AN OPEN-ENDED ASSESSMENT PLATFORM IN A HOSTED E-LEARNING PORTAL

Patrick Stalljohann
Center for Innovative Learning Technologies (CiL), RWTH Aachen University, Aachen, Germany

Keywords: e-Assessment, Assignment submission, Open assessment, LMS, Portal technology, SharePoint, Hosting.

Abstract: Assessment activities play a central role in university courses. Assessment for learning during a semester can support students' performance and motivation, but needs a lot of time and effort for organizational issues. This paper presents a system which tries to combine generic applicability across different domains with the support of highly specialized processes and with the knowledge of specific domains. Portal Technologies are used to provide a university-wide hosting of assessment tools within a single architecture. The connection to cloud-services facilitates the integration of open assessment activities into formal assessment processes. A prototypical implementation demonstrates possible points of extension.

1 INTRODUCTION

Technology enhanced assessment is a topic for many years. Several systems have been developed to support either a specific scenario or common tasks generally. None of these systems seems to be well suited as a hosted platform for a university-wide service. With the aim of providing such a platform, investigations about adaptability for special scenarios, extensibility for future forms of assessment, and the integration of external services are needed. The paper at hand describes a concept of modular *service managers*, which allows evolutionary enhancement of integrated assessment platforms. A prototypical implementation demonstrates the modular integration of a cloud service to an assignment submission tool.

The paper proceeds as follows: Fundamentals of assessment and arising problems of current assessment systems are discussed in section 2. In section 3, an overall concept with loosely coupled components, event-based binding and integration of cloud services is presented. Section 4 describes a prototypical implementation based on a portal technology within a university-wide hosted e-learning platform. In section 5, an overview of related work concerning extensible assessment platforms is given.

2 MOTIVATION

Assessment takes a central place in higher education teaching. On the one hand, final exams are used to evaluate students' performances after a period of learning. This is called "assessment **of** learning". On the other hand, "Assessment **for** learning" improves students' performance and their motivation (Bostock, 2004) during the learning process.

Mainly, three dimensions for the implementation of assessment **for** learning can be distinguished: *type of activity*, *assessor*, and *formation of participants*. Examples for possible types of activities are assignments, tests, or presentations. Beside a teacher, who is the most traditional assessor, activities can be assessed by peers (Peer Assessment), electronic systems (Automatic Assessment), or the learners themselves (Self-Assessment) (Race, 2001). The third dimension contrasts the assessment of an individual performance (Individual Assessment) with the assessment of collaboratively performed activities within a group of students (Group Assessment). However, collaborative assessment brings along some difficulties. One difficulty to assess such group work is to evaluate not only the product as a whole, but to take each students' individual participation or the collaboration itself into account (Elliott, 2007).

Stalljohann P..
AN OPEN-ENDED ASSESSMENT PLATFORM IN A HOSTED E-LEARNING PORTAL.
DOI: 10.5220/0003474503920401
In *Proceedings of the 3rd International Conference on Computer Supported Education* (ATTeL-2011), pages 392-401
ISBN: 978-989-8425-50-8

2.1 Current Assessment Systems

A wide range of different systems have been developed to reduce the required time and effort which comes along especially with Assessment **for** Learning. Furthermore, the enhancement of assessment processes by technology can facilitate new forms of assessment. Figure 1 shows a classification of those systems which can be divided into the categories *Electronic Test Systems*, *Organizational Assignment Systems*, *Domain-Specific Systems*, and *Learning Management Systems*.

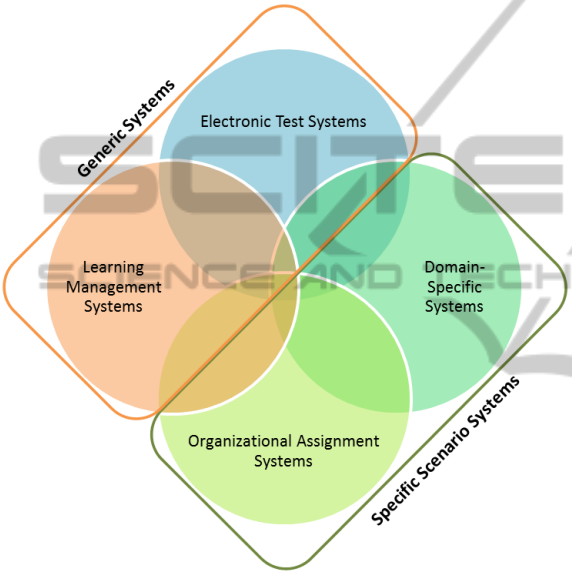

Figure 1: Classification of classic systems with assessment support.

Electronic Test Systems like Hot Potatoes[1] provide computer-based support for creation, management, execution and statistical evaluation of closed questions. Beside the simple electronic form of single- and multiple-choice tests, additional forms of interaction (e.g. drag and drop) and the integration of multimedia content (e.g. videos) are possible. The interoperability of standard tests and questions is supported by the IMS QTI[2] specification. The main advantages of such kinds of assessment are automatic output of marks and feedback as well as the capability for domain independent usage (e.g. for driving licence tests or exams in medical studies). Disadvantages are the limitation to closed questions and static feedback.

