Modular Assessment System for Modern Learning Settings: MASS

Mohammad AL-Smadi
Institute for Information Systems and computer Media,
Graz University of Technology
Brückenkopfgasse 1
8020 Graz, Austria

Christian Gütl
Graz University of Technology, Austria, School of Information Systems, Curtin University of Technology, Perth, WA.
Brückenkopfgasse 1
8020 Graz, Austria

Rajkumar Kannan
Dept. of Computer Science
Bishop Heber College
Tiruchirappalli 620017
Tamil Nadu, India

ABSTRACT

In the last three decades, the field of e-assessment has become more important. Several universities and higher education institutes have started to provide online assessments. The variance in the e-assessment application domains as well as the market competition has caused several e-assessment systems to be developed. As the universities cover different subjects and courses, they may have more than one e-assessment system based on the department and the course. Therefore, more resources and budgeting planes are required. This paper discusses the possibility of a generic and flexible e-assessment system as a way to solve this problem. Furthermore, it investigates the generality and flexibility requirements in the field of e-assessment and provides appropriate architecture for a modular assessment system with reference to these requirements.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Computer-assisted instruction (CAI);

General Terms

Management, Performance, Design, Reliability, Security, Human Factors, Standardization.

Keywords

E-assessment, Assessment framework, Modular assessment system, Assessment services, Service-oriented architecture for assessment.

1. INTRODUCTION

In the 21st century a new age of information has appeared where information and communication technology plays a main role in education and learning society. The option of e-learning and computer-based/assisted assessment packages has started to be promising. Therefore, universities and higher education institutes has become more and more interested in using computers to assist the educational process. Several Learning Management Systems (LMS) and Computer-Based/Assisted Assessment systems (CBA/CAA) have been developed after wards.

CAAs have been developed either based on the items' types they provide or based on the application domains they serve. For the first one, standards and specifications has been published to represent the different types of the assessment items. Although, some of these items suits the application domains such as knowledge assessment in different subjects but there is also a lack in different application domains such as mathematics and skills assessment. Therefore, some assessment systems have been developed to serve specific application domains. Examples of such assessment systems are: mathematics [7], chemistry [5], Algebra [3], and programming assignments [8].

The variance in application domains caused a variance in the developed CAA systems too. Therefore, many of these systems are limited to a specific domain or sometimes to a specific activity in that domain. Universities are teaching several topics that cover different application domains. Having more than one CAA system in the same university for each application domain or at least for related application domains, requires more resources and running costs. Some of the CAA systems don't have a proper design and architecture to get benefit from others. For instance, for each of these systems a student record is required to store the assessment and progress results during the course. If these systems don't have the ability to integrate with the student information system (SIS) in the university all of them will have the same redundant data about students.

Having a generic and flexible CAA system that can integrate with other systems without a need for external interventions or redevelopments, sounds promising to solve this problem. The following research questions will be discussed in this paper:

- Is it possible to have a generic and flexible CAA?
- What are the generality and flexibility requirements?
- What is the appropriate architecture for such system?

To this end, this paper is organized as follows: section 2 presents the assessment process handled by MASS in two sub-sections. MASS context diagram in section 2.1 and MASS main modules in section 2.2. Section 3 shows the variance in the e-assessment application domains as a set of personas. Section 4 investigates the generality and flexibility requirements for an e-assessment system in general and for MASS in specific. Section 5 suggests a service-oriented architecture for MASS as a generic and flexible assessment system. A show case for the eligibility of MASS suggested architecture is presented in section 6. Finally, conclusion and outlook represents the content of section 7.

2. OVERVIEW OF THE ASSESSMENT PROCESS

2.1 MASS Context Diagram

In the context of e-assessment, MASS will interact with its environment as shown in figure 1. The interaction will be handled through the following interfaces:

- **User Agents:** this interface represents MASS possible users, tools, and systems that may interact with the e-assessment system. For instance, assessment systems, LMS, and any other authoring tools can be user agents.
- **Administration:** although the administration is part of the user agents, but it encompasses administration planes and regulations. Therefore, this interface will serve the users responsible for administration as well as for internal business rules and regulations.
- **National bodies:** as e-assessment could be used to serve the national planes and to achieve national goals, this interface has been added for such purposes.
- **Organization and Infrastructure:** this interface will handle the interaction between the system needs and the staff who are responsible for further developments and enhancements. Tasks such as data backup and recovery, system and data security are parts of this interface services.
- **Business Model:** MASS should also have the ability to exchange its assets with other systems such as LMS. For instance tests and test items can be sold to other systems through an e-commerce system.

