Framework for Parallelized Hybrid Strategies in CAD-Based Disassembly Sequence Planning

Sören Münker, Amon Göppert and Robert H. Schmitt

Abstract

Circular economies require efficient disassembly processes for end-of-life products from various product series, variants and with various product states. CAD-based Disassembly Sequence Planning aims to efficiently create feasible disassembly sequences to minimize the needed effort in disassembly execution. Manual, collision-based and feature-based Disassembly Sequence Planning approaches are predominant in research. However, there is no tool covering multiple of these techniques simultaneously. In this paper, we aim to enable the combination of these techniques in a parallelized and hybrid manner. A Service-Oriented Model-View-Controller Framework is presented as an overarching software architecture. The architecture was implemented and tested on an exemplary use case. The validation proved the potential quality and efficiency increase of CAD-based disassembly sequence planning. The framework contributes to choosing the right disassembly sequence planning strategy and thus increasing the efficiency of the circular economy.

Keywords

Disassembly sequence planning • Computer-aided design • Software architecture • Circularity

S. Münker (✉) · A. Göppert · R. H. Schmitt
Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany
e-mail: s.muenker@wzl-mq.rwth-aachen.de

R. H. Schmitt
Fraunhofer Institute for Production Technology IPT, Steinbachstraße 17, 52074 Aachen, Germany

S. Ihlenfeldt et al. (eds.), *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2023*, https://doi.org/10.1007/978-3-031-74010-7_8

1 Introduction

With the rising demand for circular economies, disassembly optimization gains more importance. Crucial for disassembly efficiency is the disassembly structure representation and the planning of disassembly sequences [14]. The application of Computer-aided Deacsign (CAD) not only during product design but also for automated (dis-)assembly planning is an active field of research [4]. However, the applicability of this approach is still low due to the low quality of automatically generated sequence results and high computational demands.

This work aims to develop a framework combining multiple (Dis-)assembly Sequence Planning (DSP) algorithms from recent research and previous work on Assembly-by-Disassembly (AbD) approaches [11, 12] and, thus, make them more applicable for industrial use cases. We propose using parallelized hybrid strategies based on gathered CAD data. *Hybrid* refers to using a combination of different methods (e.g., manual and automatic DSP approaches). Hybridization aims to increase the quality of DSP results. *Parallelized* refers to the ability to perform multiple tasks simultaneously by dividing a large optimization problem into smaller independent subproblems. The parallelization aims to accelerate the DSP process and improve the overall efficiency of CAD-based DSP.

Relevant requirements for a CAD-based DSP software framework with parallelizable components and hybrid strategies are:

- Independence of proprietary CAD software
- Integrateability with (dis-)assembly planning tools or modules (e.g., Job Scheduling software tools)
- Portability of software modules utilized in framework to other frameworks
- Outsourceability of parallelizable computational efforts
- User interface for interacting with CAD and graph environment

This paper first reviews the related work for CAD-based (dis-)assembly planning frameworks and common software architecture patterns. A new framework combining a Service-Oriented Architecture (SOA) pattern and a Model-View-Controller (MVC) pattern is proposed and implemented. The implementation is validated on an exemplary industrial use case.

2 Review of CAD-Based (Dis-)Assembly Planning Frameworks

The “Auto-Assem” system by Xu et al. (2012) is an early framework for combined Assembly Sequence Planning (ASP) and Job Scheduling (JS) with sequential optimisation. The authors’ idea was to build a CAD/CAM-like tool with primary usage for assembly planning to assist engineers in making assembly plans. The proposed framework covers five steps: 1. Assembly modelling; 2. ASP; 3. Path planning; 4. Process planning and

visualization; 5. Assembly simulation [8]. However, most steps are executed manually rather than automatically.

Li and Lockett (2017) proposed a framework for designing final assembly lines based on product CAD models, including ASP and JS. Their motivation was to overcome the knowledge gap between product designers and final assembly process designers in the aerospace industry. The implementation is done in CATIA V6. Semantic data (e.g., from Failure Mode and Effects Analysis (FMEA) analysis, Bill of Materials (BOM) etc.) are combined in the "Unified Master Data", which is then utilized for sequencing and assembly line design [10]. However, the paper does not provide further details on possible automation of ASP and JS steps.

