A Generic Platform for Open Assessment Management in Higher Education

A modular architecture for assessment management within a centrally hosted learning and teaching portal

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

Diplom-Informatiker
Patrick Wilhelm Stalljohann

aus Ibbenbüren

Berichter:
Prof. Dr. Ulrik Schroeder
Prof. Dr. Bernhard Rumpe

Tag der mündlichen Prüfung: 17.10.2012

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.
Meiner geliebten Frau
Meinen Kindern
Meinen Eltern
Abstract

Solving tasks in everyday life and work requires continuous learning of facts, processes, and related skills. This is a lifelong process. On the one hand there are informal learning activities, i.e. they are happening unplanned during a normal activity. On the other hand, the first quarter of a human life is strongly affected by formal learning, e.g. in schools or at university. With the increasing use of Web 2.0, its possibilities for knowledge acquisition, communication, and collaboration facilitates the creation of personal learning environments. While these approaches mostly affect informal learning scenarios, their usage in formal settings is an open research question. A more specialized research topic in the context of combining formal and informal learning focuses assessment, the evaluation of learning activities, which takes an important part of formal learning.

Several methods for electronic support of assessment are in use for several years. Related software systems for the management of assessments are available and in use, but the integration of common services from the internet is still an open research question. Especially, the combination of different forms of assessment is difficult. In this context, the thesis at hand tries to cover the question, how the management of assessment scenarios in higher education can be supported technologically, with respect to traditional and new forms of assessment. The main objective of the related project was to build a web-based platform, which provides modular extensible support mechanisms for assessment management as a central service for different courses. This platform is called AMSeL (Assessment Management Services in eLearning systems).

A major problem with present assessment systems is that they are either too generic or too specialized. Generic systems provide basic support of assessment processes regardless of their domain, so that this kind of support is methodically limited. Specialized assessment systems provide extensive support for a specific type of question or a single domain. Therefore, current systems are especially not applicable to new forms of learning, e.g. informal learning or self-directed learning.

The thesis at hand tries to develop a new approach for technology enhanced assessment, which also allows applying current and new assessment and feedback processes to new forms of learning. Thus, it is discussed, how current assessment systems have to be extended or improved to support informal and self-directed learning as well. Especially, self-direction requires intelligent feedback. With the discrepancy of current generic and specialized assessment systems, the new approach demands the development of an architecture, which allows the integration of specific services into a generic platform.
These requirements have been used to build a modular software design, which takes account of incrementally growing requirements for a centralized service in a heterogeneous environment. A Software-as-a-Service (SaaS) architecture has been realized based on portal technologies to allow modular extension of functionality and integration of services. The implementation of AMSeL and its modules has been realized with portal technology mechanisms and techniques. Modules for assignment management, assessment of wiki contributions, the integration of cloud services to formal assessment processes, management of results and grading criteria, and others have been realized. In conjunction, they show how different forms of assessment can be combined in a whole assessment management platform. In this way, the advantages of the flexible service-oriented architecture have been demonstrated. As a result, the integrated and powerful assessment management platform AMSeL has been realized as a reference implementation. This platform provides currently unique enhancements related to present assessment systems. It especially addresses the emergent importance of new learning forms as well as social media services in the context of institutional assessment in an unprecedented way.
# Contents

## I. Fundamentals

1. Introduction
   1.1. Motivation ................................................................. 3  
   1.2. Objectives ................................................................. 6  
   1.3. Outline ................................................................. 7

2. Assessment ................................................................. 9 
   2.1. Definition and Classification ................................................ 9 
      2.1.1. Purposes and Dimensions ................................................. 9  
      2.1.2. Assessment Activities ................................................... 12  
      2.1.3. Feedback ................................................................. 17  
      2.1.4. Assessment Methods ................................................... 19  
      2.1.5. Classification by Bloom’s taxonomy .................................... 25  
      2.1.6. Assessment in higher education ......................................... 26  
   2.2. Technology Enhanced Assessment ........................................... 29  
      2.2.1. Electronic Tests ....................................................... 30  
      2.2.2. Generic Assessment Support in LMS ..................................... 33  
      2.2.3. Process-specific Systems ................................................. 35  
      2.2.4. Domain-specific Systems ................................................. 37  
      2.2.5. Assessment 2.0 ......................................................... 39  
      2.2.6. Classification by learning theory ....................................... 42  
      2.2.7. Standards and Reference Models ......................................... 43  
   2.3. Conclusions ................................................................. 52

## II. Analysis

3. Assessment Tools ............................................................ 59 
   3.1. e-Test Systems .......................................................... 60  
   3.2. Learning Management Systems (LMS) ...................................... 64  
   3.3. Process-specific Systems ................................................. 67  
      3.3.1. Assignment Creation ..................................................... 67  
      3.3.2. Assignment Management ................................................ 69
7. Portal Technologies

7.1. Definition and Classification

7.1.1. Web Portals

7.1.2. Enterprise Portals

7.1.3. Components and Functionalities

7.1.4. Products

7.2. The Learning and Teaching Portal L²P

7.3. Microsoft SharePoint

7.3.1. Server Architecture

7.3.2. Site Architecture

7.3.3. Customization, Development and Deployment

7.4. Conclusions

8. Portal-based Implementation

8.1. Development Patterns

8.1.1. Event-driven Service Delivery

8.1.2. Declarative Reflection Factories

8.2. Assessment Base

8.3. Assignment Management

8.3.1. L²P Assignments V1

8.3.2. L²P Assignments V2

8.3.3. Modularized Assignment Management Tool

8.4. Group Management

8.4.1. The Grouping module

8.4.2. An Example Scenario

8.4.3. Grouping as a Service

8.5. Correctionflows

8.5.1. The Correctionflow Component

8.5.2. Example Scenarios

8.5.3. Correctionflows in Assignment Management

8.6. Integration of Cloud-Services for Assessment

8.7. Wiki Pages with Assessment-Support

8.8. Assessment Planning and Criteria Management

8.8.1. The L²P Gradebook

8.8.2. An example scenario

8.8.3. Integration to AMSeL

8.9. Summary
Part I.

Fundamentals
Chapter 1.

Introduction

The thesis at hand discusses unsolved research questions about how new forms of assessment for formal and non-formal learning can be developed and how they can be applied to institutional learning scenarios. Therefore, it presents an approach to consolidate the technological support of assessment processes in higher education. Therefore, the platform AMSeL (Assessment Management Services in eLearning systems) has been developed. It provides a flexible and modular environment for various assessment processes. Modern forms of learning, which are especially facilitated by emergent technologies, as well as traditional ones have been analyzed according to possibilities for technological support and enhancement. Thus, the presented efforts have to be seen as a first step in the realization of a comprehensive technological support of assessment and feedback within lifelong learning.

1.1. Motivation

Technical innovation, process optimization, and in general, the change to knowledge societies requires a lifelong learning process for the population. The relevance of this topic as well as the need for research in this area is underpinned by The Lifelong Learning Programme of the European Commission, which is funded “with a budget of nearly €7 billion for 2007 to 2013” [1]. Beside traditional institutional learning at school or university, “new dimensions of learning such as: (1) self-directed learning, (2) learning on demand, (3) informal learning, and (4) organizational learning” [Fischer, 2001] (see figure 1.1) are increasingly required.

The utilization of new media provides an opportunity to support these new dimensions of learning. The explicit realization is challenging and raises several research questions. Especially, the new possibilities of participation in the Web 2.0 or rather social media facilitate a low-threshold access to information, communication, and collaboration services. Many people of nearly all ages and from all social classes
1. Introduction

are already using these media in their private context. On-line discussion forums, instant messaging services (e.g. Skype or ICQ), and micro blogs (e.g. Twitter) are used for communication. Shared knowledge systems (e.g. Wikipedia or Wiktionary), media services (e.g. YouTube or Flickr), and social networks (e.g. Facebook, Xing, or Google+) are used for knowledge acquisition, collaboration, networking, and last but not least for fun and entertainment. Additional benefits are possible by mobile access to these services. Smartphones and tablets achieve high acceptance and allow additional personalization features like location based services. While people are using new media in a private context they are performing several activities, which can lead to incidental learning. This informal learning experience is very authentic, but mostly neither planned nor purposeful.

"Lifelong learning requires a deeper understanding of the co-evolutionary processes between fundamental human activities and their relationships with new media. It requires an integration of new theories, innovative systems, practices, and assessment", [Fischer, 2001].

Additional effort has to be made to achieve effective learning activities by the use of social media. Several open questions about how new media can be used to support effective learning have been a topic of research in the last decade. New learning theories, e.g. Connectivism [Siemens, 2005] and LaaN [Chatti et al., 2010], try to explain learning processes in social networks and other digital environments. From a technological perspective, learning is supported by the social software itself, which can be orchestrated in so called Personal Learning Environments (PLEs). With use of a smartphone or tablet the mobility and personalization of such an environment can be increased. These theories and systems are strongly focused on the dimensions of non-institutional learning (see above).

In the overall process of lifelong learning there are mainly three different categories of systems which should support the organization of learning: Learning Manage-
1.1. Motivation

Learning Systems (LMS) or rather Virtual Learning Environments (VLE), Knowledge Management Systems (KMS), and PLEs. LMS are optimized for institutional learning, KMS are strongly used for organizational learning, and PLEs are designed for self-directed and informal learning [Kalz et al., 2011]. These systems are designed according to different learning theories. Thus, a combination of LMS, KMS, and PLEs within a mixed Platform for Lifelong Learning seems reasonable. Because of the oppositional approaches, it is still an open research question how a combined or rather integrated platform for lifelong learning support should be designed and realized.

A first step to answer this question is to analyze the requirements for a platform to support multiple dimensions of learning in the context of higher education. On the one hand, some teachers already utilize social media in their courses. For instance, learners share videos about study projects on YouTube, provide presentations via SlideShare, create documents collaboratively online with Google Docs, or keep a diary about their project with help of a blog. On the other hand, supporting presence courses with educational technology, so called blended learning, has quite a long tradition. Computer-based and Web-based Trainings (CBT/WBT), electronic Tests, LMSs as well as specific systems for formal learning aid have been topics for research. Especially, LMS are currently in use at several single institutes as well as central services at universities. Main objective of such platforms is to make standard tools available for communication, collaboration, and distribution of learning material. These are mostly online discussion forums, wiki pages, document and content management functionalities, and administrative processes. Usage statistics indicate for broad acceptance and usage of those platforms by teachers and students (see section 7.2). From a technical perspective, the combination of LMS-supported processes and social media services, which are the components of PLEs, has to be investigated.

A main difference of institutional learning in contrast to the other dimensions is the strong role of assessment (see chapter 2), i.e. the evaluation and grading of learning activities. Electronic Tests (E-Tests), the most simple form of electronic assessment, are part of most LMS. The more complex and time-consuming management of advanced assessment activities are mostly not or only rudimentary included. Nevertheless, several standalone systems are originated by research in the last years. Mostly, each of them have been developed for a single form of assessment. Some of them support one specific assessment process, e.g. peer assessment or group assessment. Others facilitate the automatic correction of submissions in one specific domain, e.g. programming assignments. Additionally, advanced features like adaptivity or parametrized questions have been added to electronic test systems. All in all, the functionalities of each system could be valuable in an integrated learning platform. But each system is not suitable as a such a platform on its own. Furthermore, social media facilitates new forms of online learning activities. How these
1. Introduction

activities can be assessed and what new forms of assessment are possible are current questions of research.

1.2. Objectives

The main objective of the thesis at hand is the modernization of technological assessment management support in higher education. This should be seen as a first step towards a life long learning platform, which has to integrate formal assessment processes with social media activities. Therefore, a model has to be developed which respects traditional as well as new forms of assessment. Methods of assessment related to social media have to be integrated. Additionally, the new forms have to be facilitate by the combination of both. An open platform for assessment management should demonstrate possibilities for an integrated technological support. The main research question in this context is:

**Research Question**

How could a harmonized model for technological support of assessment management, which allows the integration and combination of traditional and new forms of assessment, look like?

Several difficulties and subquestions have to be solved to answer this question. For instance, the handling of various assessment strategies in different learning scenarios within a single platform or the integration of cloud services with respect to authority and authentication.

**Hypothesis**

An assessment management platform requires a highly modular architecture and flexible deployment mechanisms to allow extension and adaptation for various assessment scenarios.

The thesis at hand describes a technical approach to gain scientific results about the above mentioned research question. Contributions of this project are the following:

- A model for assessment management platforms.
- Software architecture patterns for extending a portal technology to an assessment management platform.
- An architecture which allows the deployment of assessment modules to the learning and teaching portal L²P.
- Selected modules for assessment management processes, social media integration, and combined forms of assessment.
- Evaluation of the platform and its components as a university-wide service.
1.3. Outline

The development of an open assessment management platform requires to deal with theoretical foundations of the assessment domain. Therefore, basic definitions and classifications of methods for the assessment of learning activities are presented in chapter 2. In addition, the state-of-the-art in technology enhanced assessment shows current approaches for the technological support of formative assessment tasks in higher education. Different categories of systems are mentioned as well as standards and reference models.

The second part of the thesis at hand contains the requirement analysis of the development project. An evaluation of state-of-the-art assessment tools as well as current prototypes from research in chapter 3 contributes possible functionalities, which have to be considered as possible modules for the platform. A survey conducted at a technical university as well as interviews with several educators collect real world scenarios (chapter 4) which have to be covered by the support mechanisms of the intended platform. The results of literature review and interviews lead to the formulation of requirements in chapter 5.

The realization of AMSeL (Assessment Management Services in eLearning systems), which is based on the gathered requirements, is described in part III. Firstly, the overall software design and architecture for the platform is discussed (chapter 6). Portal technologies, which have been chosen as a foundation for the implementation, are introduced in section 7. The concrete implementation of AMSeL and its modules are presented in chapter 8.

The evaluation of AMSeL has been done in form of theoretical requirement validations, pilot installations with accompanying surveys, static system analysis, and user tests. Related results and conclusions concerning this thesis project are discussed in chapter 9 in part IV. A summary of the project as well as perspectives for future research are given in chapter 10.

Note: Used references throughout the thesis at hand are marked in four different ways. Firstly, scientific publications like articles, papers, and books are referenced in apalike-style (e.g. [Race, 2001]) and can be found in section “Bibliography”. Hyperlinks to online resources are numbered (e.g. [1]) and are listed in section “Links”. References to the author’s pre-publications concerning the topic of the thesis at hand are numbered, prefixed with the letter “P” (e.g. [P5]), and are listed in section “Publications”. Finally, references to thesis that have been supervised during the related project of the thesis at hand are numbered with the prefix “T” (e.g. [T6]) and can be found in section “Supervised Thesis”.

2.1. Definition and Classification

Assessment is a term which is used in several disciplines and always deals with the evaluation of gathered information. In the thesis at hand this term is used in the context of education. This educational assessment deals with the analysis and evaluation of learner performances. It is “the process of evidencing and evaluating the extent to which a candidate has met or made progress towards the assessment criteria” [2]. This criteria defines “what the learner is expected to do during the assessment in order to demonstrate that a learning outcome has been achieved” [2].

2.1.1. Purposes and Dimensions

Several perspectives on this evaluation process are possible, thus assessments are utilized for different purposes (cf. [Freeman and Lewis, 1998, p. 10]):

- to select,
- to certificate,
- to describe,
- to aid learning,
- to improve teaching.

Selection – Based on their assessment results learners are “selected for further opportunities when these are (as is often the case) rationed” [Freeman and Lewis, 1998, p. 10]. For instance, the admission to some degree program at universities are restricted, such that only learners with a minimum grade in their A levels are allowed to attend. The most common selection is performed if the assessment criteria is missed, so that the assessed activity has to be performed again or the learner has to leave the course.
Certification – It can be certified that the learner has obtained skills or competencies which are related to the performed task and the assessment criteria. Mostly, a certificate like a driving license or a university degree document that the learner reached a standard in a specific discipline proven by an official.

Description – Beside marks on a limited scale (e.g. grades, score, or percentage), the result of an assessment could be a more detailed description of the learners outcome. In this way a learners profile or portfolio (see section 2.1.2) can be created to allow a more specific view on his or her achieved competencies and skills.

Learning Aid – Assessments itself can be seen as additional learning material, especially as an opportunity to practice and to avoid tacit knowledge [Niegemann et al., 2003]. Feedback to their performance can increase their motivation and allows them to diagnose their weaknesses and strength as well as to track their personal learning progress (see section 2.1.3).

Teaching Improvement – Not only valuable for learners, teachers can also benefit from including assessments in their courses. Learners outcomes to a specific assessment can be used as a basis to judge the effectiveness of related learning material or the didactic approach. For instance, if a homework related to a specific is badly performed by most learners of a course, it indicates that the topic should be explained again, perhaps in another way.

After the purposes of applying assessments in a given context are clear, the question is, how to design the assessments to be suitable for the chosen purposes. Several dimensions of assessment have to be considered (cf. [McAlpine, 2002, pp. 6-10]):

- intention,
- transparency,
- chronology,
- target,
- convergence.

Intention of Assessment – The most common intention of assessment is the assessment of learning [Ridgway et al., 2004, Earl, 2003] or rather summative assessment [McAlpine, 2002, p. 6]. It finalizes a period of learning with a final judgment of a learners overall performance. This is especially meant for external purposes like certification. Assessment for learning [Ridgway et al., 2004, Earl, 2003] or formative assessment is designed to assist the learning process by providing feedback to the learner” [McAlpine, 2002, p. 6]. If the learner itself has a deeper integration into a formative assessment process by monitoring and evaluating himself/herself or peers, this is sometimes called assessment as learning [Earl, 2003]. Additionally, diagnostic assessment tries to predict future performances of a learner. Based on the detected strength and weaknesses either learners could be selected, e.g. an
entrance examination, or additional learning aids could be provided. Sometimes a comparison of diagnostic, formative, and summative assessment results can be used to judge the effectiveness of a course. The temporal aspect of the assessment intention is displayed in figure 2.1.

![Figure 2.1. Timing of diagnostic, formative, and summative assessments.](image)

**Transparency of Assessment** – The transparency of assessment ranges between formal and informal [McAlpine, 2002, p. 7]. Formal assessments are obviously recognizable for a learner, e.g. an exam or homework. Informal assessments are more accompanying to another activity, whereas the assessment character is more unapparent. For instance, the marking of pupils in-class participation for a whole term has a more informal character.

**Chronology of Assessment** – The chronology is somehow related to the intention of assessment, since final assessment only takes place at the end of a learning period or course and “continuous assessment is scattered throughout a course” [McAlpine, 2002, p. 8]. In addition to differences concerning the didactic approach of final and continuous assessment (cf. intention of assessment), another difference is the organizational workload, which is much higher for continuous assessment.

**Target of Assessment** – Mostly based on the idea that knowledge has to be transferred from a teacher to a learner, the classic target of assessment is a product, e.g. knowledge of a fact or a written report, which proves a certain fact. With putting more emphasis on skills and competencies, the process is the more important target of assessment, but to evaluate how a task has been solved is mostly more complex than evaluating the result of a task. For example, an essay itself can be assessed more easily than the process of collecting information and writing the essay.

**Convergence of Assessment** – If the result of a task is exactly known or limited, e.g. the result of a calculation, it is called convergent. If the range of correct answers is open and has to be analyzed individually, e.g. an essay, source code, or an oral presentation, it is called divergent. On the one hand, correcting convergent
assessments are very easy. On the other hand, it is easier to assess higher order thinking skills by use of divergent assessments.

The above mentioned relation between the covered dimensions of an assessment and its usage for a specific purpose is described in table 2.1. A check mark means that an assessment with the given value in a dimension is usable for the specific purpose. Otherwise, it is not. For instance, diagnostic assessments are used for selection, but they are not designed for certification.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Select</th>
<th>Certificate</th>
<th>Describe</th>
<th>Aid learning</th>
<th>Improve teaching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intention</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formative</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summative</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Transparency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formal</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Informal</td>
<td></td>
<td></td>
<td>(✓)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chronology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Continuous</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>(✓)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Convergence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convergent</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divergent</td>
<td>✓</td>
<td>(✓)</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1. Relation between assessment dimensions and purposes.

2.1.2. Assessment Activities

Assessment processes are strongly related to learning activities. Throughout this thesis, learning activities involved in an assessment process are called assessment activities. Each of them covers specific assessment dimensions. Thus they are suitable for different purposes. The most common types of activities are the following ones:

- attendance,
- participation,
- exams (written, oral, practical),
- questions,
• assignments,
• projects,
• portfolios.

With the selection of assessment activities for a given purpose it has to be taken into account that “[…] any assessment format or process disadvantages some candidates, and that using the same few formats disadvantages the same candidates time and time again. It is therefore important to diversify assessment, […]” [Race, 2001]. Thus it is recommended to combine the activities, which are explained in the following.

### Attendance and Participation

Probably, the most simple assessment activity is compulsory attendance to a formal learning situation, e.g. to a lecture of a university course. Based on the ideas of direct knowledge transfer from a teacher to a learner as well as intrinsic motivation of learners, attendance is sometimes equalized with the commitment of learners to learn. Neither a product of learning, not to mention the process of learning, is observed nor can be guaranteed that learning has really happened. Nonetheless, attendance is used for a kind of continuous summative assessment, mostly for organizational reasons. When used for certification, instead of a certificate of performance often a certificate of attendance is issued.

Additionally, compulsory attendance as a formal assessment activity is often combined with the, mostly informal, assessment of a learners participation to a learning situation. These activities are somehow related to attendance but mostly oppositional in their purposes. Participation is based on a kind of dialog between learners and teachers or peers which leads to a formative assessment of a real learning process. In contrast to attendance, participation can be of different quality. For instance, a student’s oral contribution in class can be assessed with regard to its content as well as its usefulness for further discussion. Table 2.2 contrasts the coverage of assessment dimensions of both activities with each other.

<table>
<thead>
<tr>
<th></th>
<th>Intention</th>
<th>Transparency</th>
<th>Chronology</th>
<th>Target</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>Summative</td>
<td>Formal</td>
<td>Continuous</td>
<td>–</td>
<td>Convergent</td>
</tr>
<tr>
<td></td>
<td>(Formative)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Participation</td>
<td>Formative</td>
<td>Informal</td>
<td>Continuous</td>
<td>Process</td>
<td>Divergent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Formal)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2. Covered assessment dimensions of attendance and participation activities.
2. Assessment

Exams

One of the most traditional assessment activities are exams. They are used as assessment of learning, especially for certification. Traditionally, exams are worked on individually, in a closed environment (e.g. a classroom), without auxiliary means, and during a given amount of time. More modern forms adjust some of these parameters [Race, 2001]:

**Exam Types**
- traditional exams,
- ‘seen’ exams (prepare for known questions),
- ‘open book’ exams (books as auxiliary material),
- ‘open notes’ exams (prepared notes as auxiliary material),
- ‘time-unconstrained’ exams.

These types try to increase the possibility to assess transfer of learning by application, analysis, or creation instead of measuring remembering of facts only. In addition, students demonstrate additional competencies by application of appropriate utilities and resources.

Furthermore, each adjusted type of exam tries to support different strengths and reduce weaknesses of learners. For instance ‘open book’ exams require less memorization, ‘open notes’ exams support systematic preparation, and ‘time-unconstrained’ exams reduce disadvantages for people with slow handwriting.

Other characteristics of exams which can vary is the way in which the performance asked for by the teacher and in which way they have to be delivered by the learner [Race, 2001]:

**Exam Delivery Types**
- written exams,
- oral exams,
- practical exams.

In written exams questions or tasks are presented to learners in written form. They have to be solved by writing down answers, results, or approaches. In oral exams questions and answers are formulated orally. Sometimes it is a kind of assessed discussion. Practical exams involve the practical creation of an artifact in the presence of a teacher. Furthermore, role plays and case studies are more specific types of practical exams, which are especially used in medical sciences. The covered assessment dimensions of exams are displayed in table 2.3.

<table>
<thead>
<tr>
<th>Intention</th>
<th>Transparency</th>
<th>Chronology</th>
<th>Target</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summative</td>
<td>Formal</td>
<td>Final</td>
<td>Product</td>
<td>Convergent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Divergent</td>
</tr>
</tbody>
</table>

Table 2.3. Covered assessment dimensions of exams.
2.1. Definition and Classification

Formative Tests, Assignments, and Projects

Knowledge that someone gained but is not able to apply this knowledge to a practical task is called tacit knowledge [Wendt, 2003, p. 38]. Formative tasks or questions are a suitable approach to support knowledge transfer and to prevent tacit knowledge [Niegemann et al., 2003, p. 112]. During the process of knowledge application in a specific task learners gain additional practical skills, which are needed to solve the task [Horton, 2006, p. 105].

The structure of a task or question can be characterized by three fields: information field, question field, answer field. Additionally, “there is a clear distinction between two groups of activities that we will call objective and applicative” [Brusilovsky and Miller, 2001, Bull and McKenna, 2004]. A distinction between three types of questions is made by [Rütter, 1973], in which two subcategories of objective activities are distinguished:

- Closed Questions,
- Semi-Open Questions,
- Open Questions.

Closed Questions  This is the first type of objective questions. For closed questions, the range all possible answers are known by teachers as well as by learners. That means a learner has to ‘choose’ the correct answers from a given list of possible answers. Wrong answers in this list are called distractors. Teachers know the correct answers to these questions. Specific types of closed questions are for instance multiple-choice and mapping questions.

Main advantage of these questions is that the identification of correct answers and therefore the correction of a students response is easily possible. Additionally, questions of this type are usable for several domains. They are used for driving license tests as well as for medical education. Disadvantages are that “objective activities (true/false questions, multiple choice questions, short-answer questions) are designed to check student understanding and involve little creativity” [Brusilovsky and Miller, 2001]. It is quite hard to design a test that calls for higher order thinking skills (see section 2.1.5) or to find good distractors. Because of the known and limited range of answers, guessing is fostered.

Semi-open Questions  The range of correct answers to a question in the second subcategory of objective questions is known by a teacher as well. In contrast to the first subcategory, the learner has to formulate an answer. A list of possible answers is not available. For instance, fill-in-the-blank questions semi-open.
Advantages in contrast to closed questions are that learner activities are increased and the chances of success for guessing are reduced. Additionally, answers can be assessed for different criteria. For instance, the correct answer is assessed as well as its spelling.

Open Questions  Open or rather applicative questions require the creation of a creative solution. “Applicative activities involve students in serious problem solving, development, or exploration”, [Brusilovsky and Miller, 2001]. A range of possible answers is not available for both, the learner and the teacher. Questions of this type foster the transformation of tacit knowledge to explicit knowledge as well as to facilitate additional knowledge acquisition with “learning-by-doing” [Brusilovsky and Miller, 2001]. Examples for applicative questions are the creation of a diagram, painting of an image, or writing source code for a computer program.

Advantages are that higher order thinking skills are better addressable than with objective questions. Especially the formulation of the question requires not such effort. Disadvantages are concerning the correction process. Individual answers have to be corrected individually and with expert knowledge in the related topic. Therefore, questions of this type are not easily transferable to other domains.

All of these questions are applicable for summative purposes, e.g. within an exam, as well as for formative purposes. Table 2.4 shows the covered dimensions of questions in a formative setting. It emphasizes that the main difference between objective and applicative questions is there convergence.

<table>
<thead>
<tr>
<th>Question Type</th>
<th>Intention</th>
<th>Transparency</th>
<th>Chronology</th>
<th>Target</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>Formative</td>
<td>Formal</td>
<td>Continuous</td>
<td>Product</td>
<td>Convergent</td>
</tr>
<tr>
<td>Applicative</td>
<td>Formative</td>
<td>Formal</td>
<td>Continuous</td>
<td>Product</td>
<td>Divergent</td>
</tr>
</tbody>
</table>

Table 2.4. Covered assessment dimensions of formative questions.

Portfolios

An educational portfolio is a set of artifacts which describe a sequence of individual learning processes [3]. The range can vary from a single learning activity (e.g. writing an essay) to a whole learning biography of a single learner. Possible artifacts are certificates, awards, or created materials (e.g. paintings, computer programs). A main objective of a portfolio is to support a systematic recording and reflection of learning experiences and outcomes. Learners have to select materials for their portfolios to illustrate and reflect personal “efforts, progress, and achievements in one or more areas” [Paulson et al., 1991] themselves. Which materials to select depends on the type of portfolio [3]:

16
2.1. Definition and Classification

- Working Portfolio,
- Assessment Portfolio,
- Best Works Portfolio,
- Process Portfolio,
- Interdisciplinary Unit Portfolio,
- Celebration Portfolio,
- Job Application Portfolio,
- Presentation Portfolio.

Portfolios, in general, are no assessment activity, but the case of assessment portfolios shows that they are applicable for assessment purposes. Therefore, criteria or rather learning objectives for such a complex activity have to be clearly defined [3].

<table>
<thead>
<tr>
<th>Intention</th>
<th>Transparency</th>
<th>Chronology</th>
<th>Target</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formative (Summative)</td>
<td>Informal (Formal)</td>
<td>Continuous (Final)</td>
<td>Process</td>
<td>Divergent</td>
</tr>
</tbody>
</table>

**Table 2.5.** Covered assessment dimensions of assessment portfolios.

2.1.3. Feedback

"Individual feedback may communicate: what is right in the correct answer, what is bad in incorrect and partially incorrect answer, provide some motivational feedback, and provide information or links for remediation" [Brusilovsky and Miller, 2001].

Feedback to learning activities is an important factor of successful learning [Gagné, 1985, Schroeder and Spannagel, 2006, Sippel, 2008, Sippel, 2009]. Especially in formative assessment settings, feedback can be used to motivate learners [Bostock, 2004] as well as to support self-reflection. Additionally, elaborated feedback can provide extra input for learning. Based on the extent of information which is provided to a learner, five levels of feedback can be distinguished (cf. [Narciss, 2006]):

- Knowledge of Performance (KP)
- Knowledge of Result/Response (KR)
- Knowledge of Correct Response (KCR)
2. Assessment

- **Answer until Correct and Multiple Try Feedback (AUC/MTF)**
- **Elaborated Feedback** (EF)

**Knowledge of Performance** – A learner gets information about his or her performance in form of a single value on a rating scale after one or more activities are finished. For instance, a learner writes an exam and after a correction period the feedback that 35 of 50 points, 75%, or a grade B has been reached.

**Knowledge of Result/Response** – The information about correct and incorrect parts of a learner’s solution is provided. For instance, a learner attends an objective test and afterwards gets his or her sheet back, in which wrong answers are marked in red.

**Knowledge of Correct Response** – All correct answers, or a sample solution in case of an applicative question, is provided. For instance, a solution to a homework is presented by the teacher the next day.

**Answer until Correct and Multiple Try Feedback** – Learners get knowledge of response and are allowed to improve their solutions for resubmission. This resubmission could either be possible until the submission does not contain any mistakes (AUC) or feedback is only given to a limited number of tries (MTF). For instance, a student prepares his or her bachelor’s thesis and sends drafts to his or her supervisor. Each feedback of the supervisor is used to improve the next draft of the thesis.

**Elaborated Feedback** – In addition to knowledge of result and knowledge of correct response, further descriptive information is provided to support correction of the current solution as well as the creation of future solutions. For instance, a learner creates a computer program. He or she gets the feedback that the program generates wrong numbers for a certain input (KR), what the expected result is (KCR), and that a used for-loop starts at the wrong index in line 45 of the source code (EF).

Which level of feedback is most suitable depends on what kind of knowledge and skills (e.g. facts, concepts, presentation skills) have been addressed. “Highly elaborated feedback is especially relevant for very complex procedural skills”, [Musch, 1999]. The purpose and effect of elaborated feedback additionally depend on the type of feedback:

**Feedback Type**

- **positive feedback**, 
- **negative feedback**, 
- **hint**.

A sentence like “Well done!” is positive and aims to increase motivation. Negative feedback like “The assumption that … is wrong, because …” tries to initiate a process of rethinking problem solving approaches [Musch, 1999]. A hint, e.g. “Recursion could help you to solve the problem of …”, aims to improve the learning
process by providing situated and individual information as additional learning aid. Furthermore, positive feedback and hints can be equipped with a rate of detail. If the feedback is presented to the learner depends on his or her response certitude [Musch, 1999, Kulhavy and Stock, 1989]. If the rate of detail is less than the certitude, the feedback is only presented on demand.

Another source of impact on the effect of feedback is based on the time when feedback is provided (cf. [Musch, 1999]):

- timely or direct feedback,
- delayed feedback,
- feedback on demand.

Some researchers argument that feedback has to be provided timely in order to facilitate learners identifying a relation to their solutions [Weicker and Weicker, 2005, Stoyan and Glinz, 2005]. Moreover, timely feedback is needed to allow multiple try feedback [Weicker and Weicker, 2005, Malmi and Korhonen, 2004]. Other studies show that delayed feedback is an additional phase of learning with repeated presentation of information [Musch, 1999] and leads to better remembering of correct answers [Niegemann et al., 2003, Musch, 1999, Kulhavy and Wager, 1993, Kulik and Kulik, 1988]. Alessi and Trollip recommend to provide feedback as soon as a learner has taken an approach which cannot lead to the correct result [Alessi and Trollip, 1985]. Because each timing has its advantages and disadvantages, a combination of immediate feedback for knowledge of (correct) response and delayed elaborated feedback is recommended [Roper, 1977]. “In this manner, students can have immediate knowledge of the correctness of their response, but still have time to think about the error before informational feedback is given”, [4].

Feedback on demand focuses on elaborated feedback to learning processes, where the demand for feedback is not only based on a submitted product, but arises during a process [Bescherer and Spannagel, 2009]. This demand can be claimed by a learner or a teacher. For instance, a learner asks a teacher for help on the homework which has to be handed in next week (demand claimed by learner). Furthermore, a teacher could present a topic in more detail again because a related homework has been solved badly by many learners (demand claimed by teacher).

2.1.4. Assessment Methods

In addition to involve different assessment activities (cf. section 2.1.2), the variation of assessment methods is another possibility to diversify assessment [Race, 2001]. Assessment methods are characterized by mainly two factors: the type of assessor and the type of assessee.
2. Assessment

The assessor is the actor who performs the correction, i.e. evaluates the learners contribution to generates feedback. Mainly four different types of assessors can be distinguished:

Assessor
- tutor (Tutor Assessment),
- computer (Automatic Assessment),
- peers (Peer-Assessment),
- self (Self-Assessment).

Assessees are the learners who are assessed by the assessor. Mainly two types of assessees can be distinguished:

Assessee
- individuals (Individual Assessment),
- groups (Group Assessment).

These different factors of an assessment method as well as related organizational aspects are presented in the following. Because of the huge amount of characteristics of automatic assessment, this topic is discussed in section 2.2 in more detail.

Tutor Assessment

“When feedback from an expert witness is really needed, there is no substitute for tutor assessment” [Race, 2001].

Formal learning scenarios are often characterized by a precise separation of learners and teachers. These roles are often adapted directly, i.e. teachers are assessors and learners are assesses. Advantages of this method are that teachers professional expertises as well as their routine with assessment facilitate high quality feedback with ensured reliability. A main disadvantage, especially in scenarios with a low teacher-learner ratio, is the huge amount of effort needed to create high quality elaborated feedback in big extent. In addition, it is somehow difficult to guaranty objectivity.

One approach to increase objectivity, efficiency, and transparency of manual correction processes is by structuring with the application of rubrics or rather marking schemes [Becker, 2003, McCauley, 2003]. Therefore, a hierarchic scheme of categories is created to identify criteria and possible ranges for marks. Teachers use this as a guideline for correction to increase objectivity by mark all submissions with the help of the same measure. Creation of a meaningful rubric for open questions is complicated.
2.1. Definition and Classification

**Group Assessment**

Another approach to reduce the workload for correction is the arrangement of groups to work on activities cooperatively. A more important objective of group assessment is to improve learning success for all group members [Hinze, 2004]. Additionally, group assessment provides opportunities to strengthen soft-skills like communication skills or conflict management [Kopp and Mandl, 2007].

"Effective collaborative learning includes both learning to effectively collaborate and collaborating effectively to learn" [Jermann et al., 2001].

According to the *Sociocultural Theory* [Vygotsky, 1978], higher order mental functions are only acquirable by social interaction. Cooperative learning is able to support learning by forcing social interaction between group members. To improve the group outcome learners have to discuss and explain approaches and facts to each other. Hence students have to teach, what is the most effective form of learning [Hinze, 2004]. In this context it can be related to *cognitive elaboration* (cf. [Kopp and Mandl, 2007]).

The theory of *Situated Learning* [Lave and Wenger, 1991] emphasizes the need to embed learning activities in related professional and social contexts. Because application contexts of modern work is strongly based on social activity, group learning is a well suited context for learning ([Hutchins, 1995] cited in [Wessner and Pfister, 2001]). The topic of *distributed cognition* deals with the idea that knowledge is not replicated to the minds of all group members, but distributed among the group members [Wessner and Pfister, 2001]. In addition, group learning takes place in terms of constructionism [Papert and Harel, 1991], because knowledge is constructed, articulated, and reflected as well as embedded to a social context [Dinse and Bonczek, 2006].

A disadvantage of group learning is that social conflicts and bad coordination processes can lead to decreased learning success. If the combined result of a group is less than the combination of all individual results is called *Ringelmanneffekt* [Hinze, 2004]. Furthermore, the distribution of workload can be imbalanced. On the one hand freeloaders benefit from extra work of the remaining group members [Gibbs, 2009]. On the other hand overwhelming people inhibit effective cooperation by wanting to do all work on their own.

Therefore, it is crucial to guide group learning. Learning is not influenced by cooperation per se, but more by social interactions which result from instructional constraints [Slavin, 1995].

Two different modes of group learning have to be distinguished [Hinze, 2004]:
2. Assessment

Group Learning Modes
- cooperative learning,
- collaborative learning.

Cooperative learning is a simple additive combination of individual learning processes which are performed in parallel. For instance, a group is working on an assignment sheet, if each team member works on a different subquestion. Activities in Collaborative Learning scenarios are performed in teamwork. For instance, all group members are involved in all subquestions. All results are discussed within the group. A strict disjunction of both methods is often either not reasonable nor possible [Dillenbourg, 1999].

Group Marking In many cases the assessment of group learning is done the same way as assessment of individual learning. That means all group members are marked the same way based on their groups’ results, independent from their individual contribution. Motivational rationals as well as institutional constraints can lead to a need of assessing the individual contributions as well. If the improvement of soft-skills is an objective, the assessment of the group work itself could be relevant, but hardly possible [Race, 2001]. [Race, 2001] describes seven different approaches to organize and assess group learning:

Group Marking Strategies
- Simple Marking – All team members get the same score based on the group result.
- Divide and Conquer – A group gets different tasks which are divided among the group members. Each learner is judged based on his or her individual contribution.
- Differentials – The group results are assessed by a tutor. The group receives a score of the group mark multiplied with the number of team members, which has to be divided among all members.
- Contribution Marks – A learners individual mark is a combination of the group mark and a mark for their individual contribution assigned by their peers within the group (Intra-Peer Assessment).
- Further Tasks – A learners individual mark is a combination of the group mark and a mark for an additional individually solved task.
- Oral Exam – A learners individual mark is a combination of the group mark and a mark for an individual oral exam.
- Written Exam – A learners individual mark is a combination of the group mark and a mark for an individual written exam.
Peer-Assessment

“Peer assessment is assessment of students by other students, both formative reviews to provide feedback and summative grading” [Bostock, 2000].

Peer-assessment, the assessment of each others activity among learners, can be differentiated into peer-review and peer-grading [Søndergaard, 2009]. The process of providing elaborated formative feedback to reviewed performance of other learners is called peer-review. Peer-grading emphasis the processes of providing summative feedback or rather grades to peers’ work. Regardless of the type of peer-assessment, there are mainly four independent factors, which influence the strategy and organization of a peer-assessment process [Millard et al., 2008]:

- number of authors,
- number of artifacts,
- number of reviewers,
- number of reviews.

**Artifacts** are the products which have been created by an author or a group of authors. These artifacts are reviewed by one or more reviewers, what leads to reviews as a resulting product of the process. In peer-assessment scenarios learners act as authors and as reviewers as well. Six different strategies to organize peer-assessment are drafted by [Millard et al., 2008] with references to more detailed descriptions of these strategies:

- **Simple** – Each learner creates a single artifact which is reviewed by a single peer. Thus, there is exactly one review created for each artifact.
- **Round Robin** – Each learner creates a single artifact and reviews the artifacts created by each peer (within a subgroup). In a setting with \( n \) learners, \( n - 1 \) reviews are created for each artifact.
- **Group Activity** – A group of learners creates a common artifact. Each learner reviews a result of another group. This strategy is also called **Inter-Peer Assessment** [Race, 2001].
- **Group Review** – A group of learners creates a common artifact. Each learner reviews the result of his or her group as well as the performance of each group member. This strategy is also called **Intra-Peer Assessment** [Race, 2001].
- **Committee Review** – Multiple independent reviews are created for each artifact. All reviews to a single artifact are combined to single review by a review committee.
2. Assessment

- **Multiplicity** – In a scenario in which each learner creates several artifacts, this strategy applies the round robin strategy to each artifact in parallel.

Which strategy is the most effective one depends on the scenario. Mostly, the review of an artifact by multiple peers in parallel is considered to be more effective [Race, 2001]. “When a large amount of peer feedback is combined with a restricted amount of tutor feedback, students can benefit from both quality and quantity” [Race, 2001]. Furthermore, learners benefit from acting as reviewers. Formulating feedback, especially applying criteria instead of only reading or examine artifacts, improves learning effects [Race, 2001].

Mainly seven challenges for the realization of peer-assessment have to be taken into account [Hamer et al., 2005]:

1. distribution and collection mechanisms,
2. validity and reliability of grading,
3. student motivation,
4. ‘rogue’ reviewers,
5. anonymity,
6. plagiarism,
7. grading disputes.

According to [Sondergaard, 2009] the challenges can be addressed as follows. Challenges (1) and (5) are solved by utilizing computer based systems (cf. section 2.2). (2) and (7) do not arise if only peer review (without grading) is used. The grading of the learners’ review performance itself reduces the value of challenges (3) and (4).

**Self-Assessment**

> “Self-assessment skills are invaluable in the context of life-long learning”, [Race, 2001].

If an assessee is his or her own assessor, this is called *self-assessment*. Reflecting and assessing their own performance supports self-critical thinking. Furthermore, “[...] giving students feedback on their self-assessment (rather than just on their work) causes deep learning for students”, [Race, 2001]. A problem of this approach is that learners are influenced to be not objective or rather honest to achieve a better grade if the amount of moderation and grading of self-assessment is to high [Race, 2001].
2.1.5. Classification by Bloom’s taxonomy

The various characteristics and possible mechanisms of assessment mentioned above can be utilized to address different levels of learning or rather cognitive skills. [Bloom, 1956] presented a taxonomy of educational objectives, which is known as Bloom’s Taxonomy. He differentiates six cognitive levels within a hierarchical order (see figure 2.2a): knowledge, comprehension, application, analysis, synthesis, and evaluation. [Bloom, 1956] assumed that a lower level has to be mastered before the next level can be reached for a specific topic.

- **Knowledge** – Terms, facts, or simple concepts are recognized and remembered by learners.
- **Comprehension** – Learners understand concepts and their relation. They are able to explain, illustrate, or paraphrase them.
- **Application** – Techniques and facts are applied to solve problems in new situations.
- **Analysis** – Learners are able to analyze problems, identify parts and distinguish between facts and derived statements.
- **Synthesis** – A creative process to create new material or knowledge by combination of know parts.
- **Evaluation** – The ability to evaluate and categorize material according to a set of criteria, which can be either given or has to be defined by the learner.