Organizational Assignment Systems are more focused to support specific organizational steps of assignment submission. Nowadays web-based systems are developed to handle the whole submission process (Jones et al., 2005). These systems emphasize only a small selection of specific functionality. Example are *Arop"a* (Hamer et al., 2007) and *PeerPigeon* (Millard et al., 2008) which have been constructed to facilitate peer review processes. Other systems are focused on group assessment. The main advantage of those systems is precise assistance for particular scenarios. In many cases, they are restricted to manual correction and feedback generation.

Although open questions cannot be corrected automatically in general, many *Domain-Specific Systems* try to use domain-specific knowledge to (pre-)correct open-ended assignments and provide intelligent feedback. Hollingsworth, for example. used automatic graders for programming classes (Hollingsworth, 1960). Later systems are often web-based and provide different static and dynamic test procedures for a specific programming language, e.g. CourseMaster (Higgins et al., 2003), TRAKLA2 (Malmi and Korhonen, 2004), PASS (Choy et al., 2007) and DUESIE (Hoffmann et al., 2008). Systems specialized to other domains, like musical dictations (Tremblay and Champagne, 2002), essays (Burstein et al., 2004), or mathematical proofs (Gruttmann et al., 2008), are available as well. The main advantage of these systems is domain-specific knowledge which increases automation possibilities for open questions.

Learning Management Systems (LMS) are designed to support a wide range of learning and teaching processes. Systems like *Moodle*[3] or *BlackBoard*[4] combine tools for course planning, communication (e.g. chats and discussion forums), and collaboration (e.g. wiki pages). Modules for the definition and execution of closed question like the above mentioned Electronic Test Systems are often included. Basic support of a standard assignment submission process is often available as well. The main advantages of LMS are the integration with other tools which are not directly related to assessment and the generic applicability for almost any course.

Elliott calls the use of technology to imitate only traditional assessment processes *Assessment 1.5* (Elliott, 2007). He recommends to extend and renew assessment strategies along with the innovative possibilities of *Web 2.0* to reach *Assessment 2.0*, also called *Open Assessment*. When students are active as Web 2.0 authors (Gray et al., 2010), they might profit from the more collaborative manner of *user-generated content*, *the power of the crowd*, *openness*, and *network effects* of Web 2.0. Thus, new forms of assessments

[1] http://hotpot.uvic.ca/ (01.02.2011)
[2] http://www.imsproject.org/question/ (01.02.2011)
[3] http://moodle.org/ (01.02.2011)
[4] http://www.blackboard.com/ (01.02.2011)

have to be developed, like the assessment of activity in online discussions (Vonderwell et al., 2007), wiki pages (Cubric, 2007; Lie et al., 2010), or blogs (Lee and Allen, 2006). The main advantage of Open Assessment is a personalised and more authentic assessment scenario utilizing online tools of students' everyday lives.

2.2 Problems and Objectives

Systems of each above mentioned category conduce a special need and have a right to exist. But the parallel usage of several systems to support the overall assessment process of a course is problematic. It is therefore desirable to have a central platform for the support of various settings and the integration of external systems, which could be provided a consistent tool for several courses. One single system could not cover all current and upcoming new forms of assessment, what caused the multiplicity of currently available systems. Therefore, *extensibility* for various purposes is a critical factor. Especially, *integration of external services* to allow for more open assessment scenarios is an objective of the work presented. Another objective is a more comprehensive usage, rather than the application for only some different scenarios in a single course or some related ones. Thus, such a platform has to be ready for a *hosting scenario*, e.g. for a whole university or perhaps as a service for other institutions.

Most LMS and some test systems provide an *extensible* infrastructure to include new modules such as additional types of tests, and especially new forms of interaction for closed questions. A more open system should be prepared for extensions at all levels of the assessment process.

One point of extension could be the integration of available cloud services for content creation and publishing on the Internet, to facilitate *Open Assessment* scenarios with common Web 2.0 tools. These services are more suitable within the overall assessment process than used separately. Furthermore, these services could be enhanced with additional functionality (e.g. peer review, grouping or marking) within a central platform.

To increase the usability, security, and reliability as well as ease the operation and maintenance of such a platform, it should be constructed to run in a *Hosting Environment*. Thus, it can be provided as part of a university-wide platform. The difficulty is to provide the best-fitting service for most courses and domains. For this purpose, the system should be extremely adaptable without overstraining the users. For instance, the extension for automatic correction of source code could be useful in the programming course but probably not in the course for English poem writing.

An approach to reach these objectives for an open-ended assessment platform will be presented in the following sections.

3 CONCEPTUAL APPROACH

For the definition of an architecture for a new assessment platform, related models or reference architectures have to be taken into account. Evaluations of some models show that they are either applicable to a specific subset of assessment processes only or they are too general to cover the specific requirements of extensibility.

3.1 Reference Models

The *Four Process Model* (Almond et al., 2002) divides assessment into the four processes *Activity Selection*, *Presentation*, *Response Processing*, and *Summary Scoring*. It allows a flexible combination of different tools for each of the four processes. This model is focused on individual test scenarios. The modularity is limited to the four processes and reuse or extension of given components are not part of the model.

The *Educational Model* (Joosten-ten Brinke et al., 2007) describes a static model of the assessment model in six parts: *Assessment Design*, *Item Construction*, *Assessment Construction*, *Assessment Run*, *Response Rating* and *Decision Making*. The static nature of the model conflicts with idea of a highly extensible platform.

