, 2022 (cit. on pp. 212, 213).
- [Mox+21] Sierra Moxon, Harold Solbrig, Deepak Unni, Dazhi Jiao, Richard Bruskiewich, James Balhoff, Gaurav Vaidya, William Duncan, Harshad Hegde, Mark Miller, Matthew Brush, Nomi Harris, Melissa Haendel, and Christopher Mungall. “The Linked Data Modeling Language (LinkML): A General-Purpose Data Modeling Framework Grounded in Machine-Readable Semantics.” English (US). In: *CEUR Workshop Proceedings* 3073 (2021), pp. 148–151. ISSN: 1613-0073 (cit. on p. 20).
- [MS03] Lars Mönch and Marcel Stehli. “An Ontology for Production Control of Semiconductor Manufacturing Processes.” In: *German Conference on Multiagent System Technologies*. Springer. 2003, pp. 156–167 (cit. on p. 154).
- [MSR02] Libby Miller, Andy Seaborne, and Alberto Reggiori. “Three implementations of SquishQL, a simple RDF query language.” In: *The Semantic Web—ISWC 2002: First International Semantic Web Conference Sardinia, Italy, June 9–12, 2002 Proceedings 1*. Springer. 2002, pp. 423–435 (cit. on p. 14).
- [MSZ01] Sheila A McIlraith, Tran Cao Son, and Honglei Zeng. “Semantic Web Services.” In: *IEEE intelligent systems* 16.2 (2001), pp. 46–53. DOI: [10.1109/5254.9205990](https://doi.org/10.1109/5254.9205990) (cit. on pp. 95, 100).
- [MT17] Riichiro Mizoguchi and Fumiaki Toyoshima. “YAMATO: Yet Another More Advanced Top-Level Ontology with Analysis of Five Examples of Change.” In: *Appl. Ontology* 17 (2017), pp. 211–232 (cit. on p. 152).

Bibliography

- [Muñ+09] E. Muñoz, G.M. Kopanos, A. Espuña, and L. Puigjaner. “Towards an Ontological Infrastructure for Chemical Batch Process Management.” In: *19th European Symposium on Computer Aided Process Engineering*. Ed. by Jacek Jeżowski and Jan Thullie. Vol. 26. Computer Aided Chemical Engineering. Elsevier, 2009, pp. 883–888. DOI: [10.1016/S1570-7946\(09\)70147-6](https://doi.org/10.1016/S1570-7946(09)70147-6). URL: <https://www.sciencedirect.com/science/article/pii/S1570794609701476> (cit. on pp. 152, 154).
- [MW13] David Milne and Ian H Witten. “An open-source toolkit for mining Wikipedia.” In: *Artificial Intelligence* 194 (2013), pp. 222–239 (cit. on pp. 194, 195).
- [n8n19] n8n GmbH. *n8n.io - Free and Open Workflow Automation Tool*. Accessed on 2024-05-23. 2019. URL: <https://n8n.io/> (cit. on p. 83).
- [NC09] Holger Neuhaus and Michael Compton. “The Semantic Sensor Network Ontology.” In: *AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany*. 2009, pp. 1–33 (cit. on p. 154).
- [Neu+22] Stefanie Neuenschwander, Patricia Romao, Jürgen Holm, and Murat Sariyar. “Developing an Ontology for Documenting Adverse Events While Avoiding Pitfalls.” In: *Studies in health technology and informatics* 289 (2022), pp. 166–169. DOI: [10.3233/SHTI210885](https://doi.org/10.3233/SHTI210885) (cit. on p. 139).
- [New+19] Daniel Newman, Mahmoud Parto, Kyle Saleeby, Thomas Kurfess, and Andrew Dugenske. “Development of a Digital Architecture for Distributed CNC Machine Health Monitoring.” In: *Smart and Sustainable Manufacturing Systems* (2019). DOI: [10.1520/SSMS20190038](https://doi.org/10.1520/SSMS20190038) (cit. on p. 84).
- [NFB96] Henrik Nielsen, Roy T. Fielding, and Tim Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.0 (RFC 1945)*. May 1996. DOI: [10.17487/RFC1945](https://doi.org/10.17487/RFC1945). URL: <https://www.rfc-editor.org/info/rfc1945> (cit. on p. 14).
- [Nic13] Nick O’Leary and Dave Conway-Jones, IBM Emerging Technologies, now part of the JS Foundation. *Node-RED*. Accessed on 2024-05-23. 2013. URL: <https://nodered.org/> (cit. on pp. 83, 84).
- [Nil08] Mikael Nilsson. *Description Set Profiles: A constraint language for Dublin Core™ Application Profiles*. Tech. rep. 2008. URL: <https://www.dublincore.org/specifications/dublin-core/dc-dsp/> (cit. on p. 210).
- [NM+01] Natalya F Noy, Deborah L McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*. 2001 (cit. on pp. 137, 138, 142, 144).
- [Noy+01] Natalya F Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W Ferguson, and Mark A Musen. “Creating Semantic Web Contents With Protege-2000.” In: *IEEE intelligent systems* 16.2 (2001), pp. 60–71. DOI: [10.1109/5254.920601](https://doi.org/10.1109/5254.920601) (cit. on pp. 20, 158, 160).
- [NP01a] Ian Niles and Adam Pease. “Towards a Standard Upper Ontology.” In: *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*. 2001, pp. 2–9. DOI: [10.1145/505168.5051700](https://doi.org/10.1145/505168.5051700) (cit. on p. 99).

- [NP01b] Ian Niles and Adam Pease. “Towards a Standard Upper Ontology.” In: *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*. FOIS '01. Ogunquit, Maine, USA: Association for Computing Machinery, 2001, pp. 2–9. ISBN: 1581133774. DOI: [10.1145/505168.505170](https://doi.org/10.1145/505168.505170) (cit. on p. 152).
- [NP12] Roberto Navigli and Simone Paolo Ponzetto. “BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network.” In: *Artificial Intelligence* 193 (2012), pp. 217–250. ISSN: 0004-3702. DOI: [10.1016/j.artint.2012.07.001](https://doi.org/10.1016/j.artint.2012.07.001). URL: <https://www.sciencedirect.com/science/article/pii/S0004370212000793> (cit. on pp. 168, 170, 178).
- [OHW22] B. Otto, M. ten Hompel, and S. Wrobel. *Designing Data Spaces*. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-93974-8 (cit. on pp. 3, 10).
- [OJ19] Boris Otto and Matthias Jarke. “Designing a multi-sided data platform: findings from the International Data Spaces case.” In: *Electronic Markets* 29.4 (2019), pp. 561–580. ISSN: 1019-6781. DOI: [10.1007/s12525-019-00362-x](https://doi.org/10.1007/s12525-019-00362-x) (cit. on pp. 7, 9, 10, 31).
- [Ont15] Ontotext. *GraphDB*. 2015. URL: <https://www.ontotext.com/products/graphdb/> (cit. on p. 20).
- [ÖÖÜ11] Tuğba Özacar, Övünç Öztürk, and Murat Osman Ünalır. “ANEMONE: An environment for modular ontology development.” In: *Data & Knowledge Engineering* 70.6 (2011), pp. 504–526 (cit. on p. 144).
- [Opa22] Kobkaew Opasjumruskit. *OntoHuman*. version 1.0.1. June 2022. DOI: [10.5281/ZENODO.6783007](https://doi.org/10.5281/ZENODO.6783007) (cit. on p. 20).
- [OPC17a] OPC Foundation. *OPC UA Part 1: Overview and Concepts*. Accessed on 2024-05-23. 2017. URL: <https://reference.opcfoundation.org/v104/Core/docs/Part1/> (cit. on p. 82).
- [OPC17b] OPC Foundation. *OPC UA Part 4: Services*. Accessed on 2024-05-23. 2017. URL: <https://reference.opcfoundation.org/v104/Core/docs/Part4/> (cit. on p. 82).
- [OPC18] OPC Foundation. *OPC UA Part 14: PubSub*. Accessed on 2024-05-23. 2018. URL: <https://reference.opcfoundation.org/v104/Core/docs/Part14/> (cit. on p. 82).
- [Ope23] OpenAI. *GPT-4 Technical Report*. Tech. rep. 2023. URL: <https://arxiv.org/abs/2303.08774> (cit. on p. 226).
- [Ope98] OpenLink Software. *Virtuoso Universal Server*. 1998. URL: <https://virtuoso.openlinksw.com/> (cit. on p. 20).
- [OPP19] Wilson Alfredo Ortega Ordoñez, César Jesús Pardo Calvache, and Francisco José Pino Correa. “OntoAgile: An Ontology for Agile Software Development Processes.” In: *DYNA* (2019) (cit. on p. 140).

Bibliography

- [Ott+16] B. Otto, J. Jürjens, J. Schon, S. Auer, N. Menz, S. Wenzel, and J. Cirullies. *Industrial Data Space - Digital Sovereignty Over Data*. Tech. rep. Fraunhofer, Jan. 2016. DOI: [10.13140/RG.2.1.2673.0649](https://doi.org/10.13140/RG.2.1.2673.0649). URL: <https://www.fraunhofer.de/content/dam/zv/en/fields-of-research/industrial-data-space/whitepaper-industrial-data-space-eng.pdf> (cit. on p. 7).
- [Ott+19] B. Otto, S. Steinbuss, A. Teuscher, and S. Lohmann. *IDS Reference Architecture Model*. 2019. DOI: [10.5281/ZENODO.5105529](https://doi.org/10.5281/ZENODO.5105529). URL: <https://zenodo.org/record/5105529> (cit. on pp. 7, 9, 28, 31).
- [OVP19] Felix Ocker, Birgit Vogel-Heuser, and Christiaan J. J. Paredis. “Applying Semantic Web Technologies to Provide Feasibility Feedback in Early Design Phases.” In: *Journal of Computing and Information Science in Engineering* 19.4 (July 2019). ISSN: 1530-9827. DOI: [10.1115/1.4043795](https://doi.org/10.1115/1.4043795) (cit. on pp. 138–140, 154).
- [Pal21] Raul Palma. *Sub-class inference using SHACL*. Accessed: 2024-04-23. 2021. URL: <https://github.com/TopQuadrant/shacl/issues/108> (cit. on p. 211).
- [Pau+18] Alexander Paulus, André Pomp, Lucian Poth, Johannes Lipp, and Tobias Meisen. “Gathering and Combining Semantic Concepts from Multiple Knowledge Bases.” In: *Proceedings of the 20th International Conference on Enterprise Information Systems*. Won an award for the **Best Paper**. Contributions: APo and APa designed, conceived, and interpreted the work. JL and LP performed, analyzed, and evaluated the work. APo, APa, JL, and LP drafted and revised the paper. TM provided funding and guidance. SCITEPRESS - Science and Technology Publications, 2018, pp. 69–80. ISBN: 978-989-758-298-1. DOI: [10.5220/0006800700690080](https://doi.org/10.5220/0006800700690080) (cit. on pp. 172, 178, 182, 183).
- [Pau+19] Alexander Paulus, André Pomp, Lucian Poth, Johannes Lipp, and Tobias Meisen. “Recommending Semantic Concepts for Improving the Process of Semantic Modeling.” In: *Enterprise Information Systems*. Ed. by Slimane Hammoudi, Michał Śmiałek, Olivier Camp, and Joaquim Filipe. Vol. 363. Lecture Notes in Business Information Processing. Contributions: APo and APa designed, conceived, and interpreted the work. JL and LP performed, analyzed, and evaluated the work. APo, APa, JL, and LP drafted and revised the paper. TM provided funding and guidance. Cham: Springer International Publishing, 2019, pp. 350–369. ISBN: 978-3-030-26168-9. DOI: [10.1007/978-3-030-26169-6_17](https://doi.org/10.1007/978-3-030-26169-6_17) (cit. on pp. 19, 165, 167, 168, 171, 174–178, 180–183, 185, 186, 194).
- [Pau+21] A. Paulus, A. Burgdorf, L. Puleikis, T. Langer, A. Pomp, and T. Meisen. “PLASMA: Platform for Auxiliary Semantic Modeling Approaches.” In: *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 2: ICEIS*. INSTICC. SciTePress, 2021, pp. 403–412. ISBN: 978-989-758-509-8. DOI: [10.5220/0010499604030412](https://doi.org/10.5220/0010499604030412) (cit. on p. 167).
- [Pau17] Heiko Paulheim. “Knowledge graph refinement: A survey of approaches and evaluation methods.” In: *Semantic web* 8.3 (2017), pp. 489–508 (cit. on pp. 24, 204).
- [PBV08] Asunción Pérez Gómez, Mcs De Figueroa Baonza, and Boris Villazón. “Neon Methodology for Building Ontology Networks: Ontology Specification.” In: *Methodology* (2008), pp. 1–18 (cit. on p. 144).