![Figure 2.2. Six cognitive levels of Bloom’s taxonomy.](image)
2. Assessment

- **Conceptual knowledge** – Models, theories, structures, and concepts.
- **Procedural knowledge** – Mechanisms, methods, or techniques and their application.
- **Meta-cognitive knowledge** – Knowledge about learning process themselves.

The Cognitive Process Dimension of Bloom’s Revised Taxonomy is strongly based on the six levels in the original taxonomy. Nouns have been changed to verbs and to two highest levels have been swapped (see figure 2.2b). The categories of this dimension are as follows [Anderson et al., 2001]:

- **Remembering** – Keeping factual knowledge in long-term memory.
- **Understanding** – Gaining conceptual knowledge from oral, written, or graphical material.
- **Applying** – Using procedural knowledge on given material.
- **Analyzing** – Material is analyzed to identify parts, their relation as well as their overall structure.
- **Evaluating** – Criticize material based on criteria.
- **Creating** – Reorganize parts of available knowledge from all levels to create new facts, structures, or techniques.

Assessment activities and methods, which utilized to reach different purposes and cover different dimensions of assessment, can be characterized by the levels of knowledge and cognitive process they are addressing. For instance, objective questions are mostly used to assess remembering of factual and conceptual knowledge. Applicative questions are focused on application and creation of procedural knowledge. Meta-cognitive knowledge is addressed by portfolios. Self- and Peer-Assessment provides a learning situation to foster analyzing and evaluating performance of peers or oneself.

2.1.6. Assessment in higher education

“Assessment influences not only what parts of a course get studied, but also how those parts are studied”, [Kirkwood and Price, 2008, p. 5].

Courses at university are a specific form of formal, institutional learning. As mentioned above, this form of learning is heavily affected by assessment. It is not only a concomitant phenomenon, but “most students’ learning is driven by assessment”, [Race, 2001]. As an institution of qualification and certification, summative
2.1. Definition and Classification

assessment is formally organized and documented in official examination regulations, which are proven by accreditation body. The relevance of assessment has been increased by the reorganization of the European Educational System based on the Bologna Process [5] [Reinmann, 2009]. This is described in the next subsection. As an example for standard processes of formative assessment, which are not covered by examination regulations, usual assignment management processes in higher education are explained afterwards.

**Regulations and possibilities in German higher education**

“The introduction of Bachelor and Master degree programs in the context of the 6th amendment of the German Framework Act for Higher Education (Hochschulrahmengesetz) results in an extensive reorganization of examinations at German universities”, [Wannemacher, 2009].

The Bologna Process [5] started in 1999 to create a unified European Educational System. Therefore, the federal systems of the participating countries had to be reorganized [Müller and Schmidt, 2009]. For Germany, and for other countries in this context as well, especially the organization of assessments has changed significantly. The number of assessments increases enormously, because of the compulsory introduction of course-related examinations [Wannemacher, 2009, Reinmann et al., 2007]. An examination is called course-related, if it happens within short time after a course, in which competencies for the exam have been taught [Reinmann, 2007]. In this way, students gain Credit-Points (ECTS) which have to be collected to pass a course of studies.

An additional factor, which influences the complexity as well as the effort of teaching and organization, is that course-related assessments have to satisfy several formal requirements [Wannemacher, 2009]. This, together with the huge amount of assessments, mostly leads to avoidance of complex and time-consuming assessment approaches. Thus, applied assessments are reduced to some basic forms [Wannemacher, 2009]. Preferred forms are simple knowledge requests, which can be corrected quickly and easily, to reduce time and effort [Reinmann et al., 2007]. “[...] feedback is only of minor importance in German Universities. The structur-oriented approach in teaching and assessment causes the absence of feedback” [Sippel, 2009]. With a main interest of selection, university rector, politicians, and others prefer standardized exams, which allow automated correction, to verify high standards to accreditation body [Reinmann, 2007].

It seems that the relevance of assessment for learning is not addressed by the Bologna Process. This could lead to a mismatch between innovative learning approaches on the one hand and limited assessment possibilities on the other hand.
2. Assessment

[Reinmann, 2007]. Because “learning targets, learning methods and assessment methods need to be matched” [Müller and Schmidt, 2009], a scope for the needed innovative assessment approaches has to be created in form of assessment for learning. Assessment for learning and, by implication, elaborated feedback is needed to reach a shift from teaching to learning as well as to facilitate the identification of competencies [Sippel, 2009, Sippel, 2008]. This has to be designed in way, in which it is not affected by examination regulations to reduce organizational effort as well as voidability [Wannemacher, 2009].

The mentioned reorganization, especially in terms of assessment, is still a topic of current research as well as of political discussion [Reinmann, 2009].

Assignment Management in Higher Education

According to experience, weekly assignments are one of the mostly established forms of formative assessment in German higher education. This is especially true for natural science, math, and technical education. Assignments are carried out related to a lecture to provide an opportunity for putting knowledge into practice. This is often realized by periodically published assignments, mostly every week, which have to be solved by the students. The related organizational process is called assignment management throughout this thesis. Especially, in course with many students (e.g. 200 to 1000 and more), assignment management requires a lot of time and effort. Figure 2.3 illustrates a basic lifecycle for assignment management.

![Assignment Management Lifecycle](image)

**Figure 2.3.** Common Assignment Management Lifecycle.

In general, there are typically three different groups of people involved in this process: *lecturers, tutors, and students*. Lecturers, or more often their assistants, create an assignment and distribute it. For instance, printed assignment sheets are
2.2. Technology Enhanced Assessment

laid out during a lecture. Students start to create a solution and hand them in, after they have finished their work. In huge science classes this has often to be done before a specific deadline. With pen and paper created solution sheets are often handed in to physical boxes, which are emptied by tutors after the deadline has been reached. Tutors, and in small classes lecturers and their assistants themselves, correct students’ submissions and provide individual feedback. This feedback can be text (knowledge of response and elaborated feedback) as well as marks or a certain score (knowledge of performance). Additionally, a sample solution can be provided (knowledge of correct response).

Extended forms of this basic lifecycle are possible as well. Groups or tutorials are applied as well as additional feedback loops. More advanced assignment management lifecycles are addressed in section 6.2. Possibilities for technological support of assignment management, and assessment in general, is described in the next subsection.

2.2. Technology Enhanced Assessment

Technology Enhanced Assessment (TEA) or e-assessment describes the utilization of technology for organization and execution of assessment activities. Depending on the degree of support Computer Assisted Assessment (CAA) and Computer Based Assessment (CBA) are distinguished [Carter et al., 2003, Baillie-de Byl, 2003]. With CAA, some parts of an assessment process are supported technologically. CBA describes the use of systems to cover a whole assessment process. If the system is web-based, the corresponding terms WAA and WBA can be used. This notation depends on technological evolution only — starting from physical technology over mainframes, personal computers, web-based variants to emergent technologies — and means the same categorization. In general, the utilization of technology for assessment can be categorized as follows.

- **Support:**
  digitalization of established processes.

- **Enhancement:**
  (semi-)automatic evaluation of students’ submissions,
  integration of multimedia content,
  integration of innovative service.

CAA encompasses the use of computers to deliver, mark or analyse assignments or exams. [Sim et al., 2004]
2. Assessment

Related to these categories, there are several objectivities why assessment processes are supported and enhanced by technology:

- reduce resources,
- produce timely feedback,
- facilitate self-assessment,
- increase objectivity,
- allow multimedia content,
- support every time and everywhere access,
- increase variety of assessment forms.

Beside these functional objectives, with the use of e-assessment it is possible to increase students’ motivation [Bostock, 2004]. And “appropriately designed assessment that exploits the potential of ICT can change students’ approaches to learning”, [Kirkwood and Price, 2008, p. 5]. Thus, e-assessment can be used as a didactic tool which facilitates additional forms of learning.

A lot of different systems have been developed to reduce the required time and effort which comes along especially with assessment for learning. Figure 2.4 shows a classification of those systems which can be divided into four categories Electronic Test Systems, Process-Specific Systems, Domain-Specific Systems, and Learning Management Systems (LMS). More general, there are generic systems, which are design for basic support of different learning scenarios and different domains. Scenario-Specific Systems provide more specialized and comprehensive support of some fixed scenarios. The main characteristics and functionalities of systems in each category are described in the following subsections. With emergent technologies, namely Web 2.0 or Cloud Services, new forms of learning and assessment are possible. These are presented in a separate subsection as well. Afterwards, the different approaches will be classified by common learning theories. This section closes with an overview of selected standards and models for the realization of e-assessment systems. An analysis of related tools follows in chapter 3.

2.2.1. Electronic Tests

Electronic Tests (e-tests) in their simplest form are a direct electronic representation of paper-based tests, which are a sequential combination of objective questions. These are presented on screen as a form, which has to be filled and submitted by a learner. The kind of interaction to answer a question depends on the question type. Simple choices from Yes or No, single or multiple selection of multiple choice, or fill-in the blank questions are digitalized presentations of the related paper interaction. Advanced forms facilitated by technology, e.g. drag-and-drop questions, are
2.2. Technology Enhanced Assessment

available as well. A detailed overview of processes, tools, and advanced features of e-test systems can be found in [Brusilovsky and Miller, 2001, pp. 11-18]. Based on that, a brief overview of main characteristics is presented in the following. Related tools and classifications are analyzed and classified in section 3.1.

Several models for the description of assessment processes, and mainly e-test processes, are separately presented in section 2.2.7. In general, the process for application of e-tests is divided into three steps (see figure 2.5).

**Authoring**  Firstly, an educator has to create each question. A question text, the type of interaction, a set of possible answers, feedback, and additional metadata has to be defined. The list of answers contains correct answers as well as wrong answers as distractors. Feedback in form of static multimedia content (e.g. content, images, or hyperlinks) is attached to the whole question as well as for each possible answer.
2. Assessment

Some tools allow simple distinction of answers to choose or generate feedback content. A guide to design good objective questions can be found at [6].

Questions are stored in pools, in which they can be found by declared metadata, e.g. keywords, level of difficulty etc. The method of storing questions can be either in presentation format (e.g. HTML) or in an internal format (e.g. XML). Questions in presentation format are static, whereas questions stored in internal formats are dynamically transformed into presentation formats. Advantages of the latter approach are that the type of interface (e.g. radio buttons, drop-down menus etc.), the technical environment (e.g. HTML, desktop application etc.), and other options (e.g. order of answers) can be dynamically determined within the transformation. A common internal format called QTI is presented in section 2.2.7.

A test is composed of several questions. Simple tests are a static sequences of predefined questions, which are manually chosen from the question pool (authoring time flexibility). Advanced tests can be created automatically generated from the pool (delivery time flexibility). In this case, questions are randomly selected based on given metadata to create personalized tests. This can be increased with use of parameterized questions. Another approach of personalization is the introduction of adaptivity. Therefore, selection of questions can be based on a learner model. Additionally, rules have to be defined, which control the sequence of questions based on selected answers.

Delivery In this stage of an e-test learners create a new personal attempt to an e-test. Answers to a question are saved locally or posted to the server. This depends on the delivery medium. Web-based systems presenting HTML as well as client software are possible. The dependency on additional software (test-players) or browser plug-ins (e.g. Flash or Silverlight) has to be considered.

Although, the possibilities for navigation through a test dependent on the type of delivery as well as of test configurations. For instance, the navigation could be limited to a sequential flow of questions one after the other. It is possible as well to allow skipping between questions and annotating them. A limitation of time to answer a test could be set up as well.

Evaluation Submitted answers are evaluated by the test system. Afterwards feedback is presented to the learner. Several methods for feedback delivery, singular or in combination, are possible:

- **Direct feedback page** – Feedback for a single question or a page of questions is presented after each intermediate submission.

- **Summative Result** – The overall result, e.g. ’75% reached’, is presented after finishing a test.
Feedback walk-through – After a test is finished, the feedback for all questions can be reviewed in a sequential manner. The kind of presented feedback can vary according the levels of feedback (cf. section 2.1.3). Thus, correctness of selected answers can be presented, correct answers can be highlighted, and static feedback to justify correct and incorrect answers can be given.

Research Topics  Advanced approaches for e-test systems facilitate the combined evaluation of related questions [Altenbernd-Giani et al., 2008] or realize domain-specific forms of interaction [7]. In general, there are mainly four technical research topics in the area of electronic test systems [Brusilovsky and Miller, 2001]:

- Metadata – Tools are in development to support easy specification of metadata for questions. Other approaches try to facilitate searching and sharing of questions for manual test composition as well as for automatic test generation based on metadata.
- Question types – More interactive question types and advanced evaluation technologies have to be investigated.
- Parameterized questions – Parameters are used in different domains, e.g. math, to individualize questions and decrease plagiarism.
- Adaptive testing – Artificial intelligence technologies are used for automatic test generation out of knowledge bases as well as adaptive sequencing of questions.

Further didactic research topics are related to questions about how e-test have to be designed to allow assessment beyond remembering of facts: how feedback for the process of answering a question can be generated or how higher order thinking skills can be addressed by use of e-tests.

E-examinations  Further challenges arise when electronic tests are used for examinations. Additional issues like security, privacy, reliability, and others have to be considered. An overview of approaches and technologies can be found in [Vogt and Schneider, 2009].

2.2.2. Generic Assessment Support in LMS

Learning Management Systems (LMS) are centralized systems which are designed to support a wide range of learning and teaching processes, especially for managing courses [Schulmeister, 2005, Bäumer et al., 2004]. “At nearly every European university, school, or other educational institution learning management systems are in
2. Assessment

use”, [Kalz et al., 2011]. They are either used by each institute or centrally hosted for all institutes of a university or other institution. But the variety of systems is large. There exist a huge range of different, mostly web-based, systems, e.g. Moodle [8] or BlackBoard [9] (see section 3.2). Although, most of them provide the same common functionalities, which can be classified as follows (cf. [Schulmeister, 2005, pp. 5-15], [Bäumer et al., 2004]):

- administration (user management, course management, role and rights management etc.),
- communication (chat, forums etc.),
- collaboration (wikis, whiteboards, shared documents etc.),
- presentation of content (slides, videos, scripts etc.),
- assessment (e-tests, exercises etc.).

These functionalities are mostly organized within a structure of virtual course rooms (see figure 2.6). Each virtual course room is related to one specific course and is only accessible by enrolled students for that course as well as related teachers or tutors. Thus, the provided functionalities are used in context.

These systems are developed for generic application to traditional, institutional learning in almost any course. Therefore, the integrated assessment functionalities are mostly rudimentary and focused on generic application. Definition and execution of electronic tests are mostly supported by integrated test modules. Basic support of a standard assignment submission process is often available as well. Main advantages of assessment modules within LMS are the integration with the other above listed functionalities, which are not directly related to assessment, as well as their contextual usage. Support of specific assessment methods are mostly missing.

Future challenges for further development of LMS, which influences the capabilities of assessment support are the following ones:

- Extensibility – Current LMS are closed environments which have to be opened to allow extension by additional assessment tools and functionalities. This is important for a sustainable support of learning management.
• **Service-integration** – Various tools and services are provided on the Internet which support content management, communication, collaboration, and other functionalities that are meaningful in the context of learning as well.

• **Personalization** – Possibilities for adaptability as well as adaptivity of an LMS to provide a personalized support experience have to be investigated.

• **Service-orientation** – Modularity as well as the possibility for integration of tools of an LMS into other systems could be facilitated by service-oriented approaches.

• **Mobile access** – The integration of LMS functionality into the context of mobile learning could be achieved by facilitation of mobile access. Further possibilities by using additional context information have to be considered as well.

• **Cloud scenarios** – Currently, each institution uses its own LMS. In terms of lifelong learning as well as regarding to cooperation across institution, cloud hosting scenarios have to be evaluated.

### 2.2.3. Process-specific Systems

Commercial learning management systems still have less than adequate support for peer assessment, but a large number of purpose-built tools have been created in the recent years. [Sondergaard, 2009]

**Process-specific systems** are more focused to support specific organizational steps of assessment, especially of assignment submission. Both generic systems, e.g. e-mail or file server, and specific tailor-made systems are part of this category. A huge variety of tools are available, which have advantages as well as disadvantages in different categories of application [Heinrich et al., 2009, Milne et al., 2008, Milne et al., 2007] (cf. section 3.3). The practical usage of these systems has been approved in many courses. An evolution of online assignment management, like described in [Jones et al., 2005b, Jones et al., 2005a, Jones and Behrens, 2003], probably happened at different educational institutions:

• 1994-1995 – manual e-mail,
• 1995-1996 – automated e-mail,
• 1996 – unintegrated web,
• 1997-2000 – integrated web,
• 2001-now – evolutionary development.
2. Assessment

Initially, assignments as well as student’s submissions had been directly send via e-mail. Afterwards e-mails had been collected in web-based platform. Standalone submission platforms had been developed to provide advanced assignment management mechanisms. These systems have typically been integrated into a universities technical infrastructure, e.g. a centralized identity management. Nowadays web-based systems are developed to handle the whole submission process [Jones et al., 2005b]. These systems emphasize only a small selection of specific functionality. For instances, systems are focused on individual assignment submission [Six et al., 2001, Brunsmann et al., 1999], group assessment [Behringer et al., 2004], the (collaborative) creation of exercise sheets [Wimmer et al., 2006], or peer review [Al-Smadi et al., 2010, Millard et al., 2008, Hamer et al., 2007]. Thus, each of them is only applicable to a set of specific scenarios, but to which they provide precise assistance in depth.

Grouping  Letting students work on assignments in small learning groups is common approach to increase efficiency (cf. section 2.1.4). Thus, assignment management systems which are supporting group assessment typically include digitalized grouping mechanisms. Mainly, there are three approaches for digitalized grouping mechanisms [Haake et al., 2004, Haake and Schümmer, 2003]:

- free access,
- assigned by a coordinator,
- self-directed.

The creation of a group can be initialized by a supervisor or the group members themselves. The allocation of a person to a group can be done manually or with the help of an automatic allocation strategy. Self-directed access can be realized in two directions. Either new group members are asked to join by invitation or they request access to the group actively by themselves. The latter has to be confirmed by the group.

Most of the mentioned functionalities are already supported by right and role management mechanisms of common content management systems. Since specialized assessment tools are mostly monolithic systems, they are often re-implementing a subset of the possible grouping functionality for their own usage.

Marking and Grading Common spreadsheet software (e.g. Microsoft Excel, OpenOffice Calc, or Numbers) as well as grade book software or rubric systems can be used to support bookkeeping of students’ results. Spreadsheets are mostly used to ease use for teachers and allow the creation of reports, which can be published separately. Specialized web-based tools additionally allow students
2.2. Technology Enhanced Assessment

to view their personal results online and reflect them in relation to average results of a course. “The ability to see their grades online is the most student-appreciated feature”, [Brusilovsky and Miller, 2001]. By publication of applied marking rubrics to students the transparency of the marking process can be increased [Ahoniemi et al., 2008] (see section 3.3.4).

Advantages and Disadvantages Most intended advantage of these systems is to increase efficiency of related management processes. Especially complex workflows, e.g. the allocation of peer reviewers, are firstly really applicable in larger courses. Added value, for instances time and place independent access to feedback, let most students prefer online submission system against pen and paper [Barker et al., 2008, Bridge and Appleyard, 2008]. Disadvantages associated with new aspects have to be taken into account, e.g. privacy, security, reliability, or even costs for hardware and software [Ridgway et al., 2004]. Only partial support of assessment processes as well as other usability aspects are barriers to continued use of online assignment submission systems [Geri and Naor-Elaiza, 2008]. Especially in mathematics and other disciplines which require students to submit graphical artifacts, e.g. diagrams, graphs, drawings, and formula, offline creation conflicts with online submission. Other disadvantages are non-general usage, such that different tools have to be combined, as well as no specific support related to the content.

In many cases, process-specific tools are restricted to manual correction and feedback generation. A technique called “semi-automatic phrasing”, which allows re-use of text modules, aims to decrease effort for the creation of common feedback [Ahoniemi and Reinikainen, 2006, Ahoniemi et al., 2008]. Additional support for feedback generation is possible by automating domain-specific knowledge.

2.2.4. Domain-specific Systems

Although applicative questions cannot be corrected automatically in general, many domain-specific systems try to utilize specific knowledge for (pre-)correcting open-ended assignments and provide intelligent feedback. These systems can be sub-classified related to the target of assessment:

- Intelligent Tutorial Systems (ITS),
- Automatic Correction Systems.

**Intelligent Tutorial Systems (ITS)** target learning processes during the creation of assignment solutions [Bescherer et al., 2011, Herding et al., 2010]. Students can be guided through the process of problem solving and specific theory inputs can be given. This requires specific knowledge about expected approaches for solving a
2. Assessment

given problem. Furthermore, the whole creation process has to be handled by an authoring tool within the system, to allow analysis as well as guiding mechanisms. For instances, environments for learning to write Java source code within an online tutorium [Bieg and Diehl, 2003] are provided as well as intelligent assessment during mathematical tasks are solved [Bescherer et al., 2009].

Automatic Correction Systems focusing the automatic correction of assignment solutions in a specific domain. Differently to objective questions solutions to applicative questions cannot be evaluated by comparison with an explicit sample solution, due to absence of a unique solution. However, in-depth knowledge is able to facilitate automatic (pre-)corrections by the use of separate evaluation steps. In 1960 Hollingsworth already presented automatic graders for programming classes to analyze and mark students’ source code [Hollingsworth, 1960]. In the course of time several systems have been developed, which apply and enhance the example of the early system in the domain of programming as well as to other domains. Nowadays, systems for source code correction often provide different static and dynamic test procedures for a specific programming language (cf. [Hoffmann et al., 2008, Choy et al., 2007, Malmi and Korhonen, 2004, Higgins et al., 2003]). All of them share common functionalities like submission queuing or sandboxing the evaluation process to decrease vulnerability. Further domains covered are for instances, **musical dictations** [Tremblay and Champagne, 2002], **essays** [Burstein et al., 2004], or **mathematical proofs** [Gruttmann et al., 2008b].

In most cases each new system adds a certain aspect for its domain, e.g. the utilization of JUnit for dynamic code testing [Tremblay and Labonté, 2003]. Basic functionality, e.g. user management or submission handling, which is required in all systems, are mostly re-implemented for each of them. Thus, newer approaches try to developed **extensible systems**, which can be modularly equipped with domain-specific add-ons, while common functionalities are shared [Gruttmann, 2010, Amelung et al., 2008].

Gained advantages by utilization of automatic correction systems are additional elaborated feedback for learners (effectiveness) as well as reduced correction effort for tutors (efficiency). Studies in the domain of programming show that students judge the quality of generated feedback at least as high as manually provided feedback [Striewe and Goedicke, 2009]. By allowing students to resubmit reworked solutions based on generated feedback the lines between automatic correction systems and tutorial systems are blurred [Malmi and Korhonen, 2004].

Because not all aspects of solutions to applicative questions can be corrected automatically in general, a combination of automatic and human correction has to be considered.
“Process-oriented feedback is very task-specific and must be implemented for each class of problems separately. Therefore it is useful to have frameworks which facilitate the implementation of process-oriented feedback [...]”, [Bescherer and Spannagel, 2009].

2.2.5. Assessment 2.0

Systems in the four above mentioned categories have been developed to support common institutional assessment processes. Elliott calls the use of technology to imitate only traditional assessment processes Assessment 1.5 in contrast to paper and classroom-base Assessment 1.0 [Elliott, 2008a, Elliott, 2007]. He recommends to extend and renew assessment strategies along with the innovative possibilities of Web 2.0 to reach Assessment 2.0 [Elliott, 2008a].

When students are active as Web 2.0 authors [Gray et al., 2010], students might profit from the “six big ideas behind Web 2.0” [Anderson, 2007]: user-generated content, the power of the crowd, data on an epic scale, architecture of participation, network effects and openness. Thus, new forms of assessments have to be developed. These forms, which are categorized as assessment 2.0, are authentic, personalized, negotiated, engaging, problem oriented, collaboratively produced, peer and self assessed, supported by tools, recognizing existing skills, and assess deep knowledge [Elliott, 2008a].

Several Web 2.0 services are applicable for evidence generation, which is an interpretation of assessment. Thus, assessment can be divided into five steps in a life-cycle of evidence generation (see figure 2.7).

Table 2.6 illustrates examples of common Web 2.0 services and their application to assessment generation. Some Web 2.0 tools can be understand as learning aids (e.g. social bookmarking) or learning environments (e.g. virtual worlds), rather than learning activities [Gray et al., 2010]. Suggestions for each step are available, except the subprocess of evidence appraisal. Especially in context of institutional learning or certification it is an open research question how to consider students’ Web 2.0 activities.

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Web service</th>
<th>Example</th>
<th>Use(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>RSS</td>
<td>Bloglines</td>
<td>Subscribing to evidence sources</td>
</tr>
<tr>
<td></td>
<td>Instant Messaging</td>
<td>MSN</td>
<td>Discussion; group work; collaboration</td>
</tr>
<tr>
<td></td>
<td>Search Engine</td>
<td>Bing</td>
<td>Locating evidence</td>
</tr>
<tr>
<td></td>
<td>Online encyclopedia</td>
<td>Wikipedia</td>
<td>Finding and publishing evidence</td>
</tr>
<tr>
<td></td>
<td>Social network</td>
<td>Facebook</td>
<td>Collaborating and</td>
</tr>
</tbody>
</table>
2. Assessment

<table>
<thead>
<tr>
<th>Creation</th>
<th>Wiki</th>
<th>Wikispaces</th>
<th>Collaborative writing; projects; research findings; group work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture</td>
<td>Social bookmarking</td>
<td>Del.icio.us</td>
<td>Capturing sources of evidence</td>
</tr>
<tr>
<td></td>
<td>VOIP</td>
<td>Skype</td>
<td>Capturing audio evidence; evidence authentication; oral assessment</td>
</tr>
<tr>
<td></td>
<td>Data capture</td>
<td>Clipmarks</td>
<td>Selecting and storing evidence</td>
</tr>
<tr>
<td>Organisation</td>
<td>Personal homepage</td>
<td>Netvibes</td>
<td>Combining evidence sources on single page</td>
</tr>
<tr>
<td></td>
<td>Blog</td>
<td>Wordpress</td>
<td>Logbook/diary; e-portfolio; authentication</td>
</tr>
<tr>
<td>Storage</td>
<td>E-mail</td>
<td>Google Mail</td>
<td>Storing (and searching for) evidence</td>
</tr>
<tr>
<td></td>
<td>Video upload</td>
<td>YouTube</td>
<td>Selecting and storing evidence</td>
</tr>
</tbody>
</table>

Table 2.6. Usage of web 2.0 tools for evidence generation (cf. [Elliott, 2008a]).

"Many educators concede that student Web 2.0 authoring in higher education raises significant challenges for assessment, posing a barrier to further adoption. [...] Good practice in assessing student Web 2.0 authoring may be inferred from existing general guides to assessment and to assessing group learning, but how to apply them appropriately to the assessment of student Web 2.0 authoring is not always obvious for technical, logistical or pedagogical reasons", [Gray et al., 2010].

Some field reports are available which illustrate how to assess online collaborative assessment [Elliott, 2008b]. It includes the assessment of activity in online discussions [Vonderwell et al., 2007], wiki pages [Cubric, 2007], or blogs [Lee and Allen, 2006]. Current tools are mostly focusing quantitative measures, since it is more difficult to realize qualitative measurement. Further details on specific tools can be found in section 3.5.

**Open Badges** [10] is a project of the Mozilla Foundation which tries to introduce a certification process, especially based on Web 2.0 activity, in a non institutional context. Learners are able to gain so called badges by demonstration of certain skills through online activity. Differently to more abstract certificates, for instances a university degree in computer science, badges intent to express specific competencies or rather skills explicitly. Figure 2.8 displays the Open Badge Backpack, which is a personal storage of gained badges. In this example the user got a red badge for JavaScript-development from the School of Webcraft. He or she can integrate the
badges to his or her web profiles which allows a presentation of skills. Open Badges can be described as a kind of informal certification portfolio.

This project opens the possibility for certification to everyone, not just established institutions. What a learner has to do for gaining a specific badge is not defined in general. In institutional learning, this approach is a common practice. Several further learning programs provide certification logos, which can be used on web pages, business cards etc. after solving a formal examination. For instances, software developers who passed exams related to development of application for a Microsoft SharePoint Server 2010 are allowed to use the certification logo of figure 2.9.
2.2.6. Classification by learning theory

“It would be foolish to suggest that the assessment of student Web 2.0 authoring should totally replace other assessment strategies; it is important to consider that it offers some valuable new assessment options”, [Gray et al., 2010]. Each kind of assessment has its value. This can be justified related to the cognitive levels in Bloom’s Taxonomy (see section 2.1.5) and by learning theories as well.

Several different theories try to explain what learning is. All of them have a different understanding of learning and teaching situations as well as corresponding roles of involved persons. Common theories are the following [Chatti, 2010, Arnold, 2005]:

**Common Learning Theories**

- Behaviorism,
- Cognitivism,
- Constructivism,
- Connectivism,
- Learning as a Network (LaaN).

**Behaviorism** – In this theory acting, thinking, and feeling are interpreted as behaviors. Learning occurs by stimulation to change a learner’s behavior [Skinner and Skinner, 1991]. Facts of a stimulus are memorized and recalled by a learner.

**Cognitivism** – The brain is understood as an active processor to transform new information together with prior knowledge to new knowledge [Arnold, 2005]. A learner is able to analyze and solve problems.

**Constructivism** – Knowledge is a result of an individual process of construction in a specific situation [Arnold, 2005]. It is not possible to transfer knowledge simply from a teacher to a learner. The main idea is that tacit knowledge – the problem of not knowing how to apply a fact to a task in situation [Wendt, 2003, p. 38] – is avoided by active work on tasks.

**Connectivism / LaaN** – The theory of connectivism emphasizes that knowledge is not only present in individuals but spread in multiple sources of information [Siemens, 2005]. The most important questions for learning are where information can be found or who can be asked. Learning is defined as a process of connecting...
sources of information respectively forming networks. *Learning as a Network* (LaaN) is based on connectivism and recognizes learning as continuous creation of *personal knowledge networks* (PKN) [Chatti, 2010, Chatti et al., 2010].

![Classification of assessment systems and functionality by learning theories.](image)

Figure 2.10. Classification of assessment systems and functionality by learning theories.

The choice which of the above mentioned assessment systems is utilized strongly bases on the didactic approach associated to a learning theory (see figure 2.10). Electronic tests are related to the strong aspect of memorization in behaviorism. Applicative tasks, related process-specific and domain-specific systems respectively, are based on the idea of constructivism. Learning management systems combine these aspects. Emergent learning theories like LaaN try to explain the processes which are taking place in new forms of learning, and assessment as well.

2.2.7. Standards and Reference Models

No matter which approach of learning or rather assessment should be supported or enhanced, the question arises if existing models and tools can be used or may be enhanced. This leads to the topic of standardization to classify common processes and functionality. This is an important topic in most areas of information and communication technology (ICT), e.g. network protocols, character sets, file formats etc.

The main purpose of standardization is *interoperability*, which can be defined as possibility to exchange information and services between different systems in a common format [Bull and McKenna, 2004, p. 102]. In case of technology enhanced
learning (TEL) and assessment (TEA) the absence of interoperability makes it difficult to change the used system. Especially, this could occur during a job change to another university. In these cases documents, assessments etc. have to be recreated by a lecturer. Additionally, a lack of integration into other platforms requires redundant content storage and synchronization [AL-Smadi et al., 2009]. Standards for data exchange and integration are needed to avoid these problems. Standardization in TEL aims to unify user interfaces, to be independent of proprietary technology, increase reusability as well as shared usage of learning content from different sources and contexts [Niegemann et al., 2008, pp. 603-604]. A brief overview of standards in terms of TEL and TEA is given in the following.

Classification of standards

Four levels of standards can be distinguished, based on their acceptance and focus [Devedzic et al., 2007]:

- specification,
- de facto standard,
- official standard,
- reference model.

A specification is a document which contains definitions, requirements, formats and terms of reference. It can be seen as suggestion for a future standard. A de facto standard is a specification which is widespread and commonly accepted. An official standard is approved and accredited by an official institution (e.g. IEEE or ISO) additionally. Reference models are more focusing architectural aspects, terms and relations as assistance to design and develop new tools.

Only a few de facto standards in the domain of TEL have reached the status of an official standard. In some domains there is even no de facto standard established so far [Devedzic et al., 2007, pp. 605-607]. Due to multiple intersecting specifications and absence of official standards, a categorization of specifications makes sense [Niegemann et al., 2008]:

- Material packaging and exchange
  IMS Content Packaging, SCORM, AICC/CMI etc.
- Course plan / didactic scenarios
  IMS Learning Design
- Learning content
  IMS QTI, elcoML, FuXML, eLML etc.
• Content metadata
  LOM, Dublin Core etc.
• User metadata
  IMS LIP, IMS RDECEO, IEEE PAPI
• Quality assurance
  ISO/IEC 19796-1, PAS 1032-1

Some of these categories, course planning for instances, are not directly applicable for assessment. A brief explanation of selected relevant specifications (SCORM, IEEE LOM, IMS LP, IMS QTI) is described in the following. An overview of appropriate reference models for assessment systems (Evidence Centered Design, Four-Process Model, Educational Model, FREMA) is given afterwards.

Shareable Content Object Reference Model (SCORM)

SCORM is a collection of different specifications to allow a consistent implementation of learning platforms [11]. It describes form and usage of replaceable objects. Objectives are accessibility, adaptability, affordability, durability, interoperability, reusability of web-based learning content [Vossen and Westerkamp, 2006]. SCORM is based on four documents [Jayal and Shepperd, 2007]: overview, Content Aggregation Model (CAM), Run-Time Environment (RTE), and Sequencing and Navigation (SN).

The overview document contains an introduction to developments, objectives, and relations. The content aggregation models describes organization and representation of resources, e.g. pictures or multimedia. How learning units are processed is covered in the run-time environment, including the data model and the application programming interface (API). Sequencing and navigation contains handling of navigation paths and ordering of content presentation.

“Scorm does not support assessment very much”, [Chang et al., 2004].

Assessment is treated as a kind of learning content in a limited way. Nevertheless, several available tools allow the integration of simple tests into SCORM packages [12].

IEEE Learning Object Metadata (IEEE LOM)

IEEE LOM is used to categorize learning objects or rather content via metadata. Objectives are a structural description of resources, interoperability of resource management as well as a common vocabulary [Barker and Campbel, 2010, Barker, 2005].
Concerning assessment, questions, assignments, tests, and related resources can be identified as reusable learning objects. In practice, at most a reduced form of LOM is applied, because an explicit definition of all metadata is mostly not feasible [Niegemann et al., 2008, pp. 609-610].

**IMS Learner Information Package (IMS LP)**

The IMS LP Specification [13] is used for communication between a learner information server and an e-learning platform. General information of a user or group as well as notes about learning achievements and educational objectives are covered. Privacy and integrity are in the main focus of the specification. Thus, specific privacy settings are assigned to each property. IMS LP has no noteworthy implementation in an assessment system, even if the provided data could be quite relevant for adaptivity. This is the case, because adaptivity features are mostly provided by research prototypes, which realize a self-made learner management, since IMS LP requires to much implementation effort for tools which are not used in productive use.

**IMS Question and Test Interoperability (IMS QTI)**

The most important specification in the area of assessment is IMS QTI [14], which is a de facto standard for interoperability of objective questions, tests, and results. It is used for communication between authoring tools, questions pools, LMS, and e-test systems.

QTI is used to define reusable questions, called items, which can be composed to tests. The definition of such items and tests is done based on XML [Paar, 2005]. Core of the specification is an XML-Schema, which defines allowed structures for items, tests, interactions, feedback, response processing and dependencies. An example of an QTI-based definition for an item is displayed in listing 2.1. It defines a simple multiple choice question with three choices and two correct answers, which is presented with choice-boxes for interaction.

```xml
<?xml version='1.0'?>
<!DOCTYPE assessmentItem SYSTEM 'imsqti_v2p0.dtd'>
<assessmentItem identifier='EX1' title='Which planets are immediate neighbors of planet earth?' adaptive='false' timeDependent='false'>
  <responseDeclaration identifier='R-EX2' cardinality='multiple'>
    <correctResponse>
      <value>choice1</value>
    </correctResponse>
    <value>choice2</value>
  </responseDeclaration>
  <itemBody>
    <itemStatement>
      Which planets are immediate neighbors of planet earth?
    </itemStatement>
  </itemBody>
</assessmentItem>
```
2.2. Technology Enhanced Assessment

Interactions are used to determine the communication between users and items. Answers are mapped to variables. The evaluation strategy is defined with rule definitions as response processing [Dienst, 2008]. Interactions and evaluation are limited to objective questions. Multiple types of objective questions can be evaluated automatically. More complex scenarios could handled hardly or only partially. The following types of interactions are available in QTI:

- Simple interaction:
  - `choiceInteraction` – selecting given choices.
  - `orderInteraction` – ordering values.
  - `associate Interaction` – connecting elements from to sets.
  - `gapMatchInteraction` – fill in the blank by choice.

- Text-based interaction:
  - `textEntryInteraction` – text input.
  - `hottextEntryInteraction` – choice of text parts.

- Graphic interaction:
  - `hotspotInteraction` – selection of points in a picture.
  - `graphicOrderInteraction` – ordering picture.
  - `positionObjectInteraction` – positioning in a given picture.

- Miscellaneous interaction:
  - `uploadInteraction` – uploading a single file.
  - `drawingInteraction` – modification of a given picture.
  - `customInteraction` – any other form of interaction.

Import and export of items and tests are supported by a huge amount of systems. Due to compatibility issues of different versions of the specification and limitations to
2. Assessment

only basic functionality, decreases a convenient usage of QTI [Gorissen, 2003]. Further limitations, which are mainly based on the high complexity of QTI, are discussed in [Piotrowski and Fenske, 2007, Sclater, 2007] in detail. Combinations with other specifications like discussed in [Miao et al., 2007] are possible, but require additional concepts. Covering advanced assessment methods, e.g. peer- or self-assessment, can hardly be realized with QTI.

Evidence Centered Design

_Evidence Centered Design_ (ECD) is a design concept for the realization of assessments [Williamson et al., 2003]. Main objectives of this approach is to increase reutilization of questions. Four types of re-use are distinguished:

Types of re-use

- create new questions by modification of existing ones,
- use a question for another purpose,
- use a question for other skills,
- use a question in another domain.

Reusable elements have to be identified and isolated during an iterative creation of an assessment. The amount of details according to ten key questions are increased with each iteration. The key questions are [Williamson et al., 2003]:

Key Questions

- Why are we assessing? (Purpose Definition)
- What will be said, done, or predicted on the basis of the assessment results? (Claims and Prospective Score Report)
- What portions of a field of study or practice does the assessment serve? (Domain Analysis)
- Which knowledge and proficiencies are relevant to the field of study or practice? (Domain Model)
- Which knowledge or proficiencies will be assessed? (Student Model)
- What behaviors would indicate levels of proficiency? (Evidence Models)
- How can assessment tasks be contrived to elicit behavior that discriminates among levels of knowledge and proficiency? (Task Models)
- How will the assessment be conducted and at what point will sufficient evidence be obtained? (Assembly Models)
- What will the assessment look like? (Presentation Models)
- How will the assessment be implemented? (Delivery System Model)
The objective is a mapping to the Conceptual Assessment Framework (CAF), which defines several models for characterization of an assessment. These are student model, task model, evidence model, assembly model, presentation model, and delivery model (see figure 2.11). Each model contains isolated properties, which can be reused. They are defined during the iterative creation related to the key questions.

*Figure 2.11. Conceptual Assessment Framework [Williamson et al., 2003].*

The student model defines the assessed skills and competencies. It contains the definition of related indicators and scales. A task model is a template for the creation of explicit questions. A set of rules to define a mapping from results to the student model is contained in the evidence model. It is dependent on the student model and the task model. Composition or sequencing of questions are regulated by an assembly model. A presentation model is used to define interaction and design of an assessment. It covers technology enhanced approaches as well as pen and paper based ones. A separate delivery model covers organizational and technical aspects of publication. Reutilization can be reached by exchange of a model. For instance, different evidence models are applied to multiple contexts with students of different levels of prior knowledge.

**Four-Process Model**

The *Four Process Model* is a reference architecture for delivery of assessments. It tries to enable the combination of different tools, which are required for different scenarios. Therefore, it divides assessment into the four processes [Almond et al., 2002]:
2. Assessment

**Four Processes**

- Activity Selection,
- Presentation,
- Response Processing,
- Summary Scoring.

Figure 2.12 shows the relation of the four processes, related users, and a Task / Evidence Composite Library.

![Image of Figure 2.12](image)

**Figure 2.12.**

The four process model [Almond et al., 2002].

In the activity selection process, an administrator determines tasks to solve by participants. Task description data is taken from the Task / Evidence Composite Library. Instructions are sent to the presentation process which displays the presentation material to participants. They submit a work product which is passed to the response processing, which does a task-based evaluation and generates task level feedback. A final evaluation is done by the summary scoring process, which scoring record can be used for selection of the next activity.

The four process model allows a flexible combination of different tools for each of the four processes. This model is focused on individual test scenarios, especially based on objective questions. The modularity is limited to the four processes and reuse or extension of given components are not part of the model. The above mentioned IMS QTI specification is based on the four process model.

**Educational Model**

The Educational Model is a conceptual framework based on static model for a description of assessment [Joosten-ten Brinke et al., 2007]. The main objective is
an increasing interoperability of assessment tools beyond objective questions. The framework contains six static UML-diagrams, where each characterizes elements and relations of a subprocess. These six models are:

- **Assessment Design** (AD),
- **Item Construction** (IC),
- **Assessment Construction** (AC),
- **Assessment Run** (AR),
- **Response Rating** (RR),
- **Decision Making** (DM).

The assessment design covers the conceptual development of an assessment. A plan, target audiences as well as outcomes to measure are defined. The next step is the explicit construction of items (task / questions). Indicators for evaluation as well as feedback information are linked with an item. The composition of item to a unit of assessment is explained by the assessment construction model. A response to an item is created and submitted during an assessment run. The evaluation of a response by an assessor of arbitrary kind (tutor, peer, computer etc.) is modeled in the response rating. The last step is the decision making, which combines the evaluation of all items to an overall decision for a unit of assessment, e.g. passed or not passed.

**e-Framework Reference Model for Assessment (FREMA)**

FREMA is a “Reference Model for the Assessment Domain; a guide to what resources (standards, projects, people, organisations, software, services and use cases) exist for the domain” [Wills et al., 2009]. An overview about current dimensions of assessment is given using topic maps (see 2.13) as well as within a semantic wiki [15]. First of all, FREMA does have an implicit technological aspect. It tries to explain all currently known aspects of technology enhanced assessment, which have to be considered for creation of a new assessment tool. Nonetheless, a service-oriented architecture is suggested, which allows an easy replacement of different services. The modularity is limited to traditional forms of assessment in formal, institutional settings. The integration of emergent technologies or fine-grained extensions to existing functionalities are not covered by this reference model. Although this model does not describe explicit approaches for a technical realization of an assessment platform, it provides a collection of basic aspects which have to be considered for a technical realization.
2. Assessment

Figure 2.13. FREMA process-based topic map [Wills et al., 2009].