The *e-Framework Reference Model for Assessment (FREMA)* (Wills et al., 2009) tries to describe the whole assessment domain. An overview about current dimensions of assessment is given using topic maps. While the suggested service oriented architecture allows an easy replacement of different services, the modularity is limited to the currently known domains of assessment. The integration of emergent technologies or fine-grained extensions to existing functionalities are not covered by this reference model.

Even though the above mentioned models are not directly applicable, several of their process descriptions can be adopted.

3.2 Evolutionary Extension Modules

Evolution of technology and services leads to the need of evolutionarily extensional systems. Those systems

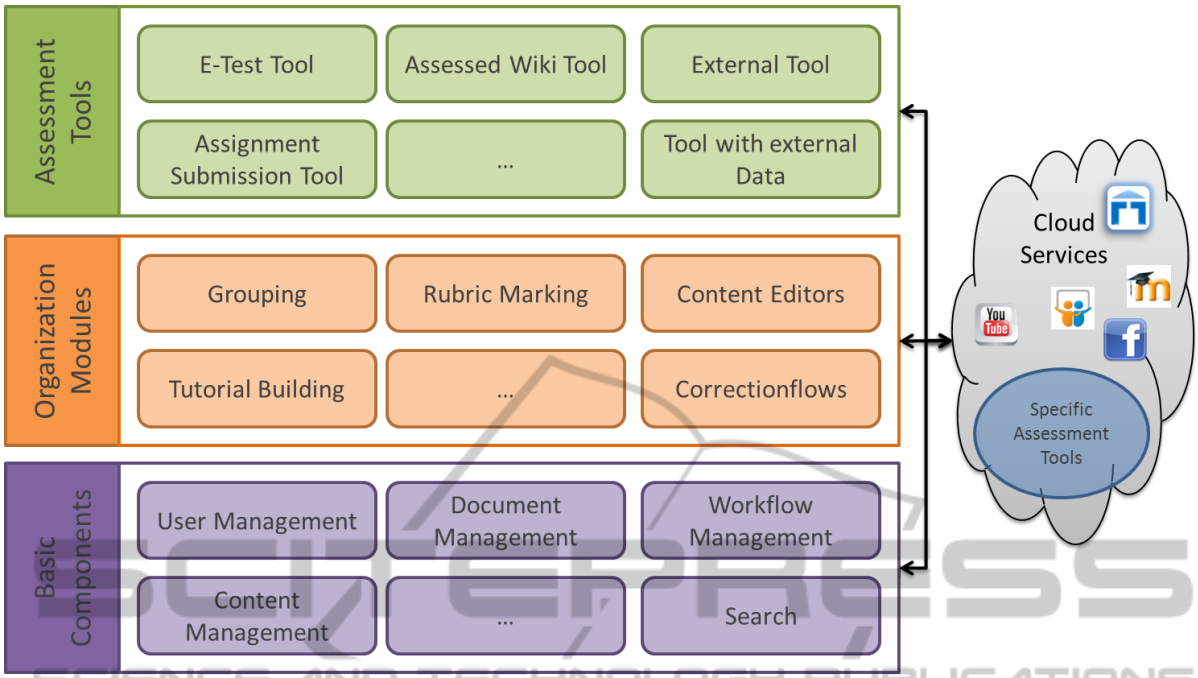

Figure 2: Open Ended Assessment Platform.

have to allow flexible points of extension to avoid the development of completely new systems with a single advancement and several reimplementations of available functionality.

Figure 2 presents an architecture with modularity of mainly three dimensions: *assessment tools*, *organization modules*, and *basic components*. The most general forms of reuse and extensibility are achieved by the integration of *assessment tools* as a whole, either internal or external ones. For instance, an existing test tool can be integrated into the platform as well as a new tool for the assessment of collaborative activity in wiki pages.

Organization modules provide reusable functionality for a special purpose of a typical step in an assessment process. For example, 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. (Semi-)Automatic feedback and peer review as special kinds of correction could be integrated with *correctionflows*, which are modular definitions of correction processes like within the eAixessor-system (Altenbernd-Giani et al., 2008) with the use of a workflow engine. These modules for organization can reuse basic components like *user management* or *document management*. Many available platforms provide different kinds of those components, which should be extensible as well. The integration of external cloud services is considered on all levels of the platform, e.g. with use of a whole external system or integrated with an organization module.

An architecture is needed to organize and integrate functionality among modules (Baldwin and Clark, 2000), which are weakly connected by definition. However, it is difficult not to limit the extensibility within the architecture. Therefore an approach for the modular integration of services, which themselves can handle modules in their specific area of functionality, has been developed.

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 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.

Figure 4 shows an example of a *video service*

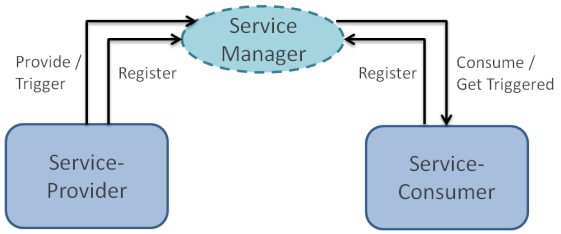

Figure 3: Event-based services for encapsulation of functionality.

manager, which can be used by a module for the extension of an assignment submission tool with videos from different services. Providers for this service could be an *internal video storage* as well as *cloud connectors* for the integration of YouTube[5] or MyVideo[6], for instance. Other examples could be services for grouping or assessed activities in general.