- [PDK22] Foivos Psarommatis, Paul Arthur Dreyfus, and Dimitris Kiritsis. “Chapter 9 - The Role of Big Data Analytics in the Context of Modeling Design and Operation of Manufacturing Systems.” In: *Design and Operation of Production Networks for Mass Personalization in the Era of Cloud Technology*. Ed. by Dimitris Mourtzis. Elsevier, 2022, pp. 243–275. ISBN: 978-0-12-823657-4. DOI: [10.1016/B978-0-12-823657-4.00012-9](https://doi.org/10.1016/B978-0-12-823657-4.00012-9). URL: <https://www.sciencedirect.com/science/article/pii/B9780128236574000129> (cit. on pp. 137, 140).
- [PDT12] Hervé Panetto, Michele Dassisti, and Angela Tursi. “ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment.” In: *Advanced Engineering Informatics* 26.2 (2012), pp. 334–348 (cit. on p. 154).
- [Pel+16] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. “From Freebase to Wikidata: The Great Migration.” In: *Proceedings of the 25th International Conference on World Wide Web*. WWW ’16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 1419–1428. ISBN: 9781450341431. DOI: [10.1145/2872427.2874809](https://doi.org/10.1145/2872427.2874809) (cit. on p. 169).
- [Pen+19] Jan Pennekamp, Rene Glebke, Martin Henze, Tobias Meisen, Christoph Quix, Rihan Hai, Lars Gleim, Philipp Niemietz, Maximilian Rudack, Simon Knape, Alexander Epple, Daniel Trauth, Uwe Vroomen, Thomas Bergs, Christian Brecher, Andreas Buhrig-Polaczek, Matthias Jarke, and Klaus Wehrle. “Towards an Infrastructure Enabling the Internet of Production.” In: *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 31–37. ISBN: 978-1-5386-8500-6. DOI: [10.1109/ICPHYS.2019.8780276](https://doi.org/10.1109/ICPHYS.2019.8780276) (cit. on pp. 25, 58, 96, 133).
- [Per16] Silvio Peroni. “A Simplified Agile Methodology for Ontology Development.” In: *OWL: Experiences and Directions—Reasoner Evaluation*. Springer, 2016, pp. 55–69 (cit. on pp. 140, 144).
- [Pet+12] David Peterson, Shudi Sandy Gao, Ashok Malhotra, C. Michael Sperberg-McQueen, Henry S. Thompson, and Paul V. Biron. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2012. URL: <https://www.w3.org/TR/xmlschema11-2/> (cit. on p. 13).
- [Pet+17] Florian Pethig, Sebastian Schriegel, Alexander Maier, Jens Otto, Stefan Windmann, Björn Böttcher, Oliver Niggemann, and Juergen Jasperneite. *Industrie 4.0 Communication Guideline Based on OPC UA*. Frankfurt: VDMA Verlag GmbH, Sept. 2017. ISBN: 9783816307099 (cit. on p. 66).
- [Pet77] James L Peterson. “Petri nets.” In: *ACM Computing Surveys (CSUR)* 9.3 (1977), pp. 223–252. DOI: [10.1145/356698.356702](https://doi.org/10.1145/356698.356702) (cit. on p. 87).
- [Pha+16] Minh Pham, Suresh Alse, Craig A Knoblock, and Pedro Szekely. “Semantic labeling: a domain-independent approach.” In: *The Semantic Web—ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I* 15. Springer, 2016, pp. 446–462 (cit. on p. 195).

Bibliography

- [Pin09] Pinto, H. Sofia and Tempich, C. and Staab, Steffen. “Ontology Engineering and Evolution in a Distributed World Using DILIGENT.” In: *Handbook on Ontologies*. Ed. by Steffen Staab and Rudi Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 153–176. ISBN: 978-3-540-92673-3. DOI: [10.1007/978-3-540-92673-3_7](https://doi.org/10.1007/978-3-540-92673-3_7) (cit. on p. 144).
- [Pla19] Plattform Industrie 4.0. *Project GAIA-X: A Federated Data Infrastructure as the Cradle of a Vibrant European Ecosystem*. Tech. rep. D-11019 Berlin, Germany: Federal Ministry for Economic Affairs and Energy (BMWi), Oct. 2019 (cit. on pp. 7, 9, 31).
- [Pla21] Plattform Industrie 4.0. *RAMI4.0 – A Reference Framework for Digitalisation*. Accessed on 2024-05-23. 2021. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=3 (cit. on p. 140).
- [PLM19a] Andre Pomp, Johannes Lipp, and Tobias Meisen. “You are Missing a Concept! Enhancing Ontology-Based Data Access with Evolving Ontologies.” In: *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. Contributions: JL and AP designed, conceived, and interpreted the work. JL performed and evaluated the work. JL and AP drafted and revised the paper. TM provided funding and guidance. IEEE, 2019, pp. 98–105. ISBN: 978-1-5386-6783-5. DOI: [10.1109/ICOSC.2019.8665620](https://doi.org/10.1109/ICOSC.2019.8665620) (cit. on pp. 201, 203, 205).
- [PLM19b] André Pomp, Johannes Lipp, and Tobias Meisen. “Enabling the Continuous Evolution of Ontologies for Ontology-Based Data Management.” In: *International Journal of Robotic Computing* (Oct. 2019). Contributions: JL and AP designed, conceived, and interpreted the work. JL performed, analyzed, and evaluated the work. JL and AP drafted and revised the paper. TM provided funding and guidance., pp. 19–43. DOI: [10.35708/TAI1868-126244](https://doi.org/10.35708/TAI1868-126244) (cit. on pp. 167, 192, 194, 200, 202).
- [PLS14] Eric Prud’hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. “Shape expressions: an RDF validation and transformation language.” In: *SEMANTICS*. 2014, pp. 32–40. DOI: [10.1145/2660517.2660523](https://doi.org/10.1145/2660517.2660523) (cit. on pp. 16, 133, 134, 210).
- [PM13] Li Peng and Yuan Man. “A Novel Data Exchange Model Based on P2P and Ontology.” In: *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*. 2013, pp. 1693–1696. DOI: [10.1109/MEC.2013.6885329](https://doi.org/10.1109/MEC.2013.6885329) (cit. on p. 138).
- [PMS10] Antonio Paredes-Moreno, Francisco J Martínez-López, and David G Schwartz. “A Methodology for the Semi-automatic Creation of Data-driven Detailed Business Ontologies.” In: *Information Systems* 35.7 (2010), pp. 758–773 (cit. on p. 144).
- [Pol22] Policy Rules Committee. *Gaia-X Trust Framework - 22.10 Release*. Gaia-X Documentation. Accessed on 2024-05-23. Gaia-X European Association for Data and Cloud AISBL, 2022. URL: <https://docs.gaia-x.eu/policy-rules-committee/trust-framework/22.10/> (cit. on p. 7).

- [Pom+18a] André Pomp, Alexander Paulus, Sabina Jeschke, and Tobias Meisen. “Enabling Semantics in Enterprises.” In: *Enterprise Information Systems*. Ed. by Slimane Hammoudi, Michał Śmiałek, Olivier Camp, and Joaquim Filipe. Cham: Springer International Publishing, 2018, pp. 428–450. ISBN: 978-3-319-93375-7 (cit. on pp. 173, 183).
- [Pom+18b] André Pomp, Alexander Paulus, Daniel Klischies, Christian Schwier, and Tobias Meisen. “A web-based UI to enable semantic modeling for everyone.” In: *Procedia Computer Science* 137 (2018), pp. 249–254 (cit. on p. 201).
- [Pov+22] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. “LOT: An Industrial Oriented Ontology Engineering Framework.” In: *Engineering Applications of Artificial Intelligence* 111 (2022), p. 104755 (cit. on p. 146).
- [Pre+09] Valentina Presutti, Enrico Daga, Aldo Gangemi, and Eva Blomqvist. “eXtreme Design with Content Ontology Design Patterns.” In: *WOP*. 2009 (cit. on p. 144).
- [Pre+20] Sebastian Pretzsch, Holger Drees, Lutz Rittershaus, Christoph Schlueter Langdon, Christoph Lange, and Christian Weiers. *Mobility Data Space - Ein Sicherer Datenraum für die souveräne und plattformübergreifende Bewirtschaftung von Mobilitätsdaten*. Tech. rep. Mar. 2020. URL: https://www.mobility-data-space.de/content/dam/ivi/mobility-data-space/documents/Mobility_Data_Space_DE_20220603_web.pdf (cit. on p. 28).
- [PRG12] Sara Paiva, Manuel Ramos-Cabrer, and Alberto Gil-Solla. “GSSP—A Generic Semantic Search Platform.” In: *Procedia Technology* 5 (2012), pp. 388–396 (cit. on p. 199).
- [Pri22] The Trustees of Princeton University. *WordNet - A Lexical Database for English*. Accessed on 2024-05-23. 2022. URL: <https://wordnet.princeton.edu> (cit. on pp. 152, 178).
- [Pru04] Eric Prud’hommeaux. *Algae RDF Query Language*. Author work in progress. Accessed on 2024-05-23. 2004. URL: <https://www.w3.org/2004/05/06-Algae/> (cit. on p. 14).
- [PS07] Simone Paolo Ponzetto and Michael Strube. “Knowledge derived from Wikipedia for computing semantic relatedness.” In: *Journal of Artificial Intelligence Research* 30 (2007), pp. 181–212 (cit. on p. 195).
- [PT16] Pieter Pauwels and Walter Terkaj. “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology.” In: *Automation in Construction* 63 (2016), pp. 100–133 (cit. on p. 117).
- [Pub12] Publications Office of the European Union. *EU Open Data Portal*. Accessed: 2020-10-21. 2012. URL: <https://data.europa.eu/euodp> (cit. on p. 26).
- [QHV16] Christoph Quix, Rihan Hai, and Ivan Vatrov. “Metadata Extraction and Management in Data Lakes With GEMMS.” In: *Complex Systems Informatics and Modeling Quarterly* 9 (2016), pp. 67–83. DOI: [10.7250/csimq.2016-9.04](https://doi.org/10.7250/csimq.2016-9.04) (cit. on p. 82).
- [Qui03] Christoph Quix. “Metadata Management for Quality-Oriented Information Logistics in Data Warehouse Systems.” PhD thesis. Aachen, Germany: RWTH Aachen University, 2003 (cit. on p. 82).