2.3. Conclusions

The question after this introduction to the domain of (technology enhanced) assessment is, how technology can be utilized to enhance or facilitate assessment in terms of life long learning. In the dimension of institutional learning, assessment is already well established [Müller and Schmidt, 2009]. Therefore, most of the above mentioned assessment approaches are designed for institutional settings. Resulting topics of research are, if and how established forms of assessment can be applied or transformed to emerging, informal, and non-institutional learning scenarios. With these novel forms of learning becoming more and more present in institutional learning as well, new forms of assessment have to be developed. It has to be investigated, how these parts of learning can be assessed. Especially, the combined assessment of formal and informal learning in an institutional context has to be studied. Development of completely new forms of assessment as well as transformation of existing forms are relevant, and the integration of both.
Open Assessment, as used in the thesis at hand, describes a term to cover several aspects of assessment related to all dimensions of lifelong learning (see figure 2.14), except learning on demand. The latter is included and represented by all other dimensions. The relation between the types of assessment and the types of learning is the following one:

- Institutional Assessment ⇔ Institutional Learning
- Network Assessment ⇔ Organizational/Network Learning
- Self-Assessment ⇔ Self-Directed Learning
- Assessment 2.0 ⇔ Informal Learning

Figure 2.14. Classification of open assessment in context of lifelong learning.

Approaches for supporting the management of institutional assessment technologically are already available to a large extent (cf. figure 2.4). Network learning, self-direct learning, and informal learning are facilitated by distributed services as well as personal learning environments. Characteristics of assessment or mostly not considered by these tools.
2. Assessment

With an increasing importance of lifelong learning as well as experiences and research results concerning the importance of assessment as a driver for learning success, the research of combined assessment support for all aspects of lifelong learning is very relevant. Complexity and diversity of the different approaches for learning as well as the hitherto contrary technological approaches (closed LMS vs. open PLE) makes a unified and integrated approach for technology enhanced assessment very difficult. Therefore, a process of approximation from two directions is a possible approach to create such a complex solution successively. Either the new dimensions of learning can be included in the management of institutional assessment processes or institutional assessment processes can be applied to non-institutional learning. New forms of assessment are expected to arise in both cases.

### Definition

**Open Assessment Management** describes the combined support of assessment processes containing aspects within several categories of open assessment. It covers the organizational aspects of assessment design and execution across several activities, methods, and approaches.

**Direction A: Institutional Perspective**

Include new dimensions of learning in institutional assessment processes.

**Direction B: Non-Institutional Perspective**

Building assessment processes for non-institutional learning.

### Objectives

*Direction A* has been chosen as the development approach to approximate an open assessment management platform in the thesis at hand. Thus, the following objectives are pursued:

- describe a model for open assessment management platforms,
- develop a platform for open assessment management,
- provide an integrated service,
- enable cross domain usage,
- cover heterogeneous approaches.

A big challenge in this project is the technological consolidation of various approaches to respect different activities, methods, and even methodologies for assessment. Due to an institutional perspective this is already challenging for higher education scenarios, as subsets of life long learning scenarios.

A first idea could be to utilize a common learning management system (LMS), what has been done by several universities. These kind of platforms are designed
2.3. Conclusions

for generic support of different scenarios. They mostly lack in flexibility and extensibility [16] to cope with advanced functionality, variable requirements in different courses, and emerging forms of assessment.

An alternative is the development of a new platform, not as a monolithic system, but as an extensible environment. The above mentioned reference models (see 2.2.7) are not directly applicable for procedure, but several of their process descriptions can be adopted. Available standards are problematically as well, due to their current limitations as well as to their static nature. Service-orientation could help to allow integration and interoperability anyway [Vossen and Westerkamp, 2008].

The basic functionality that should be provided by the intended platform is based on characteristics of 21st Century Assessment [17] (see figure 2.15). Thus, collaborative assessment, self- and peer review, transparent rubrics, and timely feedback should be applicable in combination.

Figure 2.15. Characteristics of 21st Century Assessment [17].
Part II.

Analysis
Chapter 3.

Assessment Tools

This chapter is intended to link the theoretical results of research with their technological realizations. Therefore a broad analysis of assessment tools has been made. The main objective was to prove them for usage as an open assessment management tool. Therefore, standard and especially specific advanced functionality of these tools have been investigated.

Since there was no schema for a categorization of assessment tools available, which addresses the broad range of open assessment management, a new schema has been developed based on literature review and investigation of state-of-the-art tools. Eleven categories (see figure 3.1) have been defined to allow a systematic analysis of assessment tools. Firstly, the types of activities which are supported by a tool as well as the extensibility for new types have been considered. Additionally, controlling mechanisms, e.g. adaptivity or publication timing, have been taken into account. The marking process — including available scales, management of rubrics, and definition of rules for decision making – is covered by the second category. Group assessment has been investigated regarding the collaboration process itself, the provide mechanisms for grouping, and related marking strategies. In the case that peer assessment is realized by a tool, implemented strategies, their flexibility (i.e. adjustability and extensibility) as well as possibilities for reutilization are observed. The fifth category is self-assessment. Automatic assessment emphasizes possibilities for general as well as domain-specific automation of correction. With this, flexibility for adjustment and parametrization of evaluation processes as well as extensibility for new evaluation approaches or other domain-knowledge have been investigated. Provided feedback is classified by dimension, timing, and personalization. The latter distinguishes static feedback and dynamic feedback that depends on a learner’s submission and his or her individual profile. Social media-based approaches, which utilize distributed services in a mostly open environment, are covered by the category Assessment 2.0. Tools are characterizes according to their supported process steps related to the six models within the educational model (cf. 2.2.7). This model has been chosen for a category, because it describes the assessment process...
for generic systems as well as for scenario-specific systems (cf. figure 2.4). Finally, a tool’s suitability for a hosting scenario has been estimated. This is done based on the possibility for integration into a centralized identity management system as well as with additional functions for learning or content management, which are not related to assessment. Adaptability for various different courses in parallel have been considered as well. The last category deals with interoperability via supported standards and specifications or via service-oriented approaches.

**Figure 3.1.** Categories for analysis of assessment tools.

In addition to the above mentioned categories, the analysis is divided by classic assessment system categories (see figure 2.4) and assessment 2.0 approaches. Advanced features of specific tools, which are not covered by the categories for analysis, are named explicitly.

### 3.1. e-Test Systems

Four types of electronic test-systems — authoring tools, test players, test suits as well as specialized systems — have been analyzed. Advantages and disadvantages of each tool type are presented based on some selected tools.
3.1. e-Test Systems

Authoring Tools

Several authoring tools are available which are focused on the item construction process, i.e. the definition of single questions including the type of interaction, possible answers, and static feedback.

**Hot Potatoes** [18] is an authoring software for questions of selected types that is composed of six modules. These modules allow the creation of multiple-choice questions (*JQuiz*), jumbled-sentences (*JMix*), crossword puzzles (*JCross*), matching/ordering questions (*JMatch*), fill-in-the blank questions (*JCloze*), and combinations of multiple questions (*The Masher*). Questions and tests can either be created as full web pages or exported as SCORM.

**Aqurate** [Campos et al., 2007] [19] is a JAVA-based open source desktop application. It currently supports seven item types. The items can be exported to QTI 2.1. The item banking in a large pool is supported by tools like **Minibix** [Lay et al., 2008, Campos et al., 2007]. In this case, item banks can be filled and items can be taken in a service oriented manner via REST-services. The composition of items to tests (assessment construction) can be done with **AsDel** [Wills et al., 2008, Campos et al., 2007].

Authoring tools for web based trainings (WBT), for instance **Adobe Captivate** [20], allow the inclusion of objective tests into tutorials and training solutions. Questions are directly integrated to the user interface. The resulting WBT can be exported as SCORM-package for execution in a test environment, e.g. within an LMS.

Test Player

**QTIEngine** [21] and its predecessor **R2Q2** [Wills et al., 2008] provide a service for execution of items or rather tests (assessment run). Questions are rendered as full html-pages. Students’ responses are send to the service to generate feedback and score (response rating). The integration of such a service into a learning environment can be realized by appropriate plug-ins to the corresponding system.

Test Suites

In contrast to components for each sub process of electronic test processing, test suites unify the whole process within one platform.

**TOIA** [22] (Technologies of Online Interoperable Assessment) is a fully web-based test suite. It is standalone and therefore has its own authentication process as well as role and rights management. Nine question types are supported. Interoperability is address by IMS QTI, IMS CP and LOM.
3. Assessment Tools

**Onyx** [Winkelmann, 2011] is a test suite which is design for integration scenarios with any LMS. It consists of an *editor* for question and test creation, a *player* for execution and evaluation of tests, a *reporter* for result statistics, a *converter* for import of LMS content and transformations from and to QTI 2.1, and finally a plugin module for LMS integration using a web service API.

**Questionmark Perception** [23] is a commercial tool to manage the entire assessment process of electronic tests including several advanced features. For instances, it provides blended delivery modes to browser, smartphones, paper etc., including printing and scanning test sheets. A role-based security mechanism is suitable for multi-author environments. Translation management facilitates multilingual assessments. Results, reports and item analysis are accessible on-demand. It allows randomized presentations of questions and choices as well as adaptive branching based on how questions are answered. Integration to LMS or enterprise systems (for further learning) is possible using industry standards (including QTI and SCORM), Perception’s web services API and using ”Connector” software available from Questionmark.

**Specialized Test Systems**

Test suits already provide a huge amount of features according to standard functionality of electronic test systems. However, there are several specialized systems available, which address advanced possibilities related to current research topics (cf. 2.2.1) as well as to pragmatic needs.

**SKA** is an interactive online tool to allow automatic generation of feedback to exercises in the domain of propositional logic and first-order logic [Schulz-Gerlach and Beierle, 2006]. For instances students have to transform a logic term into its presentation in conjunctive normal form (KNF). They can enter their solution directly into a web form, submit it or request a hint.

**SIETTE** [Conejo et al., 2004] is a test engine based on JAVA-Applets to realize adaptive test. ”Questions are selected intelligently to fit the student’s level of knowledge. In this way, we obtain more accurate estimations of student’s knowledge with significantly shorter tests”, [Conejo et al., 2004]. SIETTE allows the presentation of objective questions in form of multiple-choice as well as custom forms of interaction, which have to be developed within the applet.

**Dynexite** (DYNamic EXercises in an Internet Environment) [7] is a tailor-made system that has been developed at the School of Business and Economics at RWTH Aachen University as an automated testing solution for student exercises complementing their economics courses. The special structure of economic calculation models (e. g., accounting records) and their complex input requirements have been addressed by Dynexite. The main difference compared to standards systems is that
items are authored manually in PHP source code that utilizes a specific item API. This approach allows to implement custom logic, e.g. to realize randomized variables, to make the interaction as well as the evaluation more flexible.

**Individual Spreadsheets** [Blayney and Freeman, 2004] allow complex calculation-based questions, individualized tasks, and feedback on-demand. Each student is provided with a spreadsheet for calculation containing individualized values preset. Questions have to be answered by application of calculation rules. The main objective of this approach was to facilitate self assessment and provide more feedback. The feedback is generated on demand using scripts (VBA) that are included in the spreadsheet. The creation of such a task requires specific knowledge in script development. The process of delivery has to be handled additionally. A submission by students as well as corrections by tutors are not intended.

**Discussion**

An overview of the provided functionality of the selected e-test tools can be found in table 3.1. It shows that each tool focuses different aspects and provides different features. Because all of them are limited e-test activities, some categories (e.g. peer assessment) are neither suitable nor listed. Each tool that has influence on the item construction allow feedback possibilities, which are similar to QTI: static feedback per item, per test, timely or after test.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Process Steps</th>
<th>Interoperability</th>
<th>Hosting</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hot Potatoes</td>
<td>IC,(AC)</td>
<td>SCORM</td>
<td></td>
<td>5 item types</td>
</tr>
<tr>
<td>Aqurate</td>
<td>IC</td>
<td>QTI 2.1</td>
<td></td>
<td>7 item types</td>
</tr>
<tr>
<td>MiniBix</td>
<td></td>
<td>QTI 2.x</td>
<td>web services</td>
<td>item banking</td>
</tr>
<tr>
<td>AsDel</td>
<td>AC</td>
<td>QTI 2.x</td>
<td>web services</td>
<td></td>
</tr>
<tr>
<td>Adobe Captivate</td>
<td></td>
<td>SCORM</td>
<td></td>
<td>WBTs with tests</td>
</tr>
<tr>
<td>QTIEngine</td>
<td>AR,RR,DM</td>
<td>QTI 2.x</td>
<td>web services</td>
<td>LMS plugins</td>
</tr>
<tr>
<td>TOIA</td>
<td>IC,AC,AR,RR,DM</td>
<td>QTI,CP,LOM</td>
<td></td>
<td>9 item types</td>
</tr>
<tr>
<td>Onyx</td>
<td>IC,AC,AR,RR,DM</td>
<td>QTI 2.1</td>
<td>web services</td>
<td>LMS plugins</td>
</tr>
<tr>
<td>Questionmark Perception</td>
<td>IC,AC,AR,RR,DM</td>
<td>SCORM,QTI</td>
<td>web services</td>
<td>LMS connectors</td>
</tr>
<tr>
<td>SKA</td>
<td>IC,AC,AR,RR</td>
<td></td>
<td></td>
<td>blended delivery</td>
</tr>
<tr>
<td>SIETTE</td>
<td>IC,AC,AR,RR</td>
<td></td>
<td></td>
<td>statistics</td>
</tr>
<tr>
<td>Dynexite</td>
<td>AC,AR,RR,DM</td>
<td></td>
<td></td>
<td>logical formulas</td>
</tr>
<tr>
<td>Individual Spreadsheets</td>
<td>IC,AC</td>
<td></td>
<td></td>
<td>feedback on demand</td>
</tr>
</tbody>
</table>

**Table 3.1.** Functionalities of analyzed e-test systems.
A conclusion that can be drawn as a result of this analysis is the fact that there is no e-test system available which allows all kinds of test scenarios or support features. Nevertheless, a lot of complex processes are realized by these tools. Therefore, a combination of these tools has to be considered for an integrated support of e-test functionality.

### 3.2. Learning Management Systems (LMS)

As mentioned in section 2.2.2, multiple different LMS are available, which provide almost the same core functionality. Thus, a consolidation of those standard platforms can be observed in during the last decade (cf. figure 3.2). From an international perspective, as shown in the figure, there are two big players left. *Blackboard* as a commercial offer, and *Moodle* as an open source alternative. Additionally, learning platforms in use at German universities are *ILIAS* [24], *Stud.IP* [25], *OLAT* [26] and others. Furthermore, some research projects, e.g. iPAL [Pinkwart et al., 2005] or EduComponents [Amelung and Rösner, 2008], try to enhance common content management systems (CMS) or portals to be suitable as a learning platform. One representative for each mentioned category (open source, commercial, German university, CMS extension) is described in the following.

**Moodle** (*Modular Object-Oriented Dynamic Learning Environment*) [8] is an open source platform with a huge amount if installations\(^1\). It provides standard functions for course management, a complex role and right management system as well as various features for communication and collaboration [Cole and Foster, 2005].

Assessment concerning modules are [Cole and Foster, 2005]: e-tests, assignment management, and grading. The e-test module provides typical functionality of an e-test suite with QTI support as mentioned above. Assignments can be published at a specified publication date. Submissions are possible by individual users until a given deadline has been reached. Possible types of submissions to an assignment are either **offline activity** (no submission), **online text**, **single file submission**, or **multiple file submission**. A combination of submission types for a single assignment is not possible. The grading engine provides several numeric, non-numeric, and custom scale types, which are used to define marking rubrics for assignments. Combination of results are possible within gradebook module, which provides several advanced calculation methods (e.g. weighting) and personalized views for students.

Furthermore, Moodle provides a plug-in architecture which facilitates developers to implement modular extensions for Moodle. The development base is PHP and

\(^1\) As of November 2011 Moodle was used for 5,876,819 course at 72,162 registered websites in 223 countries [28].
3.2. Learning Management Systems (LMS)

Concerning assessment, plug-ins for new question types for e-tests are possible. Custom developed assessment tools can be deployed as closed and isolated components.

In contrast to the various set of possibilities, main disadvantages of Moodle are its complexity as well as missing possibilities for structured extension and combination of assessment tools. For instance, users prefer the usage of Moodle for online assignment submission against emergent collaborative online tools like Writely (now part of Google Docs) [Petrus and Sankey, 2007]. There are no default mechanisms available to extend the default assignment submission component with new functionality. Instead a new component has to be developed that realizes old as well as new functionality.

BlackBoard [9] is a commercial course and learning management system. As well as Moodle, it provides various components for content delivery, communication and collaboration with in a modular server platform. Supported assessment activities are electronic tests and assignments with individual submission and manual cor-
3. Assessment Tools

The detection of plagiarism is provided via SafeAssign [Blackboard, 2009] as advanced feature for assignment management. A transparent overview of all students’ results, a personal overview respectively, is provided by a flexible gradebook module. All in all, assessment functionality of Blackboard has no big difference to Moodle’s capabilities.

**OLAT** (*Online Learning And Training*) [26] is an award-winning learning management system. It has been developed at *University of Zürich* and is used as a central platform at several German Universities, for instances provided by *Bildungsportal Sachsen* for all Universities in Saxony[^3]. Its assessment support is quite similar to Moodle and Blackboard.

**iPAL** [Pinkwart et al., 2005] extends the standard functionalities of the *PostNuke* [29] CMS with modules for building tutorials, as well as handling assignments and lecture notes. The assignment management tool includes publication and deadline timing, only submission of single files, and manual feedback definition. Additionally, a mechanism for partition of students to tutorials, which are each supervised by assigned tutors, is possible. Tutors get a notification when a student from his or her tutorial submitted a solution for correction.

**Discussion**

The descriptions of selected LMS should have emphasized, that LMS indeed contain good support for standardized objective testing, but provide no or only basic support for assignment management, group assessment, peer assessment, self-assessment, and assessment 2.0. Automatic assessment only applies to electronic tests. Activities are limited to objective tests and assignments. In contrast to the assessment support of standard LMS, which are closed system that provide basic or rather generic support by a fixed set of tools, a platform for open assessment management has to be open for the integration and combination of different specialized tools to provide advanced support for various scenarios.

An overview of the selected systems is presented in table 3.2. Feedback options are only listed for assignment management, because all platforms that contains e-tests, provide static feedback as defined in QTI questions. According to the intention of being a single platform for all learning management tasks, mainly all process steps for assessment are covered somehow. Except iPAL, all platforms are already used in hosting scenarios at/for multiple universities.

[^2]: OLAT has won MedidaPrix 2000, Swiss Open Source Award 2008, and IMS Learning Impact ‘Leadership Award’ 2009

[^3]: In Saxony OLAT is called OPAL.
3.3. Process-specific Systems

While LMS only provide basic support of assessment processes for common usage, other approaches have to be chosen to support context specific processes on more depth. On the one hand generic tools can be utilized for a specific sub-process. For instance, E-Mail is used to handle assignment submission and individual feedback delivery [Huett, 2004]. On the other hand several systems for detailed support of specific organizational tasks concerning assessment have been developed. A selected set of tools with different focuses (e.g. assignment sheet creation or peer assessment) have been analyzed and are described in the following.

### 3.3.1. Assignment Creation

Typically, lecturers of scientific and technical courses typically provide weekly assignments on paper sheets to their students (cf. section 2.1.6). Because the authoring process of these sheets itself requires a lot of effort, some tools try to reduce the effort with use of technology.

**xGen (Exercise Sheet Generator)** is a web-based tool for the creation of assignment sheets, which utilizes the typesetting system LaTeX as a back-end [Wimmer et al., 2006]. Exercises are defined by LaTeX-source and are stored in a pool. An assignment sheet can be composed of existing exercises together with some metadata (e.g. submission deadline or maximal score per exercise) (see figure 3.3). A PDF-File of the sheet is generated directly online. Advantages of the system are the central storage of exercises, their re-usability for several assignment sheets, as well as well typed output files. Disadvantages are that users need to be familiar with LaTeX typesetting. Additionally, it only covers one small part of the overall assessment process, such that delivery and submission mechanisms are not included.
3. Assessment Tools

**Figure 3.3.** The page for the preparation of a new exercise sheet in xGen [Wimmer et al., 2006].
3.3.2. Assignment Management

In addition to basic assignment management support within learning management systems, there are several tools available which are specialized on technological support of assignment management processes.

**CVS** *(concurrent versioning system)* is a tool to keep track of updates and changes to files and documents, especially in distributed teams. It is mainly used for handling source code in software development projects, but is generally allows arbitrary file types. Experiences in courses with more than 200 students show, that CVS is suitable to let students develop and submit their software development assignments [Piekarski, 2005]. Although it is transferable to other domains, its usage seems to be difficult for non technical people.

**FAsT** *(Flexible Assignment System)* [Topcuoglu, 2006] tries to provide a flexible support for different assessment scenarios based on different possible collaboration scripts. Students can be assessed individually or in groups and they can use an integrated workspace to create their solutions collaboratively. The process how their collaboration is managed and how they are assessed, by a tutor or by peers, is adjustable according to different collaboration scripts. “Collaboration is structured by defining tasks, deadlines, roles, how to form groups, etc.” [Topcuoglu, 2006]. Therefore, a script defines an assessment process in several phases. An example for a peer assessment process in eight phases can be found in figure 3.4. In this case the phases are conceptual formulation, grouping, editing exercises, assign marking groups, assessing submissions, inspection, backward-feedback, and revision. This is a very structured and flexible approach for organizational issues, but limited to integrated mechanisms like the grouping tool, that allows only grouping by tutors. FAsT is a closed and monolithic platform which does not allow integration of or into other platforms.

3.3.3. Peer Assessment

Even though systems like FAsT already facilitate peer assessment as one possible setting, there are various tools which are specialized to peer assessment. PeerPigeon [Millard et al., 2008] as well as Aropā [Hamer et al., 2007] are explained in more detail. Other tools of that kind are referenced in [Millard et al., 2008].

**PeerPigeon** [Millard et al., 2008] PeerPigeon is a web-based application which contains a peer review engine to interpret peer patterns. These patterns are based on peer review cycles, which are each a simple combination of three actions: generate,
submit, and distribute. Cycles are connected by peer review transforms “that dictate how documents move between peers within each stage” [Millard et al., 2008]. Patterns can be defined with a domain specific language (DSL). An example of a pattern which defines a review process using the simple peer assessment strategy (cf. section 2.1.4) is displayed in figure 3.5.

Based on the specified pattern, a review process is generated. Metadata for a such a review process, including participants and roles as well as schedule plans, can be adjusted via a web form (see figure 3.6).

Although this approach allows flexible adjustments to support multiple scenarios, security issues as well as usability issues have to be solved before the system can be used in a broad productive setting.

“Giving users the ability to add new plans is problematic, as allowing any user to insert arbitrary script into the system would represent a serious security hazard, and in any case the DSL is non-trivial to write and should be tested properly before being deployed. [...] a solution would be to create a graphical authoring front end to the DSL system” [Millard et al., 2008].
3.3. Process-specific Systems

Figure 3.5. DSL for a simple review pattern in PeerPigeon [Millard et al., 2008]

Aropä [Hamer et al., 2007] is a web-based peer assessment tool which can be adjusted according to a set of properties. For instance, it can be configured, if reviewers are grouped or if the review is done anonymously. The feedback is provided by use of formalized rubrics (see figure 3.7), which does not require students to type free text and provide a base for more objective review.

3.3.4. Marking

Assigning scores or grades to students submissions, or performances in general, is a task that is required, if the performances are involved in deciding about certification. Furthermore, marks are a special kind of feedback that is used for motivation and other purposes of formative assessment. Technological support in this context is available for the marking process itself as well as for the bookkeeping of students result and decision making concerning multiple activities. The latter is often managed by use of common spreadsheet software (e.g. Microsoft Excel, OpenOffice.org Calc, or Numbers). Programs of this kind are very powerful because of their large pool of different formulas which can be used to calculate decisions. Other advantages are the easy data modification possibilities through the client software. Disadvantages are the rising complexity of formulas as well as missing possibilities for delivery of personalized results to students.
3. Assessment Tools

Figure 3.6. PeerPigeon edit review page [Millard et al., 2008]
### Figure 3.7. A formalized grading rubric in Aropä [Hamer et al., 2007].

<table>
<thead>
<tr>
<th>Marking for allocation #1 for assignment “Assignment”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTRACT (25 Marks)</strong></td>
</tr>
<tr>
<td>✓ The abstract is over 250 words</td>
</tr>
<tr>
<td>✓ The abstract is less than or equal to 250 words</td>
</tr>
<tr>
<td><strong>WORD COUNT (5)</strong></td>
</tr>
<tr>
<td>✓ No structure present</td>
</tr>
<tr>
<td>✓ Some structure apparent but largely gobbled and/or illogical</td>
</tr>
<tr>
<td>✓ Structure apparent but may not be entirely appropriate</td>
</tr>
<tr>
<td>✓ Clear, logical structure</td>
</tr>
<tr>
<td><strong>STRUCTURE (7)</strong></td>
</tr>
<tr>
<td>✓ Large amounts of irrelevant information. Little or no mention of key points.</td>
</tr>
<tr>
<td>✓ Important aspects ignored or receive minimal attention. Some irrelevant information.</td>
</tr>
<tr>
<td>✓ Appropriate weighting provided for most aspects of the article. Purpose and outcomes of study explained.</td>
</tr>
<tr>
<td>✓ Appropriate weighting provided for all aspects of the article. Purpose and outcomes of study clearly explained.</td>
</tr>
</tbody>
</table>

---

3.3. Process-specific Systems
3. Assessment Tools

**ALOHA** [Ahoniemi and Reinikainen, 2006, Ahoniemi et al., 2008] aims to increase consistency and objectivity of manual correction processes. This is especially a problem in large courses with multiple assessors. The use of marking rubrics is utilized to reduce this issue. A tutor has to grade each subcategory within the rubric as displayed on the left-hand side in figure 3.8. The scales for grading are not limited to scores, but can be configured by the teacher. Based on configurations of the assignment creator and the tutors grades for each subcategory, an overall result is suggested by the system. Feedback can be given to learners for each subcategory as well. It is distinguished by positive, negative, and neutral. A technique called “semi-automatic phrasing” [Ahoniemi et al., 2008] allows to reuse and afterwards personalize often used feedback phrases.

![Figure 3.8. Screenshot of the grading view in ALOHA [Ahoniemi et al., 2008].](image)

**eduViz** [Friedler et al., 2008] is a specialized gradebook software, which provides advanced statistical functionalities to teachers (see figure 3.9). It allows definition of weighted categories to automatically compute overall grades and create grade reports. Final grade levels can be dynamically defined according to results of other activities, which are clustered by category. A single student’s performances as well as the average result for specific tasks can be explored. Web-based tools for grade booking, e.g. Gradekeeper [30] and others, additionally provide access for pupils and their parents to get a personalized view of their own results. But gradebook
software is optimized for school scenarios and is often not applicable to special needs of German universities. Limitations are mostly restrictions concerning available rating scales and possibilities for advanced calculations.

![Grading and exploration panel of EduViz](Friedler-et-al-2008)

**Figure 3.9.** Grading and exploration panel of EduViz [Friedler et al., 2008].

### 3.3.5. Discussion

Various tools have been presented in this section, which each provides advanced support for very specific organizational aspects within a whole assessment process. Therefore, continuous and integrated support of whole process life cycles in different scenarios is not possible with a single of these tools. Each single approach or functionality for specific advanced assessment support has to be considered for an open assessment management platform. The integration and combination of them are major challenges.

An overview of the selected systems is presented in table 3.3. Most of them are standalone systems which do not provide any possibility for interoperability or integration. A classification of these systems is quite difficult, because their functionalities are overlapping. Therefore, tools which support group assessment for example, can be found in the next section 3.4 as well, because they provide domain-specific features as well.
### Table 3.3: Functionalities of analyzed process-specific systems

<table>
<thead>
<tr>
<th>Tool</th>
<th>Activities</th>
<th>Marking</th>
<th>Group Assessment</th>
<th>Peer Assessment</th>
<th>Feedback</th>
<th>Process Steps</th>
<th>Wiki</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>xGen</td>
<td>assignments</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>IC,AC</td>
</tr>
<tr>
<td>FAsT</td>
<td>assignments</td>
<td>rubrics</td>
<td>tutor assigned</td>
<td>custom strategies</td>
<td>simple marking, collaborative scripts</td>
<td>–</td>
<td>IC,AC,AR,RR,DM</td>
<td>–</td>
</tr>
<tr>
<td>PeerPigeon</td>
<td>assignments</td>
<td>custom strategies</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>IC,AC,AR</td>
</tr>
<tr>
<td>Aropä</td>
<td>assignments</td>
<td>rubrics</td>
<td>assigned groups</td>
<td>static</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>IC,AC,AR,RR,DM</td>
</tr>
<tr>
<td>ALOHA</td>
<td>assignments</td>
<td>manual</td>
<td>assessed scores</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>IC,AC,AR,RR,DM</td>
</tr>
<tr>
<td>CSV</td>
<td>individual</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>FAsT</td>
<td>individual</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PeerPigeon</td>
<td>individual</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ALOHA</td>
<td>individual</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ALOHA</td>
<td>individual</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CSV</td>
<td>manual</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

76
3.4. Domain-specific Systems

In addition to generic as well as specific support of organizational process for assignment management, several systems include automatic correction of applicative questions. Because this can not be realized in general, each of these tools is focused on a domain for which specific knowledge is available. An overview of possible functionalities and domains are given in the following.

3.4.1. Source-Code

Especially in the domain of programming, there exist a huge amount of tools which try increase quantity, quality, and delivery speed of feedback by automatic (pre-)correction of source code. Brief descriptions of selected tools are presented in alphabetical order.

**Agar** [Winters et al., 2006] is a desktop application that provides automatic compilation and testing of source code in C/C++. A fixed set of test steps is available for combination to build a marking rubric (see figure 3.10). Student submissions are collected and each put to a folder on the file system. They are automatically corrected, such that the marking rubric is filled and feedback is generated. Manual feedback can be defined afterwards. Results of the correction process are send to the students via e-mail.

**ASAP** [Douce et al., 2005] follows a service-oriented approach with the major aim to get usable from within other platforms (see 3.11). A tool called *Automatic Java Marker* (AJM) encapsulates the correction of Java source code to provide it via web services. Functionalities of other tools, e.g. a gradebook, can be provided the same way. The integration of these services are realized by a custom developed portlet (cf. 7.1), which can be integrated into a LMS or other platform. The correction process itself follows an test-driven approach, i.e. unit tests have to be developed and attached to each assignment by a teacher. The creation of tests is supported by templates.

**ASSYST** (*A*ssessment *S*YSTM) [Jackson and Usher, 1997] is a desktop application with a graphical user interface, which allows the automatic correction of source code in Ada and C. Firstly, a basic block analysis is done. With conjunction with an output specification, tutor’s test data, and student’s test data, five evaluation steps are executed in a fixed sequence. These steps are evaluations of *correctness, efficiency, style, complexity*, and *test data adequacy*. Results of all steps are collected in a final report.
3. Assessment Tools

Figure 3.10. Creating a rubric related to automatic evaluation in Agar [Winters et al., 2006].

Figure 3.11. System architecture of ASAP [Douce et al., 2005].
3.4. Domain-specific Systems

**ASB** *(Automatische Software Bewertung)* [Morth et al., 2007] is a web-based application for handling standard assignment management capabilities with additional features for multi-staged peer assessment and automatic correction of Java source code. The correction process is realized in a fixed order of steps, which are executed by use of other programs. For instances, *FindBugs* is used to find common mistakes and *Abbot* is used to allow unit testing of applications with GUI.

**AT(x)** *(Analyze & Test for language X)* [Beierle et al., 2003] extends the assignment management system *WebAssign* [Six et al., 2001, Brunsmann et al., 1999] with correction processes for arbitrary programming languages. Students submissions are automatically corrected with use of black box tests. Feedback is generated and send back to the authors. Students are allowed to rework their solutions based on feedback and resubmit the new versions again. Automatically generated results are used as pre-corrections for a final correction by tutors.

**DUESIE** *(Das UEbungSystem der Informatik Einführung)* [Hoffmann et al., 2008] is a web application based on PHP that provides assignment management functions for individual submissions in combination with automatic correction of source code in Java, SML as well as UML class diagrams. Static and dynamic tests of Java and SML programs are facilitated by integration of specific tools (see 3.12). Simple multiple-choice questions are handled as well.

![Figure 3.12. Architecture of DUESIE [Hoffmann et al., 2008].](image)

**EClaus** *(Electronic Correction of onLine Assignments at the University)* [Behringer et al., 2004] allows the management of four different types of activities: *essay type assignments* (online text or document), *multiple-choice questions*, *programming assignments* and *votings*. The latter type is used to organize the presentation of solutions in a lecture. Furthermore, EClaus allows self-directed grouping of students for each assignment.
3. Assessment Tools

**Praktomat** allows automatic correction of programs written in **Java**, **C++**, and **Haskell** [Zeller, 2000, Zeller, 1999, Krinke et al., 2002, Eichelberger et al., 2003]. The source code is compiled to a program, on which a sequence of functional test steps is executed afterwards. A step can be either mandatory or optional. A submission is only accepted if all mandatory tests are finished successfully. Reworked versions of accepted solutions can be resubmitted as well, e.g. to pass more optional tests. Additionally, optional tests can be either public or hidden. Hidden test steps are used to ensure, that students do not only develop against known test cases. Style checks and plagiarism checks are possible as well. Another approach to avoid plagiarism is implemented with a macro mechanism to create exercise variations.

**Other** tools for automatic correction of source code have been investigated as well:

- **AUTOMATION** [Linden et al., 2008],
- **BOSS** [Heng et al., 2005, Joy et al., 2000, Joy et al., 2005],
- **CourseMarker/CourseMaster** [Higgins et al., 2003, Higgins et al., 2002],
- **GUI_Grader** [Feng and McAllister, 2006],
- **Jack** [Striewe and Goedicke, 2009]
- **JOSH** [Bieg and Diehl, 2003],
- **Kassandra** [Von Matt, 1994],
- **LlsChecker** [Rösner et al., 2005],
- **Ludwig** [Shaffer, 2005],
- **Marmoset** [Spacco et al., 2006],
- **OTO** [Tremblay et al., 2005],
- **OCETJ** [Tremblay and Labonté, 2003],
- **PASS** [Choy et al., 2007, Yu et al., 2006],
- **PSGE** [Jones, 2000],
- **Scheme-Robo** [Saikkonen et al., 2001],
- **Submit!** [Pisan et al., 2003],
- **TRY** [Reek, 1989],
- **WBGP** [Juedes, 2005].

Each of these systems have similar functionality. They only differ by correctable languages, set and sequence of applied test steps, additional assessment methods (e.g. group assessment), and especially by some small settings (e.g. feedback timing, re-submission). General handling of automatic correction processes, e.g. queuing new
correction requests or sequencing evaluation steps, is almost the same, but reimple-
mplemented in each system. They are standalone systems, which are neither integrable
to other infrastructure systems nor provide possibilities for modular extensions.

3.4.2. Other Domains

Semi-automatic correction of applicative questions is not only possible for program-
ming, but for nearly all domains which deal with well formalizable solutions.

TRAKLA 2 [Laakso et al., 2005], as well as its predecessor TRAKLA
[Korhonen and Malmi, 2000], is a “visual algorithm simulation exercise systems”
[Laakso et al., 2004]. Students have to solve personalized, algorithmic exercises that
are implemented as Java Applets. After a task, for instance processing a sorting
algorithm manually, has been done, the process is automatically corrected at server
side and immediate feedback is generated.

E-Rater [Attali and Burstein, 2006], Criterion [Burstein et al., 2003,
Burstein et al., 2004], IEA [Foltz et al., 1999], and Apex
[Lemaire and Dessus, 2001] facilitate semi-automatic evaluation of essay type
questions. Realized are mainly detection of misspellings and grammar mistakes as
well as analysis of text structure, word lengths, or used terminologies.

XLX (eXtreme eLearning eXperience) has been designed to support assignments in
course on database development [Vossen et al., 2001, Vossen and Westerkamp, 2004,
Schwieren et al., 2006]. Therefore it is specialized to correct query and transforma-
tion languages: SQL, XQuery, and XSLT.

Music dictations are another domain in which automatic correction can enhance
the assessment process [Tremblay and Champagne, 2002]. By use of an editor which
stores a music dictation in MusicXML, a student’s solution can be compared with the
original to find mistakes (see figure 3.13). Furthermore, the edit distance algorithm
is used to find typical mistakes and provide specific feedback on that.
3. Assessment Tools

![Architecture of musical dictation marking tool](image)

**Figure 3.13.** Architecture of musical dictation marking tool [Tremblay and Champagne, 2002].

### 3.4.3. Extensible Systems

**EduComponents** is a set of modules to enhance the content management system (CMS) **Plone** [31] with additional functionality to support different learning and assessment activities [Amelung and Rösner, 2008, Amelung et al., 2007, Amelung et al., 2006]. The modules are **EC Lecture**, **EC Quiz**, **EC Assignment Box**, **EC Review Marking**, and **EC Auto Assignment Box**. **EC Lecture** integrates general course information and allows the management of course specific components. **EC Quiz** is a module for handling multiple-choice tests. **EC Assignment Box** is an assignment management module, which collects students’ submissions and facilitates tutor marking. **EC Review Marking** adds peer review processes to the assignment management. **EC Auto Assignment Box** is another add-on to **EC Assignment Box**. It realizes an extensible, service-oriented environment for automatic marking processes (see figure 3.14). The domain-specific logic for the correction of submissions is encapsulated in different **backends**. Each backend represent a specific correction process, for instances one backend for Java source code correction and another one for evaluation of UML diagrams. These backends are connected to the **EC Auto Assignment Box** via **ECSpooler**. This middleware is called via XML-RPC methods, to register new backends, ask for available backends and, first of all, queues and transmits submission between the platform and the backend. Restructuring an available process, for example by adding a style check and removing a black box test, requires the development of a whole new backend. Reusing single evaluation steps for new backends is not possible directly.
3.4. Domain-specific Systems

**EASy** provides a flexible approach concerning domain-specific assessment support [Gruttmann, 2010, Böhm, 2008, Eilers et al., 2008, Gruttmann et al., 2008a, Gruttmann et al., 2008b]. Started as a system for the definition, submission and (semi-)automatic correction of mathematical proofs, it has been extended to a platform for hosting modules if different domains. The assessment process is divided into **item construction**, **response construction**, and **correction and marking** (see figure 3.15). A newly created module has to provide an editor for each of these steps. Modules for **programming assignments**, **mathematical proofs**, **verification proofs**, and **multiple-choice tests** are already available. Assignments can be solved individually or within learner groups. Peer assessment, advanced grading methods, and other forms of assessment are neither available nor modularly integrable. The integration of collaborative activities or open assessment is not covered either. Furthermore, **EASy** is a standalone system, which implements its own user management. Hence, it is not directly integrated into an e-learning system to support other purposes than assessment as well.

### 3.4.4. Discussion

A lot of systems, which support assessment process very precisely by utilizing specific domain knowledge have been introduced in this section. An overview is listed in table 3.4 and table 3.5. It can be noticed that all of the presented systems are very similar according to the underlying process. Because of this, systems like **EduComponents** and **EASy** try to provide a flexible and extensible platform, which can be extended for arbitrary domains.
Support of domain-specific processes is important to provide real advanced enhancement in context of open assessment management. Similar to the analyzed extensible platforms, the support has to be domain-specific on the one hand, but a platform for open assessment management must be applicable across various domains on the other hand. This combination of both contrary requirements as well as the integration with other aspects (e.g. different process-specific approaches) is a major challenge for the system architecture as well as for the underlying model.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Group Assessment</th>
<th>Peer Assessment</th>
<th>Automatic Assessment</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agar</td>
<td>-</td>
<td>-</td>
<td>C/C++</td>
<td>Test step combination</td>
</tr>
<tr>
<td>ASAP</td>
<td>-</td>
<td>-</td>
<td>Java</td>
<td>fixed process, dynamic tests</td>
</tr>
<tr>
<td>ASSYST</td>
<td>-</td>
<td>-</td>
<td>C, Ada</td>
<td></td>
</tr>
<tr>
<td>ASB</td>
<td>-</td>
<td>multi-staged</td>
<td>Java</td>
<td>fixed process</td>
</tr>
<tr>
<td>AT(x)</td>
<td>-</td>
<td>-</td>
<td>Java</td>
<td>arbitrary programming languages</td>
</tr>
<tr>
<td>DUESIE</td>
<td>-</td>
<td>-</td>
<td>Java, SML, UML</td>
<td></td>
</tr>
<tr>
<td>EClause</td>
<td>self-directed grouping</td>
<td>simple marking</td>
<td>-</td>
<td>Java</td>
</tr>
<tr>
<td>Praktomat</td>
<td>-</td>
<td>-</td>
<td>Java, C++, Haskell</td>
<td></td>
</tr>
<tr>
<td>TRAKLA (2)</td>
<td>-</td>
<td>-</td>
<td>algorithms, data structures</td>
<td></td>
</tr>
<tr>
<td>E-Rater</td>
<td>-</td>
<td>-</td>
<td>Essays</td>
<td></td>
</tr>
<tr>
<td>XLX</td>
<td>-</td>
<td>-</td>
<td>SQL, XQuery, XSLT</td>
<td></td>
</tr>
<tr>
<td>Music Dictations</td>
<td>-</td>
<td>-</td>
<td>MusicXML</td>
<td></td>
</tr>
<tr>
<td>EduComponents</td>
<td>tutorials</td>
<td>separately available multiple, extensible</td>
<td>multiple, extensible</td>
<td></td>
</tr>
<tr>
<td>EASy</td>
<td>assigned groups, tutorials</td>
<td>-</td>
<td>-</td>
<td>multiple, extensible</td>
</tr>
</tbody>
</table>

Table 3.4. Functionalities of analyzed domain-specific systems (part 1).
3.5. Assessment 2.0

<table>
<thead>
<tr>
<th>Tool</th>
<th>Feedback</th>
<th>Process Steps</th>
<th>Interoperability</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agar</td>
<td>automatic, manual</td>
<td>(AR),RR,DM</td>
<td>automatic results assigned to rubric</td>
<td></td>
</tr>
<tr>
<td>ASAP</td>
<td>automatic</td>
<td>RR</td>
<td>web services QTI, IMS CP</td>
<td>SOA, testing</td>
</tr>
<tr>
<td>ASSYST</td>
<td>automatic</td>
<td>RR</td>
<td>--</td>
<td>external programs</td>
</tr>
<tr>
<td>ASB</td>
<td>manual</td>
<td>--</td>
<td>GUI tests</td>
<td></td>
</tr>
<tr>
<td>AT(x)</td>
<td>timely</td>
<td>RR</td>
<td>Plug-in for WebAssign</td>
<td></td>
</tr>
<tr>
<td>DUESIE</td>
<td>automatic</td>
<td>IC,AC,AR,RR,DM</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>E Claus</td>
<td>automatic</td>
<td>IC,AC,AR,RR,DM</td>
<td>--</td>
<td>Voting</td>
</tr>
<tr>
<td>Praktomat</td>
<td>timely, sample solutions</td>
<td>IC,AC,AR,RR,DM</td>
<td>--</td>
<td>Self-directed grouping</td>
</tr>
<tr>
<td>TRAKLA (2)</td>
<td>timely</td>
<td>IC,AC,AR,RR,DDN</td>
<td>--</td>
<td>optional/mandatory</td>
</tr>
<tr>
<td>E-Rater</td>
<td>timely</td>
<td>RR</td>
<td>test steps, public/hidden test steps, variations</td>
<td></td>
</tr>
<tr>
<td>XLX</td>
<td>timely</td>
<td>IC,AC,AR,RR</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Music Dictations</td>
<td>timely</td>
<td>IC,AC,AR,RR</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>EduComponents</td>
<td>timely, automatic</td>
<td>IC,AC,AR,RR</td>
<td>QTI, XML-RPC</td>
<td>SOA for backends</td>
</tr>
<tr>
<td>EASY</td>
<td>timely, automatic</td>
<td>IC,AC,AR,RR</td>
<td>--</td>
<td>module approach</td>
</tr>
</tbody>
</table>

Table 3.5. Functionalities of analyzed domain-specific systems (part 2).