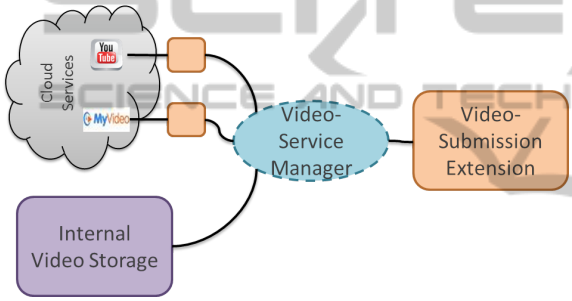

Figure 4: An example service for video integration from the cloud.

An example for the use of services, basic components and modules for the evolutionary development of an assignment submission tool is illustrated in figure 5. It is assumed that a *basic assignment submission tool* has been developed to handle the publication of assignments and the acceptance of individually submitted responses. This tool serves as a provider of an *assignment service* which is linked to a more generic *assessment service*. Functions for getting a list of assignments are provided by this service. Events for reaching a publication date or a deadline for submission can be forwarded to consumers. The main idea is to enhance the basic tool with additional functionality without modifying its basic source code. Several modules can be created to build an extended tool, which now works as a provider for the assignment service as a whole. Advantages of this modular approach are that several services, modules, and components can be used with less dependency. For example, the basic assessment tool and the grouping tool are not directly linked to each other, but the knowledge of groups is needed for *group assessment* as well as for *peer assessment*. Thus, corresponding extensions are consumers of a *group service*, which can be provided by the grouping tool. This tool can be replaced by another tool without having to alter the extensions. Another new requirement for the assignment submission tool could be that students are organized in *tutorials*, in which only an associated tutor is allowed to correct their submissions. Therefore, a *tutorial service*, which is linked to the group service, is used to provide an extension for *tutorial management* with the partition of students into tutorials. Because the basic tool is built on the basic components of a platform (compare figure 2), a *Java editor extension* can be integrated into the basic content management component what leads to integration to the assignment submission tool as well. The integration of external services and content from the cloud works the same way. As shown in figure 4, an open video assessment extension can be used to enable students to include their online published videos in their assignment submissions.

4 IMPLEMENTATION

Since 2007 the tailor-made, university-wide e-learning portal L^{2}P supports presence teaching and is available for all students and teachers at RWTH Aachen University as a central service[7] (Schroeder et al., 2008). A so called virtual course room can be set up to enhance a lecture with information, communication and collaboration functionality. Started with 270 virtual course rooms in summer term 2007, there are about 2.200 lectures from eight different faculties in winter term 2010/10 which are supported by a virtual course room. Overall, the portal reached a number of about 11.000 virtual course rooms during the last eight terms. Approximately 25.000 enrolments are registered per term, what means that every user is enrolled in six course rooms on average. 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.

L^{2}P has been developed based on Microsoft SharePoint Server 2007, a portal technology which contains several standard functionalities such as user management, document management, content management, workflow management and search. Using such a well established platform facilitates the usage as a hosting service and increases security, reliabil-

[5]http://www.youtube.com/ (01.02.2011)
[6]http://www.myvideo.de/ (01.02.2011)
[7]http://www.elearning.rwth-aachen.de/ (01.02.2011)