Bibliography

- [Ram+14] Luis Ramos, Richard Gil, Dimitra Anastasiou, and Maria J. Martin-Bautista. “Towards a Machine of a Process (MOP) Ontology to Facilitate e-Commerce of Industrial Machinery.” In: *Comput. Ind.* 65.1 (Jan. 2014), pp. 108–115. ISSN: 0166-3615. DOI: [10.1016/j.compind.2013.07.012](https://doi.org/10.1016/j.compind.2013.07.012) (cit. on p. 154).
- [Ram+15] S Krishnamurthy Ramnandan, Amol Mittal, Craig A Knoblock, and Pedro Szekely. “Assigning semantic labels to data sources.” In: *The Semantic Web. Latest Advances and New Domains: 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31–June 4, 2015. Proceedings*. Springer. 2015, pp. 403–417 (cit. on pp. 19, 165, 195).
- [Ram+21] Borja Ramis Ferrer, Wael M. Mohammed, Mussawar Ahmad, Sergii Iarovyi, Jiayi Zhang, Robert Harrison, and Jose Luis Martinez Lastra. “Comparing Ontologies and Databases: A Critical Review of Lifecycle Engineering Models in Manufacturing.” In: *Knowledge and Information Systems* (2021). DOI: [10.1007/s10115-021-01558-4](https://doi.org/10.1007/s10115-021-01558-4) (cit. on p. 138).
- [RAT13] Hajo Rijgersberg, Mark van Assem, and Jan Top. “Ontology of Units of Measure and Related Concepts.” In: *Semant. Web* 4.1 (Jan. 2013), pp. 3–13. ISSN: 1570-0844. DOI: [10.3233/SW-2012-0069](https://doi.org/10.3233/SW-2012-0069) (cit. on pp. 99, 152).
- [Rat16] Rat für Informationsinfrastrukturen (RfII) Head Office. *Enhancing Research Data Management: Performance Through Diversity*. Tech. rep. Papendiek 16, 37073 Göttingen, Germany, May 2016. URL: <https://rfii.de/?p=2075> (cit. on pp. 8, 9).
- [Ray14] Steve Ray. *Quantities, Units, Dimensions and Types QUDT*. 2014. DOI: [10.25504/FAIRsharing.d3pqw7](https://doi.org/10.25504/FAIRsharing.d3pqw7). URL: <http://www.qudt.org/> (cit. on pp. 99, 100).
- [RBI20] Víctor Julio Ramírez-Durán, Idoia Berges, and Arantza Illarramendi. “ExtruOnt: An Ontology for Describing a Type of Manufacturing Machine for Industry 4.0 Systems.” In: *Semantic Web* 11 (2020), pp. 887–909 (cit. on p. 154).
- [RDS17a] Jonathan Robie, Michael Dyck, and Josh Spiegel. *XML Path Language (XPath) 3.1*. W3C Working Group Note. Accessed on 2024-05-23. W3C Working Group, 2017. URL: <https://www.w3.org/TR/xpath-31/> (cit. on pp. 13, 14, 66, 67).
- [RDS17b] Jonathan Robie, Michael Dyck, and Josh Spiegel. *XQuery 3.1: An XML Query Language*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2017. URL: <https://www.w3.org/TR/xquery-31/> (cit. on pp. 13, 14, 67).
- [ReS12] ReSpec Editors. *ReSpec: A tool for creating technical documents and web standards*. W3C Documentation. <https://www.w3.org/respec/>. 2012. URL: <https://www.w3.org/respec/> (cit. on p. 20).
- [Rey12] Dave Reynolds. *The Organization Ontology*. W3C Recommendation. Accessed on 2024-05-23. overnment Linked Data Working Group, 2012. URL: <https://www.w3.org/TR/vocab-org/> (cit. on pp. 56, 152).
- [RG15] Hans-Jürgen Rennau and Christian Grün. “XQuery as a data integration language.” In: *Proceedings of Balisage: The Markup Conference 2015*. Ed. by B. Tommie Usdin, Deborah A. Lapeyre, James David Mason, C. M. Sperberg-McQueen, and Norman Walsh. Balisage Series on Markup Technologies. Mulberry Technologies, Inc. Rockville, Maryland, 2015. DOI: [10.4242/balisagevol15.rennau01](https://doi.org/10.4242/balisagevol15.rennau01) (cit. on p. 67).

- [Rix+16] Michael Rix, Bernd Kujat, Tobias Meisen, and Sabina Jeschke. “An agile information processing framework for high pressure die casting applications in modern manufacturing systems.” In: *Procedia CIRP* 41 (2016), pp. 1084–1089. DOI: [10.1016/j.procir.2015.12.134](https://doi.org/10.1016/j.procir.2015.12.134) (cit. on p. 79).
- [RKT05] David J Russomanno, Cartik Kothari, and Omoju Thomas. “Sensor Ontologies: From Shallow to Deep Models.” In: *Proceedings of the Thirty-Seventh South-eastern Symposium on System Theory, 2005. SSST’05*. IEEE. 2005, pp. 107–112. DOI: [10.1109/SSST.2005.1460887](https://doi.org/10.1109/SSST.2005.1460887) (cit. on p. 99).
- [RL+02] Stephen L Reed, Douglas B Lenat, et al. “Mapping ontologies into Cyc.” In: *AAAI 2002 Conference workshop on ontologies for the semantic Web*. 2002, pp. 1–6 (cit. on p. 152).
- [Rom+15] Dumitru Roman, Jacek Kopecký, Tomas Vitvar, John Domingue, and Dieter Fensel. “WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs.” In: *Journal of Web Semantics* 31 (2015), pp. 39–58. DOI: [10.1016/j.websem.2014.11.006](https://doi.org/10.1016/j.websem.2014.11.006) (cit. on p. 100).
- [RRR00] Dominick V. Rosato, Donald V. Rosato, and Marlene G. Rosato. *Injection Molding Handbook*. Third Edition. Boston, MA and s.l.: Springer US, 2000. ISBN: 9781461545972. DOI: [10.1007/978-1-4615-4597-2](https://doi.org/10.1007/978-1-4615-4597-2) (cit. on p. 73).
- [RT20] Cecilia Reyes Peña and Mireya Tovar Vidal. “Ontology Reusing: A Review.” In: *Res. Comput. Sci.* 149.4 (2020), pp. 107–113. URL: <https://rcs.cic.ipn.mx/2020.149.4/Ontology%20Reusing-%20A%20Review.pdf> (cit. on p. 140).
- [RTM09] H. Rijgersberg, J. Top, and M.B.J. Meinders. “Semantic Support for Quantitative Research Processes.” English. In: *IEEE Intelligent Systems* 24.1 (2009), pp. 37–46. ISSN: 1541-1672. DOI: [10.1109/MIS.2009.17](https://doi.org/10.1109/MIS.2009.17) (cit. on p. 99).
- [Rud+16] Ron Rudnicki, Barry Smith, Tanya Malyuta, and William Mandrick. *Best Practices of Ontology Development*. Tech. rep. CUBRC, Inc., Oct. 2016 (cit. on pp. 137–139, 143).
- [Rüe+14] Janine Rüegg, Corinna Gries, Ben Bond-Lamberty, Gabriel J Bowen, Benjamin S Felzer, Nancy E McIntyre, Patricia A Soranno, Kristin L Vanderbilt, and Kathleen C Weathers. “Completing the data life cycle: using information management in macrosystems ecology research.” In: *Frontiers in Ecology and the Environment* 12.1 (Feb. 2014), pp. 24–30. DOI: [10.1890/120375](https://doi.org/10.1890/120375) (cit. on pp. 33, 41).
- [Rum16] Bernhard Rumpe. *Modeling with UML: Language, Concepts, Methods*. Springer International, July 2016. DOI: [10.1007/978-3-319-33933-7](https://doi.org/10.1007/978-3-319-33933-7). URL: <http://www.se-rwth.de/mbse/> (cit. on p. 117).
- [RWT11] Hajo Rijgersberg, M.L.I. Wigham, and Jan Top. “How Semantics Can Improve Engineering Processes: A Case of Units of Measure and Quantities.” In: *Advanced Engineering Informatics* 25 (Apr. 2011), pp. 276–287. DOI: [10.1016/j.aei.2010.07.008](https://doi.org/10.1016/j.aei.2010.07.008) (cit. on p. 99).
- [RWT19] RWTH Aachen. *Internet of Production*. 2019. URL: <https://iop.rwth-aachen.de/> (cit. on p. 25).
- [Rym14] Arthur Ryman. *Resource Shape 2.0*. W3C Member Submission. W3C Working Group, 2014. URL: <https://www.w3.org/submissions/shapes/> (cit. on p. 210).