3.5. Assessment 2.0

There are various web 2.0 applications available on the internet, which can be utilized for informal learning. Sharing platforms, for instances YouTube for Videos and SlideShare for Slides, allow their users to share their content and get feedback from people all around the world. Thus, availability of collaborative ratings and comments facilitate assessment 2.0 in these informal learning situations.

Usage of web 2.0 tools has been adopted to formal learning scenarios as well, such as using wikis for collaborative creation of learning artifacts [Panke and Thillosen, 2008, Parker et al., 2007, Mindel and Verma, 2006]. Additional questions arise, when students’ contributions created with these tools have to be assessed. It follows an explanation of problems and approaches by the example of assessing students’ contributions to wiki pages. Details about assessing contributions to blogs can be found in [Lee and Allen, 2006] as well as for online discussions in [Vonderwell et al., 2007]. The integration of electronic tests to the virtual learning environment Second Life with the tool quizHUD is described in [Bloomfield and Livingstone, 2009].
Wiki systems are well known web applications in the Web 2.0. The contents of wiki systems consist of many interlinked wiki articles. These articles are web pages. A wiki web application is easy to use and allows its users to edit, update, and organize contents collaboratively through a web browser [Duffy and Bruns, 2006, Bruns and Humphreys, 2005]. Wikis have a versioning system that allows every content modification that is made by a user to be automatically recorded as a new version of the content itself. This functionality of wikis makes it possible for an author to access an older version of the wiki content, edit and save it as the new version of the content.

Many scenarios based on wiki-usage in education are possible, such as a documenta-
tion platform of research projects, for mind mapping, distance learning, and also as a medium for knowledge transfer [Parker et al., 2007, Duffy and Bruns, 2006]. With wikis students can easily manage their research evaluations, results and documenta-
tions of their work in one medium. Analogously to the well known Wikipedia [32], wiki pages can also be used to create a specific encyclopedia in the learning context [Bruns and Humphreys, 2005]. Students edit and manage the encyclopedia and teachers assess, re-edit and publish it at the end of the semester. In some sce-
narios, the use of wiki systems as a collaborative tool is even more efficient than working collaboratively without a wiki system [Mindel and Verma, 2006].

The problems of assessing student activities like those mentioned above have been recognized by [Cubric, 2007]. She calls it “working collaboratively but assess individu-
ally”. To obtain the quality of each student’s contribution, a teacher has to navigate through many pages. This process is time-consuming and error prone. However, a lot of information needed to do the assessment of the learning activity is already stored within the wiki system. The problem is that this information is not clearly laid out for the assessment process. Other problems may occur as well. How should the marking be processed? How about group assessment? And how can peer marking be supported?

There are several suggestions on how to do and manage assessment of collabora-
tive work in wiki systems. One of them is automatic calculation. [Hoisl, 2007, Hoisl et al., 2007] introduce an algorithm to calculate a peer marking method within a wiki system (see figure 3.16). The algorithm is divided into two main parts. The first part is to calculate the score for each revision (version) of a wiki article based on the amount of references to this article and the amount of views and rating of the articles. Secondly, the algorithm allocates the score of each revision to the au-
thor based on the actuality and the size of their contribution. In this approach, the quality of an article is defined through quantitative information about it. However, there seems to be no reliable relation between the number of references in an article and the quality this article possesses.
3.6. Conclusions

Another suggestion is to provide a “student social graph” [Saltz et al., 2004], which shows the information of the students’ interactions. In our case an example would be a presentation showing to which articles a student made his or her contributions. This is a good solution to provide a tutor a better understanding about the connection and the involvement of the student in the learning process.

As introduced above, the explanation of assessing collaborative learning within wiki systems is just an example for approaches and challenges in the context of assessment 2.0. These have to be considered by models and architectures for an open assessment management platform. To achieve sustainability, it is crucial that such a platform allows the integration of social media services in a generic way, so that currently available services can be integrated as well as upcoming ones.

3.6. Conclusions

According to the objective of providing a platform for open assessment management supporting 21\textsuperscript{th} Century Assessment (cf. section 2.3), each of the analyzed tools contributes some very useful functions. But none of them provides all required features nor seems a seamless integration of different tools possible. Therefore, a new approach for the consolidation of technology enhanced assessment systems is needed, particularly with regard to a centralized open assessment management platform, rather than a platform for lifelong learning.

Furthermore, all of the above presented systems lack in some possibilities, which are considered as very important for the intended platform. Current main limitations are:
3. Assessment Tools

- integration with general e-learning functions and available system infrastructure,
- coverage of specific organizational processes,
  \( \text{(e.g. rubrics, peer assessment strategies, and grouping mechanisms)} \)
- reutilization of single evaluation steps,
- combination of different approaches,
  \( \text{(e.g. dynamically grouped peer assessment of wiki contributions with automatic pre-correction)} \)
- adaptability for various specific scenarios in parallel.
Chapter 4.

Real World Scenarios

In addition to possibilities provided by current and future tools, a sustainable open assessment management platform designed for practical (real world) use has to consider real world scenarios. Each lecturer, with it each university course, has a unique didactic approach. Furthermore, the assessment processes of courses differ in several details. Typically, these settings are not static, but can be adjusted dynamically by the lecturers. For instances, rhythm of assignment publication or settings for sample solutions can have influences on learning behavior and service acceptance [Altenbernd-Giani et al., 2009].

Therefore, three different measures have been adopted to get an overview of current scenarios. Firstly, a university-wide survey has to be conducted to get a broad view across different faculties and disciplines. An in-depth analysis has been made by use of interviews in the field of computer sciences. Requirements of other domains have been collected during iteratively conducted pilot installations (see section 9.2). The third analysis emphasizes the assessment of students’ contributions to wiki pages as an example for assessment of collaborative learning using a common web 2.0 tool.

4.1. University-wide Survey

In summer of 2009 a survey about assessment activities and related criteria has been conducted at RWTH Aachen University. All of the approximately 435 chairs have been called for participation, of which 162 chairs have answered the questionnaire. That is a response rate of about 37 percent. The distribution of participants over departments (see figure 4.1) shows, that all departments are covered. Some participants did not name their department. Others have not been official members of a department.
4. Real World Scenarios

Results of selected questions are presented in the following. To put this survey in the context of technology enhanced learning, lecturers have been asked if and how they utilize learning platforms to support their teaching.

Are features of a learning platform used?

Which learning platforms are in use?

73% of the participants definitely support their courses with use of one or more learning platforms (see figure 4.2a). Mainly six different platforms have been mentioned. The best coverage with 92% of all mentioned platforms has L²P, which is the centrally hosted platform of RWTH Aachen University (see section 7.2 for details). The other systems are used by less then 5%. MyReiff is a specialized system for the department of architecture, BSCW is a groupware platform, and Okuson is a custom system for individual tests and publication of results. The learning management platform Moodle has been used by single institutes and is currently reduced successively. The tailor-made test system Dynexite is used in addition and fully connected to L²P.
4.1. University-wide Survey

What kind of formative assessment activities are used?

Several types of assessments are currently in use (see figure 4.3). The most frequently mentioned assessment method is the use of weekly assignments. The other classical assessment methods, like in-class activity, compulsory attendance or exams, are often mentioned as well. More modern types like evaluating quantity and/or quality of activities in discussions in a forum, wiki pages or articles in a blog are also in use. Other mentioned types have been special types of exercises, like preparing and presenting seminar papers. Lecturers are often using combinations of different assessment types in their classes.
4. Real World Scenarios

Are formative assessment processes enhanced by technology?

Which activities are conducted with (partial) system support?

Each type of activity seems to be supported by technology in some scenarios (see figure 4.4). Assessment of contributions to blogs and wikis is completely based on technology, because the activities are technology-based. More traditional activities, which are based on direct communication (e.g. oral exams or in-class activity), are less supported. Using digital communication channels (e.g. video/audio conferencing tools) is an alternative. Surprisingly, only 50%-60% of assignments as well as tests are supported by technology, although a lot of systems are already available (cf. chapter 3).

Figure 4.4. Rate of technology enhancement used to support assessment activities.

What kind of rating scales are used to grade students’ performances?

The students’ performances in assessments are rated on different scales (see figure 4.5). Those could be numeric score, a percentaged value, German grades, a boolean result or ranked position relative to the results of all students in the course. Some custom scales and text based ratings are also in use.
What are formal goals to archive by performing formative assessment activities?

Beside summative assessments, performance on formative assessments are taken into account for reaching certain formal goals. According to figure 4.6, these goals could be a certificate of attendance, a certificate of performance, admission to a final exam, or others (e.g. bonus score for the final exam). The overall result of students’ assessment performances throughout a whole semester is mostly rated with a grade or with a boolean result, i.e. passed or failed. This overall outcome is a calculated result based on the single outcomes.

Which criteria have to be fulfilled by students to reach a formal goal?

The approaches for the calculation of the final course result vary in a wide range. They are mostly related to different concepts for extrinsic motivation to increase continuous learning. About 40% of the chairs define a maximal number a student is allowed to be absent from a lecture date (see figure 4.7). About 25% sum up the scores a student gained over all assignments and define a rate to pass relative to the maximal score possible. Other methods like building an average grade are used. Often those different criteria are combined to build the whole criterion for a course. For instance, a student has to reach 60% of the possible assignment score.
4. Real World Scenarios

at minimum, must have no more than two assignments with zero points, and must present one of his or her solutions in the tutorial or problems class.

Figure 4.7. Criteria to fulfill for reaching a course goal.

What kind of tools are used for bookkeeping and publishing students’ results? Which tools are used to define and calculate assessment criteria?

One question in the survey was asking for the kinds of utilities the lecturers use to report the students score, define criteria to pass and calculate related outcomes (see figure 4.8). More than 55% are using spreadsheet software (e.g. Microsoft Excel, OpenOffice.org Calc, or Numbers). Handwritten notes are made by about 35%. Less than 10% are using the integrated gradebook functionalities of a learning management system. Some chairs (about 6%) are using custom developed databases or applications for that purpose. Specific gradebook software has not been used by the participants of the survey.

Figure 4.8. Tools used to support criteria management and bookkeeping of students results.
4.2. Specific Scenarios in Computer Sciences

Assessment scenarios and their management in current practice have been analyzed in more depth by interviews with representatives of multiple chairs from the department of computer sciences at RWTH Aachen University. Lecturers from 17 of 27 chairs (63%) have been surveyed. The questions have been asked about management as well as realization of assessment activities per course. Thus, 66 of about 120 course (approx. 55%), which are offered computer science degree programs, have been covered. The used field manual for the interview can be found in appendix A.

The main focus has been set to offered assessment activities, utilized assessment methods, related management processes, and applied criteria, which have to be passed by students. The combination of activities via a specific criterion is a key fact. Some standard criteria, which are used in several courses, have been identified:

- The admission to a final exam requires $X\%$ of the overall score for weekly assignments.

- A certificate of attendance is gained if not more than $Y$ appointments in a series of tutorials during the semester have been missed.

Further criteria of selected courses, whose scenarios deviate from standard cases, are presented in the following. An overview of the result from all interview is given afterwards.

**Complexity Theory – Berechenbarkeit und Komplexität (BuK)**

*Complexity Theory* is a required course of bachelor’s studies in computer science at RWTH Aachen University. A total of approximately 250 to 300 students of computer science program as well as from other programs (e.g. communication science or teacher training) are attending this course every second term.

The assessment scenario of the course contains multiple activities, which can be used by students to collect score. Assignment sheets containing several exercises are published weekly. Common tasks are proofs, calculations, graphs, and essays. Some exercises are for self-assessment purposes only. Others are labeled, such that two points can be gained by submission of a correct solution. Further two points are granted for presentation of an individual solution to other students during a tutorial. By solving two midterm exams, additional 60 points per each can be collected. The criterion is fulfilled, and in this way admission to the final exam in accordance with the examination regulations granted, if a student achieves 60 points at minimum. Students are allowed to choose their preferred activities by themselves.
4. Real World Scenarios

BuK Criterion

Collect 60 points at minimum.
- 2 points per labeled exercise (max. 22 points),
- 2 points per presentation (max. 22 points),
- max. 60 points per midterm exam.

Assignment sheets are created digitally by use of \LaTeX{} and are published on the institutes web pages. Solutions have to be submitted on paper by single students or teams. Teams can be build dynamically by the students themselves with a limit of three students per team. Corrections are prepared by tutors with pen and paper. Feedback is provided by handing back the corrected solution sheet a week after submission. Sample solutions are only provided for tutors digitally. Students have to write them down during a tutorial. The bookkeeping of students results is done distributed by each tutor himself or herself. A centralized point of information for lecturers or personalized access for students is not provided.

Object-oriented Software Construction (OOSC)

Presenting solutions for weekly assignments is a central component of the assessment scenario in OOSC. This course is attended by about 40 students of the master program in computer science. Each participant has to present one of his solutions per term at minimum. To avoid that students only prepare one solution and do not work any further after this presentation, they have to vote each week if they want to present. They have to vote at least two times per term. Thus, multiple students prepare their solutions multiple times and only present, if they have been chosen.

OOSC Criterion

- present at least one solution for an assignment,
- vote at least two times to prepare a presentation.

Assignment sheets are created with MS Word and afterwards are printed on paper as well as published via the learning platform L²P. Common tasks are the development of source code, essays, and diagrams. For some assignments additional material (e.g. templates for source code or text forms) is provided. Management of students’ results are done with a spreadsheet file on a server at the institute. Students with poor chances of success are informed via e-mail.
4.2. Specific Scenarios in Computer Sciences

Distributed Systems – *Sichere Verteilte Systeme (SVS)*

Another strategy to increase continuous learning and well prepared presentations has been utilized in the course of distributed systems. Students are solving weekly assignments in small groups with up to three students. Solutions are presented in weekly tutorial classes. Attendance of at least one student per group is required at each appointment. Each participant of the tutorial class has to be prepared for presentation, because the presenting team is chosen randomly. The interviewed lecturer referred about increasing quality of presentations.

- at least 65% of the overall assignment scores,
- one presentation,
- required attendance of a team member (max. one missed appointment).

Assignment sheets are published as pdf files, via a self-hosted wiki system formally, now via L²P. Solutions are handed in on paper or digitally via e-mail. Corrections are made by tutors on paper. Digital submissions are printed before correction. Intermediate results are not published.

**Introduction to Databases (IDB)**

The assessment process in the database course, which is obligatory for bachelor students, is totally based on assignments. The difference to a default criterion is, that two different requirements are attached to the results for assignments in parallel. On the one hand a certain percentage of the overall score has to be reached. On the other hand submissions for at least 9 of 12 assignments have to be made successfully, i.e. those have to reach more than 0 points. In this way students are forced to work on as much as possible different assignments. Additionally, students are motivated to work on most questions by providing bonus score for the final exam, if at least 80% of the assignments score has been reached.

- at least 50% of the overall assignment score,
- at least 9 of 12 accepted solutions,
- bonus score at more than 80% of assignment score.
4. Real World Scenarios

Introduction to Programming (Progra)

A very complex scenario has been found at the introductory course on programming, which is obligatory for bachelor students in their first term. This course is attended by approximately 500 students per term, which have to solve weekly assignments in groups up to three students. At least one solution has to be presented. Furthermore, one extra assignment has to be solved individually in a presence class. To get admission to the final exam, 50% of the first part of assignments as well as another 50% of the second part have to be achieved. The extra assignment has to be passed with at least 50% as well. Missing score at the extra assignment can be balanced with additional score from the second part of assignments. The course is organized by two different chairs each year in alternation. Thus, the criterion is adjusted each year, for instance with an additional amount of extra assignments or required presentations.

<table>
<thead>
<tr>
<th>Progra Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• at least 50% of the first half of assignment score,</td>
</tr>
<tr>
<td>• at least 50% of the second half of assignment score,</td>
</tr>
<tr>
<td>• one presentation,</td>
</tr>
<tr>
<td>• at least 50% of an extra assignment or</td>
</tr>
<tr>
<td>(at least 33% of the extra assignment and</td>
</tr>
<tr>
<td>at least 50% of the combined score from the extra assignment and the second half of assignments).</td>
</tr>
</tbody>
</table>

Submission of students’ source code has to be done on paper as well as via e-mail. Tutors are compiling the digital version of the program and write down comments on the printout, which is handed back to the students. Although a huge number of systems for semi-automatic correction of programming assignments is already available, the correction is done completely manually. A self-made database system is the only technical support for the assessment process. It is used for the management of students results. Students have personalized access to this database via a web-based user interface. Thus, they are able to have a view on current results permanently. The evaluation of the criterion is done not until the end of term by a complex SQL-Query on the database. Adjustments of the scenario or criterion require direct adjustments on this query.
4.2. Specific Scenarios in Computer Sciences

Discussion

Utilization of a self-made system for result management is an exception. It has only been done at 13% of the courses. In most cases, spreadsheet software is used (72%). Handwritten notes are used in 6% of the courses. No precise statement can be made for 13% of the courses, since each tutor manages the process himself or herself.

Data for collection is produced within different activities. Weekly assignment are used in more than 98% of courses. Students submissions, which are corrected individually, are expected in 94%. They are graded with score (80%), German grades (5%), passed or failed (5%), or stay ungraded (10%) with elaborated feedback. Objective Tests are used in 15 courses (approx. 23%), but only very infrequent in textual form on assignment sheets. Solutions have to be presented at half of the courses. Voting for presentations is used in 10%. Midterm exams are used for 8% of the courses. Attendance is required at 12%. Assessment of contributions to wiki pages, blogs or discussion forms, or other emergent technologies has not been realized.

A very interesting results is that assignment management is still strongly based on paper in most courses for computer science. Publication of assignment sheets is increasingly done via websites or the learning management system L²P (see figure 4.9a). In most cases students have to submit their solutions on paper (see figure 4.9b). Interviewees argued that digital creation of solutions for several assignment types (e.g. graphs or mathematical proofs) is more difficult than with pen and paper. Creation of source code for instance is obviously done digitally and mostly has to be submitted electronically. Sometimes a printout has to be handed in additionally.

Figure 4.9. Applied approaches for assignment management.
The correction process depends on the type of submission and is therefore mostly done on paper as well (see figure 4.9c).

4.3. Scenarios with Wikis

One result of the broader survey has been, that some lecturers already assess students’ contributions within collaborative scenarios with wiki pages. Three selected scenarios have been investigated in more depth by semi-structured interviews [T6]. Current approaches, their strengths and weakness as well as suggestions for process improvements have been collected according to a prepared field manual (see table 4.1).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning scenario</td>
<td>What is the setting of the course?</td>
</tr>
<tr>
<td></td>
<td>Number of participants?</td>
</tr>
<tr>
<td></td>
<td>Final exam?</td>
</tr>
<tr>
<td>Wiki usage</td>
<td>How are wiki contributions integrated with other course activities?</td>
</tr>
<tr>
<td></td>
<td>In which way does students have to contribute?</td>
</tr>
<tr>
<td></td>
<td>Why are wikis used?</td>
</tr>
<tr>
<td>Assessment</td>
<td>How are students graded?</td>
</tr>
<tr>
<td></td>
<td>What are important criteria for grading?</td>
</tr>
<tr>
<td>Problems</td>
<td>What are current problems of the process?</td>
</tr>
<tr>
<td></td>
<td>What are currently applied workarounds?</td>
</tr>
<tr>
<td>Suggestions</td>
<td>How to improve grading?</td>
</tr>
<tr>
<td></td>
<td>Which features would ease the process?</td>
</tr>
</tbody>
</table>

Table 4.1. Field manual for interviews according to assessment of wiki contributions.

First Interview

The first interviewee lectures a course about linguistics, especially English grammar. The course is attended by 35 to 40 students in their first term of their studies. The lecturer has had some difficulties to find appropriate assignments in grammar books, which fit to the needs of her students. Therefore, she developed an approach to solve this problem using a wiki system.

The idea was that students construct a common collection of grammar exercises on wiki pages collaboratively. Each chapter of the collection is filled with exercises to a certain topic from each student. Contributions are peer reviewed and corrected. The approach has been realized with the integrated wiki system of L²P.
4.3. Scenarios with Wikis

The course consists of 14 weekly classes with a length of 90 minutes each. A new topic has been taught every week. The transfer of knowledge about a topic has been treated as successful, if students are able to phrase their own new exercises about it.

The related collaborative learning scenario can be described as a sequence of weekly processes. Such a weekly process is structured as follows:

1. New topic (lecturer)
   - prepare topic
   - create wiki page
   - provide introductory information with examples

2. Create exercises (each student)
   - write new exercise to wiki page
   - correct exercises of peers

3. Create solutions (each student)
   - solve exercise on paper
   - submit paper solution (outside wiki pages)

4. Create sample solutions (each student)
   - write sample solution of own exercise to wiki page
   - correct sample solutions provided by peers

Only step three has been done without support of the wiki platform. The other steps are possible or rather realizable by use of wikis within L²P. Thus, wiki pages seem to be a good choice to support the mentioned scenario. The edit functionality of the L²P-wiki pages facilitate students to submit their own contributions as well as to correct solutions of the peers in an easy way.

Motivation for active participation is intended by a final exam about exercises, which are contained in the collaboratively developed collection. An additional oral exam is provided to students with poor results in the exam. A detailed analysis of a student’s participation to the wiki-based exercise collection is needed to prepare the oral exam. Quality of contributions as well as weaknesses in topics have to be examined. All versions of all wiki pages have to be read to find a student’s contributions and evaluate the according to quality criteria. It was not possible to note results of the time-consuming evaluation directly within the system with relation to the contributions.

A possibility to mark weekly performances of students with either active or not active directly within the system has been suggested as an improvement. Another
suggestion is related to the parallel achieved scores gained from the submitted solutions outside the systems. These results should be displayed associated to the other marks as well as to the contributions themselves.

**Second Interview**

In this case students have to create an essay collaboratively as a project in parallel to a course. The 100 students of the course have been partitioned into twenty groups with five members each. Different possible topics for an essay are provided at the beginning of the term. Each group got a topic they have to work on during the whole term. A wiki page per group has been used for creation of the essay.

Furthermore, each student has to provide feedback to essays of other groups. Therefore, an additional wiki page for feedback has been created for each group or rather essay. Comments, suggestions for improvements as well as critique can be posted to these pages.

Same as the first scenario, this scenario has been realized within the wiki pages of L²P as well. The process of this course is structured as follows:

1. Assign groups of 5 students each. (Lecturer)
2. Assign a topic to each group. (Lecturer)
3. Create a wiki page for an essay for each group. (Lecturer)
4. Create a wiki page for feedback for each group. (Lecturer)
5. Create an essay for the given topic on the wiki page. (Members of a group)
6. Provide feedback to peers on their feedback page. (Each student)

Students are assessed individually at the end of term according to their work on the essay. They are either *pass* or *fail* with regard to the following criteria:

- The essay a student created with his group is of good quality.
- The student has contributed own ideas to the essay.
- The student have actively written constructive feedback to essays of other groups.

The evaluation of the first criterion can be done reading the final essay. Former versions and individual contributions are ignored. Second and third criterion are more difficult, because every version of an essay page as well as of all feedback pages have to be analyzed. The interview described a very time-consuming process, in which the overview is often lost.
Third Interview

The third scenario has been realized with wiki pages in L²P for a course with about 25 students. Similar to the second scenario, the lecturer wants to increase students’ continuous activity for the course. Therefore, each students have to write ten short essays about given topics during a time slot of three weeks. Elaborated feedback is provided by peer review as well as by the lecturer. Misspellings as well as linguistic mistakes are corrected by the lecturer additionally.

The learning and assessment process is organized as follows.

1. A wiki page is created for each student. (Lecturer)
2. Ten short essays are typed to the personal wiki page. (Each student)
3. Feedback is provided directly into the wiki pages of peers. (Each student, Lecturer)
4. Linguistic correction are made to all essays. (Lecturer)

Main drawbacks of the realization are that feedback and corrections from the lecturer is hardly recognizable or rather distinguishable from peer feedback. Every reviewer, peers as well as the lecturer, have to analyze all versions of a wiki page to distinguish the authors contributions from contributions of an assessor.

4.4. Discussion

All in all, there exists a variety of didactic concepts and related assessment scenarios which are in practical use. Same objectives, mostly of motivational nature, are address with different very different approaches (e.g. voting for presentations vs. required team attendance). Most complex criteria for extrinsic motivation of continuous learning have been found in courses of first terms in a program of study. The management of such complex assessment processes are very complex and time-consuming, especially in these lectures with several hundreds of students as well as multiple tutors and assistant professors.

The definition of an initial criterion for a course, the organization of involved persons and systems, configuration of tools, and its execution is quite difficult. Different study programs as well as certification regulations have to be respected. Using a former criterion again does not reduce required effort very much, because a lot of organization, administration, and configuration have to be done time and time again.
4. Real World Scenarios

Main findings of the scenario analysis are:

- Each applied scenario has a specific variation regarding other scenarios.
- Other assessment methods than tutor assessment are rarely applied.
- Weekly assignments are the most important assessment activities.
- Collecting assignment submissions is often done on paper or by mail.
- Grouping of small teams for collaborative work has often several constraints.
- Domain-specific systems are not in use.
- Centrally hosted services are well accepted.
- Application of web 2.0 services for assessment can be found only in few cases.
- Assessing individual contribution in collaborative environments is not well supported.
Chapter 5.

Requirements

With the intention to build an integrated platform for open assessment management, several requirements have been collected based on related objectives, approaches and tools from literature, and current real world scenarios. These are mostly focusing the consolidation of current practices. New and innovative approaches – beside the innovation of an open assessment management platform itself – have been respected within an additional requirement elicitation in two phases, including a SWOT-Analysis (see section 5.1). A categorized specification of requirements and use cases is summarized in section 5.2.

5.1. Elicitation

As results of above presented survey and interviews show, there only few scenarios in practice, which utilize social media. Neither scenarios with integration of new dimensions of learning nor corresponding assessment practices could be identified in detail. Therefore, requirements for integration of new learning dimensions with institutional assessment processes (according to open assessment management, see section 2.3) have been surveyed in two phases [T3].

5.1.1. First Phase

The main goal in the first cycle of requirement elicitation was to find the strengths and weaknesses of current assessment process according to the main stakeholders, which are academic staff and students. Opportunities to enhance assessment processes with the usage and integration of external web 2.0 services within a university’s learning platform are of main interest. Thus, these questionnaires aim to reveal the thoughts of the stakeholders regarding assessment 2.0 and to gather more specific ideas of opportunities offered by web 2.0 applications and services. Results
5. Requirements

are presented in a SWOT analysis of assessment processes which might give a more clear overview of main conceptual requirements towards a generic open assessment environment. In order to have as many ideas, the survey has been formulated with open questions. Examples are given to orientate the interviewees or explicitly ask them to brainstorm and write down the ideas.

Another important result that has been derived from the questionnaire is a comparison between the ideas of students and academic staff about assessment or rather open assessment. Thus, two separate questionnaires have been conducted, one for students and the other for academic staff. Questions are substantially the same and can be easily confronted.

Questionnaire Outcomes

The survey was developed with the help of Google Docs and has been distributed mainly online. Eleven open questions have been asked to our interviewees with very small distinctions between the teacher and students questionnaires. The participants filled the questionnaire individually, with only 3 personal interviews, one to a teacher and two to students. 38 persons (27 students and 11 teachers) have taken part at the surveys. We had a diversity of participants according to age, gender and study/teaching fields as shown in figure 5.1:

![Diagrams of participant diversity](image)

(a) Student age  (b) Student gender  (c) Teacher age  (d) Teacher gender

**Figure 5.1.** Diversity of participants of the requirements survey in phase one.

Comparison of teachers’ and students’ views on Open Assessment

Selected questions and respective answers are presented with differentiation by students and academic staff. Some questions are presented separately because they are more relevant to a SWOT analysis, which is described subsequently. Several outcomes of the following questions are relevant in terms of new opportunities and threats for assessment. They will be represented in the related sections of the SWOT analysis as well.
5.1. Elicitation

Do you think that using web 2.0 content and applications for assessment tasks is a good opportunity?

74\% of the surveyed students share the opinion, that web 2.0 services should be utilized for enhancement of assessment processes (see figure 5.2a). 3.7\% are not sure about it and another 3.7\% does not think, that it is a good idea. Students in the field of computer sciences does not differ from other students. Lecturers of this discipline are not sure about the opportunities. The distribution of pro and cons among lecturers of other departments are balanced (see figure 5.2b).

Can you mention at least 3 web 2.0 applications, that you use the most during your assessment tasks?

Even if not already integrated, students as well as teachers stated, that they use different web 2.0 services during their assessment tasks (see figure 5.3). Some these services or used for information gathering (e.g. digital libraries, forums, or Wikipedia) and communication, while others directly affect content creation (e.g. Google Applications or wikis) and material sharing and submission (e.g. Dropbox and YouTube). Furthermore, the result shows that various distributed services are already used, especially by students.

If you were asked to submit a material (document, audio, video, code etc.), how do you think is the most appropriate way to submit them?

Most students state that e-mail is still an accepted way for submission (see figure 5.4). Teachers prefer submission handling via the learning platform L²P as well as by hand. Submission via Dropbox or YouTube have been mentioned by teachers and students similarly. Using social networks has been suggested by students only.
5. Requirements

**Figure 5.3.** Currently used web 2.0 applications during assessment tasks.

**Figure 5.4.** Students’ and teachers’ suggestions for material submission processes.
What common features of those applications do you think are useful for the assessment activity?

Generally web 2.0 applications are not designed for assessment purposes, but the survey participants identified some features and interactions which are already available to serve for assessment (see figure 5.5). Ratings (37%), comments (25.9%), grouping processes (18.5%), and file uploading mechanisms (14.8%) are the most mentioned features.

What other features do you think are needed in such applications in a way that they can fully support the assessment activity?

Missing features which have been thought of useful for assessment purposes have been identified (see figure 5.6). Main points are security and privacy settings as well as integrated communication mechanisms. Teachers are mainly interested on organizational features for assessment (e.g. deadline handling).

What are the main threats of using these external services for assessment?

Identified threats of opening assessment by integration of external services are privacy, plagiarism, security, reliability, and usability (see figure 5.7). Furthermore, teachers worry about increasing effort for organizational tasks.
5. Requirements

Figure 5.6. Missing features of web 2.0 tools for assessment.

Figure 5.7. Threats of using external services for assessment.
Can you envision how assessment can look like in 10 years from now? What aspects of traditional assessment will remain and what will change?

Figure 5.8 presents opinions about the future development in the area of assessment. One interpretation of the answers is that basic assessment processes will stay the same, but publication and submission processes as well as communication are mostly supported online, with integration of automatic evaluation mechanisms. Facilitated by the technical support of a web-based system, institutional learning and assessment will be opened beyond university borders, especially formative assessment processes, which become more important in contrast to summative exams.

SWOT Analysis

A SWOT analysis is an effective method to identify and plan innovation decisions in many areas [Pearce and Robinson, 2003]. It was conceived as a tool for business planning but it can be used in many other fields, for example research, as well. It can help to determine the strength and weaknesses of a certain solution and identify possible opportunities of research. The SWOT has four main divisions Strengths and Weaknesses which come from internal factors as well as Opportunities and Threats which come from external factors. It is usually represented in a 2x2 matrix. To find strengths and weaknesses three initial questions have been ask in the questionnaire.
5. Requirements

In which ways have you been assessed during your study experience?
In which ways do you assess your students during your teaching experience?

These questions have been a prerequisite for the next questions. Students and teachers current experiences with different assessment activities have been surveyed to serve as a reference to identify strengths and weaknesses. All of them are familiar with oral and written exams (see figure 5.9), because these are required in examination regulations. Most students have been assess by multiple choice and other objective questions, while only a few teachers still apply them.

![Figure 5.9. Survey participants’ experiences with assessment.](image)

The results of the following questions about strengths and weaknesses have been analyzed in combination with above mentioned questions aiming opportunities and threats. A conclusion is represented in the respective fields of the SWOT table 5.1.

Can you mention some advantages of known types of assessment?
Can you mention some disadvantages of known types of assessment?

Conclusions and consequences

The results of this questionnaire represent the fundamentals of the work on eliciting the requirements of an open assessment environment. It is clear that technology has affected the way students prepare their assessment tasks and generally the distribution of the assessment material. On the one hand great majority of students support the idea of making use of technologies during assessment tasks. They have plenty of ideas and names of applications which are useful for specific tasks. On the other hand academic staff is not familiar with these technologies and consequently is skeptic about their utility. Results of two additional questions, which have been asked to academic staff only, are useful to sum up the conclusions.
### 5.1. Elicitation

#### Table 5.1. SWOT table for integration of external services to assessment processes. [T3]

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Has high social value</td>
<td>• One point in time view</td>
</tr>
<tr>
<td>• Improves communication skills</td>
<td>• Hard to assess individual work (Group work)</td>
</tr>
<tr>
<td>• Improves team work skills</td>
<td>• Workload division is not equal</td>
</tr>
<tr>
<td>• Offers concrete implementations (Projects)</td>
<td>• Don’t encourage regular studying (Examinations)</td>
</tr>
<tr>
<td>• Leads to self-assessment</td>
<td>• Lack of intensity</td>
</tr>
<tr>
<td>• Learn from peers (Group work)</td>
<td>• Easy to lose focus</td>
</tr>
<tr>
<td>• Is uniform</td>
<td>• Lack the personal aspect of learning</td>
</tr>
<tr>
<td>• Determines progress</td>
<td>• Time consuming</td>
</tr>
<tr>
<td>• Evaluates commitment</td>
<td>• Resource consuming</td>
</tr>
<tr>
<td>• Interactive way of showing knowledge</td>
<td>• Individual skills are hidden (Group work)</td>
</tr>
<tr>
<td>• Structured and focused learning</td>
<td>• Not enough time (Examinations)</td>
</tr>
<tr>
<td>• Focus on the subject</td>
<td>• Long and exhausting assessments</td>
</tr>
<tr>
<td>• Harder to cheat</td>
<td>• Not clear or tricky questions</td>
</tr>
<tr>
<td>• Indicate the level of the class</td>
<td>• Influenced by the assessor opinion</td>
</tr>
<tr>
<td>• Informal feedback</td>
<td>• Assuming the majority position</td>
</tr>
<tr>
<td></td>
<td>• Not enough or continuous feedback</td>
</tr>
<tr>
<td></td>
<td>• Group coordination and motivation problems (Group work)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Make use of web 2.0 applications</td>
<td>• Usability</td>
</tr>
<tr>
<td>• Reduce time and effort for material</td>
<td>• Reliability</td>
</tr>
<tr>
<td>submission</td>
<td>• Privacy</td>
</tr>
<tr>
<td>• Continuous feedback</td>
<td>• Security</td>
</tr>
<tr>
<td>• Enhance peer-review</td>
<td>• Plagiarism</td>
</tr>
<tr>
<td>• Strong social interactions</td>
<td>• Cheating</td>
</tr>
<tr>
<td>• Automatic assessment options</td>
<td>• False information spreading</td>
</tr>
<tr>
<td>• Integrate web 2.0 services into the</td>
<td>• Reputation problem</td>
</tr>
<tr>
<td>university learning platform</td>
<td>• Loosing pen and paper touch</td>
</tr>
<tr>
<td>• Easy authorship identification</td>
<td></td>
</tr>
<tr>
<td>• Offer anytime and anywhere teaching</td>
<td></td>
</tr>
<tr>
<td>• Offer personalized assessment</td>
<td></td>
</tr>
</tbody>
</table>
According to your experience, are your students using web 2.0 services and applications during assessment tasks? If yes, do you think this produces a more qualitative or quantitative work?

The outcome seems to contradict what was sustained by the academic staff in the previous questions in which they did not appreciate the usage of web 2.0 applications as useful. Here almost 75% of the teachers are aware that their students make use of web 2.0 applications during their assessment tasks and this according to most of them improves mainly quantity but increases also quality in a considerable way. A possible explanation of the skepticism of academic staff towards the use of technology is their lack of knowledge of the potential of such applications. This conclusion is supported by the lack of ideas that the academic staff had on each of the questions where they are asked to name applications and related features.

(a) Students’ usage of web 2.0 services during assessment.  
(b) Expected effects of web 2.0 usage. 

Figure 5.10. Effects of using web 2.0 tools for assessment.

5.1.2. Second Phase

The first round of interviews gave important conclusions which can be found summarized in the SWOT table 5.10. Since the aim was to evaluate requirements for a generic open assessment system two chosen web 2.0 applications have been evaluated as a demonstrative scenario. They have been chosen based on the interviewees’ preferences (cf. figure 5.3). Google applications and mainly Google Docs are stated to be the most used applications by the students, while no academic staff mentions any of them. Anyway, they do not have been chosen, because they are mostly limited to content creation, which takes only a small part within an assessment management process. Wikis are also largely available and used already for assessment purposes. The next applications mentioned as very useful for assessment tasks are YouTube and Dropbox.

Findings of the first questionnaire will serve as a starting point of the second questionnaire which should offer a view on the functional requirements of an assessment
environment with the help of *YouTube* and *DropBox*. Within the questionnaire, an overview of the results of the previous survey as well as brief descriptions of the two chosen applications are given.

The objectives of this survey were to determine the functional requirements that applications such as *YouTube* and *DropBox* should have to support an assessment activity. Mainly multiple choice questions or rating grids of all of the gathered ideas from the previous questionnaire have been conducted. Since the focus is set on two specific applications, it was intended to extract which functionalities apply to these tools. Another advantage of this approach is to have the list of the requirements already prioritized for the implementation.

**Questionnaire Outcome**

The questionnaire was developed with the help of a web application *SurveyGzimo* and was distributed online. Again, two separate questionnaires, one for students and the other for academic staff, have been developed in order to address specific questions. Results of common questions should have been compared. In this round of interviews 32 persons participated, 22 of them were students and 11 teachers. The questionnaire had 7 questions for the academic staff and 6 questions for the students. Diversity of participants according to age, gender and academic fields are very similar to the survey of the first phase. We noticed no big difference in the answers related to the academic field, age and gender of both students and teachers.

How do you think is the most appropriate way to use these applications for assessment?

The most important question for the thesis project was to know the preferences of our users regarding the environment and origin of the applications which are use. For example *YouTube* and *DropBox* can be used for assessment already as *distributed services* or can be *integrated* into the university platform with possible additional functionalities. Furthermore, the re-implementation of *own services* as tailor-made and homogeneous applications within a universities’ platform can be an option (e.g. *RWTHTube* as a YouTube re-implementation for usage at RWTH Aachen University solely). Figure 5.11 shows the opinions of students and academic staff towards these three ideas represented.

Answers of the students and teachers are slightly different (see figure 5.11). Most students prefer the option to integrate existing web 2.0 applications into the university platform. Teachers prefer the implementation of separate services. If we analyze the overall data from both students and teachers then the option to integrate the existing applications is the most coined with 53% of the preferences against 25% of the third option and 22% of the first one. Additionally, an integration scenario is adoptable for combination of a separate services with a learning platform.
5. Requirements

Figure 5.11. Types of using web 2.0 services for assessment.

? Please select the relevance of the following features for an assessment activity using YouTube and DropBox.

To analyze this rating grid of questions points have been assigned to each rating option starting from 1(not important) to 5(very important). The charts below are represented the scores of each functionality for our applications according to the survey outcome.

Here there are also relative diversities in the answers between students and teachers, but as we can notice privacy and security are common important requirements from both parts (see figure 5.12). Comments, rating, group creation and the possibility to upload the video from our interface are the other most important requirements from the students. The teachers on the other hand think that embedding the video into the assessment platform interface is more important together with comments. Less important functionalities include number of views information and a feature for polls and questionnaires.

The next question is based on the ideas that users gave on the previous questionnaires about additional functionalities that are required from these tools. These functionalities are not present in the applications and are intended as enhancements to better support the assessment activity.

? Please evaluate the importance of the following features for an assessment activity using YouTube and DropBox.

According to both students and academic staff an option to send private messages is very important to support assessment (see figure 5.13). A deadline functionality and correction tools are required as functionalities in the assessment process as well. Furthermore, less important but still relevant are considered the options to collect entries from one person and to visualize or provide an assessment schema for both parts.
5.1. Elicitation

**Figure 5.12.** Relevance of assessment features for YouTube and DropBox.

**Figure 5.13.** Importance of assessment features for YouTube and DropBox.
5. Requirements

? Consider group works. How can we distinguish the individual work?

The majority of interviewees agree that everyone can see the contributors name after each action (see figure 5.14). Possible actions of a user might be for example uploading, commenting, rating, viewing etc. This would solve the authorship problem mentioned in the previous survey as a disadvantage of current assessment systems and would allow transparency not only for the teachers but also for students themselves.

Figure 5.14. Visibility of individual authorship or contributions to group work.

? How would you like to give/receive feedback to the students with this tool?

As displayed in figure 5.15), there are differences in the answers between students and teachers. The first ones think that the option to have feedback from teachers in an identical way as the feedback from peers is leading while the teachers think that is better that they give feedback privately with a messaging option. The other fields were left as empty to gather ideas and all the comments were suggesting that it is important to have both of these options.

? What specific privacy options do you need from these two applications?

This question was also left open, to see what are the main concerns of the stakeholders about privacy. The interviewees did not have a clear idea of the environment in which such activity will take part, since this was still an open question in this survey. Most of the answers were similar. Both students and teachers agreed that they like to have the possibility to select who views their actions with defining some predefined groups with different roles and rights. Few others asked the possibility to define remain anonymous or to be able to define for each post the audience. In this way everyone can select with whom to share the information.
5.1. Elicitation

Concluding we asked to the academic staff also to select what automatic assessment functionalities did they need from such tools, giving three options and leaving an open field for other possible ideas.

What kind of (automated) processing for assessment would you need from this tool?

As we can see the most requested feature is to have a separated view when clicking on the name of a student to see all his contributions (see figure 5.16). Also the deadline functionality has been mentioned as an important assessment support mechanism. No one of the interviewees gave any idea in the other field, so no important option seems to be left out.