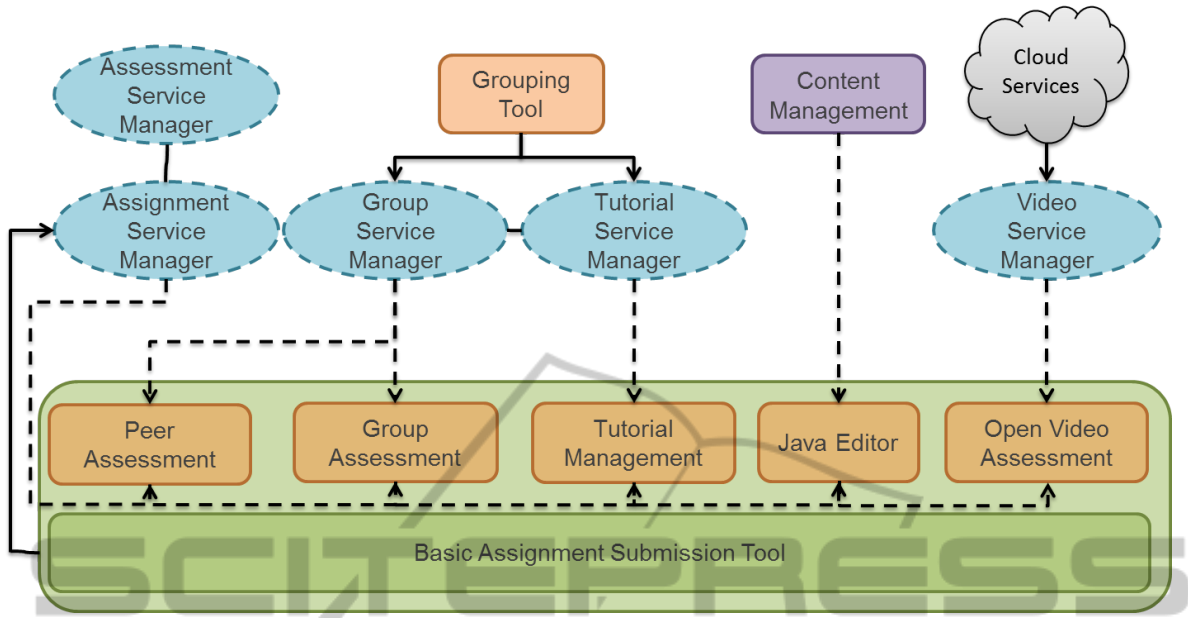

Figure 5: Modular architecture for assignment submission with exemplary extensions and cloud-service integration.

ity, and maintainability of the system. Additionally, it provides multiple deployment mechanisms to add user defined functionalities, which allow the development of custom integrated applications. Among others, these possibilities has been used to integrate specific assessment tools. A custom assignment submission system (Dyckhoff et al., 2008) has been developed, which has been improved stepwise with several functionalities like the opportunity to provide direct feedback (Stalljohann et al., 2009). The management of students' marking results and criteria for passing a lecture is handled by a custom integrated grade book (Stalljohann and Schroeder, 2010), which currently runs in several pilot installations.

4.1 Portal Deployment Mechanisms

SharePoint employs a hierarchical architecture starting with a whole *farm*, which contains several *web applications*. Within these web applications there are multiple *sites*, which build independent workspaces. In L^{2}P , a separate site is used for each virtual course room. These sites are structured itself by a tree of so called *webs* or rather *sub webs*. They can be used to define different areas, e.g. for information, learning materials, literature management, or assessment. The 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, and others. Lists can handle meta data and files as their items. *Fields*, corresponding to columns in a SQL table, represent options of an items property, like the name, data type, rendering and others. To handle different types of items within a list, they can have different *content types*, which define the properties (fields) of an item. The content types are organized with help of inheritance. A list for literature management for example, 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. The *authorization* of users can be handled in different ways, like with an active directory, forms authentication with a custom database, or custom providers. The *authorization* is managed with the user of *groups*, *role definitions*, which are assigned to a whole site, a web, a list, or a single list item. The combined presentation of information, actions, and settings from several sources is realized by *webparts*[8]. They encapsulate specific data, logic, as well as its form of presentation. A weather webpart for example can be placed on every page within the system beside other webparts. A connection to webparts which handles a list of cities could facilitate a kind of filter.

There are several points for extension already available within SharePoint which allow for the development of evolutionary extensions, which is intended

[8]Names for webparts in other systems are portlets, gadgets, or widgets.

related to the concept in section 3. *Custom field types* allow the definition of new data types, their validation and presentation for usage as a field in a list. Fields, content types, and templates for lists can be defined declaratively. The deployment of these elements is managed with so called *features*, which can be installed to a SharePoint farm and activated in a specific scope. Possible scopes are the entire farm, a web application, a site or a single web. Others elements that can be deployed with a feature are for example *webparts*, *event receivers*, *workflow definitions*, *custom actions*, or *external content types*. *Webparts* contain code for the presentation of information of data and the execution of actions. *Event receivers* are classes which can be bound to a specific component, e.g. a list item or a web, to execute custom logic, when an event was triggered. *Workflows* and related *workflow activities* can be developed to specify a modular process, which can be executed for an item. 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, ...), and the corresponding action (execution of server code, client script or a link). *External lists* or rather *External content types* are a mechanism to integrate external systems by consuming a web service, a database, or a custom connector to arbitrary sources. This integration allows the presentation and handling of external data just like data in an ordinary list. All of these components and several more ones allow the creation of modular extensions for the platform, which can packaged as a *solution* for reuse in different farms.

4.2 Developing Modular Tools

As mentioned above, those deployment techniques have been used to build different custom applications for assessment purposes on top of SharePoint. By using these mechanisms, the whole 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. The main problems with the first implementations of the systems are the reusability of functionalities for other tools and the extensibility with new functionalities, which are the result of frequently added requirements. For example, the tool for assignment submission contains a grouping functionality, which allows the dynamic building of groups for cooperative assignment submission. Adding other strategies for grouping as well as using these groups for other tools is not straight forward.

A restructuring of these applications with the use of *service managers* (see figure 3) provides a way to solve these problems. Some aspects for the development of a service manager architecture are already available. The paradigm is similar to *event receivers* in SharePoint, but these are limited to a fixed set of elements and events that are triggered (e.g. an item was added to a list). The same strategy is now used to extend additional services with related semantic events, like an assignment service which triggers an event when the deadline was reached or a student's submission was handed in. A new service manager is build as a class which is connected to a feature. With this, the functionality of the service manager can be (de-)activated at a specific scope. A provider as well as a consumer reference the service manager project and register themselves with their fully qualified names. If the service of a provider is consumed or an event is triggered, the service manager creates an instance of the corresponding classes by reflection. This way, new services, providers and consumers can be deployed, installed, and activated in a SharePoint farm modularly.