Beck et al. (2021) developed a framework to integrate (Dis-)assembly Sequence Planning (DSP) modules with robotic path-planning modules. The framework considers information extracted from the product's CAD model. The framework was implemented using the Octopuz Python API, which can automatically read and remodel STEP files for disassembly. A JSON file is used to construct robot programs for any type of robot. The robot-specific execution code can be exported into native robot languages using Octopuz's internal program exporter [6]. However, this approach is not transferable to sequence planning for large-scale assemblies.

Barbu et al. (2022) focus on precise assembly sequence generation based on various optimization criteria without relying on a specific CAD system. The approach includes a reverse engineering algorithm and an additional user interface utilizing a game engine to add the missing information. The user can select and move the product to disassemble parts. The generated information about disassembly paths is saved in a neutral exchange format. This exchange format can then be used for automatically generating assembly animations and videos [5].

Yao et al. (2022) propose a comprehensive ASP system to be applied in practice. The system utilizes CAD files to create Liaison and Interference Matrices representing the mathematical relationships between parts. An adapted Ant Colony Optimization algorithm is then used to generate an optimized assembly sequence based on these relationships. The system can be accessed through a web-based application where users can upload files and interact with the system. The concept and functionality of the system have been validated using a CAD file of an electric motor product as an example [15]. However, the concept lacks modularity, application of hybrid strategies and parallelization of computationally expensive tasks.

The authors' previous work primarily focused on fully automated pipelines to generate Precedence Graph (PG) or AND/OR Graph (AOG) from CAD files. Both approaches utilized a collision-based Assembly-by-Disassembly(AbD) planning approach and were implemented as a CATIA v5 Add-On [11, 12].

Currently, CAD-based DSP approaches are mostly fully manual or fully automated. The automated approaches have feature-based or collision-based approaches with use-case-dependent advantages and disadvantages. There is no approach explicitly considering the

parallelization of DSP approaches yet. A framework is needed to ensure the integrability of existing DSP and JS modules, solve the hybrid application of different DSP approaches and enable performance improvement by parallelization. To address these research gaps, in chapter 'Exploratory Pilot Study Investigating Effects of Exoskeletons on Movement Patterns', the parallelizable hybrid DSP approach is introduced, and the accompanied framework is covered in chapter 'Robot-Based Assembly of Hydrogen Tube Fittings for Large-Scale Electrolyzers'.

3 Parallelizable Hybrid Disassembly Sequence Planning

The concept of parallelizable hybrid DSP aims to optimize the disassembly process by identifying which steps can be done in parallel and choosing the right DSP strategy for parallelizable subassemblies. Parallelizable subassemblies can be identified, e.g. through community detection [16], similarity detection or selected manually. Community detection identifies groups of parts or subassemblies that can be disassembled together. In contrast, similarity detection is the process of identifying parts or subassemblies that have similar disassembly requirements.

There are three main (Dis-)assembly Sequence Planning (DSP) strategies to be utilized hybridly: manual DSP, automatic collision-based DSP and automatic feature-based DSP. Manual DSP allows human operators to plan the disassembly process by manually selecting the order of disassembly for the identified communities and similar parts. Automatic collision-based DSP uses algorithms to plan the disassembly process by simulating the movement of parts and identifying potential collisions. It is based on the principle that the disassembly process can be optimized by minimizing the number of collisions between parts. Automatic feature-based DSP uses algorithms to plan the disassembly process by taking into account the characteristics of the parts, such as size, shape, and connections. It is based on the principle that the disassembly process can be optimized by considering the unique characteristics of the parts. The overall concept of parallelizable hybrid DSP is visualized in Fig. 1.