Conclusions

Results of the second survey define special needs for integration of external services to institutional assessment processes. Main functionalities and use cases for a related open assessment management platform have been extracted. A list of functional requirements has been extracted and is presented in the next section.
5. Requirements

5.2. Specification

Scenarios, problems, objectives, possibilities, and opportunities of technology enhanced assessment, and especially an open assessment management platform, have been analyzed to collect functional and non-functional requirements for such a system. Pragmatic requirements are respected in equal measure to scientific research about “what are the features of an ideal e-assessment system” [MacKenzie, 2003] or more advanced approaches for flexible and service-oriented systems like in [AL-Smadi et al., 2009, AL-Smadi and Gütl, 2008]. The requirements analysis presented in the thesis at hand goes beyond known analysis by consideration of lifelong learning and social media as well as centralized hosting situations, since aspects across all dimension of open assessment management have been specified for the first time.

General nonfunctional requirements are mostly the same as for all systems which provide mechanisms for assessment management, learning management, or document and content management:

- provide a sustainable system base,
- allow web-based access,
- allow modular extensions,
- allow integration from and with other systems,
- allow flexible adjustments,
- allow workspaces for parallel scenarios,
- be scalable for hosting usage,
- provide a good usability,
- secure private data.

Gathered functional requirements are specified in form of use cases, which are used to describe possible interactions of different actors or roles with a system [Cockburn, 2003]. For reasons of clarity and readability these uses cases are presented in form of use case diagrams as defined in the Unified Modeling Language (UML). Furthermore, each use case is briefly explained User Stories, which are mostly used in agile development processes like SCRUM or Extreme Programming (XP). They are defined by short sentences which mainly comply with the following template [Cohn, 2004]:

*User Story Template*

```
<Role> wants <goal/desire> so that <benefit>.
```

or

```
<Role> wants <goal/desire>.
```
5.2. Specification

Identified roles which appear in this context are listed below.

- Student,
- Tutor,
- Teacher,
- Administrator,
- Developer,
- External Service.

Institutional roles like dean or academic advisors are relevant for organizational tasks across several lectures. Since, they are not directly involved to formative assessment management in a single lecture, such roles are not considered here.

Because, main objective is to provide a generic and flexible platform which allows further extension to cope with emergent approaches for assessment, 90 top level requirements are specified at this point. More specific requirements related to a certain module are presented during presentation of these modules in chapter 8. Same categories which have been applied for tool analysis (cf. chapter 3 and figure 3.1) are used to prepare the requirements systematically.

Activities

Figure 5.17 presents the use case diagram containing use cases which are related to the publication of activities, related solution creation, and correction by a teacher or tutor. Associated requirements are comparable with those of monolithic but extensional platforms for tutor assessment like EASY (cf. [Eilers et al., 2008]). User stories for short description of the uses cases are listed below.

**UC 1.1:** A teacher wants to plan the assessment scenario.

**UC 1.2:** A teacher wants to create an assessment activity so that it is provided for students online.

**UC 1.3:** A teacher wants to choose the type of activity so that different types can be combined in a scenario.

**UC 1.4:** A student wants to create a solution for a published activity so that it is accessible online.

**UC 1.5:** A student wants to submit a solution online so that he or she has a single point of information.

**UC 1.6:** A tutor (and implicit a teacher) wants to assess a submission online so that a feedback and marks can be accessed from anytime and anywhere.
5. Requirements

UC 1.7: A developer wants to develop an activity editor so that it can be used to enter specific metadata during creation of certain activities.

UC 1.8: A developer wants to develop a correction editor so that it can be used to enter specific metadata during the assessment of a submission.

UC 1.9: A developer wants to develop a solution editor so that it can be used to enter specific metadata during creation and submission of a solution.

UC 1.10: A developer wants to develop activity type modules so that related editors are be packed for deployment.

UC 1.11: An administrator wants to deploy an activity type module so that the system supports new types of activities.

Marking

Tasks related marking processes are contain in figure 5.18. Corresponding user stories are listed in the following:

UC 2.1: A teacher wants to define a criterion for the course so that it can be published to students.

UC 2.2: A teacher wants to calculate students’ status related to the course criterion so that it is transparent to the teacher as well as to the students.
5.2. Specification

Figure 5.18. Use cases concerning marking.

**UC 2.3:** A teacher wants to create a marking scheme so that the marking becomes more objective and transparent.

**UC 2.4:** A teacher wants to define an indicator so that results on a certain subcategory (e.g. exercise or competency) are presented explicitly.

**UC 2.5:** A teacher wants to choose a rating scale for an indicator so that results can be presented in different dimensions.

**UC 2.6:** A teacher wants to define a custom scale so that it can be used for indicators.

**UC 2.7:** A tutor wants to fill the marking rubric so that measures of students' performances are protocolled.

**UC 2.8:** A tutor wants to publish marking results so that they are published for students individually.

**UC 2.9:** A student wants to view personal results so that the own performance can be appraised by a student.

**UC 2.10:** A student wants to view the defined course criterion so that it is transparent what to do for passing the course.

**UC 2.11:** A student wants to view the personal status related to the course criterion and compared to other participants so that own results can be self-assessed.
5. Requirements

**Group Assessment**

Requirements for handling group assessment, especially the grouping mechanism, are covered by the use case displayed in figure 5.19 as well as in the related user stories.

![Open Assessment Management Platform – Group Assessment](image)

**Figure 5.19.** Use cases concerning group assessment.

**UC 3.1:** A teacher wants to manage a group context so that assessment activities can be done collaboratively.

**UC 3.2:** A teacher wants to choose the grouping strategy so that different scenarios are supported.

**UC 3.3:** A teacher wants to assign a grouping rule to a group context so that groups can only be created regarding certain regulations.

**UC 3.4:** A teacher wants to create a grouping so that it can be used for restrictions.

**UC 3.5:** A tutor wants to assess group work so that collaboratively created work can be assessed.

**UC 3.6:** A tutor as well as a student want to create a group within specific context so that students can be grouped in different context for different scenarios in parallel.

**UC 3.7:** A tutor wants to assign students to a group.

**UC 3.8:** A student wants to participate group work.

**UC 3.9:** A student wants to invite new members to a group so that a group can be build dynamically.
UC 3.10: A student wants to answer an invitation to a group so that invitations can be accepted or declined.

UC 3.11: A student wants to request an invitation so that it can be asked to enter a group.

UC 3.12: A developer wants to develop grouping rule types so that custom business logic for specific reusable rules can be extended.

UC 3.13: An administrator wants to deploy a grouping rule type so that it can be made available for teachers.

Peer Assessment

Requirements concerning peer assessment processes as well as extensional mechanisms for automating distribution strategies are formulated below (cf. figure 5.20).

![Peer Assessment Diagram](image)

**Figure 5.20.** Use cases concerning peer assessment.

UC 4.1: A teacher wants to manage a peer marking strategy so that different types of peer assessment can be carried out.

UC 4.2: A teacher wants to assign a peer distribution strategy so that students are assigned to submissions of peers in a selected way.

UC 4.3: A teacher wants to create a peer marking scheme so that students can report marks about their peers work.

UC 4.4: A student wants to view an assigned solution so that it can be assessed.

UC 4.5: A student wants to mark an assigned solution so that additional valuations are provided.
UC 4.6: A student wants to comment an assigned solution so that additional elaborated feedback is provided.

UC 4.7: A developer wants to develop a peer distribution module so that a new approach of assigning students to submissions of their peers are realized in a reusable way.

UC 4.8: A developer wants to develop a peer distribution strategy so that peer distribution modules can be used to realize the assignment in a given environment.

UC 4.9: An administrator wants to deploy peer distribution strategies so that they can be used by teachers.

Self-Assessment

Situations of self-marking, self-commenting, and self-evaluation are described in figure 5.21 and explained in the user stories below.

Figure 5.21. Use cases concerning self-assessment.

UC 5.1: A teacher wants to provide activities for self-assessment so that students are motivated to apply and to gather knowledge.

UC 5.2: A student wants to work an provided activities so that specific techniques are internalized and knowledge is deepened.

UC 5.3: A student wants to review solutions from peers so that they can be compared with own approaches.

UC 5.4: A student wants to view the own level of performance so that former efforts can be reflected and further ones are planned.

UC 5.5: A student wants to assess an own submission so that the own performance has to be reflected and evaluated.
UC 5.6: A student wants to view feedback about the self-assessment so that the self-assessment itself can be reflected on a meta-level again.

UC 5.7: A tutor or peer wants to comment a student’s self-assessment so that different views on a performance can be compared.

Automatic Assessment

Uses cases for a modular and extensible approach for (semi-)automatic assessment processes are defined from development and deployment to association and execution in figure 5.22.

![Open Assessment Management Platform - Automatic Assessment](image)

**Figure 5.22.** Use cases concerning automatic assessment.

UC 6.1: A developer wants to develop an automatic evaluation module so that domain-specific knowledge about a specific evaluation task can be encapsulated in a reusable way.

UC 6.2: A developer wants to develop a (semi-)automatic correction process so that modules are composed to use them combined for a specific kind of activity.

UC 6.3: An administrator wants to deploy correction processes and module so that they can be used in the assessment environment.

UC 6.4: A teacher wants to assign a correction process to an activity so that it can be executed on related submissions.

UC 6.5: A teacher wants to configure a correction process for an activity so that it can be adjusted for specific settings.
5. Requirements

**UC 6.6:** A teacher wants to compose new correction processes with use of given modules so that the correction is more flexible.

**UC 6.7:** A student as well as a tutor want to start a correction process for a specific submission so that feedback can be generated automatically.

**UC 6.8:** A student as well as a tutor want to parametrize a correction process so that it is executed with individual options for each submission.

**Feedback**

Requirements for provision of feedback respecting the different possibilities of timing as well as different feedback level are displayed in figure 5.23.

![Open Assessment Management Platform - Feedback](image)

**Figure 5.23.** Use cases concerning feedback.

**UC 7.1:** A teacher wants to create a sample solution so that a correct approach is explained for students.

**UC 7.2:** A tutor wants to create an individual correction of a students’ submission so that individualized feedback on the applied approach is given.

**UC 7.3:** A tutor wants to create feedback text so that hints and elaborated descriptions can be provided.

**UC 7.4:** A tutor wants to choose a publication timing for all kinds of feedback so that timely given hints and delayed instructions of an alternative approach of solution can be distinguished.

**UC 7.5:** A student wants to view a published sample solution so that the intended solution approach can be understood.

**UC 7.6:** A student wants to view a published personal correction so that individual mistakes are described.
5.2. Specification

**UC 7.7:** A student wants to view published feedback text so that hints can be respected for further work.

**Assessment 2.0**

Most important functional requirements for integration of external services to submission processes can be found in figure 5.24. They are explained in form of user stories below.

**Figure 5.24.** Use cases concerning integration of Assessment 2.0.

**UC 8.1:** A teacher wants to activate an external service for an assessment context so that external entities can be used for submission.

**UC 8.2:** A student wants to link the account for an external service so that the platform is able to collect the student’s external entities for integration.

**UC 8.3:** A student wants to create new entities from within the assessment platform so that content creation is done at a single place but propagated to involved services.

**UC 8.4:** A student wants to select external entities so that they are integrated to a submission.

**UC 8.5:** A tutor wants to view entities which are selected for a submission so that they can be respected for assessment.

**UC 8.6:** A tutor wants to filter entities by selected options so that authorship and relevance can be evaluated easier.
5. Requirements

**UC 8.7:** A tutor wants to assess a submission including external entities so that feedback and/or marks are attached.

**UC 8.8:** A developer wants to develop specific external service connectors so that external entities are specified and specific connections are established.

**UC 8.9:** An administrator wants to deploy external service connectors so that they can be used within the platform.

**Process Steps**

All of the above mentioned requirements are somehow related to a specific step in the assessment process. The following list contains a relation of process steps with selected User Stories. It shows that the whole assessment process should be covered by the collected requirements.

- **Assessment Design:** UC 1.1.
- **Item Construction:** UC 1.2.
- **Assessment Construction:** UC 1.2.
- **Assessment Run:** UC 1.4, UC 1.5, UC 3.8, UC 8.4.
- **Response Rating:** UC 1.6, UC 2.7, UC 3.5, UC 4.5, UC 4.6, UC 5.5, UC 5.7, UC 7.1, UC 7.2, UC 7.3, UC 8.7.
- **Decision Making:** UC 1.6, UC 2.1, UC 2.2, UC 2.7, UC 8.7.

**Hosting**

In addition to many non-functional requirements, e.g. security and privacy settings, which arise in hosting scenarios, several functional requirements have been identified as well (see figure 5.25).

**UC 9.1:** A teacher wants to manage a learning context so that a (closed) set of students is provide with common e-learning functionality.

**UC 9.2:** A teacher wants to manage the assessment scenario for a specific context so that settings for the scenario are applied for the selected scenario only.

**UC 9.3:** A teacher wants to activate an assessment tool within a learning context so that it can be used to support a certain type of assessment activities.
5.2. Specification

**Figure 5.25.** Use cases concerning hosting scenarios.

**UC 9.4:** A teacher wants to configure an assessment tool so that it is adjusted for the scenario in a specific learning context.

**UC 9.5:** A developer wants to develop assessment tools so that specific activities or management tasks are enhanced by technology.

**UC 9.6:** An administrator wants to deploy an assessment tool so that it can be integrated within several different learning contexts.

**UC 9.7:** An administrator wants to configure an assessment tool so that common settings for the hosting environment are defined.

**UC 9.8:** An administrator wants to integrate the platform with available IDM\(^1\) and ERP\(^2\) systems so that already available technical infrastructure can be reused.

**Interoperability**

Issues of interoperability, covering standards as well as service-oriented approaches, are summarized in 5.26.

**UC 10.1:** A teacher wants to export assessment related information to standard formats so that it can be reused in other platforms.

**UC 10.2:** A teacher wants to import assessment related information from standard formats so that it can transported from other platforms.

**UC 10.3:** An external system wants to provide an assessment service so that it can be consumed by the platform.

---

\(^1\)Identity Management System.

\(^2\)Enterprise Resource Planning system.
5. Requirements

**UC 10.4:** An external system wants to consume an assessment service so that functionalities of the platform can be integrated to other platforms.

**UC 10.5:** A developer wants to develop converters for standard formats so that new standards for importing and exporting can be extended.

**UC 10.6:** An administrator wants to deploy standard converters so that they can be used within the platform.

**UC 10.7:** An administrator wants to configure the service integration so that, for security and privacy reasons, only trusted systems are connected.

*Figure 5.26. Use cases concerning interoperability.*
Part III.

Realization
Chapter 6.
Conceptual Design

Closed monolithic systems are not suitable to provide a sustainable service for open assessment management, since the number of new requirements and new services rises gradually. However, the provision of a single point of service is best practice, e.g., the implementation of a Single Point of Contact (SPOC) according to the IT Infrastructure Library (ITIL) [33], because a manual composition of distributed services is exhausting. Therefore, evolution of services and technology leads to the need of evolutionarily extensional systems. Those systems have to allow flexible points of extension to avoid the development of completely new systems for only a single advancement while already available functionality has to be reimplemented.

6.1. General Design of AMSeL

A design for an open assessment management platform called AMSeL (Assessment Management Services in eLearning systems) is presented in this section. Regarding the requirements which have been defined in section 5.2, it has been attached great importance on flexibility, extensibility, and sustainability. A modular approach has been developed, which allows dynamic composition of tools and components to support different scenarios. Standard mechanisms for assessment management are provided. Furthermore, a mechanism for extension allows modular development of additional tools and components. In this way sustainability is increased, because a consolidation of different assessment tools as well as the integration of innovative ones is facilitated.

The technical design of AMSeL is based on a classic 3-tier architecture with a data layer, a logic layer, and a presentation layer. The presentation layer should allow to provide different front-ends in parallel, e.g. web pages as well as web services. Services within all layers should be scalable within a server farm to provide a centralized service for a whole university or bigger scenarios.
6. Conceptual Design

6.1.1. Modularization

Furthermore, the conceptual architecture of AMSeL is structured by three layers as well (see figure 6.1): assessment tools, organization modules, and basic components. Additionally, cloud services can be integrated on each layer.

![Conceptual Architecture of AMSeL](image)

**Figure 6.1.** Conceptual architecture of AMSeL.

**Assessment tools** provide functionality to handle an assessment process for selected types of assessment activities and methods. For instances, one tool allows creation, execution, and evaluation of e-tests. Another tool could be used to manage assignments. The set of supported activities, methods, and specific scenarios is the combination of all possibilities of all tools. Integration of further tools, newly developed within the platform as well as externally connected ones, provide the most general form of reuse and extensibility. For instances, an existing test tool can be integrated into the platform as well as a newly developed tool for the assessment of collaboration in wiki pages.
Organizational modules provide reusable functionality for a special purpose of a typical step in an assessment process. For instances, a grouping module allows the arrangement of students, which is part of the assessment design for group assessments. Additional content editors (e.g. for mathematical equations or music notation) can be used as part of item construction, response submission, or feedback. The creation of marking rubrics is part of the assessment design, whereas the marking with rubrics is part of response rating and decision making. Providing these subprocesses aims for reducing the creation process of new tools by avoiding same mechanisms time and time again. Additionally, assurance of quality as well as richness of features can be increased.

Basic components are used to manage generic processes, which are needed for assessment management but which are not directly related to it. For instances, user management is needed for authentication and authorization of participants. Content management as well as document management facilitate publication of learning material, students’ submissions etc. Communication tools, search, and others are needed in any platform which tries to support collaboration. They could be either used in parallel to an assessment tool or are directly integrated for explicit support of an assessment process.

Cloud services of various functionality are already available on the Internet. Some of them provide basic functionality, e.g. document management with YouTube or Dropbox as well as communication via Facebook. Others can be used to handle organizational tasks of assessment. Assessment tools as a whole are available as well. Therefore, the integration of external cloud services is considered on all levels of the platform.

Layer connection is the most difficult tasks within the conceptual design of AMSeL. Because all layers allow to add or interchange elements, connections between elements across layers have to be loosely coupled. Especially, an element of a lower layer does not know by which element of the higher level it is used. For instances, it has to be avoided that a grouping module has to know that it is use by an assignment management tool, since this behavior requires a modification of the grouping module for each new assessment tool. The other way round, the assessment tool should be extensible with new concepts, provided by new organizational modules, without changes to the basic tool itself. Figure 6.2 displays an approach using extensions within an assessment tool for communication across layers. These extensions are used for specific adjustments of the assessment tool needed integration of an organization tool. For instances, a group extension contains additional logic about
how group assessment in the specific tool is implemented, while the grouping process is used from the organization grouping module.

![Figure 6.2. Modular extension within an assessment tool.](image)

### 6.1.2. Event-based Service Managers

“A module is a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units. [...] The system as a whole must therefore provide a framework - an architecture - that allows for both independence of structure and integration of function.” [Baldwin and Clark, 2000]

The above mentioned connection of elements from different layers by extension has a generic perspective. In more detail, an architecture is needed to organize and integrate functionality these elements [Baldwin and Clark, 2000]. But it is difficult not to limit the extensibility within the architecture. For instance, an mechanism for extension of content editors is helpful only for that purpose. Therefore an approach for modular integration of services, which themselves can handle modules in their specific area of functionality, has been developed. That means, extension mechanisms themselves are modularized and integrable.

In the generic approach, a service manager works as a kind of broker between one or more service-providers and several service-consumers (see figure 6.3). The main differences to service-orientation in general are that the consumer does not call the service-provider directly and that events can be forwarded to a consumer. A
service manager is used for the registration of providers and consumers and serves as a *facade* for specific functionality a provider can offer. This event-based approach aims to reduce the dependencies between modules (here consumer and provider). Providers can be added, removed, or exchanged without any change to the service manager or a consumer. The main advantage of this approach is that a platform can be extended with multiple service managers of different type, which each allow the extension of modules to provide or consume services for a specific topic.

Figure 6.3. Event-based services for encapsulation of functionality.

Figure 6.4 shows an example of a *video service manager*, which can be used by a module for the extension of an assignment submission tool by videos from different services. Providers for this service could be an *internal video storage* as well as *cloud connectors* for the integration of *YouTube, MyVideo*, or other video services. Service managers to handle interchangeable grouping mechanisms or correction processes are similar.

Figure 6.4. An example service for video integration from the cloud.

### 6.1.3. The Assessment Service Manager

The registration and management of all assessment tools, which are not statically known but extended continuously, is handled by an *AssessmentManager* for each assessment context (see figure 6.5). Such a context could be an institutional course as well as every other context. Service providers are the assessment tools within this context. Assessment consumers are triggered on predefined events regarding activities, indicators, and result. Furthermore, they can pull data (e.g. a list of published activities) from the assessment manager, which aggregates the requested
information from all registered tools. These information is gathered according the structure of the basic tool as well as from all its registered extensions.

Each tool, and extension, is expected to conform to a basic structure of assessed activities. Namely, a tool contains an arbitrary number of activities, which are either published or hidden. In general, several students create common submissions to each activity. Each student is only allowed to participate in one submission per activity. Multiple feedback can be created per submission. The specific grading process is handled by each assessment tool itself, but it they are expected to provide a set of DecisionIndicators, which can be used for global decision making across activities. Results or rather values of a grading process are stored for each indicator and per student.
6.2. Assignment Management

Assignments are the most used activities in current institutional learning scenarios according to the initial survey (cf. figure 4.3). Furthermore, there exist a variety of different systems supporting different aspects of an assignment process (see section 3). Thus, an assignment management tool for AMSeL has been developed from scratch to provide a base for consolidation of existing approaches.

6.2.1. General architecture

To redevelop yet another monolithic assignment management would neither facilitate the consolidation of all possible scenarios nor allows a sustainable enhancement with new features. Therefore, the presented assignment management system has been developed according to the conceptual architecture as well as the approach of loosely coupled modules with AMSeL. Figure 6.6 displays the general architecture of this tool. Basic functionality of tutor assessed assignment management is provided by a basic tool. Advanced functionality is explicitly not contained to facilitate more possibilities for adaptation. Building of groups, online creation of domain-specific content as well as (semi-)automatic correction of students’ submission can only be attached with related extensions. The overall tool provides its own assignment service manager, which allows the extensions to get triggered by events which happen within the basic tool are other extensions. For instance, the extension for automatic assessment could be triggered when the deadline is reached, such that the processes are started for a final pre-correction. This allows to initialize the tool in one context with group assessment and automatic correction of Java source code while it is configured for peer assessed video submissions in another context.

6.2.2. The Basic Assignment Management Tool

The basic tool related common assignment management lifecycle in higher education (cf. figure 2.3) which has been explained in section 2.1.6. Therefore, the timing process of assignments and solutions schedules three to four main milestones (see 6.7): assignment creation, date of publication, an optional deadline, and the final correction. If students are allowed to submit their solutions during the submission phase, which starts with the date of assignment publication and ends either with the defined deadline or is manually determined by the student.

In addition to this classic approach, the basic assignment management tool already supports some advanced features: rubric marking, timely feedback, and solution workspaces. The resulting design of the basic tool is displayed in figure 6.8.
6. Conceptual Design

Figure 6.6. Modular architecture for assignment submission with exemplary extensions and cloud-service integration.

Figure 6.7. Basic timing process for assignments and solutions.
Assignments are the assessed activities to which students create solutions as submissions. The item construction process\(^1\) is realized by the creation of documents, i.e. several assignment documents (e.g. exercise description or solution template documents) can be associated to an assignment for publication. Solutions are constructed the same way. They are not only final submissions of students, but allow incremental work by attaching new documents or updating existing ones. This means intermediate states or rather drafts of solutions can be stored on the platform during the submission phase (*solution workspaces*). Therefore, it is possible to provide feedback to the students already during this phase by investigating drafts as well (*timely feedback*). To some extend, this facilitates the assessment of processes in addition to the final product only. Because some feedback could be a hint for further processing while other feedback would betray the final solution before the deadline is reached, the timing of feedback publication can be set to: *directly, after deadline, after correction finished, or after deadline and correction finished*. The same is true for marking results, which are special kind of feedback. They are provided according to a marking scheme which is composed by several indicators. They are named, having a rating scale, and it can be chosen if they are used for decision making.

\(^1\)The creation of items, tasks, or questions according to the *Educational Model* (cf. section 2.2.7).

---

**Figure 6.8.** Design of the assignment management base tool.
6. Conceptual Design

6.3. Group Management

An example for an organization module, which can be reused in several assessment tools, is a group management tool to provide grouping-as-a-service (cf. [P5] and [T6]). The main features of this module are:

- support of tutor processed as well as self-direct grouping (cf. section 2.2.3),
- customizable restrictions for grouping,
- collections of groups with different restrictions.

These features are achieved by the main concept of organizing groups of users within group sets to which rules can be defined and applied. Self-direct grouping is handled by use of invitations and requests. The corresponding data structures and there associations can be found in figure 6.9.

![Figure 6.9. Design of the group management module.](image)

Groups are sets of users, mostly defined within a certain context. For instances, a group might consist of all students working together on the same solution, while groups of other students working on their own solutions in parallel.

Group sets are collections of groups. That means, a group set could contain all groups working on assignments. Group sets also contain three sets of users which reflect the roles of students, tutors and managers. In order to become a member of a group assigned to a group set, a user needs to be assigned to one of these roles within that group set.

Rules are formalizations of constraints which need to be considered for group formation. Rules define which persons are allowed to join a certain group, based on variables available within the system, such as meta data. They are defined on the group set level and apply to all groups assigned to a specific group set. Rules may refer to distinct properties of groups as well as properties of a group set as a whole. An example for the former would be a restriction of group sizes. An example for the latter would be the requirement that all groups within a group set are disjoint, i.e. every user is at most member of one group within the same group set. The system
actively ensures that all rules are being enforced all of the time by monitoring and, if necessary, preventing all actions which might lead to breaking a rule, except a manager explicitly decides to override certain rules.

Invitations and requests are employed for self-directed grouping. Students who are members of a group may invite others to their group by creating an invitation, which can be accepted or rejected by the person being invited. On the other hand, students looking for joining a group may ask for group membership from a certain group by creating a request, which can be accepted or rejected by the members of that group. Managers can, however, decide to turn off self-directed grouping in order to manually assign students to groups.

By use of a service manager for grouping, according processes can be reused by different tools. Required events are group created, group updated, group deleted, use joined group, and user left group. More events, e.g. concerning group sets, are possible but not necessary.

6.4. Correction Processes

Modules for handling correction processes aim to make these processes more reusable and adaptable. By providing them as an organizational module of AMSeL, they are reusable for several tools. In addition, the processes themselves are modeled to be adaptable by composition of reusable subcomponents. Domain-specific and (semi-)automatic correction of applicative questions are addressed as well as organizational processes, here peer assessment.

(Semi-)automatic correction

To allow consolidation as well as enhancement of domain-specific correction processes, a module or rather a framework for handling those processes for various domains has been developed [P12, P13]. Major objectives of the presented module are to increase reusability, extensibility, flexibility, versatility, and maintainability. Therefore the module follows an approach that brakes up correction processes into atomic evaluation steps (cf. [P1] and [Stalljohann, 2007]). Each step evaluates a certain aspect of the submission and generates corresponding feedback. For instance, an orthography test checks for spelling mistakes and can return a string describing the mistakes, along with a grade. In addition, certain evaluation steps may generate additional resources. For instances, an evaluation step for a compilation test can return an executable program.

A new correction process for specific types of exercises is made up of evaluation steps, control structures (e.g. loops, conditions, or parallelizations), and corresponding transitions. The correction process has a defined entry point, and the outcome
of an evaluation step can cause the process to terminate. A correction of a specific submission can be regarded as a walk through the corresponding graph, i.e. a sequence of evaluation steps.

Because the processes are designed to correct solutions to open-ended tasks, which in general cannot be corrected fully automatically, each evaluation step can either be an automatic or a manual step. The latter involves a human corrector, e.g. a tutor or a peer. Like any other evaluation step, manual ones result in feedback text and a score. In comparable systems, tutors are only allowed to influence the grading after all automatic evaluation steps are finished (cf. analysis in section 3.4). In contrast, Person-in-the-Loop steps may be inserted into the correction process at any position. In effect, the tutor’s evaluation may influence the rest of the correction process.

Figure 6.10 shows an example process for the semi-automatic correction of source code. The first evaluation step is a compiler which produces feedback about compiler warnings and exceptions as well as the compiled program, if it was successful. Depending on the result, a condition evaluates whether a unit test step was executed or a tutor has to be consulted. The tutor is able to give manual feedback, scores, and correct the source code so that it is compilable afterwards.

![Figure 6.10. A graph showing an exemplary correction process for source code correction.](image)

A correction process is predestined to be a long-running process, as certain evaluation steps (such as a compilation test) require computation time. Similarly, manual steps are long-running, as it takes some time until a tutor or peer gets notified, checks the submission, and enters appropriate feedback. Thus, the engine for execution of correction processes has to allow persisting and restoring a process while waiting for the end of the current task.

**Allocation of Peers for Review**

While above mentioned correction processes are focused on evaluation of a single solution, other processes need a more global perspective. For instances, processes for generation of statistics could collect information concerning all submissions for
6.5. Integration of Cloud Services

An important feature of a platform for open assessment management is the support for integration of cloud services. These services already provide several functionality that can enhance learning and teaching processes. In some cases they are directly usable for assessment tasks (see section 2.2.5). Since the assessment management for all of these processes should be unified within AMSeL, an approach for the integration of cloud services into assessment tools has been designed.

The main abstraction for the designed model has been that a cloud service is composed by a set of entities (see figure 6.12). Entities could be of different types. For example, a video service like YouTube consists of multiple video entities, comment entities etc. Furthermore, a cloud service has registered users. Entities of the service could be related to one or more users. This relation mostly represents...
6. Conceptual Design

authorship, e.g. a user has uploaded a video or a group of users wrote an online article.

Figure 6.12. Design for integration of cloud services to assessment tools.

Entities of the cloud service are mapped to entities within AMSeL. They facilitate direct access to the cloud service through a connector, which allows CRUD-operations (Create, Read, Update, Delete) by using APIs, web services, direct database access, or other connection techniques. To use these entities in context of assessment, they could be associated to assessment activities as well as to submissions. Their type of association, that means what type of element they represent (e.g. a whole submission, a solution document, a correction document, a feedback element etc.), is defined by the assessment tool. Furthermore, the mode of integration, linked integration or synchronization has to be distinguished. For instances, videos from YouTube could be embedded with a player linking to the original source. An approach for integration of DropBox could be that files are synchronized between the local data storage of AMSeL and DropBox-storage of related users.

Security, privacy, and traceability of authorship are main challenges for the utilization of social services (cf. sections 2.2.5 and 5.1). Therefore, the access to linked entities, and with it to cloud entities, through the connector is limited by personal permissions of a user or explicit release by another user. Since a users account and credentials within AMSeL do not have to be identical with those in a specific cloud
service, a mapping of accounts for each user and service is required to enable and control access for the connector.

Another point to consider is how to select linked entities for a specific submission or activity. A user could have authored multiple entities within a cloud service, but only some of them relate to a certain element in AMSeL. If the authorship of an entity is not required for some reasons, e.g. students are asked to collect open accessible information about a certain topic or a teacher wants to integrate a freely available script, a selection by explicit choice or filtering have to be possible.

6.6. Assessment of collaborative wiki contributions

In addition to an assignment management system, which handles a well defined submission process itself, a tool for the assessment of collaborative contributions in a wiki system has been developed (see [P5] and [T6]). The wiki system itself is categorized as a basic component within the platform of AMSeL. Therefore, the wiki system has to be incorporated into the assessment tool to extend its functionality with assessment related processes.

According to the analyzed scenarios (see section 4.3), three steps of configuration for different scenarios have been identified to build a related model:

A: Selection of a scenario for using the wiki system with $N$ students.
   • Students work on their own wiki article(s) individually.
   • All students work on all pages collaboratively.
   • Students work with members of their group on some shared wiki article(s) collaboratively.
   • Each student contributes to other articles then his/her group members, but with students from other groups collaboratively.

B: Decision which of and how the elements are to be assessed.
   • Marking of individual contributions per article.
   • Marking of group contributions per article.
   • Total rating of individual contributions across all articles.
   • Total rating of group contributions across all articles.

C: Differentiation by whom the assessment is to be done.
   • A teacher or tutor (Tutor Assessment).
   • Some peers (Peer Assessment).
6. Conceptual Design

- The student or group itself (Self-Assessment).
- The system (Automatic Assessment).

A collaborative assessment scenario can be constructed by diverse combinations of these steps. For instance, the first one of the analyzed scenarios (see section 4.3) can be built through the combination of A.2, B.1, and C.1. Figure 6.13 shows a model for this scenario. A Student works on several WikiArticles and writes several WikiArticleVersions. The assessment is done by a Tutor who creates a Marking for the combination of a Student and a WikiArticle. The Marking is based on a MarkingScheme just like in the assignment management tool.

![Figure 6.13. Static model for the example scenario with wiki contributions.](image)

To support different scenarios that can be described by a combination of the above defined steps, a general model has been developed, which can be used for the implementation of an appropriate system (see figure 2).

The general functionality of wiki pages is presented by WikiArticles, its WikiArticleVersions, and their association to a Student. To realize different scenarios in step A, Students can be organized in Groups. The formation of groups can be managed by above presented grouping module. The different assessable elements for a scenario are represented by an Assessee, which is the object to be assessed. For each scenario, there exists a specific model which references the involved elements in the assessment:

- GroupPerArticle: The combination of a Group and a WikiArticle.
- StudentPerArticle: The combination of a Student and a WikiArticle.
- GroupOverall: A Group.
- StudentOverall: A Student.
Figure 6.14. Design of the assessment tool for wiki contributions.
6. Conceptual Design

In general, the Marking of an Assessee can be done by an Assessor. In the case of TutorMarking, it is done by a Tutor. PeerMarking can be used by several students. Both marking alternatives are based on a specific MarkingScheme. Self-Assessment (C.3) is not part of the model in figure 6.14 due to readability issues. It can be added in the same way the peer marking is modeled.

The integration of automatic rating processes, i.e. an automatic assessor, is possible as well. Therefore, an extension can be used to calculate certain results which can be written to the marking data structure in relation to the marking scheme.

6.7. Result and Criteria Management

Since formative assessment aims to support learning processes continuously, an assessment process utilizes various assessment activities of different types in sequence as well as parallel. Within AMSeL these activities could be scattered across several assessment tools. Therefore, global tasks like assessment planing and criteria management are not covered by a single assessment tool.

In this section a design for gradebook module is presented, which illustrates the integration of a global assessment tool by consuming information of the assessment service manager. Main objectives of the gradebook are to record students results and to allow definition of complex course criteria. Students’ personal statuses according to the defined criteria have to be calculated, such that they can follow their own development of marks successively.

6.7.1. Criteria graphs

Management of students results and calculation of outcomes related to defined assessment criteria is mostly done with spreadsheet software or handwritten notes (see section 4.1). However, requirements like personal access of individual results by students, transparent publication of criteria descriptions, or reuse of already defined criteria demand for a new approach. The aim is to facilitate the definition of models that are not strictly bundled with a result table which is individual for each course. Therefore three types of elements can be differentiated in this approach: indicators, categories, and rules (see figure 6.15).

An indicator (see figure 6.15a) describes a task, skill, or competency a student demonstrates by performing a specific assessment task. This can be a single assignment, the active participation in a class or a presentation of a homework etc. Results for an indicator are typed on a rating scale, e.g. score, bool, grade, etc. In cases of scores the scale has to be limited by a maximum value.
6.7. Result and Criteria Management

Categories (see figure 6.15b) can be used for the hierarchical organization of indicators and other categories. This is similar to other approaches like in Moodle. Categories can contain several indicators of different types.

A rule (see figure 6.15c) describes a calculation based on outcomes of indicators or other rules. Cyclic references are not allowed. To select the operands of the rule, indicators can be referenced directly or indirectly through referencing a category. The type of a rule defines how the calculation is done and can be e.g. SUM for a summation or CONDITION for a specification of a condition over the referenced outcomes. Other more complex rule types can be composed. Some rules can reference multiple indicators, while others may only reference a single one.

An exemplary graph representation of an assessment criterion, which combines several indicators by rules, is displayed in figure 6.16. Decisions are made based on a set of assignments that are contained in a category named Assignments. It is assumed that they are rated with scores. The presentation in a tutorial is modeled by an indicator named # Presentations. Students’ results in a Mock Exam, rated with a German grade, are integrated to the criterion as well.

In the example, six rules are used to decide whether admission to the final exam, which is defined in the examination regulations, is granted to a student or not. The overall result is calculated by a logical AND that evaluates to true iff all referenced elements are true. The referenced rule to decide, if a student has Enough Presentations is a parametrized CONDITION related to # Presentations. Parameters are, that the rule evaluates to true, if the element has a score that is greater than or equal to 2. Another CONDITION decides whether the Mock Exam is Passed or not. In this case it is passed, if the grade is 4 or better. The calculation of a sub-criterion regarding assignments, is a combination of three rules. A CONDITION that decides on the percentage RATE of the overall score for assignments, which is summed by the according rule based on all assignments within the related category.

Further criteria graphs for the scenarios surveyed in section 4.2 can be found in appendix B. They show that this approach allows to model very different and complex scenarios. Because indicators and categories can be referenced by several rules in parallel, the whole criterion is more expressive than tree-based models, which are used in Moodle for example.
6. Conceptual Design

6.7.2. Tool Design

The *gradebook* module in *AMSel* has been designed for realization of the presented criterion graph approach. It acts as a consumer of the *AssessmentManager*, to recognize creation and changes of activities, indicators, and results within all connected assessment tools (see figure 6.17). The structure of a criterion graph is presented by elements, whose derivations are rules, indicators, and categories. All elements can be organized in hierarchical categories. Each indicator is directly mapped to a decision indicator that is used within an assessment tool. Rules are used to calculate (sub-)criteria and can have various implementations. Individual results for each student are related to the defined elements and contain values from indicators as well as calculated ones from rules.

The way the tool is designed as a separate module, which integrates with the assessment management service, it allows flexible integration of indicators and results from several tools, even newly integrated ones. Therefore, the tool is suitable for the overall planning of an assessment scenario.

![Figure 6.16. An example for a criteria graph.](image)
6.8. Discussion

The system AMSeL has been designed as a modularized platform which is composed of three conceptual layers and a mechanism for integration of cloud services. An approach of service managers tries to facilitate a loosely coupled connection of modules across layers. Flexibility as well as extensibility of the overall design is demonstrated by exemplary modules for each layer. Additional design possibilities for adaption and extension within such modules has been explained as well.

Although the technical as well as the conceptual architecture of AMSeL seems to be suitable for evolutionary but modularized development processes, there still challenges concerning the implementation of the defined approaches. Especially, because the platform should be based on web technologies, questions about how to deploy new components have to be answered. Furthermore, the realization of web-based event mechanisms for service managers have to be addressed. Finally, the possibility for modular integration of new interactions to the web-based front-end have to be considered.
Chapter 7.

Portal Technologies

7.1. Definition and Classification

“A portal is a web application, which offers an entry point to a business, a specific topic, or a user’s workplace. Therefore, it combines several blocks of information and functionality.” [Zörner, 2006]

Nowadays, a huge amount of information and services are available through the internet. These services are distributed across billions of different websites. Therefore, acquisition of adequate information is challenging. There are mainly two approaches to ease information access over the internet. On the one hand online search engines (e.g. Google[34] or Bing[35]) allow to filter information by user defined keywords. Thus, users have to know what they are searching for. On the other hand portal applications offer a predefined selection of information for a specific purpose or topic. Different categories of portals can be distinguished (see figure 7.1). While web portals [36] offer an entry point to the whole internet, enterprise portals [37] aim to support different target audiences in an intranet of an enterprise [Schulz, 2007]. Subcategories are discussed in the next sections in more detail.

7.1.1. Web Portals

Like mentioned above, web portals are mainly web applications which provide a selection of information sources. They facilitate browsing and accessing theses sources from a single entry point. Web portals can be subcategorized regarding their content, i.e. how the selection of linked sources is made: horizontal or vertical.

Horizontal portals combine information about a wide range of different topics. Mostly, latest news from different areas (e.g. politics, life style, and sports) are presented on the start page. Further information is divided into areas of interest and can be browsed by users. Personalizable horizontal portals allow users to customize
a portal for their needs. For instance, they can choose which information sources are selected (e.g. the local weather and selected stock prices). The portal can learn from the users behavior to show personalized advertisement or to recommend information and services. An example for such a personalizable web portal is *iGoogle* [38] (see figure 7.2a).

Figure 7.2b shows a web application with information and services about cooking. This is an example for a *vertical portal*. A portal of such kind is mostly a service for a specific audience. It provides deep information about a specific topic or area of interest. *Web shops* are vertical portals as well, because they are providing specific services concerning shopping.

### 7.1.2. Enterprise Portals

*Enterprise portals*, also called *intranet portals* [40], build a second category of portals. These are web based applications that are focused on information and services in context of a specific enterprise. Thus, from a global perspective, they are a special kind of vertical portal. Related to the local intranet of an enterprise, they have a more horizontal character. A more suitable characteristic for distinction of enterprise portals is the target audience. In this way, there are four subcategories: *employee portals*, *business portals*, *supplier portals*, and *consumer portals* [Vlachakis et al., 2005].

*Employee portals* are used as a central entry point to services and systems, which an employee needs for his or her everyday work. *Business portals* handle processes with business partners, like marketing or support delivery. *Supplier portals* support the management of all processes concerning suppliers, e.g. tendering procedures or billing. *Consumer Portals* provide information and services for end costumers. For
7.1. Definition and Classification

(a) A personalized horizontal portal page [38]
(b) A vertical portal about cooking [39]

Figure 7.2. Examples of web portals: iGoogle and Cooks United

instance, a web shop enables customers to browse and order from the portfolio of products and services, as well as paying for them.

7.1.3. Components and Functionalities

Portals, especially enterprise portals, offer a lot more than just a set of links to several sources of information. Mostly, they are highly modular and extensible platforms, which provide several services. Fraunhofer IAO [41] developed a reference architecture for portal software called Portal Analysis and Design Method (PADEM 2.0) (see figure 7.3). Most modern portal systems contain standard components, which are described in the 3-tier reference architecture (backend, logic, presentation).

The backend is built by an internal backend system (e.g. system databases) as well as external data sources and services. The logic layer is based on an application server which is used to provide the services via http or webdav, thus they can be accessed with a webbrowser. Creation, controlling, and management of these services is done by the portal software. Standard services as well as custom created services are strongly integrated with application modules by a portal API. In the following, an overview of standard services and application modules is given, which is strongly based on [Vlachakis et al., 2005].

Authentication and Authorization Management is needed to facilitate a personalized preparation of information. For this reason different authentication mechanisms (e.g. form based authentication or basic authentication) as well as the man-
7. Portal Technologies

Figure 7.3. Fraunhofer PADEM portal software reference architecture 2.0 [Vlachakis et al., 2005]
agement of users (e.g. via LDAP) are basic portal services. A Single-Sign On service allows the access of different internal and external services with a singular login process. Additionally, the management of users within groups or roles as well as the assignment of different rights for information and service objects are covered.

**Content Management** facilitates its users to manage and organize content and metadata. Especially, it is used for publication of hypertext (HTML) with localization in different languages. Related processes like content approval is covered as well.

**Document Management** focuses content in files of various formats. Mostly, the file handling is more advanced than a simple file space. For instance, version management, check-in and check-out, or information rights management is provided. Additionally, documents can be enriched with additional metadata and alternative structuring methods like keywords, tags and ratings.

**Communication and Collaboration** are central tasks in every team project, whether in professional or educational context. Technological support of communication is realized textually (e.g. text chat or online forum), auditorily (e.g. voice chat), or visually (e.g. video chat). More constructive processes of teams, like text creation, are supported by online collaboration tools (e.g. wiki pages). Group calendars, common contact management, and other groupware tools facilitate online management of organizational processes.