Bibliography

- [Sah19] Sahar Vahdati. “Collaborative Integration, Publishing and Analysis of Distributed Scholarly Metadata.” PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, Feb. 2019. URL: <https://hdl.handle.net/20.500.11811/7874> (cit. on p. 35).
- [SAS11] R Subhashini, J Akilandeswari, and V Sinthuja. “A review on ontology ranking algorithms.” In: *International Journal of Computer Applications* 33.4 (2011), pp. 6–11 (cit. on p. 166).
- [Sat+20] Abdul Sattar, Ely Salwana Mat Surin, Mohammad Nazir Ahmad, Mazida Ahmad, and Ahmad Kamil Mahmood. “Comparative Analysis of Methodologies for Domain Ontology Development: A Systematic Review.” In: *Int. J. Adv. Comput. Sci. Appl* 11.5 (2020), pp. 99–108 (cit. on p. 141).
- [SB04] Sebastian Schaffert and François Bry. “Querying the Web Reconsidered: A Practical Introduction to Xcerpt.” In: *Extreme Markup Languages*. Citeseer, 2004 (cit. on p. 14).
- [SC18] Ashley Sommer and Nicholas Car. *A Python validator for SHACL*. Accessed: 2024-04-23. 2018. DOI: [10.5281/zenodo.4750840](https://doi.org/10.5281/zenodo.4750840). URL: <https://github.com/RDFLib/pySHACL> (cit. on p. 214).
- [Sch+02] Mario Schlosser, Michael Sintek, Stefan Decker, and Wolfgang Nejdl. “A Scalable and Ontology-based P2P Infrastructure for Semantic Web Services.” In: *Proceedings. Second International Conference on Peer-to-Peer Computing*. IEEE, 2002, pp. 104–111. DOI: [10.1109/PTP.2002.1046319](https://doi.org/10.1109/PTP.2002.1046319) (cit. on p. 100).
- [Sch+13] Craig Schlenoff, Tsai Hong, Connie Liu, Roger Eastman, and Sebti Foufou. “A Literature Review of Sensor Ontologies for Manufacturing Applications.” In: *2013 IEEE International Symposium on Robot and Sensors Environments (ROSE)*. Oct. 2013, pp. 96–101. ISBN: 978-1-4673-2938-5. DOI: [10.1109/ROSE.2013.6698425](https://doi.org/10.1109/ROSE.2013.6698425) (cit. on p. 99).
- [Sch+14] Guus Schreiber, Yves Raimond, Frank Manola, Eric Miller, and Brian McBride. *RDF 1.1 Primer*. W3C Recommendation. Accessed on 2024-05-23. W3C Working Group, 2014. URL: <https://www.w3.org/TR/rdf11-primer/> (cit. on p. 18).
- [Sch+19] Rainer Schiekof, Stephan Grimm, Maja Milicic Brandt, and Michael Weyrich. “A formal mapping between opc ua and the semantic web.” In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. Vol. 1. IEEE, 2019, pp. 33–40. DOI: [10.1109/INDIN41052.2019.8972102](https://doi.org/10.1109/INDIN41052.2019.8972102) (cit. on p. 67).
- [Sch73] Edward W Schneider. *Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis*. Tech. rep. 1973 (cit. on p. 24).
- [Scr+22] Mario Scrocca, Antonia Azzini, Petr Bures, Marco Comerio, and Peter Lubrich. “Towards napDCAT-AP: Roadmap and Requirements for a Transportation Metadata Specification.” In: *Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems (SEMANTiCS 2022), Vienna, Austria, September 13th to 15th, 2022*. Ed. by Umutcan Simsek, David Chaves-Fraga, Tassilo Pellegrini, and Sahar Vahdat. Vol. 3235. CEUR Workshop Proceedings. CEUR-WS.org, 2022. URL: <https://ceur-ws.org/Vol-3235/paper22.pdf> (cit. on p. 16).

- [SDV22] Prashant Kumar Sinha, Biswanath Dutta, and Udaya Varadarajan. “Ranking the Ontology Development Methodologies Using the Weighted Decision Matrix.” In: *Data Technologies and Applications* (2022). DOI: [10.1108/DTA-05-2021-0123](https://doi.org/10.1108/DTA-05-2021-0123) (cit. on p. 140).
- [SE11] Pavel Shvaiko and Jérôme Euzenat. “Ontology matching: state of the art and future challenges.” In: *IEEE Transactions on knowledge and data engineering* 25.1 (2011), pp. 158–176 (cit. on p. 199).
- [Sem09] Semantic Web Company. *PoolParty Semantic Suite*. 2009. URL: <https://www.poolparty.biz/product-overview> (cit. on p. 20).
- [Sem22] Semantic Arts. *gist*. Accessed on 2024-05-23. 2022. URL: <https://www.semanticarts.com/gist/> (cit. on p. 152).
- [SFK19] Gernot Steindl, Thomas Frühwirth, and Wolfgang Kastner. “Ontology-Based OPC UA Data Access via Custom Property Functions.” In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2019, pp. 95–101. DOI: [10.1109/ETFA.2019.8869436](https://doi.org/10.1109/ETFA.2019.8869436) (cit. on p. 67).
- [SH+12] Robyn Speer, Catherine Havasi, et al. “Representing general relational knowledge in ConceptNet 5.” In: *LREC*. Vol. 2012. 2012, pp. 3679–86 (cit. on p. 196).
- [SH19] Cogan Shimizu and Karl Hammar. “CoModIDE - The Comprehensive Modular Ontology Engineering IDE.” In: *ISWC 2019 Satellite Tracks co-located with 18th International Semantic Web Conference (ISWC 2019)*. Vol. 2456. CEUR-WS. 2019, pp. 249–252. URL: <https://ceur-ws.org/Vol-2456/paper65.pdf> (cit. on p. 160).
- [SH99] A Stanford-Clark and U Hunkeler. *MQ Telemetry Transport (MQTT)*. Accessed on 2024-05-23. 1999. URL: <http://mqtt.org/> (cit. on p. 79).
- [SHH19] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. *MODL: A Modular Ontology Design Library*. 2019. arXiv: [1904.05405](https://arxiv.org/abs/1904.05405) [cs.AI] (cit. on pp. 139, 140).
- [SHH21] Cogan Shimizu, Karl Hammar, and Pascal Hitzler. “Modular Ontology Modeling.” In: *Semantic Web Preprint* (2021), pp. 1–31 (cit. on pp. 140, 144).
- [Sil02] N Silva. *Discovering mappings between distributed ontologies*. Tech. rep. Internal Report, University of Karlsruhe, July 2002, 2002 (cit. on p. 198).
- [Sin+12] Amit Singhal et al. “Introducing the Knowledge Graph: Things, not Strings.” In: *Official Google Blog* 5.16 (2012), p. 3. URL: <https://blog.google/products/search/introducing-knowledge-graph-things-not/> (cit. on pp. 24, 168, 170, 178).
- [Sin02] Push Singh. “The open mind common sense project.” In: *KurzweilAI.net* 143 (2002) (cit. on p. 196).
- [SKB05] Barry Smith, Anand Kumar, and Thomas Bittner. “Basic Formal Ontology for Bioinformatics.” In: *IFOMIS Reports 2005* (2005) (cit. on p. 152).
- [SKR18] Román Sosa, Csaba Kiraly, and Juan D Parra Rodriguez. “Offloading execution from edge to cloud: a dynamic node-red based approach.” In: *2018 IEEE International Conference on Cloud Computing Technology and Science (Cloud-Com)*. IEEE. 2018, pp. 149–152. DOI: [10.1109/CloudCom2018.2018.00039](https://doi.org/10.1109/CloudCom2018.2018.00039) (cit. on p. 84).

Bibliography

- [SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. “Yago: A Core of Semantic Knowledge.” In: *Proceedings of the 16th International Conference on World Wide Web*. WWW ’07. Banff, Alberta, Canada: Association for Computing Machinery, 2007, pp. 697–706. ISBN: 9781595936547. DOI: [10.1145/1242572.1242667](https://doi.org/10.1145/1242572.1242667) (cit. on pp. 168, 170, 196).
- [SL22] Anja Strunk and Christoph Lange. *Self-Description of Resources, Service Offerings and Participants within Gaia-X Ecosystems*. Tech. rep. Accessed on 2024-05-23. Avenue des Arts 6-9, 1210 Brussels, Belgium: Gaia-X European Association for Data and Cloud AISBL, Aug. 2022. URL: https://gaia-x.eu/wp-content/uploads/2022/08/SSI_Self_Description_EN_V3.pdf (cit. on pp. 7, 9, 32, 46).
- [SM09] Gunther Schadow and Clement J McDonald. “The Unified Code for Units of Measure.” In: *Regenstrief Institute and UCUM Organization: Indianapolis, IN, USA* (2009). URL: <https://ucum.org/ucum> (cit. on pp. 99, 100).
- [Smi+11] A. Smirnov, A. Kashevnik, N. Shilov, S. Balandin, I. Oliver, and S. Boldyrev. “Principles of ontology matching, translation and interpretation in smart spaces.” In: *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. 2011, pp. 158–162. DOI: [10.1109/CCNC.2011.5766443](https://doi.org/10.1109/CCNC.2011.5766443) (cit. on p. 165).
- [Sol12] K. R. Sollins. “Pervasive persistent identification for information centric networking.” In: *Proceedings of the second edition of the ICN workshop on Information-centric networking*. New York, NY, USA: ACM, 2012, pp. 1–6. ISBN: 9781450314794. DOI: [10.1145/2342488.2342490](https://doi.org/10.1145/2342488.2342490) (cit. on p. 23).
- [SP21] Daniele Spoladore and Elena Pessot. “Collaborative Ontology Engineering Methodologies for the Development of Decision Support Systems: Case Studies in the Healthcare Domain.” In: *Electronics* 10.9 (2021). ISSN: 2079-9292. DOI: [10.3390/electronics10091060](https://doi.org/10.3390/electronics10091060). URL: <https://www.mdpi.com/2079-9292/10/9/1060> (cit. on pp. 139, 140).
- [Spo+19] Manu Sporny, Dave Longley, David Chadwick, Grant Noble, Daniel C. Burnett, Brent Zundel, and Kyle Den Hartog. *Verifiable Credentials Data Model*. W3C Recommendation. The W3C Verifiable Credentials Working Group, Nov. 2019. URL: <https://www.w3.org/TR/vc-data-model/> (cit. on pp. 7, 46).
- [Spo+20] Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Pierre-Antoine Champin, and Niklas Lindström. *JSON-LD 1.1 – A JSON-based Serialization for Linked Data*. W3C Recommendation. The W3C JSON-LD Working Group, July 2020. URL: <https://www.w3.org/TR/json-ld11/> (cit. on pp. 13, 23).
- [SS14] A. Singhal and J. Srivastava. “Leveraging the web for automating tag expansion for low-content items.” In: *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*. 2014, pp. 545–552. DOI: [10.1109/IRI.2014.7051937](https://doi.org/10.1109/IRI.2014.7051937) (cit. on pp. 165, 197).
- [SS15] I.O Sanya and E.M Shehab. “A Framework for Developing Engineering Design Ontologies Within the Aerospace Industry.” In: *International Journal of Production Research* 53.8 (2015), pp. 2383–2409. DOI: [10.1080/00207543.2014.965352](https://doi.org/10.1080/00207543.2014.965352) (cit. on pp. 137, 140).