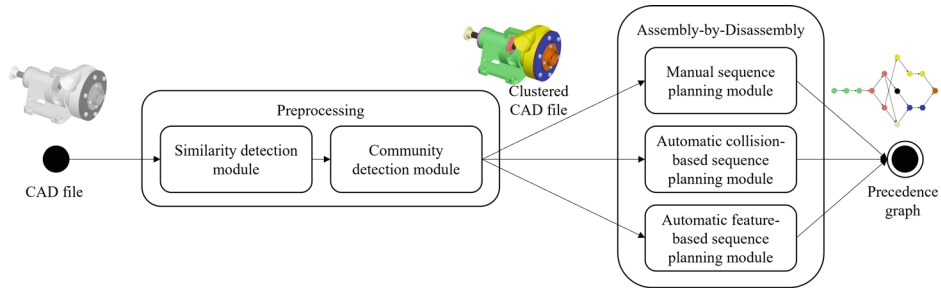

Fig. 1 Exemplary process pipeline of a parallelized hybrid strategy disassembly sequence planning system

4 Service-Oriented Model-View-Controller Framework

Software architecture patterns are design principles that provide a common language and a proven solution for common software development challenges [7]. Different software architecture patterns can be used to design and implement a framework for the parallelized hybrid (Dis-)assembly Sequence Planning (DSP) system. In this chapter, the most common patterns are compared hence their suitability for the problem and a framework with combined patterns is proposed.

4.1 Preselection of Software Architecture Patterns

Many software architecture patterns are intuitively not suitable and can be excluded from consideration (e.g., Peer-to-peer, Mater-Slave, Event-Bus and Broker—all of them focus on communication in networks which is less relevant for user interactive 3D tools). The remaining suitable patterns based on Alebrahim and Heisel (2017) or Bi et al. (2018) [3, 7] are summarized by key characteristics, advantages, and disadvantages in Table 1.

In conclusion, the Model-View-Controller (MVC) pattern is well-suited, as shown in the Table 1. However, it has a disadvantage in limited scalability. It may be an issue for more computationally intensive CAD-based DSP tasks (e.g., for complex engine parts). On the other hand, SOA is a modern approach that offers high scalability and the ability to outsource tasks. However, it requires standardized services with clear interfaces, which may be less suitable for dynamic user intervention during CAD-based DSP analyses because response times could be too high.

The idea of this work is to combine the advantages of both MVC and SOA by integrating both patterns in one framework. This approach is inspired by the "Model-View-Controller-Service-Paradigma" that uses an inner and outer MVC structure to include service-based modules [9]. This combination could provide a more comprehensive solution for software architecture design in CAD-based DSP and is introduced in the following.

4.2 Concept of Service-Oriented Model-View-Controller

The resulting Service-Oriented Model-View-Controller (SOMVC) framework incorporates the advantages of SOA and MVC in a web-based application for CAD-based DSP. It is designed to provide a system-independent way for users to interact with software tools through a browser interface. The main logic of the DSP system is handled by one server, while computationally heavy tasks are outsourced to cloud-based High-Performance Computing (HPC) servers. The framework is illustrated in Fig. 2.

The inner circle of the framework, which is located on the server, is responsible for low-performance tasks. These tasks include manually manipulating the CAD model (e.g.,

Table 1 Comparison of software architecture design patterns

Name	Typical use cases	Advantages	Disadvantages
Layers	• General desktop applications • e-commerce web applications	• Reusability of layers • Potability and exchangeability of layers	• Strict separation of functions in layers hard to achieve • Possibly degradation of performance through multiple layers
Pipes and filters	Linear workflows (e.g., semantic analysis, parsing)	Suitable for multiple transformations on same data	• Not suitable for interactive applications • Filters not directly reusable for other purposes
Model-View-Controller	Interactive applications (e.g., for web applications)	• Well suited for any visual application • Multiple views of same model possible • Easy to develop collaboratively • Easy to maintain • Easy to exchange components	• Structure may lead to overhead in updating each component for every user action • Limited scalability
Client-server	Online applications (e.g. e-mail, document sharing)	• Easy to integrate new servers and functionalities • DevOps possibility with parallel implementation	High dependability on server (single point of failure)
Interpreter pattern	Interpreting programs in dedicated language (e.g., SQL database queries)	High portability (easy replacement of interpreted programs)	• Performance issues for interpreted software • Highly dynamic patterns might lack understandability
Service oriented architecture	Distributed architecture consisting of multiple services with defined interface description	• Maintainability • Independent on platform • Scalability for multiple users or outsourcing computing performance	• High bandwidth server needed • Only integratable with standardized services with clear interfaces