The shift to this new paradigm has been demonstrated by a reimplementation of the assignment submission system on base of SharePoint 2010. Figure 6 illustrates the workspace for a submission from the perspective of a teacher or tutor. The whole tool is integrated as a separate sub web and can be reached from the left navigation bar (entry *assessment*). The main area is a combination of several webparts. By default, there are webparts for metadata (e.g. the student, the related assignment, and the submission state), documents in the solution, correction documents created by a tutor, a list of textual feedback, and the given marks. Additionally, an extension for the integration of videos from YouTube places an extra webpart related to an external list for the student's videos on the page. New videos for this submission can be directly added, deleted, and modified within the platform. Using the same service on the YouTube webpage is also possible, by tagging the videos as a submission. Other extensions, such as providing a list for sample solutions for each assignment or allowing the building of groups, have been separated into different modules as well. The main point here is that these extensions are able to consume the assignment service. The grouping extension could for example synchronize the composition of groups from the grouping service. After the event that the deadline has been reached is triggered, the groups for this assignment are fixed within the submission tool.

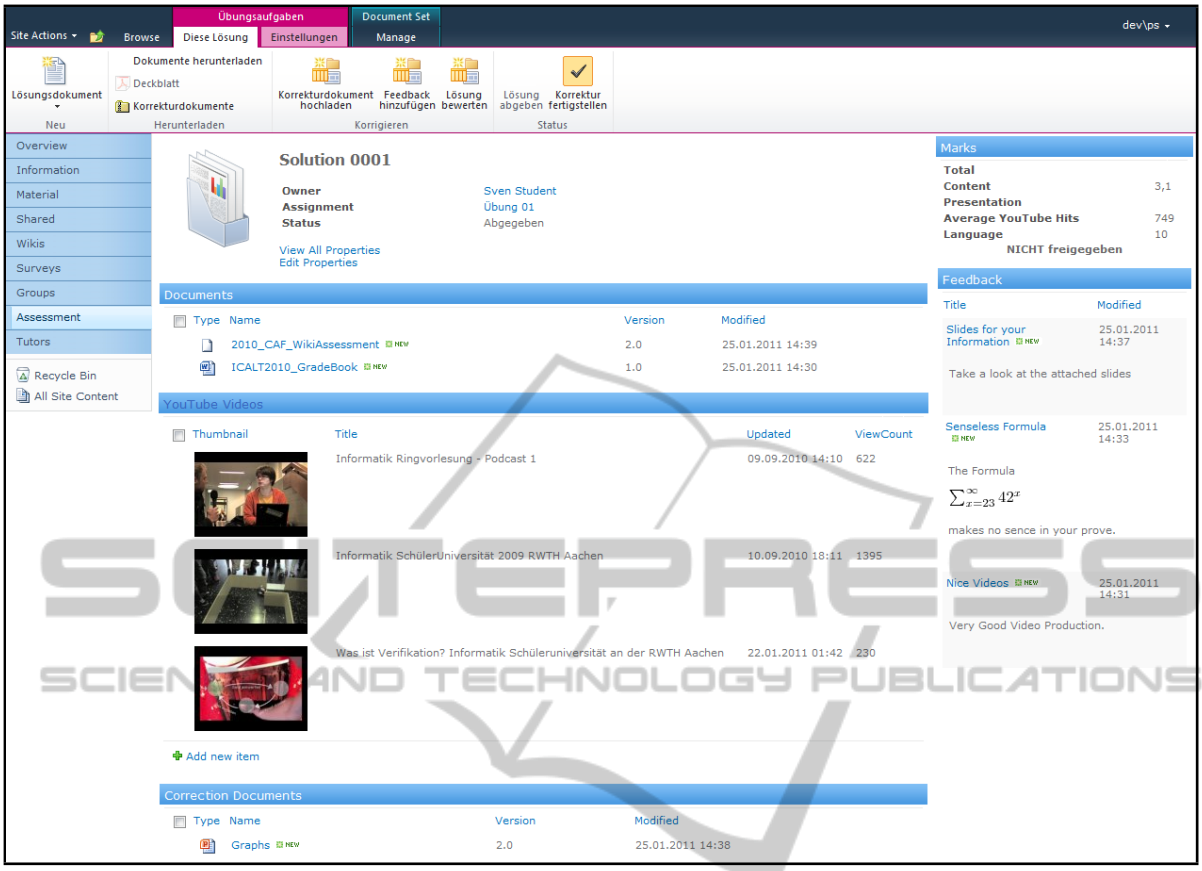

Figure 6: Using YouTube Videos in Assignment Submissions.

5 RELATED WORK

The *Flexible Assignment System (FAST)* tries to provide a flexible support for different assessment scenarios by the use of collaboration scripts (Topcuoglu, 2006). 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, can be defined with the collaboration script. But this flexibility is limited to this process description, such that additional functionality like the integration of external services or self-regulated grouping has to be added within the basic system.