- [ŠS19] Dušan Šormaz and Arkopaul Sarkar. “SIMPm – Upper-level ontology for manufacturing process plan network generation.” In: *Robotics and Computer-Integrated Manufacturing* 55 (2019). Extended Papers Selected from FAIM2016, pp. 183–198. ISSN: 0736-5845. DOI: [10.1016/j.rcim.2018.04.002](https://doi.org/10.1016/j.rcim.2018.04.002). URL: <https://www.sciencedirect.com/science/article/pii/S0736584517302119> (cit. on p. 152).
- [SSS04] York Sure, Steffen Staab, and Rudi Studer. “On-to-knowledge methodology (OTKM).” In: *Handbook on ontologies*. Springer, 2004, pp. 117–132 (cit. on p. 144).
- [SSS09] York Sure, Steffen Staab, and Rudi Studer. “Ontology Engineering Methodology.” In: *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 135–152. ISBN: 978-3-540-92673-3. DOI: [10.1007/978-3-540-92673-3_6](https://doi.org/10.1007/978-3-540-92673-3_6) (cit. on pp. 140, 144).
- [Ste+21] Sebastian Steinbuß, Andreas Eitel, Christian Jung, Robin Brandstädter, Arghavan Hosseinzadeh, Sebastian Bader, Christian Kühnle, Pascal Birnstill, Gerd Brost, Mark Gall, Fabian Bruckner, Norbert Weißenberg, and Benjamin Korth. *Usage Control in the International Data Spaces*. Tech. rep. Mar. 2021. DOI: [10.5281/zenodo.5675884](https://doi.org/10.5281/zenodo.5675884) (cit. on p. 28).
- [STM08] Peter Spyns, Yan Tang, and Robert Meersman. “An Ontology Engineering Methodology for DOGMA.” In: *Applied Ontology* 3.1-2 (2008), pp. 13–39 (cit. on p. 144).
- [Sye+10] Zareen Syed, Tim Finin, Varish Mulwad, Anupam Joshi, et al. “Exploiting a web of semantic data for interpreting tables.” In: *Proceedings of the Second Web Science Conference*. 2010 (cit. on pp. 19, 165).
- [TA07] Samir Tartir and I Budak Arpinar. “Ontology evaluation and ranking using OntoQA.” In: *International conference on semantic computing (ICSC 2007)*. IEEE. 2007, pp. 185–192 (cit. on pp. 166, 167).
- [Tah+14] Mohsen Taheriyan, Craig A Knoblock, Pedro Szekely, and José Luis Ambite. “A scalable approach to learn semantic models of structured sources.” In: *2014 IEEE International Conference on Semantic Computing*. IEEE. 2014, pp. 183–190 (cit. on pp. 194, 195).
- [Tah+15] Mohsen Taheriyan, Craig A Knoblock, Pedro A Szekely, José Luis Ambite, and Yinyi Chen. “Leveraging Linked Data to Infer Semantic Relations within Structured Sources.” In: *COLD*. Citeseer. 2015 (cit. on pp. 19, 165, 195).
- [Tah+16] Mohsen Taheriyan, Craig A Knoblock, Pedro Szekely, and José Luis Ambite. “Learning the semantics of structured data sources.” In: *Journal of Web Semantics* 37 (2016), pp. 152–169. DOI: [10.1016/j.websem.2015.12.003](https://doi.org/10.1016/j.websem.2015.12.003) (cit. on pp. 19, 165, 195).
- [Tay95] Barry Taylor. *Guide for the Use of the International System of Units (SI): The Metric System*. DIANE Publishing, 1995 (cit. on p. 99).
- [TBS16] Eike Tauscher, Hans-Joachim Bargstädt, and Kay Smarsly. “Generic BIM queries based on the IFC object model using graph theory.” In: *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering, Osaka, Japan*. 2016, pp. 6–8 (cit. on p. 117).

Bibliography

- [TD16] Guenter Toesko and Christian Dehnert. “Femtosecond laser optics combat pulse dispersion, color errors, and reflections.” In: *Pulse* 1 (2016) (cit. on p. 97).
- [Tes+08] Maurizio Tesconi, Francesco Ronzano, Andrea Marchetti, and Salvatore Minutoli. “Semantify del.icio.us: Automatically turn your tags into senses.” In: *The 7th International Semantic Web Conference*. Citeseer, 2008, p. 67 (cit. on pp. 194, 198).
- [The+22a] Johannes Theissen-Lipp, Max Kocher, Stefan Josef Decker, Patrick Sapel, Mauritius Schmitz, and Christian Hopmann. “Collecting Data and Metadata by Transforming between Differently Expressive Query Languages.” In: *Proceedings of the 16th International Conference on Semantic Computing*. Contributions: JTL and MK identified the research gap and designed and conceived the work. PS and MS provided technical infrastructure and injection molding expertise. JTL, MK, and MS performed, analyzed, and interpreted the work. JTL, MK, and PS drafted and revised the paper. SD and MS reviewed the paper and provided feedback. SD and CH provided guidance and funding. 2022. DOI: [10.1109/ICSC52841.2022.00044](https://doi.org/10.1109/ICSC52841.2022.00044). URL: <https://publications.rwth-aachen.de/record/842364> (cit. on pp. 66, 67, 69–71, 74–76).
- [The+22b] Johannes Theissen-Lipp, Max Kocher, Michael Rath, Sebastian Ulrich, Maximilian Rudack, and Stefan Decker. “Integrating an XPath-Enhanced OPC UA Data Collection Into Industrial Communication.” In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. Contributions: JTL and MK identified the research gap and designed and conceived the work. MRu provided technical infrastructure and high pressure die casting expertise. JTL, MK, SU, and MRa performed, analyzed, and interpreted the work. JTL, MK, MRa, SU, and MRu drafted and revised the paper. SD provided feedback, guidance, and funding. IEEE, 2022, pp. 1–8. ISBN: 978-1-6654-9996-5. DOI: [10.1109/ETFA52439.2022.9921562](https://doi.org/10.1109/ETFA52439.2022.9921562) (cit. on p. 77).
- [The+22c] Johannes Theissen-Lipp, Moritz Kröger, Benedikt Heinrichs, and Stefan Decker. “LISSU: Continuous Monitoring of SOA Communication with Constraint-Based Validation.” In: *SN Computer Science* 3.4 (2022). Was honored with the **Young Researcher Award 2022** by the [Cluster of Excellence](#). Contributions: JL and MK designed, conceived, and interpreted the work. BH, JL, and MK performed, analyzed, and evaluated the work. JL, MK, and BH drafted and revised the paper. SD provided funding, feedback, and guidance. DOI: [10.1007/s42979-022-01170-5](https://doi.org/10.1007/s42979-022-01170-5) (cit. on pp. 102, 104, 109–111, 113).
- [The+23] Johannes Theissen-Lipp, Max Kocher, Christoph Lange, Stefan Decker, Alexander Paulus, André Pomp, and Edward Curry. “Semantics in Dataspace: Origin and Future Directions.” In: *Companion Proceedings of the ACM Web Conference 2023*. WWW ’23 Companion. Contributions: JTL and SD identified the research gap and designed and conceived the work. JTL and MK performed, analyzed, and interpreted the work. JTL, MK, APo, and APa drafted and revised the paper. SD and EC provided feedback and guidance. SD provided funding. Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1504–1507. ISBN: 9781450394192. DOI: [10.1145/3543873.3587689](https://doi.org/10.1145/3543873.3587689) (cit. on pp. 6, 24).

- [The+24] Johannes Theissen-Lipp, Niklas Schäfer, Max Kocher, Philipp Hochmann, Michael Riesener, and Stefan Decker. “Extending Semantic RML Mappings with Additional Source Formats.” In: *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*. Contributions: JTL identified the research gap. JTL and NS designed and conceived the work. PH, JTL, and NS performed, analyzed, and interpreted the work. JTL, MK, and NS drafted and revised the paper. SD and MR provided guidance and funding. INSTICC. SciTePress, 2024, pp. 201–208 (cit. on pp. 118–120, 122, 124).
- [The04] The Linux Information Project. *GUI Definition – Advantages of GUIs*. Tech. rep. Oct. 2004. URL: <https://www.linfo.org/gui.html> (cit. on p. 158).
- [The11a] The Apache Software Foundation. *Apache Jena*. 2011. URL: <https://jena.apache.org/> (cit. on p. 20).
- [The11b] The Apache Software Foundation. *Apache Jena Shacl*. Accessed: 2024-04-23. 2011. URL: <https://jena.apache.org/documentation/shacl/index.html> (cit. on p. 211).
- [The18] The Apache Software Foundation. *Apache StreamPipes (incubating)*. Accessed on 2024-05-23. 2018. URL: <https://streampipes.apache.org/> (cit. on p. 83).
- [TK18] Silviu-Iulian Toc and Adrian Korodi. “Modbus-OPC UA Wrapper using Node-RED and IoT-2040 with application in the water industry.” In: *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE. 2018, pp. 000099–000104. DOI: [10.1109/SISY.2018.8524749](https://doi.org/10.1109/SISY.2018.8524749) (cit. on p. 84).
- [TKC18] Nikolaos Trokanas, Linsey Koo, and Franjo Cecelja. “Towards a Methodology for Reusable Ontology Engineering: Application to the Process Engineering Domain.” In: *28th European Symposium on Computer Aided Process Engineering*. Ed. by Anton Friedl, Jiří J. Klemeš, Stefan Radl, Petar S. Varbanov, and Thomas Wallek. Vol. 43. Computer Aided Chemical Engineering. Elsevier, 2018, pp. 471–476. DOI: [10.1016/B978-0-444-64235-6.50084-X](https://doi.org/10.1016/B978-0-444-64235-6.50084-X). URL: <https://www.sciencedirect.com/science/article/pii/B978044464235650084X> (cit. on p. 144).
- [Tom17] Dominik Tomaszuk. “RDF validation: a brief survey.” In: *International Conference: Beyond Databases, Architectures and Structures*. Springer. 2017, pp. 344–355 (cit. on pp. 210, 214).
- [Top15] TopQuadrant. *Topbraid SHACL API*. Accessed: 2024-04-23. 2015. URL: <https://github.com/TopQuadrant/shacl/tree/d566c6b955cb8ec63ca32129bfb41b358ac07e31> (cit. on p. 211).
- [Top17] TopBraid. *TopBraid Enterprise Data Geovernance*. Accessed: 2024-04-23. 2017. URL: <https://www.topquadrant.com/topbraid-edg/> (cit. on p. 211).
- [Top20] TopQuadrant. *TopBraid Composer*. Accessed on 2022-01-28. 2020. URL: <https://topquadrant.com/products/topbraid-composer/> (cit. on pp. 20, 160).