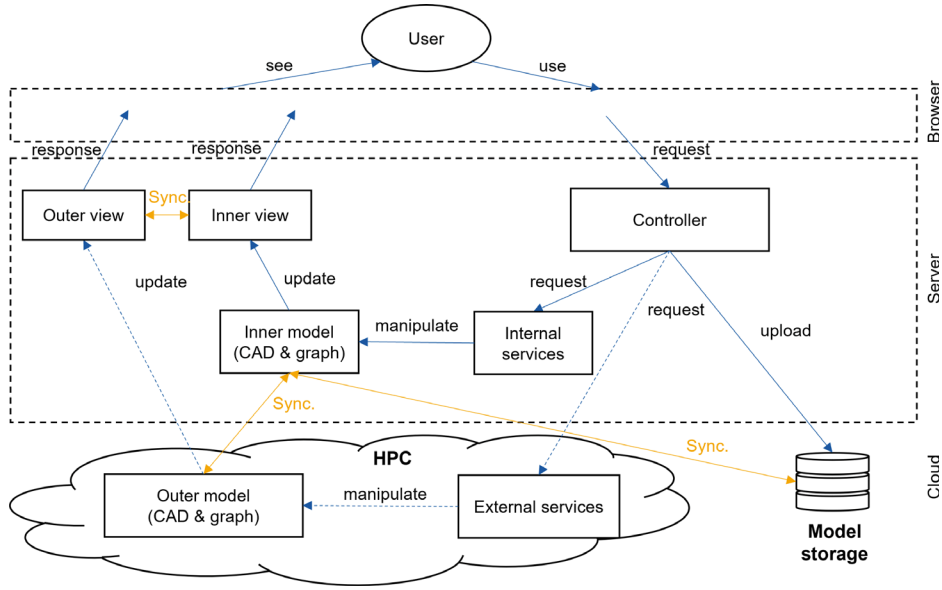

Fig. 2 Block diagram of the proposed service-oriented model-view-controller framework, with HPC = high performance computing

highlighting or excluding parts). The inner model contains the current status of the CAD model and the assembly graph model (e.g., Precedence Graph (PG) or AND/OR Graph (AOG)). The controller has access to all internal services which directly influence the inner model. All changes made by the controller are stored in the inner model, such as which parts are visible and which are invisible. The inner view component is responsible for visualizing the current status of the model, such as displaying different colours for selected parts or only displaying parts with the "visible" attribute.

The outer circle of the framework, which is mainly located on the HPC, is responsible for high-performance tasks such as collision analysis in DSP or graph clustering. The controller can be utilized to request these external services. The external services manipulate the outer model (CAD and graph models). Each model update results in an outer view update.

The user can choose between the inner and the outer view to work on both parallelly. When inner and outer circle processes are terminated, the outer model, inner model and model storage can be synchronized to merge the results of internal and external services. Additionally, the inner and outer views will be synchronized with the model synchronization.

5 Verification and Validation on Exemplary Use Case

The verification and validation of the Service-Oriented Model-View-Controller (SOMVC) framework is an essential step in ensuring that the system works as expected and meets the requirements of chapter 'Latest Challenges in the Development of Scalable Assembly

Systems for Fuel Cell Stacks'. Utilized techniques are unit testing, integration testing and acceptance testing [13]. All tests are executed with an exemplary use case as visualized in Fig. 3. The described use case scenario contains two user interactions: The upload of a CAD model and the Assembly-by-Disassembly (AbD) analysis.

Unit tests are written to test individual system components, such as the internal/external model, the internal/external view, the controller and the internal/external services. These tests ensure that each component works as expected and that the interactions between the components are functioning correctly.