**Structure and Layout Management** handle templating, navigation, and design mechanisms for processing of application pages. These mechanisms allow customized and personalized provision of content as well as processes. A central concept in this context is the modular composition of application pages by containers, which provide self-contained snippets of information and functionality. Within Java-portals these containers are called *portlets* [42]. In 2003 several software enterprises developed the JSR-168 [43] standard as a programming model for portlets [Kussmaul, 2005]. A revised specification has been published ith JSR-286 [44] in 2008. Other terms of similar information containers are for instance, *webparts* [Shepherd, 2008] in Microsoft ASP.NET applications, *widgets*, or gadgets. A detailed explanation of this concept using the example of webparts is given in section 7.3. Additionally, the OASIS network protocol *Web Services for Remote Portlets (WSRP)* and its following version WSRP 2.0 [45] “define a web service interface for interacting with presentation-oriented web services” [46]. These services can be integrated into application pages with or without the use of portlets, webparts, widgets, or gadgets.
Workflow Management facilitates definition and execution of business processes. Temporal supervision of content, e.g. publication and retention planning for official documents, as well as organizational processes, e.g. multi-tiered approval of press releases, are supported. Mostly, different user roles are involved. Different channels for communication and service integration, e.g. e-mail or web services, can be used.

Search Searching for a specific piece of information is a common task on the internet as well as within an intranet infrastructure. Big challenges for search engines within a portal are to respect individual permissions of the searching user, to cover different structuring concepts like tags and categories as well as full text search, and to include search results from integrated external systems.

API and Integration Functionalities included in delivery of standard portal software still facilitate the support of various business processes, especially through customization by administrators as well as end users. Furthermore, a big strength of portal software is their extensibility, that allows the integration of custom applications, which can use the above mentioned standard components through a special API. Special deployment techniques for those applications allow the development of custom applications as well as the use of third party components.

7.1.4. Products

Different portal software products, which offer the above mentioned functionalities, are available, commercial as well as open-source. A study from Gartner [47] compares horizontal portals of 10 different vendors to range them in a magic-quadrant (see figure 7.4). Firstly, they are categorized by their ability to execute, i.e. coverage of different scenarios, vendor viability, and expanding market presence. Another category is the completeness of vision, i.e. comprehension of customer needs as well as innovative approaches and technologies. Thus, the products can be classified as leaders, challengers, visionaries, and niche players. The following is a short overview of the 5 leader products.

Liferay Portal is an open-source web platform based on Java technology [48]. It runs in an application server (e.g. GlassFish or JBoss) with a servlet container (e.g. Apache Tomcat) and a database (e.g. MySQL). Its requirements are flexible, such that it can be deployed to very different environments with different operating systems (linux, unix or windows) as well as in cloud environments (e.g. VMWare Server or Amazon EC2). Above mentioned standard functionalities are provided. Custom applications can be build based on an SDK, especially with portlets. Web-based development can be done with Javascript, Ruby, PHP, and Python.
7.1. Definition and Classification

Figure 7.4. Magic Quadrant for Horizontal Portals [47]

**SAP NetWeaver Portal** is a component of the SAP NetWeaver product line [49]. It is based on Java/J2EE and ABAP and runs in the *SAP NetWeaver Application Server*. The whole set of portal functionalities is reached by combination with other SAP products, *SAP NetWeaver Identity Management* or *SAP NetWeaver Enterprise Search* for instance. Development processes are supported with special tools like *SAP NetWeaver Developer Studio*.

**Oracle WebCenter Portal** is another Java/J2EE-based portal technology [50]. Together with *WebCenter Spaces* and *Composite Applications* it builds the *Oracle WebCenter Suite* for enterprise content management (ECM), business process management (BPM), and social computing. Development of custom applications can be done with the *Oracle JDeveloper* tool.

**IBM WebSphere Portal Server** is JAVA-based enterprise portal within IBMs WebSphere product family [51]. It integrates with IBM Lotus products and supports latest standards like *Java Portlet Definition Standard* (JSR 168 and JSR 286) as well as *Web Services for Remote Portlets* (WSRP 1.0 and 2.0). Five editions with different sets of functionality are available: *WebSphere Portal Server*, *WebSphere Portal Enable*, *WebSphere Portal Enable for z/OS*, *WebSphere Portal Extend*, and *WebSphere Portal Express*. An application server, and LDAP directory service, a database server, website templates, as well as development tools are contained in all editions, except *WebSphere Portal Enable for z/OS*.
Microsoft SharePoint Server is the only portal software based on .NET technologies in this category [52]. Using ASP.NET this modular system is hosted in the Microsoft Information Server (IIS) on a windows operating system. The cloud-based variant SharePoint Online is provided as part of Office 365 [53], which a combination of tools for online productivity and communication. A detailed description of standard functionalities as well as development and deployment mechanisms for SharePoint is given in section 7.3.

7.2. The Learning and Teaching Portal L²P

Although intranet portals are developed to address special needs of enterprises, they are applicable for higher education institutes (HEI) as well. A study from the Northumbria University in UK shows that “most UK HEIs are using SharePoint to some extent. In the telephone survey of 40 UK HEIs, 78% said that they were making some use of SharePoint” [Lappin and McLeod, 2010]. SharePoint, as an enterprise portal, is mostly used to support administrative tasks but its usage as a virtual learning environment is increasingly growing. A survey from [Browne et al., 2010] shows that the use of Sharepoint as a virtual learning environment at UK HEIs has increased from 0% in 2008 to 13% in 2010. While portal technologies itself provide a lot of functionalities which are applicable by means of learning management systems (e.g. document management, communication and collaboration), specific enhancement of such systems for educational scenarios are possible as well.

For instances, the tailor-made, university-wide learning and teaching portal L²P [54] is based on SharePoint Server 2007. It has been carried out as a joint project of the Center for Innovative Learning Technologies (CiL) and the Center for Computing and Communication (CCC). After gathering some first experiences in pilot installations, L²P has been launched in summer term 2007. Since that date it is available for all students and teachers at RWTH Aachen University as a central service for blended learning [Schroeder et al., 2008a, Schroeder et al., 2008b]. It is fully integrated into the university’s infrastructure. It is coupled with the campus management system CAMPUS, which was already established for administrative tasks, such as course management, room management, or examination management [Bischof et al., 2005, Gebhardt and Bischof, 2004]. The authentication process is handled with an Active Directory (AD) in combination with the Tivoli Identity Management System TIM, which is centrally provide for the university. Figure 7.5 shows the general architecture and its association to CAMPUS and TIM.

Within the CAMPUS system, which usage is mandatory, a lecturer can create a so called virtual course room for each of his lectures by a single click. This course room is structured by domains, which each provide a set of functionality to support and enhance the learning scenario of a course. A lecturer is able to publish
7.2. The Learning and Teaching Portal L²P

course material (e.g. slides, scripts, or video recordings), compose announcements, send e-mails, conduct surveys etc. Furthermore, it is possible for students to take part actively by using online discussions with peers, participate in creating wiki pages collaboratively, or share documents. Beside standard functionality, custom components and applications have been developed to add more services which are related to learning and teaching. For instances, a module which allows recommendation of further reading including digitalization workflows for publications involving the university library has been integrated [Stalljohann and Rohde, 2008]). Provided content as well as services are purposively restricted to registered participants of the lecture. Some parts can be configured for open access.

Figure 7.5. Generic architecture of L²P.

Figure 7.6. Impression of a L²P-Course-Room
7. Portal Technologies

Figure 7.6 displays an example of an L²P-CourseRoom. The different provided domains can be reached by the navigation area on the left hand side. A breadcrumb navigation facilitates an additional way for navigation within the current course as well as direct access to a personalized list of enrolled courses. An overview page for each domain (in this case the information domain) contains widgets for each type of functionality within the selected domain. For instances, announcements, e-mails, surveys, and hyperlinks can be managed directly from the overview page of the information domain.

Each course is constructed based on a template, which describes the structure of a virtual course room as a hierarchical combination of domains and their functionalities. Therefore, lecturers as well as students find information and services at the same place in each of their virtual course rooms.

The acceptance of L²P is very high and the number of supported lectures rose continuously (see figure 7.7a). The portal left its pilot phase as started with 270 virtual course rooms in summer term 2007. In winter term 2010/11 alone, there have been about 2,200 lectures from eight different faculties which have been supported by a virtual course room (see figure 7.7b). Overall, the portal has reached a number of about 11,000 virtual course rooms during the eight terms from 2007 to 2010/11. Approximately 25,000 enrollments have been registered per term, what means that every user has been enrolled in six course rooms on average. In winter term 20011/12 more than 96% of all students have been enrolled to at least one virtual course room in that term. Currently the system is used by about 15,000 users per day. About 2.5 million pages and 750 GB of data are transferred per week.

![Figure 7.7. Growing usage of L²P.](image)

Initially, assessment processes have not been supported within L²P. Therefore, an approach for integration of an assessment tool has been searched. The *SharePoint Learning Kit (SLK)* [55] has been investigated and was evaluated to not suitable for the scenarios at a German university, since each assessment activity has to be
assigned to each student individually. The next approach was to integrate the e-test functionality of the learning management system Clix as an additional module. Because of several problems with that platform, it has been replaced by the e-test tool of Moodle, which is currently in productive usage. Additional modules for assessment management have been deployed as pilot installations of modules from the thesis at hand. Thus, it has to be kept in mind that the approaches presented in the thesis at hand already influenced the positive development and acceptance of L2P.

Other approaches to build a learning environment based on SharePoint are for example the project ZePeLin [Lämmle, 2009] as well as the commercial product SharePointLMS [56].

7.3. Microsoft SharePoint

The specific architecture, components, and especially development and deployment mechanisms of a selected enterprise portal is presented in this section exemplarily. Because of its possibilities compared to other portal technologies (cf. section 7.1) as well as its increasing usage at higher education institutes (see section 7.2), Microsoft SharePoint has been chosen for this purpose.

The latest version of this product is Microsoft SharePoint 2010 [52]. Its main capabilities are clustered into six categories:

- **Sites**
  A single infrastructure to manage all websites.

- **Composites**
  Components and tools (e.g. Access Services or Business Connectivity Services) to build no-code business solutions.

- **Insights**
  Reporting and business intelligence tools.

- **Communities**
  Social media and collaboration tools (e.g. wikis, profiles, and tagging).

- **Content**
  Document and content management tools (e.g. document types, version management, or retention policies).

- **Search**
  Customizable search engines and metadata filters.
Three editions are available: *SharePoint Foundation 2010, SharePoint Server 2010 Standard,* and *SharePoint Server 2010 Enterprise.* Main differences are the contained components within each of the above mentioned categories and related licenses (see figure 7.8). *SharePoint Foundation 2010* can be installed on a Windows server without additional payment. It contains basic components of all capability categories. Both server products technically identical. Which tools and components are allowed to use depends on the purchased license, standard or enterprise.

![Figure 7.8. SharePoint 2010 Platform Stack [57].](image)

Different to a standard software that is installed and afterwards applied for a restricted set of use cases, Microsoft SharePoint is more a construction kit to build highly customized portals to various needs. Therefore, it provides its own architecture for building enterprise portals in conjunction with an extended object model for development tasks.
7.3. Microsoft SharePoint

7.3.1. Server Architecture

SharePoint employs a hierarchical architecture hosted as a farm on several physical servers which provide different services (see figure 7.9). These services include Windows services (e.g. database service, search, timer service etc.) as well as web services. Each service can be run in multiple configured farmScoped instantiations (CFSIs) on different servers. At least one web service is used to handle web applications, which are hosted in an Internet Information Services (IIS) website and provide access to credentials and other farm-wide application settings. Within these web applications there are multiple site collections, which are used to build independent workspaces.

![Figure 7.9. SharePoint 2010 Server Architecture](image)

7.3.2. Site Architecture

Each site collection itself is a tree of subsites, which always contains a top level site as its root node (see 7.10). The subsites, as well as the root site, represent a set of portal pages for presentation of content. A site collection is a self-contained environment for provision of components and tools to a specific target audience. Subsites are able share several information (e.g. groups, navigation, templates etc.) provided by a parent site or the root site, which represents global information of the site collection itself.

Content and document management is handled by lists, which provide an abstracted layer of SQL tables, including mechanisms for versioning, check-in/out, controls for the user interface, presentation templates etc. Items of these lists contain
7. Portal Technologies

Figure 7.10. SharePoint 2010 Site Architecture [58].

metadata and, in case of document libraries, each item is a file. *Fields*, corresponding to columns in a SQL table, represent item properties (e.g. name, data type, rendering etc.). They are well typed so that specific data validation as well as specific presentation is possible. Lists as well as list items can be created, read, updated, and deleted (CRUD) by end users via the web front-end. Beside the differentiation of lists and document libraries, lists can have different types or rather templates. For instances, list templates for announcements, contacts as well as wiki pages are provided as standard templates.

To handle different types of items within a list, items can have different *content types*, which define the properties (fields) of an item. The content types are organized with help of inheritance. For instance, a list for management of publications contains fields for *title, author, pages, publisher, journal, and ISBN*. A content type for an abstract publication contains the first four fields. A content type for an article inherits this fields and adds the journal field. Another content type for a book inherits from the abstract content type as well and includes the ISBN field additionally. Whenever an item is created or updated, the related formula for the end user is dynamically created and contains only fields, which are relevant regarding the content type of the corresponding item.

**Roles and permissions** are needed to control access to stored documents and content within a site collection, because data could be personal or sensitive by any reason. The *authentication* of users can be handled in different ways, like with an active directory, forms authentication using a custom database, custom providers,
or federations. The authorization is managed within a site collection via assignment of role definitions on four levels: the entire site collection, a subsite, a list, or a single list item. The concrete permissions are expressed by pre-defined base permissions (e.g. View Listitems, Add Listitems, Create Groups, or Manage Subwebs) which are combined by a named role definition. For instance, the role definition Contributor contains the base permission Add Listitems, but not the permission Manage Permissions. The assignment of permission can be done for single users as well as for groups, which are named sets of users within a site collection.

WebParts are components to handle presentation and user interaction in a modular way, such that several of them can be combined to personalize portal pages. Each portal page contains a set of zones, which can be dynamically filled with instances of webparts by an end user via the web browser. A connection between webparts can be used to provide or consume parameters. For instance, a weather webpart could consume the name of city as a filter from another webpart which allows the selection of a city on a map. SharePoint provides a set of default webparts (e.g. presenting list data, images, filters etc.), which can be chosen from a catalog to add them to a portal page within a site. Adding custom webparts to this catalog is possible as well (see section 7.3.3). Additionally, the selection of available webparts in the specific catalog of a site collection is controlled by features.

Features are mechanisms to activate and deactivate certain functionality in four different scopes: Farm, Web Application, Site Collection, or Subsite. For instances, features are used to make templates for lists, content types, webparts as well as many other components available within the selected scope. SharePoint itself provides multiple standard features to modularize functionality. Thus, additional components, which are only available in the enterprise edition are deactivated in the standard edition by default. The most important aspect of features are that they also used to install and activate custom developed components for extension and enhancement of SharePoints default capabilities.

7.3.3. Customization, Development and Deployment

As already mentioned in section 7.1, SharePoint is based on Microsoft’s .Net-Framework including ASP.NET for hosting web pages within the Internet Information Server (IIS). Therefore the development is supported by well established programing languages (C#, Visual Basic etc.) and development tools (e.g. Visual Studio). Additionally, advanced development techniques for web-based applications which are already provided by ASP.NET can be applied. For instances, abstraction

---

1Names for webparts in other systems are portlets, gadgets, or widgets (cf. section 7.1).
of controls, separation of markup and code, templating with MasterPages, advanced state handling (e.g. sessions, cookies, view states, server caching etc.), dynamic data binding, an other mechanisms are available. Furthermore, SharePoint provides its own mechanisms for extending the system with custom components as well as its own object model to access available components programmatically (cf. [Pialorsi, 2011]). More specifically, three possible ways to customize the platform can be distinguished:

- **Adjustment** of various components can be realized using the web front-end, especially with tools from the composites category (e.g. creation of lists, integration of well form external services etc.).

- **Advanced Customization** can be done by power users or web designers via the free tool SharePoint Designer [59], which allows to make modifications on styles and layouts using a designer tool or markup and script (e.g. HTML, XML, CSS, XSLT, or JavaScript). Advanced administrative or constructive tasks, for instance the composition of workflows, are possible via this tool as well.

- **Development** of specific business logic in form of custom webparts, pages, controls, and other types is only possible via implementation using the .Net Framework with Visual Studio.

Some possibilities of advanced development for extending the platform with custom components and tools addressing specific business logic is outlined in the following.

**Custom Field Types** allow the definition of new data types as well as their validation and presentation for usage as a field in a list. A new field type is presented by custom field class inheriting SPField. It defines data type, validation, and other settings. Registration of this field class is done by a deployment file with a name conform to the pattern fldtypes_<name>.xml. Further possibilities are custom rendering via XSLT and custom rendering controls, custom value classes, and property editing controls. A deployed field type is available for the whole farm and can not be directly restricted to a specific site collection.

**Application Pages** are a kind of service pages which provide specific functionality for all site collections, because they are accessible via the virtual path <sitepath>/_layouts/<pagefile>. After these pages are published to the central Layouts folder in the physical SharePoint folder on all servers, it can be used within all site collections. Additional restrictions have to be implemented via code.

**Features** are the main mechanisms to activate and deactivate custom components within a selected scope (see above). They are represented by an XML-file name
feature.xml placed in a folder with a unique name. This file defines meta data like name or version of a feature and links other manifest files, which are used to define the elements, which are (de-)activated by this feature. Possible elements which are described completely declarative are for instances: field definitions, content types, list templates, list instances, or site definitions. Another important aspect is the possibility to attach custom actions to the user interface, by defining the type (e.g. button or link), its style (image, caption etc.), and the corresponding action (execution of server code, client script, or a link). Elements that are implemented by code but registered with a manifest file are for example: webparts, delegate controls, event receivers, workflow definitions, external content types.

WebParts are implemented as .NET libraries or rather assemblies, such that they are pre-compiled for execution. A new webpart has to inherit the ASP.NET WebPart class. Presentation, interaction, and business logic are encapsulated within this class. Deployed with a feature, it can be added to the webpart catalog of a certain site collection by activating the feature.

Delegate controls are used to deploy and place custom developed server controls to predefined place holders in available web pages. For instances, a portal page contains a place holder for a search field. By default this place holder is filled with a basic control, which allows rudimentary search requests. With the activation of a feature with a related delegate control, the control within the placeholder can be replaced with a more advanced one.

Event receiver are classes which can be bound to a specific content element, e.g. a list item or a web, to execute custom logic, when a pre-defined event has been triggered. SharePoint provides a static set of events, that are triggered by the SharePoint API. For instances, a custom event handler, implemented in form of a .Net class, can be registered to get trigger whenever a list item within a specific list has been modified. Custom code is then executed as pre- or post-processing of the event. In this way the deletion of items which are younger than five month can be avoided or an e-mail can be send to the creator of a file whenever it has been modified.

Workflows are used to specify modular processes, which can be executed for an item or within a subsite [Nassiri, 2007]. SharePoint provides its own implementation of a workflow runtime based on Windows Workflow Foundation (WF) [Scribner, 2007] which is part of Microsoft’s .Net Framework since .Net 3.0. It allows two types of workflows: sequential workflows and state machine workflows (see Figure 7.11). Sequential workflows are composed of different activities which are
arranged on paths from the start to the end. State machine workflows consist of several states with transitions between them, which are triggered by events. Within each state, sequential workflows are used for state initialization and when leaving a state. A workflow is either a programmed class or a piece of XML to define the structure of activities. A workflow activity is a class containing the business logic. The implementation of custom activities as well as the definition of workflows is supported by a visual workflow designer within Visual Studio.

However, the WF is not an out-of-the-box workflow runtime, but provides an API to build a runtime. Several providers for specific tasks like persistence have to be developed for a custom runtime instance. The SharePoint workflow runtime already contains all required providers, such that it is usable out-of-the-box. An additional SharePoint Workflow API is built on top of the WF. It defines a three step deployment process for workflows within SharePoint (see figure 7.12). These steps are Template Definition, Association, and Initiation. Firstly, the structure of a workflow, i.e. which activities are used and how are they are combined, is defined as a template, which is deployed to the server within a feature. Afterwards, the template can be activated for modular usage in different locations. The second step is the association of a workflow to a specific context, i.e. to a list of items, with a set a parameters which are applied to the template. The last step is the

Figure 7.11. Types of workflows in Windows Workflow Foundation and SharePoint.
7.3. Microsoft SharePoint

initialization of the workflow for a selected item. The workflow is started using
association parameters as well as additional initiation parameters.

![SharePoint workflow deployment process](image)

Figure 7.12. SharePoint workflow deployment process.

Further benefits of this platform are two additional tools for the creation of work-
flows. *SharePoint Designer* provides a wizard-like interface for the creation of work-
flows without programming skills needed. An overall graphical definition can be
done within the diagram suite *Visio*. Different audiences are addressed by different
tools: non-technical planner (*Visio*), technical designer (*SharePoint Designer*), and
developers (*Visual Studio*).

**Business Connectivity Services (BCS)** are a component for integration of exter-
nal data sources. It allows to define *external lists* or rather *external content types*
which facilitate presentation and interaction with that data just like with data in
ordinary lists [60]. Data of external systems can be consumed via web services,
direct access to a database, or a custom connector to arbitrary sources. To allow
individualized and routed authentication processes for the integrated system, a *Se-
cure Store Service* can be used to store user credentials in an encrypted database
[61]. External content types for plain connection of web services or databases can
be defined via SharePoint Designer. Custom connectors have to be developed with
Visual Studio. Adding a new external list based on an available external content
type can be directly done via web browser.

**Excel Services** facilitate a service-oriented experience of spreadsheet calculation
[62]. Core element is the *Excel Calculation Service*, which allows server-side cal-
culation of Excel-spreadsheets that are published to a trusted place. *User-defined
functions (UDFs)* can be implemented and deployed to the server and afterwards
be used as an additional type of formula within the spreadsheet. Data retrieval
as well as dynamic modification of the spreadsheet content can be done in four
ways: *Excel Web Access* (WebParts for integration in portal pages), *Excel Web Ser-
vices* (SOAP), Representational State Transfer (*REST* services), and *ECMAScript*
(*JavaScript*, *JScript*).

**Other** available services, depending on the SharePoint edition (cf. figure 7.8), are for
example: *Visio Services*, *Access Services*, or *Word Automation Service*. An
integrated *Service Application Framework* allows to create custom services which can be hosted within a farm as well. Further advanced features are for instances *multi tenancy support* and *mobile webpart adapters*.

**Solutions** are single file packages, which can contain all necessary files (including features, element descriptions, application pages, etc.) for a custom application or extension of the portal environment. These packages can be deployed to every SharePoint farm by a standardized process. Thus, custom solutions and products can be shared and reused on multiple farms. In addition to these *farm solutions*, with SharePoint 2010 the idea of *sand-boxed solution* has been introduced. These solution can contain a subset of elements, which can be published to a restricted environment within a single site collection. Implemented elements within this type of solution are limited to a subset of the SharePoint-API. For reasons of security as well as performance they are running in a different process, which can be regulated by the administrator separately.

**7.4. Conclusions**

The investigation of portal technologies in general, as well as SharePoint and L²P in particular, show that they provide a well suited environment for a centrally hosted service. Combination of different functionality, ranging from content delivery to communication and collaboration tools, as well as integration of external services are supported. The combination on several different tools for widget management, workflow management, document management etc. and related compatibility issues can be avoided with this approach of an integrated but extensible system.

Therefore portal technologies, more specifically SharePoint, have been chosen as the development platform for *AMSeL*. Additional advantages in the context of open assessment management are, that this platform is already established in enterprises and are increasingly used in higher education as well. The amount of integrated functionalities and especially the possibilities for customization have been crucial factors as well. Furthermore, L²P provides a test bed to deliver and evaluate the platform in real scenarios.
Chapter 8.

Portal-based Implementation

An implementation of AMSeL (Assessment Management Services in eLearning systems) based on SharePoint (see section 7.3) is presented in this chapter. The development is based on the conceptual design presented in chapter 6. Since AMSeL utilizes SharePoint as its development platform, the layer of basic components is already realized by the underlying portal technology. Systematically chosen assessment tools as well as organization modules have been developed to demonstrate the suitability of AMSeL as an evolutionarily growing platform for open assessment management.

A first set of tools and modules has been developed to cover three of the four categories of classic assessment systems (cf. figure 2.4). The category of electronic test systems has been skipped, since a basic integration of an electronic test engine is available for L²P (see section 7.2). A discussion about a more advanced approach for integration of e-tests to AMSeL can be found in section 10.1. Generic assessment support that is integrated into a platform with other elearning functionalities, similar to an LMS, is realized by an integrated assignment management tool (section 8.3). Possibilities to successively add support for specific processes are demonstrated by extensions for publication of sample solutions or management of self-marking processes. Additionally, a module for group management has been developed to enable group assessment for different assessment tools (section 8.4). A workflow-based module realizes a generic and modular mechanism to enhance AMSeL with arbitrary domain-specific correction processes (section 8.5).

To demonstrate that AMSeL is suitable to integrate new dimensions of learning, a tool as well as a module have been developed, each to support a social media service. A tool for assessing contributions to wiki pages shows how the assessment of collaborative processes with user generated content can be realized (section 8.7). A module that allows to integrate YouTube videos as well as to associate them to students’ assignment solutions show how cloud services, which can be used for assessment 2.0 or networked assessment, can be integrated to formal assessment processes (section 8.6).
Patterns for development mechanisms which lead to a sustainable and robust platform are presented in section 8.1. The basic platform which hosts all above mentioned tools and modules is described in section 8.2.

8.1. Development Patterns

SharePoint already provides several deployment mechanisms, which allow to extend the platform in various ways. By using these mechanisms, a custom application can be deployed to different farms. The use of *features* facilitates the (de-)activation of specific functionalities within the application separately. Thus, the modules are adaptable for different scenarios. Main problems of such applications are reusability of functionalities by other tools as well as extensibility with new functionalities, which have to be integrated according to frequently added requirements. Although the underlying portal technology itself has a modular architecture, custom developed applications on top of it do not provide possibilities for modular extensions by default.

According to experiences gathered during the implementation of several SharePoint-based tools and modules, further advanced development techniques are required to facilitate loose coupling of extensible applications and components. Therefore, two new development patterns for SharePoint-based applications have been defined to ease the management of modularity and extensibility. They are used in several tools and modules, which are presented in the thesis at hand. Furthermore, they can be adopted by any new application for *AMSeL*.

8.1.1. Event-driven Service Delivery

The first pattern addresses event-driven delivery of services, which are provided by organization modules and consumed by assessment tools, according to the concept of service managers (see section 6.1.2). Some mechanisms for the development of an architecture to handle service managers are already provided by SharePoint out-of-the-box.

The paradigm is similar to *event receivers* in SharePoint, but those are limited to a fixed set of elements and related events that are triggered. For instances, a new event receiver that derives from *SPListItemEventReceiver* can be attached by a feature to a certain type of list, such that its *ItemAdded*-method is called, every time a new item has been added to a related list.

Since the realization of service managers requires the possibility do add newly defined *semantic events* for each new service, another approach has to be chosen.
8.1. Development Patterns

This approach is based on a combination of default deployment techniques for different scopes, standard data structures and storage types as well as custom developed business logic (see figure 8.1).

A new service manager is build as a class, which defines methods for information pull as well as for triggering on events, both concerning a specific semantic for a topic that is handled by this service manager. Possibilities for extensions are achieved by storing references to providers and consumers, which have to provide specific methods. Because of stateless web-connections, the store has to be persisted. In a first approach, this has been done by writing full qualified class names of provider and consumer classes to the property bag object of a related feature, that can be activated and deactivated at a specific scope. If the service of a provider is consumed or an event is triggered, the service manager creates a new instance of the corresponding classes by reflection and calls the appropriate methods. Several problems arose, because property bags of features are cleared with every deactivation. Furthermore, they do not provide a default mechanism to store large collections of properties. A more robust approach is to use specific instance of a SharePoint list that can be instantiated with the related feature. All of these elements are packaged into a solution, such that it can be deployed and activated in several SharePoint environments.

A provider as well as a consumer reference the service manager assembly and register themselves with their fully qualified names. To allow adaptable integration of providers and consumers they are mostly activated by a feature in a chosen
scope. Therefore, default feature event receivers can be used to execute the registration process.

An example for an application of that pattern could be the handling of students’ enrollments to a virtual course room (see figure 8.2), since SharePoint does not provide handling of user related events. In this simple case, the service manager defines a method `GetEnrolledStudents` which combines the result of the same method on all registered providers. This method could be used by consumers to get an updated list of all enrolled students. Changes on the enrollment are triggered by the providers to the manager by calling `OnStudentEnrolled` or `OnStudentUnregistered`, which forward the method call to all registered consumers.

Although, the presented pattern directly realizes the design for service managers, in some tools a modified approach has been implemented. To reduce complexity, it is not distinguished between providers and consumers. Only one interface that has to be implemented is provided for consumers as well as for providers. Suitable methods are implemented while others are ignored.

8.1.2. Declarative Reflection Factories

The second pattern addresses the dynamic creation of objects for execution of specific business logic from extended modules. This is needed for the development of extensible mechanisms within an application. The pattern applies for mechanisms which allow configuration by multiple instances of various subtypes.

For example, an e-mail application allows the definition of rules for automatic organization of messages. Different templates for rules are available, e.g. `subject contains word` or `sent by user`. Multiple rules can be instantiated with specific parameters (e.g. a concrete word to filter for) based on different templates. Defined configurations are persisted for later reuse. In the case the set of rules have to
be applied to a new message, parametrized business logic for each rule has to be executed according to the related rule template.

An appropriate approach for the execution of such configuration mechanisms is already provided by the **factory method** pattern, which is a constructional design pattern defined by [Gamma et al., 1995]. It is used to abstract the creation of products (objects), especially if the type of a product is not exactly known at design time. Based on parameters passed to this **factory method**, it decides which specific type of product is created. This pattern is well suitable for the above mentioned example of message rules in a closed system. Since default implementations of factory methods have to know all available object types (e.g. message templates) at design time, dynamic and loosely coupled extension is not directly applicable. Furthermore, the pattern does not explicitly define the process of persisting parameters for specific objects.

Therefore, a new development pattern of **declarative reflection factories** for SharePoint-based applications has been defined (see figure 8.3).

**Figure 8.3.** Declarative factory pattern for SharePoint-based applications.

It combines the general factory method pattern with a persistent storage and extensible template descriptions. SharePoint lists are used to persist configuration settings for each product in a related item. Each item can have a different content type, which defines the properties used in the form of an item. Furthermore, each specific content type references a specific product type. The factory method **CreateProduct** can be parametrized with a **ProductItem**, to determine the related type of product through the specifically used content type. An instance of that specific class is then created with the property values that are stored in the **ProductItem**. With a predefined **ProductList** that contains a set of related content types, this pattern allows persistence of product properties for use in stateless web scenarios.
Because content types are already highly extensible components within SharePoint, which can be deployed and attached with default mechanisms, continuous extension of new product types is facilitated by this approach as well.

The above mentioned example of message rules would be realized according to this pattern with definition of a specific content type per rule template. This allows the creation of rules by items of such content types. The business logic is defined in corresponding class that is referenced in the content type definition. Listing 8.1 contains an example for the declarative element manifest of a `SenderFilterRule` that can be deployed by a feature. The content type references a field for the `SenderAddress` that is used as a parameter for the rule. The rule class is associated to the content type by an additional `XmlDocument`-entry that contains the qualified name of the class an the corresponding assembly name.

```xml
<ContentType Name="SenderFilterRule" ID="[...]">
  <FieldRefs>
    <FieldRef ID="[...]">Name="SenderAddress" />
  </FieldRefs>
  <XmlDocuments>
    <XmlDocument NamespaceURI="http://company.de/MailApp/RuleClasses">
      <RuleClass Name="company.mailapp.rules.senderfilter">
        <Assembly>company.mailapp, Version=1.0.0.0, Culture=neutral, PublicKeyToken=[...]
      </RuleClass>
    </XmlDocument>
  </XmlDocuments>
</ContentType>
```

**Listing 8.1** Exemplary content type definition of a message rule.
8.2. Assessment Base

In this section basic elements of AMSeL are presented. Firstly, an assessment service manager is provided to enhance SharePoint-based learning and teaching environments with a general concept of assessment. Furthermore, there exist several basic elements concerning assessment, which are reusable for several assessment tools.

With application of the above presented pattern for event-driven service delivery, the development of an assessment service manager, as designed in section 6.1.2, is straightforward. Because the same concept is applied to extensions of assessment tools, an abstract and generic base class AssessmentTool<ExtensionType> is provided. This class already implements mechanisms for registration and persistence of extensions. To allow well typed access to specifically defined extensions, the generic type descriptor can be set to a specific type of extension when inheriting the base class.

In addition to the realization of AMSeL's structural framework, default deployment mechanisms of SharePoint have been utilized to provide several basic elements (e.g. list templates, fields, content types, webparts etc.) addressing specific sub-processes which can be found in different assessment processes.

For instance, the handling of timely managed submission phases requires the definition of according dates. Since activities would be mostly presented by items within a SharePoint list, properties for deadline and date of publication can be stored in related fields of the list. To ease the integration of such fields to a list, they have been defined as site fields. Even though SharePoint provides a default field type for storing dates, a field of that type is not suitable for definition of deadlines and publication dates. This is true, because those fields are strongly depend on each other, since the deadline cannot be set to a date before the date of publication. Furthermore, values within the field have to be set to readonly after publication or after the deadline is reached. Therefore a custom field type has been developed, which respects the explained business logic.

Another essential and basic concept of AMSeL is the well-timed delivery of elaborated feedback. Therefore, the definition of multimedia feedback is combined with a timed publication process related to a submission phase for an assessment activity (cf. section 6.2.2). Elaborated feedback text is stored and published as a list item in a related feedback list, which can be instantiated for several tools as well as contexts. The structure of the related basic content type is presented in listing 8.2. It inherits the basic item content type, but removes the default field for a title, since the feedback content stored in the note field body. A field for the selection of a feedback timing, which has been realized by a custom field type, as well as default rating fields are included as well. The latter allow students to judge the provided feedback on a simple star rating scale.
A default support of rubric-based marking is provided by *AMSeL* as well. Therefore, simply content types are used to define a marking scheme. An indicator is represented by a referenced field, while the field type corresponds to the related rating scale. An item persists a filled rubric that is related to a specific assessee (a user, a group, a submission etc.). While each item can be edited separately by use of the default UI, it is not directly possible to edit a list of elements without restriction (e.g. required ActiveX controls in specific browser versions). Therefore, an approach has been developed to define editable list views, which allow utilization of web controls for bulk edit functionality. Additionally, a webpart has been written which allows basic configuration of marking schemes by users with specifically definable permissions, since users need to have high permissions to be allowed to adjust content types directly.

The choice of possible rating scales for defined indicators are defined by the available field types. While default field types for storing *numbers*, *boolean values*, *text*, or fixed defined *choices* are directly applicable as scale types, there is no default type to facilitate the definition of site-wide available custom scales. Therefore an approach for the definition of custom scales has been developed to facilitate reusing those scales by all activated assessment tools within the same site collection (see figure 8.4). Exactly one instance of a SharePoint list for custom scales is placed in
the root web of a site collection. Named items within that list are defining a scale, whose values are stored in a related ordered list per scale. These lists are dynamically created by an event handler that is triggered when a new scale item has been added. Using custom scales within rubrics is supported by a custom field type.

The user interface for the management of custom scales (see figure 8.5a) and their ordered values (see figure 8.5b) are automatically generated by SharePoint. All CRUD-operations as well as additional settings (e.g. grant permissions or configure views) are available by default. Furthermore, a custom activity to view scale values by resolving the association between a scale item and the related list is smoothly integrated into the default UI.
8.3. Assignment Management

An assignment management tool for AMSeL has been developed in iterative process with currently three cycles. In the first (V1) and in the second (V2) cycles, the focus has been set to realize assignment management application as a modular extension for SharePoint. They were both embedded to the learning and teaching platform L²P and therefore developed base on SharePoint 2007. Each of them has been tested with pilot installations (see section 9.2). Additionally, V2 is currently in productive use for the whole university. In the third iteration, the module has been adjusted for SharePoint 2010. Furthermore, it has been shifted to the framework of AMSeL to increase the possibility for deployment of uncoupled extensions.

8.3.1. L²P Assignments V1

The first implementation has been design towards an abstraction of the common assignment management lifecycle 2.3) by selecting three major elements (see figure 8.6): assignments, solutions, and corrections. These elements represent the sequential flow of the traditional process. In addition to this, a back-link from corrections to solutions aims to facilitate direct feedback.

![Image of Assignment life cycle](image.png)

Figure 8.6. Abstracted assignment life cycle in the assignment management tool V1.

According to this model, a SharePoint-based design has been developed (see figure 8.7). Elements of the three main types are persisted as list items, with fields for their specific properties. Each different type of item is stored in a separate list. The items itself only contain metadata and can be understood as containers which can be filled with related content successively.

An Assignment for example can be filled by referencing items of three different types. AssignmentDocuments are files, which contain the description of tasks or questions to solve. AssignmentAttachments are used to attach additional resources (e.g. templates) that can be used for the creation of a solution. Figure 8.8 shows...
the edit form of an assignment item, in which the metadata can be defined as well as related items are managed. While default typed properties, for instance the title or a descriptive text, can be entered by default, advanced functionality had to be implemented to facilitate this default view. Firstly, a custom field type has been developed, which allows the definition of sub-tasks with a mapping to a related maximal score, that can be achieved. This functionality facilitates rubric marking. A second custom field type allows the dynamic management of related files from other lists. The last customization for the form are the field types for definition of a publication date or deadline. The stored dates are used by an attached TimingWorkflow, which handles the publication process.

Solutions and corrections are handled as containers the same way. Both of them are directly coupled, i.e. for every solution there exists exactly one correction. A solution can be created by a student, who is enabled to invite other students to participate the same solution. An invited student can accept or decline an invitation. In the first case the student is added to the team for the solution. Each student is only allowed to participate at most one solution per assignment. If he or she already has a solution, the acceptance of an invitation lets him or her change the team, such that the student leaves the former team. With this approach, a dynamic grouping process is directly integrated to the assignment management tool. Furthermore, the team can use the solution as a workspace for cooperative construction of the solution by adding solution files.
Figure 8.8. Edit form of an assignment item (V1).
8.3. Assignment Management

All team members have write access to their solution during the submission phase. After the deadline is reached, the timing workflow sets the solution to readonly for the team members automatically. Since SharePoint does not allow to define permissions on field level, information regarding the correction are separate to the correction item. Correction documents and feedback can be created within corrections by tutors. If their publication is set to direct, student immediately get read permissions for them. Otherwise this permission is set after the deadline expired. Provided score is copied from the correction to the solution. The detail form for a solution that is displayed to the team members (see figure 8.9) contains a combination of elements from a solution and its corresponding correction.

![Figure 8.9. Display form of a solution item (V1).](image)

A main drawback of the first implementation that has been addressed by new approaches in the second iteration, was the high level of element distribution that led to a huge amount of item level permissions, what decreases performance enormously. Furthermore, pilot users of the system requested additional functionality that has been prioritized, such that the highest rated features have been chosen for the redesign in the second iteration.
8. Portal-based Implementation

8.3.2. L²P Assignments V2

After some requirements that were stated by the stakeholders were integrated to the first system incrementally, a separate implementation has been created in the second iteration cycle. The main requests that have been realized in V2 are the following ones:

- integration of tutorials,
- additional filters for the list of solutions,
- alternative rating scales and modes,
- settings to define maximal group size,
- automatic announcements for newly published assignments,
- automatic labeling of solution documents,
- automated creation of cover sheets,
- downloading multiple documents as zip-archive,
- accessing solution documents via webdav (network drive),
- online creation of documents,

Some request as well as the mentioned performance issues required an adjustment of the overall structure of elements. The revised design uses folders within the lists of related items (see figure 8.10). This allows the organization of all elements that are associated to a main element (namely assignment, solution, or correction) within one place. Moreover, permissions can be granted to a folder, such that all elements within this folder inherit these permissions and the amount of item level permissions is massively decreased. Only in the case of direct feedback or correction documents, unique permissions have to be assigned to these items temporarily.

Especially suitable for large lectures, the management of tutorials has been integrated as an optional functionality. All student members of a tutorial are organized within a corresponding group. Supervisors of that tutorial are members of another group. A solution is associated with the tutorial of its first team member to decide which group of supervisors is responsible for correcting that solution. For organizational reasons, an optional restriction to allow grouping only within a tutorial can be configured. In that case, only students which are members of the same tutorial can be members of the same solution team. The allocation of students to tutorials has been connected to the campus management system CAMPUS, which allows the distribution of students among courses by use of different algorithms, not to create redundant processes. The allocation of tutors to a group of supervisors for a tutorial can be done manually within the assignment management tool.
8.3. Assignment Management

![Figure 8.10. Structure of the assignment management tool V2.](image-url)
Overview and filtering of solutions have been increased for the overview page of the exercise course within L2P. Depending on the user's role, student or teacher, other information is displayed. Teachers have direct access to all assignments and all corrections (see figure 8.11). In addition to the default mechanisms for filtering columns of lists in SharePoint, a filter control has been developed which allows filtering corrections by selected assignments or students. If tutorials are used, an additional filter for those appears as well. Students are only allowed to see published assignments as well as their personal solutions and invitations.

![Overview page of the assignment management tool V2.](image)

Figure 8.11. Overview page of the assignment management tool V2.

Configurations for the assignment management system can be made on an extra settings page, that is linked in the left navigation bar. The selection of automated announcement creation allows the timing workflow to create an appropriate announcement in the information domain of L2P when a new assignment item has been published. By definition of a maximal group size, all unanswered invitations for a solution are canceled when the limit is reached. Setting the labeling option to true
activates a custom developed event receiver that adds the id of the current course room, the solution id, and the filename to all pdf-files in the solution documents. This process allows easy identification of printouts for handwritten corrections.

Another global setting allows to choose a rating scale and mechanism for all solutions. Available options are: *scores*, *grades*, *results* (Yes/No), or *no rating*. An advanced option in the case of scores allows the exclusion of sub-scores from the calculation of a total score for an assignment. This allows to handle alternative sub-tasks. Furthermore, bonus score as well as a level of difficulty can be defined for each sub-task in the marking scheme of an assignment (see figures 8.12a and 8.12b). Marking schemes are applied for the concrete marking of a solution (see figures 8.12c and 8.12d).

Online creation of documents in addition to uploading existing ones has been addressed by a revised field type for handling related documents (see figure 8.13) in combination with a basic editor environment.

---

**Figure 8.12.** Advanced marking schemes in the assignment management tool V2.

**Figure 8.13.** Modular extensible frontend for handling related documents in V2.
In the edit mode of the field type, an additional panel is displayed which contains links to online editors for special filetypes. This panel is realized by use of a FeatureLinks-Control with a custom location, such that the links can be deployed as custom actions modularly. The corresponding editors can be realized by implementing an IFileEditor interface to be integrable to an editor environment that handles file loading and saving including dialogs in the UI and so on. Exemplary implementations include a rich text editor, source code editors with syntax highlighting, and a graphical editor for chemical models. All of these editors are realized by embedding already available editors (mostly javascript based ones).