Bibliography

- [Tor+11] Diego Torres, Alicia Diaz, Hala Skaf-Molli, and Pascal Molli. “Semdrops: A social semantic tagging approach for emerging semantic data.” In: *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. Vol. 1. IEEE. 2011, pp. 340–347 (cit. on pp. 194, 199).
- [Tre11] Andrew Treloar. *Den Haag Persistent Object Identifier – Linked Open Data Manifesto*. Tech. rep. 2011. DOI: [10.5281/ZENODO.55666](https://doi.org/10.5281/ZENODO.55666) (cit. on p. 23).
- [TT08] Ambler Thompson and Barry N Taylor. *Guide for the Use of the International System of Units (SI)*. Tech. rep. 2008 (cit. on p. 100).
- [Tud+13] Tania Tudorache, Csongor Nyulas, Natalya F Noy, and Mark A Musen. “WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web.” In: *Semantic Web 4.1* (2013), pp. 89–99. DOI: [10.3233/SW-2012-0057](https://doi.org/10.3233/SW-2012-0057) (cit. on pp. 20, 160).
- [Tud20] Tania Tudorache. “Ontology Engineering: Current State, Challenges, and Future Directions.” In: *Semantic Web 11.1* (2020), pp. 125–138 (cit. on p. 140).
- [UG04] Michael Uschold and Michael Gruninger. “Ontologies and semantics for seamless connectivity.” In: *ACM SIGMod Record* 33.4 (2004), pp. 58–64 (cit. on p. 198).
- [UK 18] UK Data Service. *Research Data Lifecycle*. Accessed on 2024-05-23. 2018. URL: <https://ukdataservice.ac.uk/learning-hub/research-data-management/> (cit. on p. 42).
- [Usc96] Mike Uschold. “Building Ontologies: Towards a Unified Methodology.” In: *Proceedings of 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems*. Citeseer. 1996 (cit. on p. 144).
- [Van+17a] Pierre-Yves Vandenbussche, Ghislain A Atemezine, María Poveda-Villalón, and Bernard Vatant. “Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web.” In: *Semantic Web 8.3* (2017), pp. 437–452 (cit. on pp. 168, 182, 187).
- [Van+17b] Pierre-Yves Vandenbussche, Ghislain A Atemezine, María Poveda-Villalón, and Bernard Vatant. “Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web.” In: *Semantic Web 8.3* (2017), pp. 437–452 (cit. on p. 178).
- [VDI20] VDI/VDE 2193. *Language for I4.0 Components - Part 1: Structure of messages*. Düsseldorf, 2020 (cit. on p. 66).
- [Ver22] R. Verborgh. “Re-decentralizing the Web, for good this time.” In: *Linking the World’s Information: A Collection of Essays on the Work of Sir Tim Berners-Lee*. Ed. by Oshani Seneviratne and James Hendler. ACM, 2022 (cit. on pp. 8, 9).
- [VH20] Minh Hoang Lien Vo and Quang Hoang. “Transformation of UML class diagram into OWL Ontology.” In: *Journal of Information and Telecommunication* 4.1 (2020), pp. 1–16. DOI: [10.1080/24751839.2019.1686681](https://doi.org/10.1080/24751839.2019.1686681) (cit. on p. 117).

- [Vig+14] Markel Vigo, Samantha Bail, Caroline Jay, and Robert Stevens. “Overcoming the Pitfalls of Ontology Authoring: Strategies and Implications for Tool Design.” In: *International Journal of Human-Computer Studies* 72.12 (2014), pp. 835–845. ISSN: 1071-5819. DOI: [10.1016/j.ijhcs.2014.07.005](https://doi.org/10.1016/j.ijhcs.2014.07.005). URL: <https://www.sciencedirect.com/science/article/pii/S1071581914001013> (cit. on pp. 137, 139).
- [VK14] Denny Vrandečić and Markus Krötzsch. “Wikidata: a free collaborative knowledgebase.” In: *Communications of the ACM* 57.10 (2014), pp. 78–85. DOI: [10.1145/2629489](https://doi.org/10.1145/2629489) (cit. on pp. 22, 133, 168, 169).
- [Vos+08] Piek Vossen, Eneko Agirre, Nicoletta Calzolari, Christiane Fellbaum, Shu-kai Hsieh, Chu-Ren Huang, Hitoshi Isahara, Kyoko Kanzaki, Andrea Marchetti, Monica Monachini, Federico Neri, Remo Raffaelli, German Rigau, Maurizio Tescon, and Joop VanGent. “KYOTO: a System for Mining, Structuring and Distributing Knowledge across Languages and Cultures.” In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/373_paper.pdf (cit. on p. 152).
- [VS09] Panos Vassiliadis and Alkis Simitis. “Extraction, Transformation, and Loading.” In: *Encyclopedia of Database Systems* 10 (2009). DOI: [10.1007/978-0-387-39940-9_158](https://doi.org/10.1007/978-0-387-39940-9_158) (cit. on p. 82).
- [W3C19] W3C Solid Community Group. *Solid: Your data, your choice*. Accessed on 2024-05-23. 2019. URL: <https://solidproject.org/> (cit. on p. 9).
- [Wac+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. “Ontology-based Integration of Information - A Survey of Existing Approaches.” In: *Ois@ ijcai*. 2001, pp. 108–117 (cit. on p. 138).
- [Wan+12] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. “Understanding tables on the web.” In: *Conceptual Modeling: 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings 31*. Springer. 2012, pp. 141–155 (cit. on pp. 19, 165).
- [Wan+17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. “Knowledge graph embedding: A survey of approaches and applications.” In: *IEEE transactions on knowledge and data engineering* 29.12 (2017), pp. 2724–2743 (cit. on p. 24).
- [WC09] Yu Wei and Junpeng Chen. “Ontology ranking for the semantic web.” In: *2009 Third International Symposium on Intelligent Information Technology Application*. Vol. 1. IEEE. 2009, pp. 573–574. DOI: [10.1109/IITA.2009.279](https://doi.org/10.1109/IITA.2009.279) (cit. on p. 166).
- [Wil+16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, et al. “The FAIR Guiding Principles for scientific data management and stewardship.” In: *Scientific Data* 3.1 (Mar. 2016). DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18) (cit. on pp. 8, 23, 32).
- [Win18] Jeannette M. Wing. *The Data Life Cycle*. Tech. rep. Accessed on 2024-05-23. New York, NY 10027, USA: The Data Science Institute at Columbia University, Jan. 2018. URL: <https://datascience.columbia.edu/news/2018/the-data-life-cycle/> (cit. on pp. 33, 43, 44).

Bibliography

- [WLA18] Vitalis Wiens, Steffen Lohmann, and Sören Auer. “WebVOWL Editor: Device-Independent Visual Ontology Modeling.” In: *International Semantic Web Conference (P&D/Industry/BlueSky)*. 2018. URL: <http://ceur-ws.org/Vol-2180/paper-75.pdf> (cit. on pp. 20, 159, 160).
- [Woh14] Claes Wohlin. “Guidelines for snowballing in systematic literature studies and a replication in software engineering.” In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10 (cit. on p. 141).
- [Wor09] World Wide Web Consortium. *Web Standards*. Tech. rep. Wakefield, Massachusetts, USA: W3C, 2009. URL: <https://www.w3.org/standards/> (cit. on pp. 7, 12).
- [Xia+16] Liumingjing Xiao, Chong Ruan, An Yang, Junhao Zhang, and Junfeng Hu. “Domain ontology learning enhanced by optimized relation instance in dbpedia.” In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 1452–1456 (cit. on pp. 194, 199).
- [Yab09] Sergey Yablonsky. “Semantic Web Framework for Development of Very Large Ontologies.” In: *Polibits* 39 (2009), pp. 19–26 (cit. on p. 152).
- [YBA21] Muhammad Yahya, John G. Breslin, and Muhammad Intizar Ali. “Semantic Web and Knowledge Graphs for Industry 4.0.” In: *Applied Sciences* 11.11 (2021). ISSN: 2076-3417. DOI: [10.3390/app11115110](https://doi.org/10.3390/app11115110). URL: <https://www.mdpi.com/2076-3417/11/11/5110> (cit. on pp. 140, 154).
- [Yun+11] Hongyan Yun, Jianliang Xu, Jing Xiong, and Moji Wei. “A Knowledge Engineering Approach to Develop Domain Ontology.” In: *International Journal of Distance Education Technologies (IJDET)* 9.1 (2011), pp. 57–71 (cit. on p. 144).
- [Yun+19] Arda Yunianta, Ahmad Hoirul Basori, Anton Satria Prabuwono, Arif Bramantoro, Irfan Syamsuddin, Norazah Yusof, Alaa Omran Almagrabi, and Khalid Alsubhi. “OntoDI: The methodology for ontology development on data integration.” In: *International Journal of Advanced Computer Science and Applications* 10.1 (2019) (cit. on p. 144).
- [Zab+15] Fouad Zablith, Grigoris Antoniou, Mathieu d’Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, and Marta Sabou. “Ontology evolution: a process-centric survey.” In: *The knowledge engineering review* 30.1 (2015), pp. 45–75 (cit. on p. 198).
- [Zaz21] Zazuko GmbH. *ZAZUKO SHACL Playground*. Accessed: 2024-04-23. 2021. URL: <https://shacl-playground.zazuko.com/> (cit. on p. 211).
- [ZBV18] Chi Zhang, Jakob Beetz, and de Vries. “BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data.” In: *Semantic Web* (Aug. 2018), pp. 1–27. DOI: [10.3233/SW-180297](https://doi.org/10.3233/SW-180297) (cit. on p. 117).
- [Zej21] Enis Zejnilovic. “A Batch Recommender for Fast Ontology Prototyping.” Advisors: Johannes Lipp and Lars Gleim. Master’s thesis. RWTH Aachen University, 2021 (cit. on pp. 175–180, 182, 184, 187–189).

- [Zen12] Wolfgang Zenk-Möltgen. “Metadaten und die Data Documentation Initiative (DDI).” In: *Langzeitarchivierung von Forschungsdaten: Standards und disziplinspezifische Lösungen*. Ed. by Reinhard Altenhöner and Claudia Oellers. Berlin: Scivero Verl., 2012, pp. 111–126. ISBN: 978-3-944417-00-4. URL: <https://nbn-resolving.org/urn:nbn:de:0168-ssoar-46679-8> (cit. on pp. 34, 35).
- [Zhe+12] Ye-Lu Zheng, Qi-Yun He, QIAN Ping, and LI Ze. “Construction of the Ontology-based Agricultural Knowledge Management System.” In: *Journal of Integrative Agriculture* 11.5 (2012), pp. 700–709 (cit. on p. 144).
- [Zho+21] Xinjie Zhou, Guyue Gao, Xinguo Ming, Liya Wang, Dao Yin, and Xiaohong Ma. “Task-oriented Knowledge Representation and Ontology Modeling for Complex Product Design.” In: *2021 International Conference on Service Science (ICSS)*. 2021, pp. 30–37. DOI: [10.1109/ICSS53362.2021.00013](https://doi.org/10.1109/ICSS53362.2021.00013) (cit. on pp. 138–140).
- [Zie04] Cai-Nicolas Ziegler. “Semantic web recommender systems.” In: *International Conference on Extending Database Technology*. Springer. 2004, pp. 78–89 (cit. on pp. 166, 167).
- [ZMG08] Torsten Zesch, Christof Müller, and Iryna Gurevych. “Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary.” In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation*. Vol. 8. European Language Resources Association (ELRA), 2008, pp. 1646–1652 (cit. on p. 195).