A more flexible approach concerning domain-specific assessment support is provided by *EASy* (Gruttmann, 2010). Started as a system for the definition, submission and (semi-)automatic correction of mathematical proofs, it was extended to a platform to host modules for different domains (see 7). The assessment process is divided into *item construction*, *response construction*, and *correction and marking*. A new 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. Disadvantages of this system are the limitation to individual assessment by tutors. Grouping, peer assessment, or

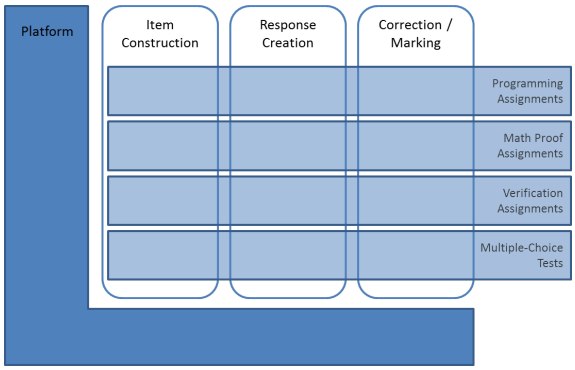

Figure 7: Platform and modules of EASy (based on (Gruttmann, 2010)).

other forms of assessment are neither available nor modularly integrable. The integration of collaborative activities or open assessment is not covered either. Furthermore, the two systems FAST and EASy are standalone systems, which implement their own user management. Hence, they are not directly integrated into an e-learning system to support other purposes than assessment as well.

More integrated approaches, which leverage the functionality of *content management systems (CMS)*, are for example the systems *iPal* (Pinkwart et al., 2005) and *EduComponents* (Rösner et al., 2007). iPal extends the standard functionalities of the *PostNuke*[9] CMS with modules for building tutorials, as well as handling assignments and lecture notes. These provide general support for typical teaching scenarios, but do not offer an additional extension model above the standards of the CMS. The aim of EduComponents was to enhance the CMS *Plone*[10] with additional components to support different learning and assessment activities. It is a collection of the tools *ECLecture* (general course information), *ECQuiz* (multiple-choice tests), *ECAssignmentBox* (assignment submission with tutor marking), and *ECAutoAssignmentBox* (assignment submission with automatic marking). The latter allows extension by different so called *back-ends*, which encapsulate the domain-specific logic for the correction of assignment submissions, e.g. one back-end for Java source code and another one for UML diagrams. The reuse of an existing back-end for building a new one on top of it is not intended.

6 CONCLUSIONS AND FUTURE WORK

Assessment takes a big part in formal learning, especially in higher education. Several tools have been developed for technology enhancement of special assessment processes. These systems, standalone as well as integrated to general e-learning platforms, support only a subset of possible assessment processes and it is obvious that they lag behind new forms of assessment. While platforms like CMS and portals use well established functionalities and deployment techniques to be highly extensible, extensions on base of these platforms do not or limitedly offer modular architectures themselves.

The paper at hand describes a concept of modular *service managers*, which allow an evolutionary enhancement of an integrated assessment platform. First parts of this concept have been proven by an implementation based on Microsoft SharePoint portal technology. The example of a stepwise enhancement of a basic assignment submission tool shows how open assessment processes, which integrates YouTube videos with students' assignment submissions, can be integrated by an extension.

Future work includes pilot installations using the new version of the system in different lectures and related evaluation. The development and evaluation of other assessment tools, e.g. for assessment of collaborative learning in wiki pages or more informal assessment settings connected to cloud services and social networks, are objectives as well. Furthermore, the usability of the contained sub-tools and extensions for general usage as well as with mobile devices needs to be evaluated. While the platform is currently hosted for multiple institutions of a single university, multi tenancy support could allow a hosting service for different universities. The approach of evolutionary module extension could perhaps be used for each tenant differently.

[9]http://www.pn-cms.de/ (01.02.2011)

[10]http://plone.org/

REFERENCES

Almond, R., Steinberg, L., and Mislevy, R. (2002). Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology, Learning, and Assessment*, 1(5):164.

Altenbernd-Giani, E., Schroeder, U., and Stalljohann, P. (2008). eAixessor - A Modular Framework for Automatic Assessment of Weekly Assignments in Higher Education. In Uskov, V., editor, *Proceedings of the 7th IASTED International Conference for Web Based-Education (WBE 2008)*, pages 70–75. ACTA Press.

Baldwin, C. Y. and Clark, K. B. (2000). *Design Rules: The power of modularity*. MIT Press, Cambridge, MA.

Bostock, S. (2004). Motivation and electronic assessment. *Effective learning and teaching in computing*, pages 86–99.

Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: The Criterion online writing service. *AI Magazine*, 25(3):27.

Choy, M., Lam, S., Poon, C. K., Poon, C., Wang, F. L., Wang, F., Yu, Y., Yu, Y. T., and Yuen, L. (2007). Towards Blended Learning of Computer Programming Supported by an Automated System. In Fong, J. and Wang, F. L., editors, *Proceedings of Workshop on Blended Learning' 2007*, pages 9–18. Pearson.

Cubric, M. (2007). Using wikis for summative and formative assessment. In *Re-Engineering Assessment Practices (REAP) International Online Conference*.

Dyckhoff, A., Rohde, P., and Stalljohann, P. (2008). An integrated web-based exercise module. In Uskov, V.,

editor, *Proceedings of the 11th IASTED International Conference on Computers and Advanced Technology in Education (CATE 2008)*.

Elliott, B. (2007). Modernising Assessment: the use of Web 2.0 for formative and summative assessment. In Khandia, F., editor, *11th CAA International Computer Assisted Assessment Conference*, pages 179–188, Loughborough.