Integration testing examines the interactions between the components of the system. The expected interactions are illustrated in Fig. 3. During the test, it was constantly monitored whether the software adhered to the given diagram. Through iterative programming, adherence was assured.

Acceptance testing is the last validation technique used to test the system from the user's perspective. To ensure objectivity, the concept of face validity was used. Face validity is a technique that involves presenting the system to experts from different fields, such as industry, research projects, working groups, and research institutes. This technique was used to gather feedback from experts in the field of (Dis-)assembly Sequence Planning (DSP) and to ensure that the system meets the needs and requirements of the target audience. Experts from research projects, such as Internet of Construction [2], AdaptAR [1], and REVAMP, were involved in the face validity process. This process helped ensure that the system is practical and usable and fulfils the requirements defined in the introduction. Screenshots of the presented software during face validity are given in Fig. 4. Thus, the resulting framework is a valid solution to increase the efficiency in DSP.

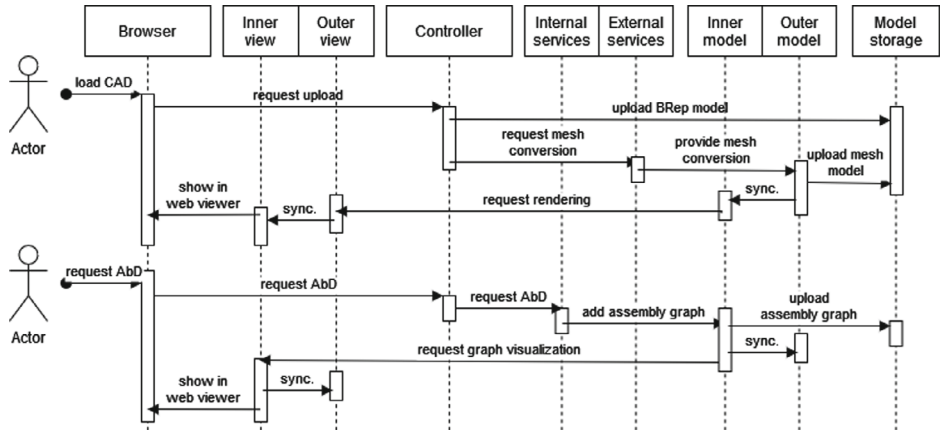

Fig. 3 Use case sequence diagram

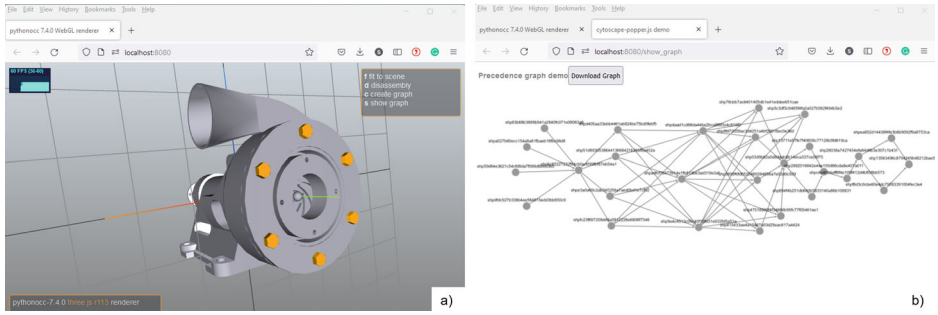

Fig. 4 Browser internal view of **a** the CAD environment and **b** the resulting graph model environment. The user can interact with the controller by keyboard commands

6 Conclusion and Outlook

In conclusion, the Service-Oriented Model-View-Controller (SOMVC) framework enables the parallelized hybrid (Dis-)assembly Sequence Planning (DSP). The use of internal and external processing circles allows for parallelizing DSP strategies (e.g., manual disassembly and automatic collision-based disassembly). Additionally, the integration of different services to identify parallelizable subassemblies, such as community detection and similarity detection modules, can be included. The SOMVC framework was verified and validated through unit testing, integration testing, and acceptance testing, which involved feedback from industry and research experts. These evaluations showed that the SOMVC framework contributes to improving the efficiency and quality of the CAD-based DSP. The contribution will also improve the efficiency and accuracy of remanufacturing planning by making "lot size one" DSP more feasible. Future research can focus on three areas:

1. The integration of the SOMVC framework with other tools and technologies, such as robot motion planning, product design feedback, and assembly instruction tools.
2. Use of artificial intelligence to improve the performance and efficiency of the services. This could include automatically deciding for the right DSP strategy based on labelled assembly topology data.
3. Increase scalability to handle large and complex assemblies with the right High-Performance Computing (HPC) infrastructure.

References

1. Adaptar - adaptive and context-specific technical instructions for the entire product life cycle in an ar environment. https://www.ipt.fraunhofer.de/en/projects/adaptar.html. Accessed 25 Jan 2023
2. Internet of construction - information networks for cross-company collaboration in the production chains of construction industry. http://www.internet-of-construction.com/. Accessed 25 Jan 2023

3. Alebrahim, A., Heisel, M.: Bridging the Gap Between Requirements Engineering and Software Architecture. Springer (2017)
4. Bahubalendruni, M.V.A.R., Biswal, B.B.: Liaison concatenation - a method to obtain feasible assembly sequences from 3d-cad product. Sādhanā **41**(1), 67–74 (2016)
5. Barbu, K., Beck, J., Schäfer, P., Neb, A.: Development of a visual assembly planning system based on neutral files. Proc. CIRP **107**, 446–451 (2022). Leading manufacturing systems transformation - Proceedings of the 55th CIRP Conference on Manufacturing Systems (2022)
6. Beck, J., Neb, A., Barbu, K.: Towards a cad-based automated robot offline-programming approach for disassembly. Proc. CIRP **104**, 1280–1285 (2021). 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0
7. Bi, T., Liang, P., Tang, A.: Architecture patterns, quality attributes, and design contexts: how developers design with them. In: 2018 25th Asia-Pacific Software Engineering Conference (APSEC), pp. 49–58. IEEE (2018)
8. Da Li, X., Wang, C., Bi, Z., Jiapeng, Yu.: Autoassem: an automated assembly planning system for complex products. IEEE Trans. Industr. Inf. **8**(3), 669–678 (2012)
9. Kowarschick, W.: Model-view-controller-service-paradigma. https://glossar.hs-augsburg.de/Model-View-Controller-Service-Paradigma. Accessed 25 Jan 2023
10. Li, T., Lockett, H.: An investigation into the interrelationship between aircraft systems and final assembly process design. Proc. Cirp **60**, 62–67 (2017)
11. Münker, S., Polikarpov, M., Schmitt, R.H.: Virtuelle demontage für xl-produkte/assembly sequence generation for xl-products – virtual disassembly for high numbers of components. wt Werkstattstechnik online **110**(11–12), 833–837 (2020)
12. Münker, S., Schmitt, R.H.: Cad-based and/or graph generation algorithms in (dis)assembly sequence planning of complex products. Proc. CIRP **106**, 144–149 (2022)
13. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2010 Winter Simulation Conference, pp. 166–183. IEEE (2010)
14. Xiao, J., Anwer, N., Li, W., Eynard, B., Zheng, C.: Dynamic bayesian network-based disassembly sequencing optimization for electric vehicle battery. CIRP J. Manuf. Sci. Technol. **38**, 824–835 (2022)
15. Yao, M., Drescher, B., Stewart, E., Manoj, L., Wittstamm, M., Henke, L., Ghodasara, S., Ge, M.: Computer-aided assembly sequence planning for high-mix low-volume products in the electronic appliances industry. In: Proceedings of the Conference on Production Systems and Logistics: CPSL 2022, pp. 91–100. Publish-Ing., Hannover (2022)
16. Zheng, Yu., Chen, L., Jiang, P., Cheng, H.: A sub-assembly division method based on community detection algorithm. Int. J. Comput. Integr. Manuf. **35**(10–11), 1133–1150 (2022)