List of Figures

1.1	In today's era of ever-increasing data, dataspace promise control over data.	1
2.1	A tweaked Semantic Web stack proposed by [Ide17], based on the original proposal of Tim Berners-Lee from 2000 [Ber00].	12
2.2	Illustration of the main Semantic Web application areas within the Internet of Production project [LS20].	26
2.3	The Mobility Data Space promotes data sovereignty through data usage control mechanisms [Dre+21].	27
2.4	Connected participants and their data offerings in the Mobility Data Space project [Dre+21].	28
3.1	Our methodology for developing a DLCM for dataspace.	31
3.2	Abstract representation of this DLCM based on the <i>Data Sharing Verbs</i> [BT09] by the Australian National Data Service.	34
3.3	Abstract representation of this DLCM based on the <i>Data Documentation Initiative's combined lifecycle model</i> [Zen12; Gro04].	35
3.4	Abstract representation of this DLCM based on the <i>Linked Data lifecycle</i> [ALN11].	35
3.5	Abstract representation of this DLCM based on the <i>capability maturity model for scientific data management</i> [CQ11].	36
3.6	Abstract representation of this DLCM based on the <i>DataOne project</i> [MJ12].	37
3.7	Abstract representation of this DLCM based on the <i>Digital Curation Centre's Curation Lifecycle Model</i> [Hig08].	38
3.8	Abstract representation of this DLCM based on the <i>GFBio</i> [GFB22].	40
3.9	Abstract representation of this DLCM based on the <i>data lifecycle for macrosystems ecology projects</i> [Rüe+14].	41
3.10	Abstract representation of this DLCM based on the <i>UK Data Archive</i> [Eyn13].	43
3.11	Abstract representation of this DLCM based on the <i>Data Science Institute of Columbia University</i> [Win18].	44
3.12	Abstract representation of our proposed DLCM based on <i>GFBio</i> [GFB22] (see section 3.2.1 for details).	49
3.13	The proposed DLCM for dataspace is addressed on three layers throughout this thesis.	50
4.1	This chapter focuses on the <i>dataspace layer</i> of the DLCM layer model presented in figure 3.13.	51
4.2	A sample instance of the IDS Information Model [Bad+20b].	53
4.3	The IDS Information Model addresses several dataspace concerns [Bad+20b]. Gray boxes indicate the reuse of existing standards.	55
4.4	The update and release process of the IDS Information Model [Bad+20b].	57
4.5	The refined <i>Digital Shadow Reference Model</i> [Mic+23].	58

List of Figures

4.6	Example application of the Digital Shadow Reference Model to a process control use case in injection molding [Bec+21; Mic+23].	60
5.1	This chapter focuses on the <i>data layer</i> of the DLCM layer model presented in figure 3.13. We explore principles for accessing data from heterogeneous sources to make relevant data available for integrated use in dataspace. . . .	64
5.2	Real-world example demonstrating the high manual effort required to inspect and manage data and metadata from OPC UA [The+22a; Koc21].	66
5.3	Our approach [The+22a; Koc21] involves mappings between differently expressive query languages and their underlying data models.	67
5.4	Implementing our approach involves mapping the graph data format (a) to the tree data format (b).	68
5.5	We demonstrate our methodology by proposing this workflow for a common manufacturing use case (see section 2.3), mapping XPath to OPC UA [The+22a; Koc21].	69
5.6	Our approach to overcoming the differences in the data structures graph and tree [The+22a; Koc21].	70
5.7	Our proposed mapping [The+22a; Koc21] of all twelve XPath axes to the two native OPC UA methods.	71
5.8	Example of transforming an XPath query into executable code for OPC UA [The+22a; Koc21].	71
5.9	Photo of an injection molding machine (a) and a cross section of the machine (b).	73
5.10	Execution times of the OPC UA export [The+22a; Koc21] for the evaluation scenarios from table 5.1.	74
5.11	Post-processing evaluation results from an OPC UA export to XPath-compatible XML [The+22a; Koc21].	75
5.12	Average execution times for <i>preceding</i> on exemplary nodes in the different scenarios [The+22a; Koc21].	76
5.13	Average runtime of the OPC UA-compliant code in listing 5.2 generated from the XPath expression of dataspace participants shown in listing 5.1.	77
5.14	<i>Frech K-Series</i> cold chamber HPDC machine (image copyright ©Oskar Frech GmbH + Co. KG).	80
5.15	The HPDC machine from our use case has many sensors, including 13 temperature sensors [Lip+20].	81
5.16	The common data integration processes extract-transform-load and extract-load-transform [Lip+20].	82
5.17	Our proposed extension to the <i>extract</i> data collection stage (see figure 5.16) is based on the phase model from listing 5.3 [Lip+20].	85
5.18	Our analysis of a sample HPDC process (see section 5.3.1) yields phases and transition conditions [Lip+20]. Image copyright ©Magmasoft 5.4 [MAG19].	86
5.19	Three-zone network architecture for the HPDC machine use case [Lip+21c].	89
5.20	Window for configuring the start and end conditions of a process phase in Node-RED [Lip+21c].	90
5.21	Configuration window of the data collection strategy [Lip+21c].	90
5.22	Example Node-RED setup with multiple nodes and data sources [Lip+21c]. .	91
5.23	Overview of the HPDC process modeled as a Node-RED flow [Lip+21c]. . .	92

5.24	A USP laser system (a) consists of several components that may be frequently replaced [Krö+24] and forms objects by (b) selectively removing material layer by layer [Mic+23].	97
5.25	Example of a semantic mismatch that occurs during the replacement of components on a USP laser system [Lip+21d].	98
5.26	Our contribution (right) adds semantic units and validation based on ontology URIs to the baseline (left) [The+22c].	102
5.27	Our approach for more reliable communication between two communication partners (e.g. client and server) [The+22c].	102
5.28	Activity diagram of our proposed initial semantic validation workflow (steps 1 to 3 in figure 5.27), which detects and recovers semantic mismatches [Lip+21d].	103
5.29	The proposed workflow for continuous monitoring of data streams [The+22c].	104
5.30	Experimental results for mapping data from streams in the continuous semantic validation service [The+22c].	113
5.31	Our approach to proposing a general workflow for extending RML with new formats [Hoc22; The+24].	118
5.32	Architecture of our extended RMLMapper, which enables the communication of IFC files, user-defined queries, and a final mapping to RDF output.	121
5.33	Extract from the production rules of the proposed context-free grammar for an RML iterator for UML class diagrams [The+24; Hoc22].	122
5.34	Visual representation of an IFC file representing a factory building [The+24; Hoc22].	124
6.1	This chapter focuses on the <i>domain layer</i> of the DLCM layer model presented in figure 3.13.	128
6.2	Semantic pitfalls of RDFS and OWL ontology reuse, illustrated by example of domain and range entailment [LGD20].	133
6.3	Our proposed modifications (marked with colors) to the semantic layer model so that it no longer merges terms, shapes, and rules into ontologies, but makes them available independently [LS20].	134
6.4	Based on [NM+01], the iterative steps required to develop an ontology [Com22a].	138
6.5	Based on [OVP19], the classification of ontologies [Com22a].	139
6.6	The methodology we use in this section to reach to the development and evaluation of best practices for vocabulary creation [Com22a].	141
6.7	Our methodology to evaluate methodologies for ontology development starts with a literature review and incrementally evaluates the findings in multiple steps [Com22a].	142
6.8	The outcome of this section is a guideline booklet for the process of ontology development, which contains fundamental concepts, methodologies, a suggested workflow, templates, and ontology catalogs [Com22a].	143
6.9	Description of criteria defined based on the literature review to evaluate methodologies for ontology development [Com22a].	145
6.10	Description of criteria defined based on our analysis of questionnaires results to evaluate methodologies for ontology development [Com22a].	145
6.11	Description of criteria defined based on literature review and our analysis of questionnaires results to evaluate tools for ontology development [Com22a]. .	151

List of Figures

6.12	The proposal of a workflow to use during the process of ontology development considering close collaboration between domain experts and knowledge engineers [Com22a].	153
6.13	Results of the <i>OOPS!</i> evaluation of the initial version of the ontology created by a domain expert in the use case of our evaluation [Com22a].	156
6.14	The proposed vocabulary drafting process is embedded into a typical user workflow [Lip+21b].	159
6.15	Neologism 2.0’s frontend communicates with a backend to persist information, and can interact with a recommender to improve the quality of drafted vocabularies by reusing existing vocabularies [Lip+21b].	161
6.16	Screenshots of class and property creation in Neologism 2.0 [Lip+21b; Lip+21a].	162
6.17	Labels (attributes) of an exemplary dataset and a corresponding semantic model [Pau+18].	172
6.18	Our approach to enable recommendations for the application areas <i>semantic modeling</i> and <i>vocabulary development</i>	172
6.19	An example execution of our approach presented in figure 6.18.	173
6.20	A screenshot of the suggestions generated by the <code>ngx-spellchecker</code> in Neologism 2.0 [Zej21] for a misspelling of <i>teacher</i> as <i>techer</i>	175
6.21	Screenshot of a vocabulary created in Neologism 2.0 on the right and controls on the left [Zej21].	176
6.22	A connector designed for each knowledge base is used to collect recommendations from it [Pau+19].	178
6.23	Extract of a response to the example query shown in listing 6.2 for the keyword <i>student</i> [Zej21].	179
6.24	The recommendation ranking task involves processing recommendations from multiple connectors and applying metrics to calculate scores for an overall recommendation score [Pau+19].	180
6.25	Example calculation of the sub-scores ω_1 to ω_5 for inter source ranking [Pau+18].	183
6.26	In the ESKAPE platform [Pom+18a], recommendations can be applied to a semantic model via a pop-up menu [Pau+19].	183
6.27	Our extension for Neologism [Zej21] includes showing additional information for each offered recommendation from the list displayed in figure 6.23.	184
6.28	Excerpt of the System Usability Scale [Bro13] questionnaire adapted and used for the second evaluation scenario <i>vocabulary development</i> [Zej21] in Neologism.	188
6.29	Excerpt of the After-Scenario Questionnaire [Lew91] used for the second evaluation scenario <i>vocabulary development</i> [Zej21] in Neologism.	188
6.30	Excerpts of the three publicly available datasets used for the second evaluation scenario [Zej21], including the domains <i>weather</i> , <i>Covid-19 test stations</i> , and <i>Open Data datasets</i> . For more details, see [Zej21].	189
6.31	Performance comparison of our recommender system and the baseline <i>Linked Open Vocabularies</i> for (a) classes and (b) properties.	190
6.32	Effectiveness of the inter and intra source rankings of our proposed recommender system.	190
6.33	The results of the <i>System Usability Scale</i> questionnaire (see figure 6.28) indicate an excellent usability of the proposed recommender system.	191
6.34	User experience and satisfaction results from the <i>After-Scenario Questionnaire</i> (see figure 6.29).	191
6.35	A cluster of related concepts in <i>ConceptNet 5</i> [SH+12].	196

6.36	The Waking and Sleeping algorithm classifies user tags into good tags (waking) and bad tags (sleeping) [Du+12; Lip17].	197
6.37	Workflow of our approach to continuously evolve a KG based on user needs during semantic annotation [PLM19a; Lip17].	201
6.38	Our approach focuses on continuously integrating information from external knowledge bases into a local KG during semantic annotation.	202
6.39	Screenshot of a sample concept generated by our proposed user wizard from a knowledge base [Lip17].	203
6.40	The average Jaccard index of each attendee’s semantic model with those of others for the (a) <i>crime</i> and (b) <i>parking</i> datasets introduced in table 6.12, comparing our assisted wizard with a no-tool baseline [PLM19a; Lip17]. . . .	205
6.41	Violin plots [Lip17] of time spent for (a) nonspecialists and (b) experts. . . .	207
6.42	The first part of our two-part approach [Moo22; Moo+24] is to introduce shortcuts for commonly used constraints that are not yet natively supported by validation languages.	212
6.43	The methodology of the first part of our two-part approach [Moo22; Moo+24], which introduces SHACL-SPARQL templates as shortcuts for commonly used constraints not natively supported by SHACL.	212
6.44	The second part of our two-part approach [Moo22; Moo+24] is to modify a current common practice for RDF validation with SHACL.	213
6.45	The complete solution of our two-step approach [Moo22; Moo+24], including scaling inference (see figure 6.44) and constraint template mapping (see figure 6.42).	213
6.46	Example execution for a constraint that all labels and comments have at least one English value, using our proposed constraint mapping tool.	215
6.47	Example execution of a subproperty inference.	216
6.48	Overview of the four example instances of providers from the Gaia-X dataspace used in this evaluation.	217
6.49	Visualization of the average execution times of our experiments (see table 6.20).	219
7.1	While conventional recommender systems (see section 6.5) could not find a concept for a specialized term, ChatGPT could.	226