Gray, K., Thompson, C., Sheard, J., Clerehan, R., and Hamilton, M. (2010). Students as Web 2 . 0 authors : Implications for assessment design and conduct. *Australasian Journal of Educational Technology*, 26(1):105–122.

Gruttmann, S. J. (2010). *Formatives E-Assessment in der Hochschullehre Computerunterstützte Lernfortschrittskontrollen im Informatikstudium*. Dissertation, Westfälische Wilhelms-Universität Münster.

Gruttmann, S. J., Kuchen, H., and Böhm, D. (2008). An E-Assessment System for Mathematical Proofs. In *Proceedings of the 11th IASTED International Conference on Computers and Advanced Technology in Education (CATE 2008)*, volume 614, pages 090–120.

Hamer, J., Kell, C., and Spence, F. (2007). Peer Assessment Using Aropä. In *Proceedings of the ninth Australasian conference on Computing education*, page 54. Australian Computer Society, Inc.

Higgins, C., Hegazy, T., Symeonidis, P., and Tsintsifas, A. (2003). The coursemarker cba system: Improvements over ceilidh. *Education and Information Technologies*, 8(3):287–304.

Hoffmann, A., Quast, A., and Wiesmüller, R. (2008). Online-Übungssystem für die Programmierausbildung zur Einführung in die Informatik. In Seehusen, S., Lucke, U., and Fischer, S., editors, *DeLFI 2008: Tagungsband der 6. e-Learning Fachtagung Informatik*, pages 173–184, Bonn. Gesellschaft für Informatik.

Hollingsworth, J. (1960). Automatic graders for programming classes. *Commun. ACM*, 3:528–529.

Jones, D., Cranston, M., Behrens, S., and Jamieson, K. (2005). What makes ICT implementation successful: A case study of online assignment submission. *Open Learning and Distance Learning Association of Australasia (ODLAA), University of South Australia*, pages 1–11.

Joosten-ten Brinke, D., Van Bruggen, J., Hermans, H., Burgers, J., Giesbers, B., Koper, R., and Latour, I. (2007). Modeling assessment for re-use of traditional and new types of assessment. *Computers in Human Behavior*, 23(6):2721–2741.

Lee, J. and Allen, K. (2006). Edublogs as an online assessment tool. In *Proceedings of the 4th International Conference on Multimedia and Information and Communication Technologies in Education*, pages 391–395.

Lie, C., Dyckhoff, A. L., and Stalljohann, P. (2010). Supporting quality assessment of collaborative learning activities with wiki pages. In Auer, M. E. and Schreurs, J., editors, *Proceedings of the 13th International Conference on Interactive Computer Aided Learning (ICL 2010)*, pages 1102–1109, Hasselt. International Association of Online Engineering.

Malmi, L. and Korhonen, A. (2004). Automatic feedback and resubmissions as learning aid. In *IEEE International Conference on Advanced Learning Technologies*, pages 186–190. IEEE.

Millard, D., Sinclair, P., and Newman, D. (2008). PeerPigeon: A Web Application to Support Generalised Peer Review. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008*, pages 3824–3836. Chesapeake.

Pinkwart, N., Harrer, A., Lohmann, S., and Vetter, S. (2005). Integrating portal based support tools to foster learning communities in university courses. In Uskov, V., editor, *Proceedings of the 4th International Conference on Web-Based Education*, volume 461, pages 201–206. ACTA Press.

Race, P. (2001). *A briefing on self, peer and group assessment*. Learning and Teaching Support Network.

Rösner, D., Piotrowski, M., and Amelung, M. (2007). A sustainable learning environment based on an open source content management. In *Proceedings of the German e-Science Conference*.

Schroeder, U., Rohde, P., and Gebhardt, M. (2008). *eStrategy der RWTH Aachen: nahtlos durchgängige eLearning-Prozesse*. Waxmann.

Stalljohann, P., Altenbernd-Giani, E., Dyckhoff, A., Rohde, P., and Schroeder, U. (2009). Feedback mit einem webbasierten Übungsbetrieb. In Schwill, A. and Apostolopoulos, N., editors, *DeLFI 2009: Tagungsband der 7. e-Learning Fachtagung Informatik*, pages 283–294, Bonn. Gesellschaft für Informatik.

Stalljohann, P. and Schroeder, U. (2010). A portal-based gradebook. In *Proceeding of the 10th IEEE International Conference on Advanced Learning Technologies (ICALT 2010)*. In Press.

Topcuoglu, H. (2006). FAST-Flexible Assignment System. In *Proceedings of E-Learn*.

Tremblay, G. and Champagne, F. (2002). Automatic Marking of Musical Dictations By Applying the Edit Distance Algorithm On a Symbolic Music Representation. In *Musical Application Using XML (MAX2002)*, pages 11–17, Milan, Italy. IEEE Computer Society Technical Commitee on Computer Generated Music.

Vonderwell, S., Liang, X., and Alderman, K. (2007). Asynchronous discussions and assessment in online learning. *Journal of Research on Technology in Education*, 39(3):309.

Wills, G., Bailey, C., Davis, H., Gilbert, L., Howard, Y., Jeyes, S., Millard, D., Price, J., Sclater, N., Sherratt, R., Tulloch, I., and Young, R. (2009). An E-Learning Framework for Assessment (FREMA). *Assessment & Evaluation in Higher Education*, 34(3):273–292.