Documents created in this way as well as uploaded ones can be access by single download, multiple within a single zip-Folder, or per webdav access. The creation of an archive has been implemented as a method that can be called for every folder of files in a SharePoint list. The access via webdav is a default feature of SharePoint, that is now suitable for documents in the assignment management system, because they are organized within folders per main element.

A podcast in plain German about how to use the assignment tool can be found at [63].

8.3.3. Modularized Assignment Management Tool

In the third iteration of the development process, the basic structures of the assignment management system has been revised again. The new implementation has been created based on the design described in section 6.2. Therefore a basic tool for individual tutor assessment has been created, which allows modular extensions to support further assessment methods and functionalities.

The Basic Tool

The new structure for the assignment management tool is still based on the idea of workspaces, which are now realized by separate sub-sites (see figure 8.14).

For each new Assignment that is added, an event handler creates a new AssignmentWorkspace as a sub-site. This contains an own list for AssignmentDocuments as well as for Markings and for Solutions. Figure 8.15 shows the default portal page for the workspace, that contains webparts to display information about related elements on a single page. Furthermore, specific actions related to the workspace, are accessible via an additional tab on the ribbon at the top the page. Examples for these actions are: modification of assignment details, creation of a new solution, or configuration of the related marking scheme. Markings are now presented by items rather then a field value. This allows a global definition of
Figure 8.14. Structure of an assignment management tool for AMSeL.
a MarkingScheme in form of a content type. Additionally, the access can be granted more precisely.

Figure 8.15. An assignment workspace in basic configuration.

Solutions are linked to specific SolutionWorkspaces that are placed as sub-sites within the AssignmentWorkspace. The owner of a solution can upload SolutionDocuments. To avoid redundancy as well to increase flexibility, a combined item for correction does not exist. Teachers and tutors can assess a solution by filling out the fields of the marking item, creating elaborated Feedback, uploading CorrectionDocuments. Students as well as teachers access the workspace via the default portal page that combines solution information with those of correction elements (see figure 8.16).

To allow extensible online editors for documents, these can be dynamically added by attaching according content types. For instance, Office WebApps can be deployed to SharePoint to allow collaborative work on word files, excel spreadsheets, and others directly in the browser.

Extension Examples

The main advantages of the new structured assignment management system are the possibilities to develop and deploy additional extension without any changes to the basic system are required. On the one hand this is achieved by default modularity
and deployment mechanisms of SharePoint. On the other hand, the dependency of process interaction are solved by the developed extension architecture.

For instance, if a mechanism to publish sample solutions has to be added to the tool to provide knowledge of correct answer feedback, this can be done by adding a new list instance for sample solution documents to assignment workspaces (see figure 8.17a). An attached workflow as well as an additional field for storing a publication date for the sample solutions are used to handle a timed publication process. The automation of this can be handled by features, that can be activated in the corresponding sub-site. To add the feature to all newly created assignment workspace, an extension class has been registered at the assignment management service manager to execute custom code when triggered.

A second example extends the assignment management process by an additional method for self-marking. Therefore, a SelfMarkingList with exactly one SelfMarking item is added to each existing solution workspace. The available indicators depend on a SelfMarkingScheme that is provisioned by a global scheme for all solutions to the same assignment. Students can fill in marks to evaluate themselves and compare to the marks given by a teacher or tutor. By default, a teacher is able to see these marks. The integration to the assignment process, e.g. to respect submission phases or get triggered when a new instance in a new solution workspace is needed, the extension is registered with a corresponding class.

Both of the explained extensions are completely packaged within a SharePoint solution, that allows easy deployment to every SharePoint farm. By bundling the elements with features, they can be activated or deactivated on demand. In this way the assignment management tool gets highly adaptable, such that different scenarios can be support in different contexts (or rather site collections) in parallel.
8. Portal-based Implementation

(a) An extension for publishing sample solutions.

(b) An extension to facilitate self-marking.

Figure 8.17. Exemplary extensions for the assignment management tool in AMSeL.

8.4. Group Management

Although SharePoint offers built-in functionality for self-directed grouping, a distinct implementation was needed to address the requirements, such as subgroup management or constraint verification [P5]. Furthermore, the grouping mechanism should be usable as a service as designed in section 6.3.

8.4.1. The Grouping module

According to the design, the main data structures have been realized as list items in separate lists (see figure 8.18). Different types of rules are represented by different content types, which define the specific properties and reference the corresponding rule classes, which implement the rule semantics programmatically. The declarative reflection factory pattern is employed in order to create instances of the rule classes (see section 8.1.2).

Two methods need to be implemented by a rule class: CheckGroupChange() and CheckRuleChange(). The method CheckGroupChange() is triggered whenever an existing group of the group set assigned to the rule is being changed or a new group is being added to that group set. It returns the information whether the changed or added group obeys to the rule. If rules are violated by creating or changing groups, the manager can define on a per-rule basis whether tutors or students can still perform the action, overriding the rule check, or if they are prevented from performing the action.
8.4. Group Management

Figure 8.18. Structure of the group management module.

CheckRuleChange() is executed if a rule is being changed or being added. In this case, all groups of the assigned group set need to be checked against that rule. This method returns a list of groups who break the rule, if there are any. This list is presented to the tutor or manager, who can decide to change or add a rule nonetheless, or rather change the groups first so that they do match the rule.

For instance, if a rule type restricts the size of groups, a parameter of a concrete rule would be the maximal group size. This parameter is a field of the rule content type but is also reflected by an attribute within the corresponding rule class (see listing 8.3). If the factory instantiates a rule object of this type, it will assign the field value of the rule list entry to an attribute of the rule class which matches the name and data type of the field value, making it accessible within the verification methods implemented by the rule class. Since all these steps are performed by the factory, no specific constructor methods for concrete rule classes are needed.

```csharp
public class LimitGroupSizeRule : Rule
{
    public int GroupSize { get; set; }

    public override bool CheckGroupChange(SPFIELDUSERVALUECOLLECTION oldGroup,
                                           SPFIELDUSERVALUECOLLECTION newGroup,
                                           IDICCTIONARY<string, string> metadata,
                                           SPWEB currentWeb)
    {
        return (newGroup == null) || (newGroup.Count <= this.GroupSize);
    }
    [...]  
}
```
8. Portal-based Implementation

**Listing 8.3** Class representation of a rule to limit the maximal group size

The system can be extended by defining new types of groups or rules based on the supplied standard types. For instance, tutorials extend groups in such a way that they are referencing an additional set of tutors. In order to create a new rule, only a new content type has to be defined that inherits from the abstract base content type for rules. Furthermore, a corresponding specific rule class has to be implemented that is derived from the abstract base class for rule implementations.

### 8.4.2. An Example Scenario

The grouping module can be used to facilitate different scenarios. For instance, a lecturer plans to involve his or her students with several collaborative learning activities: solving weekly assignments in small groups of three students, doing a bigger project in groups of ten students in parallel, and having several tutorials with about 30 students. Because the lecture is offered to students of different study courses, the small groups should always contain students of the same study course, whereas the members of the project teams should be mixed up. To ease the correction work of tutors, members of a small team have to be within the same tutorial.

The first step to organize this scenario with the presented tool is to create corresponding group set elements (see figure 8.19a). For each group set, a title has to be defined and options for the grouping mechanism have to be chosen. Because this system is a module for SharePoint-based portals, which already have a group-based role mechanism, the roles for managers, tutors, and participants have to be mapped to internal groups. If the environment, to which the module is deployed, has fixed roles, the mapping can be done programmatically. Thus, the corresponding elements can be hidden from the creation form.

Rules for each group set have to be defined as separate items. The elements of the form depend on the type of rule to instantiate. Figure 8.19b shows a form to assign a new size limitation rule to a group set. In this case, the additional property for the group size is part of the content type. The value is chosen for each rule instance and is used for the validation by the related rule class.

Overview pages for tutors (see figure 8.20) provide a direct way to manage the central grouping mechanisms. Students get an overview page for their current groups, invitations, and requests as well.
8.4. Group Management

(a) A form to define a new group set.  

(b) A form to create a new rule instance.

Figure 8.19. Creation form for elements in the group management module.

Figure 8.20. Example for a teachers’ overview of group sets and rules.
8. Portal-based Implementation

8.4.3. Grouping as a Service

An API for the service allows to use it within other tools. The implementation of a group service manager allows modular extension and loosely coupled connection to consuming tools that are triggered when group events happen, e.g. when a user joined a group. External systems can take advantage of this module by accessing the API via web service.

An example that make use of this service, is a group extension for the assignment management system of AMSeL. When the feature for this extension is activated, the content type for assignment items is extended with an additional field that references the list of group sets. In this way an assignment can be associated with a group set, when created. Furthermore, solution are extended with an additional field as well, a multi user field to store multiple owners. A new event handler that is attached to the content type of solutions, cares for attaching the team members of a solution owner to that field. Additionally, permissions are set. An extension class for the assignment service manager implements an update of the permissions for the group members whenever the state of the solution changes.

8.5. Correctionflows

The integrated workflow engine of SharePoint has been evaluated as well suitable for the implementation of the design to facilitate semi-automatic correction processes for applicative questions (see section 6.4). Development and deployment mechanisms of this basic component have been adopted to realize correctionflows, i.e. workflows for the execution of correction processes (see [P12, P13] and [T2]).

8.5.1. The Correctionflow Component

The composite structure of workflows by multiple activities which are combined in a mostly sequential order facilitate a direct mapping of evaluation steps and workflow activities. The three step deployment mechanism of workflows within SharePoint allow parametrization by teachers (in the association step) and additional choices for students (in the initiation step). Furthermore, developers are enabled to concentrate on the implementation of specific evaluation steps with activities that can be reused for different correctionflows.

There are mainly two different possible integration scenarios for using SharePoint-based correctionflows. Either the assignment management tool is also based on SharePoint, so that the correctionflows can be directly attached, or a separate assignment management system needs to use it as a service. A prototype, called
8.5. Correctionflows

eAixessor.Net, has been implemented [T2], which provides web services as an integration point. Additionally, the client system has to provide a web service itself to enable callbacks. This is needed because the workflows are potentially long running, especially if a tutor or peer is involved. The tutors response, e.g. feedback and marks, has to be handled by the consuming system.

8.5.2. Example Scenarios

Independent of both above mentioned scenarios for system integration, the extensibility, flexibility, and versatility of the approach facilitates various application scenarios.

Correction of essays

Prof. Smith provides a course about history. Students have to create essays about related topics and submit them during the term. From experience she knows that most submissions contain a lot of misspellings and grammar faults, despite current writing software already contains corresponding check mechanisms. Therefore, she decides to apply a correctionflow to all submissions, which generates score and feedback for orthography and grammar automatically (see figure 8.21a). The first evaluation step counts the number of words. The following branching element decides how to proceed in the correction process based on the determined word count and a defined threshold. If the number of words is too low, the author is informed via e-mail and the correctionflow is finished. Otherwise, the submitted essay is checked by a spellchecker and a grammar checker. Based on a given language parameter it specialized test steps for the language are chosen.

To realize this specific correctionflow, four activities have to be implemented by code. The fifth activity for e-mail notification can be taken from the default activities. The next step is to combine these activities as a sequential workflow template with the given control structures. This can be done by code or with SharePoint Designer. Beside the submitted essay itself there two additional parameters needed for this correctionflow: threshold and language. The threshold is defined as an association parameter and has to be place in the association form, such that the teacher could assign a value for all submissions to a specific task. The language is defined as an initiation parameter, though it can be set for each run of a correctionflow, i.e. it is chosen by a student for each submission.

This template for a correctionflow can be associated for several tasks. Moreover, the created activities can be reused in other correctionflows as well. New correctionflows can be created from scratch or as a copy of an existing template. In this example an additional activity could be added, which detects the language of the
essay automatically, such that the language parameter does not have to be defined manually. Furthermore, checker activities could be replaced by generic ones, which are adjustable with a language parameter themselves. In this case the control structure for language differentiation is not needed.

Figure 8.21. Exemplary extensions for the assignment management tool in AMSeL.

Correction of Source Code

Prof. Miller offers a course on Java programming. As an assignment to exercise loops, students are asked to implement the BubbleSort algorithm. When a student submits his source code, two sets of evaluations are triggered in parallel (see figure 8.21b). The first one inputs the source code to a compile activity and waits for it to finish. Depending on the compilation success, a branch takes place. In case the compilation fails (e.g. because of a syntax error), the tutor-in-the-loop activity is started. A tutor is notified and has the opportunity to correct the error, and to give the student corresponding feedback. If the compilation succeeds, there is no need to notify a tutor. Instead, automatic tests take place on the executable program that was generated by the compile activity. A unit test activity checks whether the submitted solution sorts a certain list of numbers correctly. The unsorted list and the expected outcome have been defined as parameters of the unit test activity, which are assigned by an association parameter. The unit test activity can be considered dynamic because it takes into account the output of previous activities.
8.5. Correctionflows

In contrast, the style check and keyword check activities are static because they are independent of the rest of the correctionflow. In particular, they do not need to wait for the compile activity to finish. For this reason, they are started in parallel to it. The style check validates whether the submitted code conforms to the Java Code Conventions \(^1\). The keyword check activity examines whether one of the keywords for and while occurs in the code, as they are required to correctly implement the BubbleSort algorithm. The list of expected keywords has been passed to the activity as an association parameter.

8.5.3. Correctionflows in Assignment Management

Two different correctionflows, for spellchecking (see figure 8.22a) and for html validation (see figure 8.22b), have been developed with direct integration as components for the assignment management tool as a proof of concept. Both workflows are reusing specialized activities, which allow access to elements within a solution workspace. For instance, a SolutionDocumentSelector allows selection and retrieval of solution documents filtered by file extensions. FeedbackWriter and CorrectionDocumentsWriter are used to store automatically generated results as corresponding correction elements in the solution workspace. Properties of each activity are used for either input or output and can be passed to each other. For example, the selected documents are stored in a property of the SolutionDocumentSelector which can be bound as input to a property of the SpellChecker. In this example, the SpellChecker allows to determine the language for correction. This parameter can be bind to either an association parameter or an initiation parameter, which is extracted and stored by the InitAndAssociationParametersLoader. In both cases the real evaluation is done by exactly one activity, SpellChecker or W3CValidator, but there could be more ones involved. While the SpellChecker is executing its operation by use of a method from the SharePoint API, the W3CValidator performs a web service call to pass documents to a service from the W3C consortium. By this, various different systems can be integrated as single evaluation steps. Especially, existing correction process implementations with a service-oriented interface can be used. For example, backends from EduComponents [Amelung and Rösner, 2008] (see section 3.4.3) can be integrated by an appropriate activity that sends XML-RPC calls to the provided platform.

\(^1\)The Java Code Conventions standardize how to format Java source code, e.g. how to indent lines.
8. Portal-based Implementation

(a) A correction flow for spellchecking assignment solutions.

(b) A correction flow for HTML validation of assignment solutions using W3C services.

Figure 8.22. Exemplary correction flows for the assignment management tool in AMSeL.

8.6. Integration of Cloud-Services for Assessment

With using SharePoint as the underlying technology for AMSeL, several parts for the integration of cloud services according the designed model (see section 6.5) are available by default. *Business Connectivity Services* already provide modular mechanisms to map external entities by external lists, whose items are filled by an associated service implementation which realizes the connection (see section 7.3). Standard mappings are used to integrate meta data of entities. Additionally, it is possible to implement a custom `StreamAccessor` per entity, which allows integration of binary files as well. Create and update operations are limited to meta data, by default. To bridge this gap, a workaround has been developed based on a custom field type. The field control handling the UI of the field type allows uploading files that are transferred as base64-encoded strings, such that they can be used as meta data.

As an example, figure 8.23 presents the data structures that are used to realize the integration of YouTube videos as part of students’ solutions within the assignment management system.

The mapping of users between AMSeL and YouTube, which uses Google IDs, can be realized with standard mechanisms by personalized storing of user accounts to an...
appropriate SecureStore. The BCS entity model as well as the according service that realizes CRUD-methods accessing YouTube, have been developed as a custom assembly connector. That means methods of a dedicated class are implemented to communicate with the YouTube API and map video entities within YouTube to related entities within the BCS model. The modeled entity can be used as an external content type, such that each entity is presented as a corresponding item in an external list. Because the BCS model as well as the list definition are developed to be deployable as a SharePoint features, they can be activated for each sub-site optionally. Figure 8.24 shows a screenshot of a solution workspace with a list of integrated videos from YouTube.

External lists have been optimized for meta data presentation, mostly based on simple data types such as text, numbers, or dates. Therefore further adjustments are needed to allow other forms of presentation. For instances, paths to thumbnails of referenced videos are resolved and transformed by XSLT to be displayed as images as displayed in figure 8.24. Moreover, some properties of the video entity are mapped to custom field types to allow advanced presentation and interaction in the related new form, display form, and edit form of a video item. For instances, the display view of a video item contains an embedded multimedia object to play the video directly within the environment.

Another important aspect regards the selection of entities for a certain list or rather view. For instances, a student wants to link two videos, which are uploaded by himself, as well as a freely available one, from another user. The chosen approach for a prototypical implementation has been to combine filters for keywords, users,
Figure 8.24. Screenshot of a solution workspace in the assignment management system with integrated videos consumed from YouTube.
and ids [T3]. Keywords are used to select a subset of videos from the list of a specific user. Further videos can be chosen by additional declaration of ids, by selecting videos in a list of search results. Some filter parameters can be defined by the environment in which the list has been integrated, such as the owner of the current solution. Other parameters, i.e. keywords and ids, can be assigned by the students manually.

8.7. Wiki Pages with Assessment-Support

The implementation of a tool to support assessment of collaborative contributions to wiki pages has been started with a prototype that extends the default wiki functionality within SharePoint by assessment functionalities (see [P5] and [T6]). It has been developed according to the model presented in section 6.6, such that marking schemes can be applied referencing different elements depending of the scenario of choice. Each configuration is covered by a different base type of marking scheme, that is represented by a SharePoint content type (see figure 8.25). Additional fields for specific indicators can be added as required. A list of marking items can be created in which each item is based on such a content type to store the results. Depending on the granted permissions to that list, it can be decided to facilitate tutor assessment, peer assessment, or self-assessment. Parallel list instances enable parallel approaches.

Figure 8.25. Structure of the assessment tool for wiki contributions.

The UI of the prototype consists of three main elements: a configuration page, a marking panel, and an overall marking page.

The configuration page allows teachers to define the constraints of their scenarios with respect to the three steps defined in the model (see section 6.6). In detail, they can choose if groups are used or not (A), create and assign the marking scheme to a specific marking, as well as choose the assessee types (B), and decide by whom the assessment is done (C).
8. Portal-based Implementation

An assessor can give his or her marking by using the marking panel (see figure 8.26). There is one separate marking panel for each wiki article. The marking scheme depends on the configuration which the teacher has made by using the configuration page described above. The left side of the marking panel shows all contributors to the related article. The assessor is able to select elements of that list to read the content of an article version which has been created by the selected contributors. The stored versions of the selected article are shown on the right hand side of the panel above the content. In case of a group assessment the contributors are grouped, such that all versions which are created by any author of the selected group are displayed. An assessor (e.g. the lecturer) still has to read all versions a contributor created for assessing the quality of the contributions. But by using this tool he or she does not have to search for and navigate to the relevant versions. Especially for marking groups, this kind of information collection could reduce the timely effort. Another supporting feature is the possibility of directly assigning marks to the contributors and groups respectively related to a previously defined marking scheme. The marking scheme in figure 8.26 simply consists of an indicator named Passed with a Boolean value. Beyond that, more complex schemes are definable (see section 8.2).

![Screenshot of a prototypical marking panel for wiki contributions (cf. [P5] and [T6]).](image)

**Figure 8.26.** Screenshot of a prototypical marking panel for wiki contributions (cf. [P5] and [T6]).

In scenarios with peer assessment the students are able use the marking panel as well, but they can only see those versions they have to review and use the configured marking scheme for peer marking. In self-assessment scenarios students may use the panel to reflect their own work and mark themselves. In these cases the marks from peer and self-assessment could be referenced in the tutor marking scheme as reference indicators, which are read only for the tutors.
In cases of an overall marking, the assessors can use the overall marking page. All contributions of each student (or group, respectively) in all wiki pages are presented to get an overview. The design of this page is similar to the marking panel for each article.

The prototype presented facilitates a marking process that is directly integrated into the environment of the collaborative learning scenario. With use of the above mentioned prototype, the teacher only needs to display the marking panel for the specific wiki page, read the versions per student, and directly type in the marks for that student.

8.8. Assessment Planning and Criteria Management

A tool to support the definition and publication of assessment criteria as well as bookkeeping of students results has been developed as a module for L²P (see [P14]). Although, the gradebook has been implemented based on SharePoint 2007, it is based on the modular design that has been presented in section 6.7.2 as well. Some modifications could be required for the transfer to SharePoint 2010, that is needed for productive usage of the gradebook within AMSeL.

8.8.1. The L²P Gradebook

Since L²P has been developed based on SharePoint 2007, the gradebook has been developed as a SharePoint solution for this platform. Similar to an Excel spreadsheet or a database table, students’ results are stored in a SharePoint list, in which each item is related to a specific user (see figure 8.27). Specific results of a student related to certain activity are stored in field for each corresponding indicator. A course criterion that defines which activities are used, on which scales they are rated, and how they are combined and calculated, is constructed within a second list. The three involved types of elements (indicators, categories, and rules) are presented by items with according content types. While there is exactly one type of category, indicators with different scales are presented by a set of possible content types (e.g. score and true/false). Further more, different rule types are possible, which can be easily extended with separately deployed content types at runtime. The realization of the business logic for the rule calculation is implemented in related rule classes, that are linked within a content type definition.

For instances, a rule that allows summation of results could be already available within the system while a rule for calculation of average values is missing. In this case a developer could create a new class implementing a method for calculation of average values as well as a new content representing a rule type that contains a
link to the new class. The content type definition as well as a content type binding element are combined in a new feature that can be deployed together with the rule class as a separate solution without modification the gradebook basis is needed. Activation of that feature in a sub-site allows local administrators to register new rule implementations with a single click.

The instantiation of rule classes is realized by a factory method that determines the corresponding rule class to use for a rule item by resolving the linked class name in the content type via reflection. That means, this is an implementation of the declarative reflection factory pattern as defined in section 8.1.2. More precisely, this has been the first implementation of the underlying approach, from which the pattern has been extracted.

In addition to the default mechanisms for storing students results manually, several functions have been implemented to ease the management of an according assessment process. For instances, advanced filter mechanisms can be used to selected only a subset of indicators to display. Since using required attendance to presence classes is applied in multiple scenarios (see section 4.1), the automatic creation of attendance sheets for printouts has been integrated. Since manual input of student results mostly requires much time and effort, especially in courses with many students (e.g. 200 and more), a separate view facilitates bulk editing of results from multiple students with a single post to the server. Since this is not possible in SharePoint by default, a custom approach that combines field controls in a grid view control has been implemented. To allow archiving as well as transfer of the results to the examination office, an export mechanism to create CSV formatted spreadsheet files has been developed. The most complex function has been integrated with a flexible mechanism for data import.

An add-on for the gradebook facilitates the integration of students’ results as well as descriptions of related indicators from sources at which these information are
already available. For instance, students’ solutions for published assignments are already marked within an assignment management tool, such that a tutor do not want to add these marks to the gradebook again. Another reasons are to reduce redundancy and avoid divergent results stored. Since types of possible sources are various as well as numerous, a generic approach for consuming available data has been developed. The main idea is to manage specific consumers which each handles the integration of data from a specific source (see figure 8.28). Because descriptions of available indicators are imported as well, the consumers have to be configurable for each scenario, such that a newly imported indicator can be placed in a destined category to be usable in the specific criterion. To allow persisting of these configurations as well as modular extension of new consumer types, the mechanism is build according to the declarative reflection factory pattern.

Figure 8.28. Structure of the extensible import mechanism within the L²P Gradebook.

Beside manual synchronization to pull updated information time and time again, sources can be allowed to trigger changes to the SyncManager, which triggers all registered consumers. Based on the synchronization settings, a consumer decides itself, if the provided information has to be consumed. For example, two different instances of an assignment consumer each handles a subset of created indicators, to structure them in different categories. Therefore, each instance has to decide if triggered event should be processed. The described mechanism has been constructed very similar to the event-driven service delivery pattern, that has been extracted from this one.

Exemplary consumers that have been built are for integration of the assignment management system V2 (see section 8.3.2) as well as using lecture dates defined in the campus management system for indicators, that can be used to record attendance.

A special case has been an additional function, which allows the import of CSV formatted spreadsheet files. Since there is no configuration item available for each new file for import, the configuration settings have to be specified after upload. For instances, mappings of columns to indicators or scale types for new indicators have to be defined. After this step, the import is realized by a specific consumer for CSV files.
8. Portal-based Implementation

8.8.2. An example scenario

Prof. Who holds a lecture for which the formative assessment scenario during the course involves a scenario including assignments, a mock exam, and students’ presentations (similar to the example in section 6.7.1). She requires her students to fulfill a criterion according to that defined in the criterion graph displayed in figure 6.16.

The L²P gradebook is used to support the management of her assessment processes. When the module is activated in the virtual course room that is used for the course, the result list is automatically filled with items for each student who is registered for that course. As a next step Prof. Who has to define categories and indicators according to which the students performances are measured by creating items of corresponding type in the element list. A new field in the result list is automatically created for each created indicator. These steps suffice to allow storing students results and use the additional functions. An overview of all students results that can be filtered by selecting indicators is available in the gradebook area of the course room for all involved teachers and tutors (see figure 8.29).

<table>
<thead>
<tr>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Result List" /></td>
</tr>
</tbody>
</table>

Figure 8.29. The result list in the L²P gradebook.

Mr. What, an assistant of Prof. Who, is accountable to keep record if a student has presented a solution in class. He uses the bulk edit page of the gradebook after each appointment to mark all students that have presented at that date in one process (see figure 8.30). If a student protest for a given mark, he uses the edit form of the result item for the student to only updated the personal record. Because Mr. What uses the assignment management tool in L²P to publish marks and feedback related to students solutions, he adds a new item for synchronization configuration, that is based on a content type for consuming assignments. Afterwards he starts the synchronization process, such that marks from that tool are transferred to the result list automatically.
8.8. Assessment Planning and Criteria Management

Figure 8.30. Bulk edit page in the L²P gradebook.

In parallel, Prof. Who constructs the criterion for the course by successively adding new rules which are linking categories, indicators or other rules. Students are able to see their personal results, but further more they are facilitated to comprehend their results according to the course criterion (see figure 8.31). All involved sub decisions calculated by the defined rules are transparent throughout the whole lecture.

Figure 8.31. The personalized gradebook page for a student.

8.8.3. Integration to AMSeL

The L²P gradebook is already constructed in a very modular way according to early versions of the both patterns that are used for AMSeL and its modules. Furthermore, there are defined migration paths available to shift the gradebook to a platform that is based on SharePoint 2010. The integration with the AssessmentManager requires some changes, but these should be straightforward. Furthermore, the mechanism of extensions for connection to other modules should be added. For instances, filter mechanisms based on groups or tutorials can be integrated in this way.
8. Portal-based Implementation

Nonetheless, a deeper refactoring process could be reasonable. The new platform version SharePoint 2010 offers advanced features, which have not been available when the grade book has been developed. For instances, XSLT-based list views, the ribbon-based user interface, or multi selection of list entries are well applicable to enhance the functionality. The new version of Excel Services facilitate dynamic formula modification in online hosted spreadsheets, that have not been possible in the former version. This makes an alternative approach for implementation using this technology more attractive.

8.9. Summary

A reference implementation of the conceptual design for AMSeL (Assessment Management Services in eLearning systems) has been presented in this chapter. The portal technology SharePoint has been selected as a basic platform that builds the basic components layer (see figure 8). Various components are already provided as well as deployment mechanisms that allow the extension with further components successively.

Five organizational modules have been implemented which provide reusable services: a gradebook, a grouping module, marking schemes, correction flows, and a module for integration of videos from the cloud service YouTube. These service can be used by two exemplary assessment tools: a modularized assignment submission tool and an assessment support mechanism for wiki pages.

Experience with early versions and different approaches for the SharePoint-based realization of an assignment management tool show that available portal mechanisms can be combined in different ways. Even if the underlying platform is flexible, modular, and extensible, a custom developed application on top of does not inherit these properties by default. Therefore, two patterns for SharePoint-based applications have been developed which can be applied to build modular tools. There functionality has been proven by using them for the development of the seven mentioned applications.
Figure 8.32. Exemplary tools and modules in the SharePoint-based implementation of AMSeL.
Part IV.

Review
Chapter 9.

Evaluation

The conceptual model of AMSeL as well as its reference implementation is evaluated in this chapter. Because concepts for application development and related implementation and deployment approaches are hardly testable by user tests, they are evaluated theoretically. Firstly, AMSeL is validated against previously defined requirements. To proof that the platform or rather some parts of it are functional, pilot installations of an assessment tool and an organizational module have been conducted, namely the assignment management tool (V1 and V2) and the gradebook. Results of related surveys and static system analysis are presented as well. The integration of YouTube videos to assignment management processes have been evaluated with user tests of a corresponding paper prototype so far. Finally, a discussion about the suitability of AMSeL as a platform for open assessment management closes this chapter.

9.1. Requirement Validations

The requirement specification in section 5.2 contains several functional and non-functional requirements which have to be fulfilled by a centrally hosted platform for open assessment management. How these needs are addressed within the model of AMSeL, and its reference implementation, is explained in the following. Since the functional requirements have been defined by categorized use cases, the realization of those use cases are discussed for each category separately.

Activities

Creation of activities by teachers (UC 1.2) as well as the available types of those (UC 1.3) depend on the assessment tools which are provided within the current instance of AMSeL. With the integration of an assignment management tool as well as a
tool to assess wiki contribution, it has been shown that the combination of different types of activities are possible. Moreover, the approach of a modularly extensible AssessmentManager and the abstraction of deployable AssessmentTools, allow to increase the amount of available activity types for creation by teachers successively.

Planning of an assessment scenario (UC 1.1) is supported by the gradebook that allows to define which activities and related indicators are used and how they are combined in a course criterion. The integration of activities from registered assessment tools is possible as well. That means, AMSeL provides a centrally hosted environment in which all activities can be managed within their related assessment tools. Nevertheless, it does not provide a single point of configuration for the whole assessment scenario, since the operational part, i.e. the detailed definition, publication, and execution of a related process, is not configurable directly within the gradebook.

Creation (UC 1.4) and submission (UC 1.5) of online solutions are provided by each integrated assessment tool individually. The presented assignment management tool within AMSeL shows, that it is realizable in general. The possibility of online tutor assessment (UC 1.6) is covered within this tool as well. Additionally, marking and feedback mechanisms are provided as reusable components for other tools as well.

Successive development of new editors to create different types of activities (UC 1.7), solutions (UC 1.9), and corrections (UC 1.8) is supported by both default mechanisms of SharePoint as well as additional mechanisms of AMSeL. Custom document editors can be developed and attached to newly defined content types which can be bound to lists of assignment documents, solution documents, or correction documents within the assignment management system. Furthermore, new types can be realized within a new assessment tool that registers at the AssessmentManager. In all cases a developer packs these new elements within a SharePoint solution (UC 1.10) that can be deployed by an administrator (UC 1.11).

**Marking**

The implemented gradebook allows the definition of course criteria (UC 2.1) which are created by combination of indicators and rules within categories. Rules and indicators are directly linked to a list of results, in which the rules are operationalized by calculated columns, which are used actively calculate each student’s result (UC 2.2). The defined criterion as well as the personal status according to each step within the criterion can is publish to each student (UC 2.10 and UC 2.11).

While the gradebook provides a kind of global marking scheme for a whole course, the creation of marking schemes (UC 2.3) as well as related indicators (UC 2.4) with different rating scales (UC 2.5) is provided as a central component for usage in
relation to single activities. The marking schemes are realized as content types that are kind of templates for several marking items which can be added and modified (UC 2.7) by tutors. Their publication for individual view of personal results by students (UC 2.9) can be timed related to four points in time of the assessment lifecycle (UC 2.8):

**Group Assessment**

Group sets within the group management module of *AMSeL* can be used to define a context (UC 3.1) in which groups can be created (UC 3.4) with restrictions defined by related rules (UC 3.3). Some rule types are available by default, while others can be developed (UC 3.12) and deployed (UC 3.13) successively.

The module can be configured for to support different scenarios (UC 3.2), such that it depends on settings if a group can be created by tutors or students themselves (UC 3.6). Furthermore, group members are either assigned by tutors (UC 3.7) or dynamically build by students using invitations (UC 3.9) or requests (UC 3.11) which both have to be accepted (3.10).

The participation to group work (UC 3.8) based on the defined groups as well as their assessment (UC 3.6) is realized within assessment tools, that are able to use the define groups using a provided API. For instances, a group assessment extension for the assignment management tool has been developed, that allocates students to solutions for collaborative work based on the grouping that is provided as a service.

**Peer Assessment**

As presented in section 6.4 an approach for modularization of management processes for peer assessment has been developed. It is based on the idea that each peer assessment process can be defined with a 5 step life cycle, where the underlying strategy is defined by three of these steps which build the peer allocation process. According distribution possibilities are abstracted in a step that creates the reviewer-artifact mapping. The realization in *AMSeL* can be easily done be utilization of the SharePoint workflow engine, such that the marking strategy is implemented as a workflow (UC 4.1) that use different activities as distribution modules (UC 4.2). Granting permissions to peers to view and assess assigned solutions (UC 4.4) has to be realized by separate a workflow activities which are developed for each assessment tool specifically. The definition of peer marking schemes (UC 4.3) as well as according marking and feedback lists are supported by components of *AMSeL*. Development and deployment pathes (UC 4.7 - 4.9) are directly provided by SharePoint, since standard mechanisms for workflows and activities can be used.
9. Evaluation

Self-Assessment

An exemplary self-assessment extension for the assignment management shows that the publication of activities (UC 5.1) and the creation of according solutions (UC 5.2) is directly applicable for self-assessment. The review of peer solutions for comparison (UC 5.3) can be realized analogously to the peer assessment process described above. A marking scheme for self marking is added by this extension, such that students can mark their own solution (UC 5.5), teachers can give feedback about that (UC 5.7) with the default feedback mechanism to which the student has access (UC 5.6). Further possibilities are the integration of automatic assessment, either in a domain-specific way or by integration of e-test tools.

A student can follow the own level of performance (UC 5.4) by viewing results related to local marking schemes or having a look at the overall personal results in the gradebook.

Automatic Assessment

(Semi-)automatic assessment of students solution to applicative questions are handled by correctionflows (see section 8.5). A correction process is developed as a workflow (UC 6.2) that is composed of activities which can be implemented to realize single domain-specific evaluation tasks (UC 6.1). Deployment of workflows and activities are supported by the underlying platform (UC 6.3). The composition of correctionflows using already deployed activities without programming skills (UC 6.6) is facilitated by use of available designer tools (e.g. SharePoint Designer or Visio). Further workflow engines (e.g. Nintex or K2), that are compatible with the Workflow Foundation, provide advanced features for composition directly within the browser.

Processes for configurable association of correctionflows to an activity (UC 6.4, UC 6.5) as well as parametrized initiation of correctionflows for a single submission (UC 6.7, UC 6.8) are covered by the SharePoint deployment process for workflows.

Feedback

The management and publication of elaborated feedback is a core component within AMSeL. Feedback and correction documents can be created as items in corresponding lists (UC 7.2, 7.3). The publication of these feedback elements depend on a timing process that can be configured by choosing on of four possible steps in the publication and submission live cycle (UC 7.4). The publication is based on SharePoint approval mechanisms, such that students only see elements that are already approved or rather published (UC 7.6, 7.7). A realization of assessment processes
using these feedback mechanisms are is provided with the assignment management system. An example extension for this system shows how the creation and publication of sample solutions (UC 7.1, UC 7.4) can be integrated.

**Assessment 2.0**

The requirements for the integration of external services for standard assessment processes has been realized with a prototype which allows the integration of YouTube videos for assignment submission. The module has been realized with use of SharePoint features, such that it can be activated within a chosen environment on demand (UC 8.1). Mapping of accounts for an external service (UC 8.2) to a user within AMSeL is realized by use of the *SharePoint’s Secure Store Service Application* that allows to store such mapping within an encrypted database table. By integration of videos as external lists, they can be directly integrated to solution workspaces in the assignment management tool, such that they can be investigated for the assessment of a related solution (UC 8.7). Creating and reading those videos directly from within the platform is realized by CRUD operation within a custom assembly connector (UC 8.8) that defines external entities and related operations. This connectors are directly deployable by administrators (UC 8.9).

The most difficult part is related to the question, how entities from the external platform are selected by students to be part of a solution (UC 8.4). Concepts for different approaches have been developed, which are using special keywords or storages for identifiers of explicitly chosen elements. The most appropriate way has to be evaluated in user tests.

**Hosting**

By designing AMSeL as an add-on for SharePoint-based portals, it is directly integrable to concept of virtual course rooms of L²P. This already provides a learning context (UC 9.1.) in which an assessment scenario can be managed (UC 9.2). Furthermore, this platform already demonstrates how the integration with IDM and ERP system can be managed by the example of the connection to the campus management system CAMPUS as well as the TIM system for identity management (UC 9.8).

The integration of new assessment tools to the platform is facilitated by established development tools (UC 9.5). They can be deployed and configured with standard mechanisms of SharePoint (UC 9.6, UC 9.7), while the registration and integration of assessment tools is handled by AMSeL using the AssessmentManager (UC 9.3, UC 9.4).
9. Evaluation

The main concept of AMSeL is to facilitate successively extension of the platform with new uncoupled tools and modules without changes to the basic application is needed. Thus, a distributed development process is facilitated, what means that independent groups of developers can create new applications that can be integrated independent from each other. For instances, this allows a scenario in which a university hosts the platform, while different institutes develop custom extension according to their specific needs. These extensions can be deployed by an administrator after the new development project has passed a specific process of quality assurance, such that vulnerabilities concerning security and privacy are avoided. Alternatively, sandboxed environments could be set up in which custom extension are executed. This would reduce the effort of formal review processes by the hoster. For the SharePoint-based realization this can be done by applying the approach of sand-boxed solutions.

Interoperability

Most available standards concerning assessment address information exchange on the level of specific assessment tools, e.g. data formats for electronic tests (QTI). That means each assessment tool within AMSeL is responsible for its own interoperability, since every tool would address another kind of assessment activities or processes. Since there does not exist any standard for management of specific assessment methods (e.g. peer assessment) or domain-specific correction of assignment submissions, the topic of standard conformity is not important for the presented assignment management tool. But sustainability is kept by the possibility to add a new extension that handles export and import methods of future standards. For the definition of assessment criteria or the management of groups are no standards available as well. To facilitate using available standards, additional tools for e-tests or for handling of SCORM packages should be developed and integrated as well. This is especially motivated by the need to transfer and reuse available content.

Beside standards, interoperability is supported by extensibility, i.e. various adapters for integration of or into specific systems could be created as extension or modules (UC 10.3 - 10.5). An example for a specific integration is embedded use of YouTube videos for assignment submissions.

9.2. Pilot Installations

The suitability of assessment tools and related modules within a portal platform as a centrally hosted service for a whole university has been evaluated with use of pilot installations. Prototypes of the assignment management tools V1 and V2 show that the system is very scalable and applicable for different scenarios in parallel. Processes of modular and uncoupled extensibility has been approved with pilot installations of
the gradebook, since this module already applies the two defined patterns presented in section 8.1.

9.2. Pilot Installations

9.2.1. Assignment Management

The first application module for L²P that supports assessment has been developed with an assignment management tool (V1). It has been tested with 6 pilot installations in summer term 2008 and another 31 pilots in winter term 2008/09. Topics of involved course were spread across languages courses, business sciences, chemistry, mathematics, computer sciences and others. The amount of students per course ranged from 8 to 442. Altogether, the module has been offered to 68 teachers, 38 tutors and 1,618 students in this phase.

A survey among all of them has been conducted. 114 students, 12 tutors, and 5 teachers participated the survey. 63% graded this early version of the tool as good or very good, for 18% it was ok, but still 9% did not like the tool (see figure 9.1a). 76% of the participants would recommend using the tool for other courses as well (see figure 9.1b). The 22% who would not recommend the platform. The latter case it has been often argued that some types of assignments can be easier created and corrected with pen and paper, such that no other system should be applied as well. Other causes for disapproval have been missing functions and usability issues.

![Graph](image1)

(a) Given grades from very good to very bad.

![Graph](image2)

(b) Recommendation for use in other lectures.

**Figure 9.1.** Satisfaction of students, teachers, and tutors with V1.

Based on additional requirements the test users defined as well as some performance issues, a revised version (V2) has been deployed to L²P starting with 69 pilot installations in summer term 2009. Since winter term 2009/10 the module is productive, i.e. it can be used in every virtual course room on demand. This can be configured by teachers with a single click as self-service. Since this time the assignment management tool V2 is used in about 140 virtual course rooms per term (see figure 9.2).
9. Evaluation

The huge amount of interaction by active usage of that tool is presented by high numbers of artifacts that are created within such environments. For instances, there have been up to 593 assignments that have been published in one term, such that 2.676 assignments have been published with the assignment management tool V2 in total (see table 9.1). Especially, the number of 46.199 handled solution documents shows that the platform is very scalable and highly used.

<table>
<thead>
<tr>
<th>Element</th>
<th>WS 09</th>
<th>SS 10</th>
<th>WS 10</th>
<th>SS 11</th>
<th>WS 11</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>360</td>
<td>442</td>
<td>593</td>
<td>507</td>
<td>591</td>
<td>2.676</td>
</tr>
<tr>
<td>Assignment Documents</td>
<td>385</td>
<td>423</td>
<td>641</td>
<td>445</td>
<td>614</td>
<td>2.736</td>
</tr>
<tr>
<td>Assignment Attachments</td>
<td>82</td>
<td>60</td>
<td>125</td>
<td>120</td>
<td>148</td>
<td>560</td>
</tr>
<tr>
<td>Solutions</td>
<td>4.740</td>
<td>3.760</td>
<td>6.636</td>
<td>5.511</td>
<td>5.987</td>
<td>29.011</td>
</tr>
</tbody>
</table>

Table 9.1. Created elements in the assignment management tool V2.

A variety of different configurations have been used. Different score types are applied, such that beside the default score, 15 courses have used grades, 9 ones have used results (True/False), and the marking has been deactivated in 81 courses. The possibility for exclusion of scores for sub exercises have been used in 19 courses. The support for organization of students and tutors within tutorials has been activated in 16 courses. Three main functions, which provide advanced benefits in relation to available systems, are observed separately. These functions are solution workspaces, time-controlled feedback publication [P10], and a dynamic grouping mechanism.

Solution workspace

Students have been asked about their opinion about the possibility of sharing documents in the solution workspaces. 70% stated that this function is helpful or very
helpful (see figure 9.3a). Nevertheless, only 49% of the students had uploaded intermediate versions some times or more often, while 51% never uploaded drafts (see figure 9.3b). Another 51% argued that they would not upload unfinished versions generally. Reasons are that they fear to get judged related to that versions or that they create the solution documents in one step, such that no intermediate versions exist. Furthermore, the upload process has seen as to time consuming, especially because of the mentioned usability and performance issues in V1.