List of Tables

2.1	Overview of dataspace initiatives, their scope, and goals.	9
2.2	Semantic Web tools used in dataspace initiatives such as IDS and Gaia-X. . .	20
2.3	5-star deployment scheme for (Linked) Open Data from 5stardata.info. . . .	22
3.1	Overview of general DLCMs included in our survey.	33
3.2	Evaluation of which DLCMs meet which dataspace requirements (R1 is short for requirement DLCM-R1).	44
3.3	Evaluation of which dataspace initiatives meet which DLCM requirements (R1 is short for requirement DLCM-R1).	48
4.1	Key facts about the IDS Information Model version 4.0.0 and related resources [Bad+20b].	54
4.2	Evaluation of the IDS Information Model with the approach and requirements from this chapter. Scores range from (–) to (++).	54
4.3	Evaluation of the <i>Digital Shadow Reference Model</i> with the approach and requirements from this chapter. Scores range from (–) to (++).	59
4.4	Summary of how the two example information models satisfy the requirements for dataspace information models.	61
5.1	Evaluation scenarios representing different use cases for OPC UA access. . .	72
5.2	Overview of frameworks and tools for visual programming [Lip+21c].	83
5.3	Expert knowledge about the phases of the HPDC production process [Lip+20].	93
5.4	The measured number of data points collected for a value of interest during a HPDC production cycle [Lip+20].	94
5.5	Evaluation of relevant ontologies in the areas of sensors and engineering units based on the requirements from section 5.4.1 [Lip+21d].	99
5.6	A summary of our approach to addressing requirements derived from the motivating use case (see section 5.4.1).	101
5.7	Example variables with their data types and units of measure for the real-world use case [Lip+21d].	106
5.8	Comparison of our solution and the baseline with expected values for the four evaluation scenarios [Lip+21d].	112
6.1	Reusability of the individual concepts terms, structure, and reasoning [LGD20].	134
6.2	Guidelines and principles for ontology development [Com22a], distinguishing between defined principles (PRN) and other considerations (OTH).	143
6.3	Methodologies for ontology development and their primary focus over the years [Com22a]. The last column indicates whether the methodologies meet practical requirements.	144
6.4	Evaluation of methodologies for ontology development. The final results are in descending order according to the final scores obtained by each methodology [Com22a]. See table 6.3 for details and citations for the approaches. . . .	146

6.5	Technical tools supporting the ontology development process [Com22a]. The <i>REF</i> column indicates online references of the tools.	147
6.6	Tools supporting multiple stages of the ontology development process [Com22a]. The <i>REF</i> column indicates online references of the tools.	148
6.6	The tool recommendations for the process of ontology development, grouped by their typical use [Com22a].	151
6.7	Upper and middle ontologies to consider in ontology development [Com22a].	152
6.8	Existing manufacturing domain ontologies and their addressed fields (see section 2.3) [Com22a]. Acronyms are explained in the text reference for the table.	154
6.9	Ontology catalogs with their link and description [Com22a]. The last column indicates whether the catalogs meet practical requirements.	155
6.10	Comparison of ontology drafting tools based on [Lip+21a]. Note that this comparison adds to sections 2.2.1 and 6.3.3.	160
6.11	Overview of general knowledge bases. Based on contents from [Fär+15; Pau+19]. Note that the <i>name</i> column includes links to the respective home-pages.	168
6.12	The name, source, number of labels, and description of the publicly available datasets used to evaluate our proposed recommender system for the task of <i>semantic modeling</i>	185
6.13	Heat map of how often, on average, recommendations from the top three list positions were chosen in different scenarios.	186
6.14	Detailed qualitative comparison between the results of Babelfy and our proposed recommender system for semantic modeling.	187
6.15	Examples of dataset assets that may require updates over time.	193
6.16	Related approaches in the area of semantic annotation that maintain a knowledge representation including one or more ontologies [PLM19b; Lip17]. . . .	194
6.17	Qualitative evaluation [Lip17] of search terms entered for given labels. Non-specialists tend to adopt the given label, while experts formulate a more detailed search term.	207
6.18	Comparison of validation languages based on four requirements.	210
6.19	Overview of the experiment results, determining the utility of our constraint mapping tool.	218
6.20	Execution times measured in our experiments.	219

Listings

4.1	Example policy (<code>ids:ContractOffer</code>) that grants read access to members of a particular organization for a specified price [Bad+20b]. Some details are omitted for readability.	56
5.1	XPath expression to extract all sensor nodes and their metadata at once. . .	75
5.2	For the XPath expression from listing 5.1, our approach generates executable Python code for our patched OPC UA client.	77
5.3	Our proposed model for dividing data collection into process phases [Lip+20]. Phases can be interpreted as data collection strategies that can be changed adaptively.	85
5.4	Example JSON config file for the HPDC use case based on the phases from figure 5.18 [Lip+21c].	88
5.5	Sample of a server configuration that adds semantic information via ontology URIs [Lip+21d].	106
5.6	Example of non-binary JSON data extracted from Protobuf messages in the use case [The+22c]. It includes a 3D point and laser movement speed.	109
5.7	Sample of an RML mapping definition for lifting JSON [The+22c].	110
5.8	Sample of shapes defining the constraint that x-values of positions must be within -10 and 10 [The+22c].	111
5.9	Sample of shapes that fulfill the constraint that the x and y paths of a position cannot both have values equal to 0 [The+22c].	111
5.10	Example RML mapping with an iterator (line 5) and two references (<code>{name}</code> in line 7 and <code>phone</code> in line 11).	116
5.11	Extract from the EXPRESS definition for a <code>IfcDoor</code> entity in IFC 4.3.rc.2. .	116
6.1	Concept recommendations returned by the <i>Linked Open Vocabularies</i> API for a sample query for the single label <i>student</i>	177
6.2	Example content including a domain, three class names, and three property names of a vocabulary under construction submitted to the recommender REST interface [Zej21].	179
6.3	Example SHACL-SPARQL template to validate that a certain language tag is present for a data property.	214

Appendices

A List of Acronyms

API	application programming interface	20
CSV	comma-separated values	16
DCAT	Data Catalog Vocabulary	7
DLCM	data lifecycle model	5
DOI	digital object identifier	18
EOSC	European Open Science Cloud	8
FAIR	Findable, Accessible, Interoperable, and Re-usable	iii
HPDC	High Pressure Die Casting	79
HTML	HyperText Markup Language	13
HTTP	Hypertext Transfer Protocol	14
IDS	International Data Spaces	7
IDSA	International Data Spaces Association	7
IFC	Industry Foundation Classes	115
JSON	JavaScript Object Notation	13
JSON-LD	JSON Linked Data	15
KG	Knowledge Graph	11

A List of Acronyms

NFDI German National Research Data Infrastructure	8
OPC UA Open Platform Communications Unified Architecture	26
UML Unified Modeling Language	21
OWL Web Ontology Language	3
PID persistent identifier	8
RDF Resource Description Framework	3
RDFS RDF Schema	7
REST representational state transfer	100
RML RDF Mapping Language	17
SHACL Shapes Constraint Language	15
ShEx Shape Expressions	15
SOA service-oriented architecture	95
SOLID SOcial Linked Data	8
SPARQL SPARQL Protocol and RDF Query Language	3
SQL Structured Query Language	14
URI Uniform Resource Identifier	17
URL Uniform Resource Locator	17
USP ultrashort pulse	96
W3C World Wide Web Consortium	7

XML Extensible Markup Language	13
XPath XML Path Language	13
XQuery XML Query Language	13

B Curriculum Vitae

Curriculum Vitae

Name	Johannes Theissen-Lipp
Birthday	June 7, 1991
Contact	Email: johannes.lipp@rwth-aachen.de LinkedIn: linkedin.com/in/johannes-lipp Website: johanneslipp.org

Academic Education

01/2018–Present	Ph.D. in Computer Science RWTH Aachen University Thesis: <i>Semantic Foundations of Dataspaces</i>
10/2014–12/2017	M.Sc. in Computer Science RWTH Aachen University Thesis: <i>Applying External Knowledge Bases and User-Assisted Modeling to Evolve a Semantic Knowledge Graph</i> – Grade 1.3 Final M.Sc. grade 2.0
07/2016–11/2016	Study Semester Abroad Auckland University of Technology, New Zealand
10/2011–09/2014	B.Sc. in Computer Science RWTH Aachen University Thesis: <i>Integration von Echtzeiteigenschaften in das Prozessmodell von RTAndroid</i> – Grade 1.6 Final B.Sc. grade 1.7
09/2001–06/2010	High School Diploma Europaschule Kerpen

Current Positions

12/2019–Present	Deputy Team Lead Fraunhofer Institute for Applied Information Technology FIT: Data Science and AI – Data Management • Research on dataspaces, information modeling, interoperability, ontologies, Semantic Web technologies, knowledge graphs, and the intersection of such. • Grant acquisition and project management in semantics and dataspaces (e.g., Internet of Production, International Data Spaces, Gaia-X, NFDI). • Activities to support the working of the team and department.

04/2021–Present	PhD Researcher RWTH Aachen University: Information Systems & Databases • Research on dataspace, information modeling, interoperability, ontologies, Semantic Web technologies, knowledge graphs, and the intersection of such. • Teaching and supervision of seminars and student theses.

Previous Work Experience

01/2018–11/2019	PhD Researcher RWTH Aachen University: Information Management in Mechanical Engineering – Mobility and Logistics • Research on Data Integration, interoperability, Data Lakes, ontologies, knowledge graphs, semantic computing, and the Internet of Production.
02/2017–12/2017	Student Worker IMA at RWTH Aachen University • Cognitive computing • Implementation of knowledge graphs
10/2015–01/2016	Student Worker (Tutor) RWTH Aachen University • Foundations of Data Science
03/2015–07/2015	University Internship in Oman GUtech, Muscat, Oman • Teaching in mathematical logic • Organization of programming courses
10/2012–01/2015	Student Worker (Tutor) RWTH Aachen University Teaching support in various fields: • Data structures and algorithms • Programming • Computability and complexity
05/2011–08/2011	Computer Setup and Configuration Computacenter
11/2010–04/2011	Community Service Environmental Protection at Erftverband, Bergheim
2006–2010	Several Part Time Jobs School holidays or when not at school