![Figure 9.3](image)

(a) Opinions about sharing documents in solution workspaces.  
(b) Frequency of uploading intermediate versions to solution workspaces.

**Figure 9.3.** Students’ opinions using solution workspaces and related behavior.

**Feedback**

56% of the surveyed students told that they got feedback only after the final correction has been finished (see figure 9.4a). Further 5% got direct feedback during the submission phase additionally. 19% did not get any feedback while another 19% did not know if they got feedback. In contrast to that, 81% of the students rate direct feedback as helpful or very helpful (see figure 9.4b).

![Figure 9.4](image)

(a) Experiences about when feedback has been published for students.  
(b) Opinions about direct feedback.

**Figure 9.4.** Students’ experiences with direct feedback and related opinions.
9. Evaluation

100% of the teachers and 58% of the tutors have the same opinion. The mismatch between the applied feedback strategy and the expected or rather intended one could be explained by the effort needed to provide direct feedback. Moreover, above mentioned opinions about uploading intermediated results could have been a reason as well, since they are required for direct feedback.

The static evaluation of the afterwards developed assignment management tool V2 shows, that direct feedback as well as directly provided correction documents are used in addition to delayed publication after a correction process has been finished (see table 9.2). Furthermore, directly provided feedback is mostly provided as elaborated text while feedback about the final correction is mostly published within correction documents.

<table>
<thead>
<tr>
<th>Element</th>
<th># Direct</th>
<th># Delayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction Documents</td>
<td>162</td>
<td>14761</td>
</tr>
<tr>
<td>Feedback</td>
<td>1190</td>
<td>2559</td>
</tr>
</tbody>
</table>

Table 9.2. Elements per feedback timing in the assignment management tool V2.

Grouping

Static information about group constellations within the assignment management tool V2 have been analyzed for 93 courses in summer term 2009 and winter term 2009/10 [P6]. In 13 courses (14%) the tool has only been used to publish assignment. Creation and submission of students' solutions was not allowed. Collaborative creation of solutions has been activated in 23 courses (25%). Restriction to individual submission has been configured for the 57 remaining courses (61%).

![Figure 9.5. Allowed types of submission per course.](image)

Nevertheless, the amount of collaboratively created solutions has been higher then of the individuals ones, because most courses with many participants utilize group assessment. By analysis of the grouping behavior during a semester it has been discovered that averagely, students who create and submit their solutions individually are working more continuously. That means, students who are members of a group more often leave an assignment out. Relations between quality and quantity of those submissions have not been analyzed.
9.2.2. Gradebook

The second assessment specific module that has been developed and evaluated for use in L²P was the gradebook 8.8.1. Starting in winter term, it has been tested in 75 pilot installations across 5 terms (see figure 9.6). Evaluation has been done by static system analysis as well as accompanying surveys. The design of extensible rule templates as well as import mechanisms are supported by the results of these evaluations. Furthermore, the modular integration of new rule templates has been required to allow the definition of the criterion for one of the pilot courses.

![Figure 9.6. Number of gradebook pilot installations in L²P per term.](image)

First Evaluation

The first version of the gradebook has been evaluated with pilot installations in 16 L²P course rooms of seven different lecturers from different faculties [P14]. Six different models have been constructed. One of these models has been identified to be the default model that has been used in all participating seminars of an institute for communication theory (see figure 9.7). This strengthens the idea of having templates for common criteria that can be reused in several courses, what is currently not possible in the gradebook.

After about two months, five lecturers participated a survey about their experiences using the gradebook. All of them stated that the gradebook has supported them to detect and follow the students’ performances. 60% said their amount of work has been reduced, 20% said it did not change, and 20% said it even increased. The idea of modeling criteria with blocks of rules instead of textual formulas was rated as very good by 80% of the lecturers. 80% of them would potentially use the gradebook at its current state in the next semester again. If templates were available for the criteria model, 80% of the lecturers would definitely use the gradebook in the next semester. Additionally, mechanisms for importing results from other applications or spreadsheet files were suggested as very important improvements.

Additional surveys have been provided for all students that attended the courses which have utilized the gradebook. 47 of them attended the survey with very positive
9. Evaluation

Figure 9.7. Default criterion for all seminar of an institute for communication theory.

feedback. 90% of them rated online access to personal marks and results as positive or very positive. The usage of the gradebook for other course, for the case that the lecturers manage all assessment activities within this tool, has been recommended by 76% of the students (see figure 9.8). Furthermore, they suggested improvements of usability and integration of statistical functionalities.

Figure 9.8. Students’ recommendation about using the gradebook in other courses (WS 09).

Second Evaluation

According to the described requirements of the pilot users, a revised version has been implemented. The most important one of the new features was the possibility to import results and indicator descriptions from the assignment management tool as well as the course dates that are defined in the campus management system. These
new features have been carried out for 35 pilot installation across two terms (see table 9.3). The import of course dates for indicator descriptions has been used in 10 courses. The import of assignment results was configured in 7 courses. In 13 courses the gradebook has been used for bookkeeping and publishing results only, such that no criterion has been explicitly defined. A filter mechanism by tutorials, that was added as a feature in winter term 2010, has been available in 3 courses.

<table>
<thead>
<tr>
<th>Term</th>
<th># Pilots</th>
<th>No Criterion</th>
<th>Date Import</th>
<th>Assignments Import</th>
<th>Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS 10</td>
<td>17</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>WS 10</td>
<td>18</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 9.3. Pilot installations and used functionalities in 2010.

A second survey for teachers, which has been answered by 8 persons, has been conducted in winter term 2010. 7 out of 8 teachers thought, that the gradebook improves the management of students’ results and that it supports observing their performances. 62.5% would use the gradebook in its current state again for the next term (see figure 9.9a. The same is true if templates for common criteria would be available (see figure 9.9b).

Figure 9.9. Teachers’ opinions (in WS 10) about using the gradebook in next terms again.

Third Evaluation

Finally, the gradebook has been evaluated with 24 more pilot installation in the next two years. More then half of the lectures have used the import of results from the assignment management tool (see table 9.4). The import of course dates could not be used in winter term 2011, because of technical problems. Most of the pilots have defined complex course criteria.
9. Evaluation

<table>
<thead>
<tr>
<th>Term</th>
<th># Pilots</th>
<th>No Criterion</th>
<th>Date Import</th>
<th>Assignments Import</th>
<th>Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS 11</td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>WS 11</td>
<td>15</td>
<td>4</td>
<td>0</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 9.4. Pilot installations and used functionalities in 2011.

A survey which have been answered by 60 students’ show, that more than 56% of the students have used the gradebook often or very often to take a look at their current results (see figure 9.10a). More then 72% of them would recommend to use the gradebook for other courses as well (see figure 9.10b).

![Graph a](image1.png)  
(a) Frequency of viewing their results. 

![Graph b](image2.png)  
(b) Recommendation for using in other courses.

Figure 9.10. Students’ usage of the gradebook and according recommendations for other courses.

**Add-on development**

The main concept for modeling course criteria within the gradebook was the combination of modular rules. This approach has been rated as positive by most teachers during all pilot installations. The gradebook has been deployed in winter term 2009 with an initial set of 8 rule types, which have been used for creation of new rule instances. These provide rule types have been: *Sum, Average, AND/OR, Filtered Count, Proportion of Score, Condition, Graded Partition*, and *UpGrade*. While they sufficed for construction of most criteria, a criterion that was intended to build in a course in summer term 2011 required the integration of additional rule types. This real example is used to demonstrate how new rule types can be developed and deployed without changes to the basis of the gradebook are needed.

The course in the field of mechanical engineering was provided as joined lecture of two institutes. It has been attended by 1091 students. 43 tutors are employed to
help with the organization of assignments and corrections. According to examination regulations students have to pass a final exam at the end of the term. Because of motivational reasons as well as to aid learning, students are enabled to earn some bonus score for the exam by solving weekly assignments. The specific assignments have been solved in a special lab and were managed by a custom tool of the institute. CSV-imports from this custom tools has been used, since a specific import consumer was not available so far.

In cases that a student already gained some bonus score in a former term, he or she can reuse it for the exam in the current term, if not a former try of the same exam has been already failed (see figure 9.11). Four missing rule types have been identified, which are needed to model this criterion within the gradebook: Maximum, Minimum, Conditional Score, and Percentage to Score.

Figure 9.11. Course criterion that allows recognition of assignments gathered in former terms.
These requirements have been addressed by the creation of corresponding content
types and related rule classes, which have been deployed as a separate SharePoint
solution. Listing 9.1 contains an example for the class that realizes the transforma-
tion of percentages to score according to a given maximal value, that is applied for
100%. With activation of the respective feature, content type bindings are used to
add the new content types to the elements list. In this way, they can be used as
templates for the instantiation of new rules. The rule classes are directly registered
because, they have been referenced in the content type description according to the
declarative reflection factory pattern (see section 8.1). The factory instantiates the
new rule classes, without any changes to the code of the basic system.

```csharp
public class Percent2Score : UnaryOperation<IPercentagedElement>,
  IScoredElement
{
  internal Percent2Score(SPList elementList , SPListItem elementItem ,
    SPItemEventDataCollection afterProperties)
    : base(elementList , elementItem , afterProperties)
  {
    this.Maximum = GetMetaData<double>(elementItem , afterProperties ,
      FieldId.Maximum, double.NaN);
  }

  public override string InputTypeName
  { get { return ResultTypeId.Percentage; } }

  public override string ResultType
  { get { return ResultTypeId.Score; } }

  public override string ToFormula()
  {
    IPercentagedElement operand = this.Operand;
    if (operand == null)
      return "=0";
    else
    {
      double max = this.Maximum;
      if (max != 0)
      {
        string formulaPattern = "=IF(ISBLANK([{0}]) ; 0; ROUNDUP
           ([{0}]*{1}*2; 0) / 2" ;
        return string.Format(formulaPattern , operand.ColumnName, max);
      } else return "=0";
    }
  }
}
```

Listing 9.1 Class representation of a gradebook rule to transform percentages to score
values
9.3. YouTube Prototype

The integration of YouTube videos as a modular extension for the assignment submission tool has proven the feasibility of embedding cloud services to enhance established assessment processes within AMSeL. Usability of the assignment management system in combination with the YouTube extension has been evaluated by user tests of a related paper prototype [T3].

Prototyping in general can lead to improvements of several types, such as clarification concepts and terminology, improvement of navigation structures and page layouts as well as content [Snyder, 2003, pp. 1-6]. Furthermore, missing functionality can be identified and nonessential features can be reduced in early stages, mostly before implementation. More specifically, paper prototypes are a kind of low-fidelity prototypes which have several additional advantages, for instance their cheap, quick, and easy creation as well as their independence from software and hardware [Warfel, 2009, p. 197].

The flow of interactions for the assessment of YouTube videos in solution workspaces has been modeled as sequences of paper slides, which display the user interface. Figure 9.12 contains the page for a solution workspace of the prototype.

![Figure 9.12. A paper prototype for integration of YouTube-Videos for assignment submission [T3].](image)
The user test has been done with 3 teachers and 4 students, which have to solve 3 tasks that have been defined for each role separately. They have to select a video for integration with the solution, create and view feedback for the whole solution, and add comments and a rating for a selected video directly within the solution workspace. Interactions with the prototype has been observed silently. The testing sessions lasted in average 30 minutes. The users completed the tasks with small hints when they encountered some problems. After each task the observations and the ideas of the users were gathered.

Evaluation outcomes of the tests are that several naming problems occur. Repositioning of elements for user interaction, e.g. text boxes for user input, are recommended by some participants as well. Except these observations the rest of the interface and operations were clear to the users. The steps to be performed were similar to the original YouTube interface and smoothly integrated into the existing UI of L²P. Renaming of some elements as well as restructuring some elements within the pages should increase usability. Finally, it is assumed that the extension for integration of YouTube videos as elements of submissions to weekly assignment could be well accepted by students. This has to be evaluated in combination with a large pilot installation of the whole platform of AMSeL.

9.4. Conclusion

To summarize, the validation of requirements show, that the approach of AMSeL covers all defined top-level use cases. Some limitations concerning assessment planing have been identified. Pilot installations of assignment management tools show the suitability of a portal-based approach for a university-wide hosting and for support of several different scenarios in parallel. The need for flexible import mechanisms as well as extension mechanisms are demonstrated by real needs which arise during pilot installations of the gradebook. A low-fidelity prototype has been tested to demonstrate users opinions about next steps of development.

These evaluations show that the conceptual model of AMSeL as well as its SharePoint-based reference implementation allows successive integration and combination of traditional and new forms of assessment. For instances, it is now possible to combine group assessment and self-assessment by adding both related modules to the assignment management tool. Additionally, YouTube, as a social media service, can be integrated and embedded into a formal assessment process. This kind of assessment support goes one step further compared to analyzed platforms for assessment management in higher education (cf. chapter 3). The most important aspect of the platform is its degree of extensibility and flexibility, which allows to enhance the platform with new modules successively, without changing the basic platform. In contrast to other extensional assessment systems (cf. section 3.4.3),
AMSeL does not only allow modules to extend a predefined types of processes (e.g. item authoring). It facilitates the modular integration of new process types with an event-based architecture.

Therefore, the model as well as the reference implementation of AMSeL are one possible answer to the initially formulated research question: “How could a harmonized model for technological support of assessment management, which allows the integration and combination of traditional and new forms of assessment, look like?” Although the evaluation results are very good, it has to be discussed if AMSeL is suitability as platform for open assessment management. Since it has been developed by an approach with direction A, what means with an institutional perspective, the system is well suitable for a hosting scenario within a university. The decision to realize a central service in contrast to several distributed systems, has been agreed by many users.

It has been hypothesized that “An assessment management platform requires a highly modular architecture and flexible deployment mechanisms to allow extension and adaptation for various assessment scenarios.” The variety of different scenarios that are used in different courses, the large amount of possibilities for technological enhancement of assessment processes, and the parallel usage of different configurations demonstrate at least the need for a very flexible and adaptable platform. Frequently upcoming requirements during the pilot phase, which could be very different or even contradictory, show that extensibility is a very important as well. With AMSeL or rather the prototypical use of the gradebook, which both provide a highly modular architecture and flexible deployment mechanisms, it has been shown that this approach allow extension and adaptation for various assessment scenarios. The capability of other approaches to meet the demands is not excluded. Since the concept of a modularly extensible service that allows standardized deployment mechanisms for loosely-coupled extensions facilitates distributed development for specific add-ons by multiple clients, it seems to be the most suitable way to allow this kind of scenarios.

To observe the project from a lifelong learning perspective, it could be criticized that the platform design assumes a formal learning setting for assessment support. Since the approximation of an open assessment management platform has been done with an institutional perspective, only formal learning scenarios in institutional learning has been opened to include new dimensions of learning to existing assessment processes. For instances, the idea of informal feedback for submitted videos in YouTube has been combined with the formal assessment process of weekly assignments. Nevertheless, the approach is applicable to non-institutional scenarios which are organized as formal learning. For instance, Massive Open Online Courses (MOOC) are non-institutional and can be classified as organizational learning. Their concept of virtual course rooms can be directly adopted to support their management, such that AMSeL is directly applicable depending on the provided assessment
tools. The same is true for other scenarios in which the learning can be organized as a course, such as professional learning as accompanying course. Approaches for non-institutional certification like Open Badges (see section 2.2.5) could be integrated by a module that is a consumer of the AssessmentManger and acts as a bridge between AMSeL and the Open Badges platform to fill the related portfolio successively.

For informal and self-directed learning without formal parts the concept of a centralized platform with a course-based approach is not directly applicable. In these cases assessment methods have to be directly integrated within the distributed tools that are used for the specific kind of learning. That leads to the demand for an approach that follows a non-institutional perspective to approximate an open assessment management platform from direction B.
Chapter 10. Summary and Perspectives

The thesis at hand describes conception and development of a platform for assessment management as a first approximation for a system that supports assessment processes regarding all dimensions of lifelong learning. With an institutional perspective, the approach was to build a flexible and extensible architecture for assessment management tools and modules that provide a central service that is open for the integration of new approaches. A reference implementation has been realized based on portal technologies.

Investigations of the assessment domain show that several activities and methods can be applied to assess students performances in several dimensions, such that various assessment approaches are possible to address different purposes. According to the potential of technology enhanced assessment, the concept of open assessment management has been introduced which tries to bridge the gap between established institutional assessment processes and social-media supported realizations of the new dimensions of learning (see section 2.3).

Results of a broad analysis of functionalities within existing assessment tools as well as real scenarios at a German university show, that a centrally provided platform to support assessment management processes would be preferred. Therefore, a consolidation of existing assessment tools has been identified as the major challenge and objective. Opportunities and demands of new approaches, especially the integration of cloud services to the assessment process, have been collected by use of a SWOT analysis. Based on the results of all types of analysis, 90 top-level requirements for the intended platform have been defined.

A three-layered conceptual design for the platform AMSeL (Assessment Management Services in eLearning systems) has been developed according to the previously defined requirements. Each of these layers is composed of modular building blocks, which allow to extend the platform successively. This concept requires an abstract mechanism for communication of loosely coupled modules across layers, what has been address with the concept of service managers. A reference implementation
10. Summary and Perspectives

of AMSeL has been realized based on SharePoint, which is a portal technology. The concept of the approach has been demonstrated with the development of two assessment tools and five organizational modules based on two newly defined and technology-specific patterns for the creation of extensible SharePoint-based applications. Innovative functionality, e.g. time-controlled feedback publication and dynamically definable course criteria, has been developed as well.

The evaluation of AMSeL has been done based on pilot installations of an assignment management tool and a gradebook within the learning and teaching portal L2P as well as theoretical requirement validations. The acceptance of the prototypes show, that the project has been successful according to the defined requirements. Nevertheless, it has to be taken into account that the system design is just an approximation of technology enhanced support for open assessment management that could not cover all aspects of this topic. It was suggested to try an approach from another direction with a non-institutional perspective to address assessment processes of non-formal learning, especially self-directed learning and informal learning, as well.

Concluding, the main contributions of the thesis at hand and the related projected are the following:

- A categorization of assessment tools.
- A conceptual design of an open assessment management platform called AMSeL.
- A reference implementation of AMSeL based on SharePoint.
- Patterns for development of extensible SharePoint-based applications.
- An extensible assignment management tool with time-controlled feedback mechanism.
- A rule-based group management module to provide grouping-as-a-service for other tools.
- Correctionflows, which realize semi-automatic correction processes by composition of reusable activities using a workflow engine.
- A module for integration of YouTube-videos to the formal submission process for assignments.
- An extension for wiki pages to ease qualitative assessment of individual as well as collaborative contributions.
- An extensible gradebook module that allows the definition of executable course criteria.
10.1. Conceptional and Technical Advancements

Since *AMS*e*L has been developed with the objective to unify the support for various assessment process within one platform by extension of specific tools and modules, these applications or rather connectors for existing ones have to be developed in future. This is necessary to proof its concept as well as to make it suitable for most scenarios and sustainable according to changing needs.

**E-Test Integration using UTX**

One of the first additional assessment tools for *AMS*e*L should allow the management of electronic test, because they are very suitable for learning and testing factual knowledge. Because multiple e-test systems are already available, the development of a completely new application should be avoided. Possibilities of standardized e-tests as well as advanced functionalities of specialized systems should be respected. An approach could be the integration of multiple e-test systems covered by an abstracted layer. The system *UTX* (*Uni*fid *Te*st *e*Xperience) [T4] follows this approach by creation of a service-oriented middleware that integrates existing e-test systems with specific connectors (see figure 10.1). The integration to an existing platform, a LMS or a systems like *AMS*e*L, is realized by a specific UI plug-in. Such a plug-in could be developed and deployed as a new assessment tool for *AMS*e*L. This should be straightforward for the SharePoint-based reference implementation by utilizing Business Connectivity Services.

![Figure 10.1. Architecture of the UTX e-test middleware [T4].](image-url)
Exercise Pools with XAM

In institutional context a formal learning situation (e.g. a course) and its assessment scenario is designed for repeating time and time again for different audiences with only little changes. Therefore, assessment artifacts (e.g. assignments or rather exercises) are predestined for reuse. The default approach of reuse in L²P is a semi-automatic import process of content from other course rooms. A more structured approach has been realized with a system called XAM (eXercise and Assignment Management) [T7] that has been developed at the Computer Science Department 3 at RWTH Aachen University. It is a kind of authoring tool for single exercises with document attachments which can be reused to compose assignment sheets and exams. A possible component for AMSeL could connect with the assignment pool to allow assignment reuse without implementing an own pooling mechanism. The integration should be realizable as external list.

Flexible Service Integration

The current flexibility for service integration demands the development of custom connectors (e.g. YouTube, UTX, or XAM) and related components for registration within the assignment management tool. Since multiple services could be connected on very similar ways, connector templates could allow teachers to include new services on demand as a self-service. Depending on appropriate permissions, it is already possible for end users to create new external content types which directly map standard database fields or entities consumed from web services. Furthermore, it is possible to develop new connector types. The most difficult part would be an abstraction of components for registration and instantiation of related external lists within solution and assignment workspaces.

Operational Assessmentflows

To allow various combinations of tools and modules makes the platform very flexible, but the complexity is increased with every new application as well. That means, during the process of planning the assessment for a course (e.g. the combination of several assessment activities), each required tool which is used to support at least one activity has to be configured separately. Especially, the instantiation of potentially each activity has to be done at a different place within the platform. Since the gradebook allows unified planning of assessment criteria across different types, methods, and tools for assessment, it could be enhanced to allow centralized management of distributed activities across their specific tools. Such a single point of configuration provides holistic support of technology enhanced assessment processes.
An approach which addresses these needs has been started with the concept of assessmentflows [T8], which should provide an editor for planning of assessment processes that operationalizes the process by integrated configuration of all involved tools. A specific data format that allows persisting those process definitions, including course criteria, could allow to reuse once created settings in other learning contexts again. Similar to correctionflows, a suitable approach for realization could be the utilization of workflows. A discussion about how workflows can be used to abstract and define assessment tools for multiple types of activities and scenarios can be found in [T5].

Portfolios and Competencies

Certification processes are typically based on summative assessments. For instances, a master’s degree in computer science is achieved by passing several final exams which are related to different courses. A more meaningful vehicle to show one’s competencies could be a portfolio that outlines solved tasks and gathered experiences in more detail. By using formative assessment outcomes, that are often used as a preliminary stage to final exams, as certified elements of e-portfolios, these portfolios could be a more detailed kind of competency based certificate. This highly depends on how tasks are provided and in which way their related competencies are defined.

A technical approach for automatic provision of such portfolios could be combined with AMSeL. Therefore, an appropriate module could be developed as a consumer for events of all AssessmentManagers, such that decisions for a student’s submission could be directly written into his or her portfolio. A combination with the gradebook is imaginable as well, such that competencies could be attached indicators and rules in the gradebook. For instances, a rule that combines the results of three assignment that address the same topic could be used to measure the competencies of students according to this topic. The outcome could be written to the portfolio as a certified competency.

A combination with Mozilla’s Open Badges framework (see section 2.2.5) could provide a suitable approach to include individual achievements, which have been gathered in an institutional context, to this informal platform. The integration or rather acceptance of informal badges as part of institutional certifications is possible as well, but implicates several difficulties. A related question that has to be answered in this context is: “How can informal assessment be organized to be trusted as a component for institutional certification?”.
10. Summary and Perspectives

10.2. Hosting Approaches

The next step to evaluate the suitability of AMSeL as a centrally hosted platform, would be to start a pilot installation of the whole system. Because L²P as a platform that provides appropriate learning contexts, is still based on SharePoint 2007, a migration to SharePoint 2010 or the set up of a new environment based on that technology is required.

Multi Tenancy

In the context of lifelong learning generally, and open assessment management specifically, technological support approaches for learning or assessment processes are not limited to a single institution. Therefore, it must be possible to host several custom configured instantiations of AMSeL, or comparable implementations, at different institutions. A hosting or cloud-based scenario that involves more than one university is possible as well. Furthermore, the system could be opened to allow the creation of courses to everyone. Different requirements, designs, support services, billing models, and other settings have to be configurable for each tenant individually. For instances, a university A wants to use the system such that each teacher can create own virtual course rooms, while a central administration office accounts for this at university B.

The thesis at hand explains, why and how the platform AMSeL has been modeled and implemented to allow the support of differently configured assessment scenarios in parallel. This has been realized by running contained assessment tools and modules in a certain context, e.g. a virtual course room. Multi tenancy goes one step further by clustering such contexts for each tenant. It has to be researched how this additional abstraction can be integrated to the overall model of AMSeL.

Distributed Development

Most central services are developed, maintained, and enhanced only by the development team of the service provider. In the case of assessment management, it is assumed that general mechanisms (e.g. grouping or marking) are well understood by a central team, especially since these processes are described in the thesis at hand. Furthermore, each institute has knowledge about domain-specific processes.

Since the development and deployment of new modules and tools is modularized within AMSeL, the development of new applications could be distributed to several development teams. For instances, a new authoring tool for chemical solution documents could be created at the Department for Chemistry, while an improved correctionflow for the evaluation of source code for the programming language Haskell
is developed by the Department of Theoretical Computer Science. The development of new components could be used as programming tasks for computer science students as well. Required skills for such development could be trained by developers of the hosting institute.

A kind of App Store or rather a sharing platform, could be established to allow developers and teachers to share tools, modules, and descriptions of assessment flows and correction flows. The latter idea is very interesting according to reuse of once defined assessment process in different contexts. A social component for rating and commenting could increase benefits of such a platform. Further research is required, to investigate how tools, scenarios, and corresponding assessment processes could be categorized, compared, and especially recommended to other teachers. For instances, a teacher for a programming course with 400 participants of three different courses shares his configuration of the related assessment process. Related questions are for example:

- *How could another teacher know, if this approach is applicable to his own scenario, if it is beneficial?*
- *And how could appropriate configurations be recommended automatically?*

### 10.3. Lifelong Learning Support

As already discussed in the conclusion, AMSeL only provides an approximation for an open assessment management platform with an institutional perspective. Another approach to approximate the system from direction B, i.e. the non-institutional perspective, could be used to investigate which characteristics of both approaches are different and which ones are very similar. This could lead to a better understanding of different kinds of assessment according to lifelong learning.

The objective to create a *Lifelong Learning Platform* has been the initial motivation for the development of a platform to support open assessment management. All processes that are not related to assessment have been skipped, and therefore have to be addressed in further research. The compatibility of AMSeL with other involved processes has to be evaluated as well. Since the reference implementation of AMSeL has been realized as a portal-based solution that can be deployed as additional functionality to every SharePoint server, it is suitable for combination with other processes. If this is true for all settings in the context of lifelong learning has to be evaluated.
Part V.

Appendix
Appendix A.

Interview guide

Analysis of specific scenarios in computer sciences at RWTH University (see section 4.2) has been done by interviews guided by the following questions and answers. The guideline is in German, since all interviews have been performed in German.

1. Allgemeines zu den Veranstaltungen
   a. Veranstaltungsart
   b. Veranstaltungsgröße
   c. Studiengänge
   d. Betreuungspersonen
   e. Sonstiges

2. Offene Übungsaufgaben
   a. Übungsblätter
      i. Aufgabentypen (Texte, Beweise, Grafiken, Quellcode, …)
      ii. Art der Aufgabenerstellung (handschriftlich, Word, …)
      iii. Hilfsdokumente
      iv. Bereitstellung (Papier, E-Mail, Website, LMS, …)
      v. Zugriffsbeschränkung
      vi. Zeitliche Steuerung
   b. Studentische Lösungen
      i. Entgegennahme (Zettelkasten, E-Mail, LMS, …)
      ii. Mehrmaliges Einreichen
      iii. Abgabe in Teams
A. Interview guide

1. Teamgröße
2. Teambildung
3. Strikt?

c. Bearbeitungszeiten
   i. Fristen (Verwendung, Strikt?)
   ii. Feste Taktung
   iii. Überschneidung

d. Korrektur
   i. Art der Korrektur (Papier, digital)
      1. Einscannen von Papierlösungen
      2. Ausdruck digitaler Lösungen
   ii. Korrektur durch Hiwis
   iii. Peerreview
   iv. Bewertungsschemata
   v. Feedback in Bearbeitungsphase
   vi. Textbausteine für Feedback
   vii. Automatische Vorbewertung
   viii. Bewertungsmethode (Punkte, Noten, Prozentwert, ...) / Bonuspunkte

e. Musterlösungen
   i. Zugriffsbeschränkung
   ii. Zeitliche Steuerung

3. Objektive Aufgaben
   a. Eingesetzte Fragetypen (Multiple-Choice, Lückentexte, ...)
   b. Durchführungsart (Papierbasiert, E-Tests)
   c. Software bei E-Tests
   d. Betreiber bei E-Tests
   e. Anmerkungen zu E-Tests

4. Bewertung von Diskussionsbeiträgen
   a. Durchführungsart
   b. Bewertungsmethode
c. Foren-Software
d. Foren-Betreiber
e. Anmerkungen zu Foren

5. Bewertung von Wiki-Beiträgen
   a. Durchführungsart
   b. Bewertungsmethode
c. Wiki-Software
d. Wiki-Betreiber
e. Anmerkungen

6. Bewertung von Blog-Beiträgen
   a. Durchführungsart
   b. Bewertungsmethode
c. Blog-Software
d. Blog-Betreiber
e. Anmerkungen

7. Sonstige Teilleistungen

8. Kriterien
   a. Erzielbare Prüfungsvorleistungen (z.B. Klausurzulassung)
   b. Unterscheidung nach Studiengängen
c. Konkrete Kriterien-Definition

9. Verwaltung
   a. Hilfsmittel zur Ergebnisbuchführung
   b. Veröffentlichung der Zwischenergebnisse

10. Sonstiges
Appendix B.

Criteria Graphs

Examples of real criteria graphs (see section 6.7.1) for selected lectures that have been analyzed by interviews (see section 4.2) are illustrated in the following.

Figure B.1. Criterion graph for the scenario in the course Complexity Theory.
B. Criteria Graphs

Figure B.2. Criterion graph for the scenario in the course OOSC.

Figure B.3. Criterion graph for the scenario in the course IDB.
Figure B.4. Criterion graph for the scenario in the course *Programming*.
Part VI.

Catalogs
Bibliography


Bibliography


Bibliography


Links


Links


[L18] Hot Potatoes\TM. http://hotpot.uvic.ca/


[L22] TOIA. http://www.toia.ac.uk


[L24] ILIAS. http://www.ilias.de/


[L38] iGoogle. http://www.google.de/ig

282


[L54] CENTER FOR INNOVATIVE LEARNING TECHNOLOGIES (CIL): Lehr- und Lernportal L²P. http://www.elearning.rwth-aachen.de/


**Links**


[L63] **Center for Innovative Learning Technologies (CiL):** *Podcast: Wie kann ich über L²P Übungen abgeben?* [https://www2.elearning.rwth-aachen.de/help/extern/videos/Final1310.htm](https://www2.elearning.rwth-aachen.de/help/extern/videos/Final1310.htm)
Publications


Supervised Thesis


List of Figures

1.1. Dimensions of lifelong learning ........................................ 4
2.1. Timing of diagnostic, formative, and summative assessments. .... 11
2.2. Six cognitive levels of Bloom’s taxonomy. .......................... 25
2.3. Common Assignment Management Lifecycle. ........................ 28
2.4. Classification of classic systems supporting formative assessment. 31
2.5. General stages in an e-test life-cycle. ................................. 31
2.6. Generic architecture of common lms. ................................. 34
2.7. Evidence generation life-cycle ........................................... 41
2.8. An open badge backpack. ................................................ 41
2.9. Example of an institutional certification logo. ...................... 42
2.10. Classification of assessment systems by learning theories. ....... 43
2.11. Conceptual Assessment Framework. .................................... 49
2.12. The four process model ................................................ 50
2.13. FREMA process-based topic map ....................................... 52
2.14. Classification of open assessment in context of lifelong learning. 53
2.15. Characteristics of 21st Century Assessment. ......................... 55
3.1. Categories for analysis of assessment tools. ......................... 60
3.2. History and development of the LMS market. ....................... 65
3.3. The page for the preparation of a new exercise sheet in xGen .... 68
3.4. A state-chart diagram for peer assessment in FAsT ............... 70
3.5. DSL for a simple review pattern in PeerPigeon ..................... 71
3.6. PeerPigeon edit review page ............................................. 72
3.7. A formalized grading rubric in Aropä ................................ 73
3.8. Screenshot of the grading view in ALOHA. ............................ 74
3.9. Grading and exploration panel of EduViz. ............................ 75
3.10. Creating a rubric related to automatic evaluation in Agar. ....... 78
3.11. System architecture of ASAP. ......................................... 78
3.12. Architecture of DUESIE ................................................ 79
3.13. Architecture of musical dictation marking tool. .................... 82
3.15. Platform and modules of EASy ........................................ 84
3.16. Two step calculation process for rating of wiki contributions. ... 87
4.1. Survey participants grouped by faculty. .............................. 90
4.2. Learning platforms in use to support university courses at RWTH Aachen University. ........................................... 91
4.3. Assessed activities in university courses at RWTH Aachen University. ................................................................. 91
4.4. Rate of technology enhancement used to support assessment activities. ................................................................. 92
4.5. Rating scales used for grading activities. ................................................................. 93
4.6. Goals which can be achieved by working on activities. ................................................................. 93
4.7. Criteria to fulfill for reaching a course goal. ................................................................. 94
4.8. Tools used to support grading processes. ................................................................. 94
4.9. Applied approaches for assignment management. ................................................................. 99

5.1. Diversity of participants of the requirements survey in phase one. ......................................................... 106
5.2. Opinions about utilization of web 2.0 services for assessment. ......................................................... 107
5.3. Currently used web 2.0 applications during assessment tasks. ......................................................... 108
5.4. Students’ and teachers’ suggestions for material submission processes. ......................................................... 108
5.5. Available features of web 2.0 tools for assessment. ......................................................... 109
5.6. Missing features of web 2.0 tools for assessment. ......................................................... 110
5.7. Threats of using external services for assessment. ................................................................. 110
5.8. Predictions for assessment in 10 years. ................................................................. 111
5.9. Survey participants’ experiences with assessment. ................................................................. 112
5.10. Effects of using web 2.0 tools for assessment. ................................................................. 114
5.11. Types of using web 2.0 services for assessment. ................................................................. 116
5.12. Relevance of assessment features for YouTube and DropBox. ......................................................... 117
5.13. Importance of assessment features for YouTube and DropBox. ......................................................... 117
5.14. Visibility of individual authorship or contributions to group work. ......................................................... 118
5.15. Preferences for channels of feedback delivery. ................................................................. 119
5.16. Needed process automation. ................................................................. 119
5.17. Use cases concerning assessment activities. ................................................................. 122
5.18. Use cases concerning marking. ................................................................. 123
5.19. Use cases concerning group assessment. ................................................................. 124
5.20. Use cases concerning peer assessment. ................................................................. 125
5.21. Use cases concerning self-assessment. ................................................................. 126
5.22. Use cases concerning automatic assessment. ................................................................. 127
5.23. Use cases concerning feedback. ................................................................. 128
5.24. Use cases concerning integration of Assessment 2.0. ......................................................... 129
5.25. Use cases concerning hosting scenarios. ................................................................. 131
5.26. Use cases concerning interoperability. ................................................................. 132

6.1. Conceptual architecture of AMSeL ................................................................. 136
6.2. Modular extension within an assessment tool. ................................................................. 138
6.3. Event-based services for encapsulation of functionality. ......................................................... 139
6.4. An example service for video integration from the cloud. ......................................................... 139
6.5. Design of an assessment service manager. ................................................................. 140
List of Figures

6.6. Modular architecture for assignment submission with exemplary extensions and cloud-service integration. ........................................... 142
6.7. Basic timing process for assignments and solutions. .......................... 142
6.8. Design of the assignment management base tool. .............................. 143
6.9. Design of the group management module. ....................................... 144
6.10. A graph showing an exemplary correction process for source code correction. ................................................................. 146
6.11. Abstraction of the allocation process in the peer review life cycle. ......... 147
6.12. Design for integration of cloud services to assessment tools. .................. 148
6.13. Static model for the example scenario with wiki contributions. ............... 150
6.15. Elements of a criteria graph. .................................................... 153
6.16. An example for a criteria graph. .............................................. 154
6.17. Design of the gradebook for result and criteria management. ............... 155

7.1. Classification of Portals .................................................................. 158
7.2. Examples of web portals: iGoogle and Cooks United ......................... 159
7.3. Fraunhofer PADEM portal software reference architecture 2.0 .......................... 160
7.4. Magic Quadrant for Horizontal Portals ........................................... 163
7.5. Generic architecture of L²P. ....................................................... 165
7.6. Impression of a L²P-Course-Room ................................................ 165
7.7. Growing usage of L²P. ............................................................ 166
7.8. SharePoint 2010 Platform Stack. ................................................ 168
7.9. SharePoint 2010 Server Architecture ............................................ 169
7.10. SharePoint 2010 Site Architecture ............................................... 170
7.11. Types of workflows in Windows Workflow Foundation and SharePoint 174
7.12. SharePoint workflow deployment process. .................................... 175

8.1. Event service pattern for SharePoint-based applications. ..................... 179
8.2. A service manager implementation for student events. ......................... 180
8.3. Declarative factory pattern for SharePoint-based applications. ................. 181
8.4. Elements involved to realize custom rating scales. ............................... 185
8.5. Screenshots of lists for handling custom rating scales. ......................... 185
8.6. Abstracted assignment life cycle in the assignment management tool V1. 186
8.7. Structure of the assignment management tool V1. .............................. 187
8.8. Edit form of an assignment item (V1). ......................................... 188
8.9. Display form of a solution item (V1). .......................................... 189
8.10. Structure of the assignment management tool V2. ............................ 191
8.11. Overview page of the assignment management tool V2. ..................... 192
8.12. Advanced marking schemes in the assignment management tool V2. ..... 193
8.13. Modular extensible frontend for handling related documents in V2. ....... 193

291
List of Figures

8.15. An assignment workspace in basic configuration ......................... 196
8.16. Main area of a solution workspace in basic configuration .............. 197
8.17. Exemplary extensions for the assignment management tool in AMSeL 198
8.18. Structure of the group management module ............................. 199
8.19. Creation form for elements in the group management module .......... 201
8.20. Example for a teachers’ overview of group sets and rules .............. 201
8.21. Exemplary extensions for the assignment management tool in AMSeL 204
8.22. Exemplary correction flows for the assignment management tool in 206
     AMSeL
8.23. Structure of the YouTube-Integration to solution workspaces in the 207
     assignment management.
8.24. Screenshot of a solution workspace in the assignment management 208
     system with integrated videos consumed from YouTube.
8.25. Structure of the assessment tool for wiki contributions 209
8.26. Screenshot of a prototypical marking panel for wiki contributions ..... 210
8.27. Structure of the L²P Gradebook ...................................... 212
8.28. Structure of the extensible import mechanism within the L²P Grade- 213
     book
8.29. The result list in the L²P gradebook .................................. 214
8.30. Bulk edit page in the L²P gradebook .................................. 215
8.31. The personalized gradebook page for a student .......................... 215
8.32. Exemplary tools and modules in the SharePoint-based implementa- 217
     tion of AMSeL

9.1. Satisfaction of students, teachers, and tutors with V1 ..................... 227
9.2. Number of used assignment management tools in L²P per term ........ 228
9.3. Students’ opinions using solution workspaces and related behavior. 229
9.4. Students’ experiences with direct feedback and related opinions ..... 229
9.5. Allowed types of submission per course .................................. 230
9.6. Number of gradebook pilot installations in L²P per term ................. 231
9.7. Default criterion for all seminar of an institute for communication 232
     theory
9.8. Students’ recommendation about using the gradebook in other courses (WS 09). .......................................................... 232
9.9. Teachers’ opinions (in WS 10) about using the gradebook in next terms again. .......................................................... 233
9.10. Students’ usage of the gradebook and according recommendations for other courses ................................................... 234
9.11. Course criterion that allows recognition of assignments gathered in former terms ................................................... 235
List of Figures


10.1. Architecture of the UTX e-test middleware. .......................... 243

B.1. Criterion graph for the scenario in the course Complexity Theory. ...... 255
B.2. Criterion graph for the scenario in the course OOSC. ................ 256
B.3. Criterion graph for the scenario in the course IDB. .................... 256
B.4. Criterion graph for the scenario in the course Programming. ........ 257
List of Tables

2.1. Relation between assessment dimensions and purposes. ............. 12
2.2. Covered assessment dimensions of attendance and participation. ... 13
2.3. Covered assessment dimensions of exams. ............................. 14
2.4. Covered assessment dimensions of formative questions. ............... 16
2.5. Covered assessment dimensions of assessment portfolios. ............ 17
2.6. Usage of web 2.0 tools for evidence generation ..................... 40

3.1. Functionalities of analyzed e-test systems. ............................ 63
3.2. Functionalities of analyzed learning management systems. ............ 67
3.3. Functionalities of analyzed process-specific systems. ................ 76
3.4. Functionalities of analyzed domain-specific systems (part 1). ....... 84
3.5. Functionalities of analyzed domain-specific systems (part 2). ...... 85

4.1. Field manual for interviews according to assessment of wiki contribu-
tions. ................................................................. 100

5.1. SWOT table for integration of external services to assessment processes. 113

9.1. Created elements in the assignment management tool V2. ............ 228
9.2. Elements per feedback timing in the assignment management tool V2. 230
9.3. Pilot installations and used functionalities in 2010. ..................... 233
9.4. Pilot installations and used functionalities in 2011. ..................... 234
Listings

2.1. Example of a multiple choice question in IMS QTI ...................... 46

8.1. Exemplary content type definition of a message rule .................. 182
8.2. A basic content type for timed feedback publication .................. 184
8.3. Class representation of a rule to limit the maximal group size ........ 199

9.1. Class representation of a gradebook rule to transform percentages to
     score values ................................................. 236
## Curriculum Vitae

### Persönliche Daten

<table>
<thead>
<tr>
<th>Name</th>
<th>Stalljohann</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vorname</td>
<td>Patrick Wilhelm</td>
</tr>
<tr>
<td>Geburtsdatum</td>
<td>30.11.1981</td>
</tr>
<tr>
<td>Geburtsort</td>
<td>Ibbenbüren</td>
</tr>
<tr>
<td>Staatsangehörigkeit</td>
<td>Deutsch</td>
</tr>
</tbody>
</table>

### Bildungsweg

<table>
<thead>
<tr>
<th>Datum</th>
<th>Studium/Graduation</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/2008 - 09/2012</td>
<td>Promotionsstudium Informatik</td>
<td>Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen</td>
</tr>
<tr>
<td>10/2002 - 03/2007</td>
<td>Diplomstudium Informatik</td>
<td>Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen</td>
</tr>
<tr>
<td>2001</td>
<td>Abitur</td>
<td>Städt. Johannes-Kepler-Gymnasium Ibbenbüren</td>
</tr>
</tbody>
</table>

### Beruflicher Werdegang

<table>
<thead>
<tr>
<th>Datum</th>
<th>Position/Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/2012 - heute</td>
<td>Leiter Softwareentwicklung SharePoint</td>
</tr>
<tr>
<td>03/2005 - 10/2006</td>
<td>Software-/Webentwickler (Studentische Hilfskraft)</td>
</tr>
</tbody>
</table>