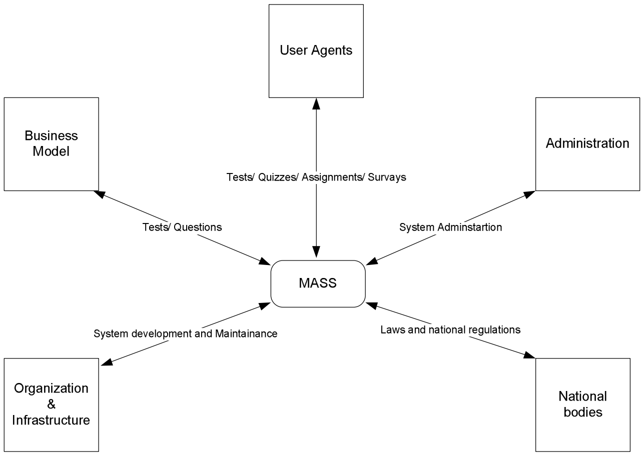
User Agents
Business Model
Administration
Tests/ Quizzes/ Assignments/ Survays
Tests/ Questions
System Adminstartion
MASS
System development and Maintainance
Laws and national regulations
Organization & Infrastructure
National bodies

Figure 1: MASS context diagram

2.2 MASS Main Modules

Similar to many e-assessment systems, MASS has four main modules: Authoring, Scheduling, Delivering, and Reporting. For each of these modules, a set of processes has been defined. The following of these processes as well as their modules during the assessment life cycle depends on the requirements of the application domain as well as the needs of the assessment system. Figure 2, shows those main phases as well as their corresponding processes, and they are as follows:

- **Authoring:** as part of this phase the author user starts with selecting the form of the assessment which can be a test, quiz, assignment, or a survey. After that, she/he defines the objectives behind the assessment items and the assessment process in general. These objectives shall guide the assessment process to achieve the learning goals. Based on those objectives the items' types are chosen. For each item, a feedback can be defined. The feedback can be pre-defined or it can be provided based on the later on marking, grading, or analysis of the learners' answers in the reporting phase. Marking criteria can be defined in this module for each of the assessment items. For instance, rubrics can be used to handle the marking process. In case of the author is preparing a test, the ability of importing items or selecting pre-prepared items from items' banks as well as arranging the items in tests is available. Finally, the items are assured for quality so they can be ready for the next phases as well as to be exchanged with other systems. This process is done by quality assurance bodies who consider matters such as learning goals and objectives and standards-conformation.
- **Scheduling:** after the assessment has been prepared and quality assured, it will be ready for the delivery phase. Before delivery, managing the users who will set the assessment as well as the assessment environment is done during this module. Such processes mainly are done by the timetabler user. The timetabler selects the place where the assessment will be taken and the timing matters such as the assessment date, how long it will stay and how many times it can be repeated or taken?. In case of the assessment will be delivered in a controlled environment the timetabler chooses the invigilators who will monitor the learners during taking the assessment. In the other case of web-based delivery matters such as security, and plagiarism detection should be part of the assessment system.
- **Delivering:** in this module assessments are delivered in different forms: paper-based, web-based, offline delivery, or via third-party such as LMS and different assessment systems. In case of some assessment items are imported from different systems or exported to different systems, matters of exchange should be considered. Web services can be used to deliver such items in both cases of import or export.
- **Reporting:** once the assessment is taken by the learners, answers are gathered and stored in a related database. The answers are then marked according to the defined marking criteria ranging from semi- to fully-automated marking with reference to items types degree of complexity. For instance, multiple choice questions can be fully-automated marked where free answers require some human interventions for final judgments. After

marking the assessment the final marks are assigned to grades and reported to the learners. The learners' results can be further analyzed and reported to decision makers as well as used to moderate and adapt the learning and assessment activities with regards to their goals and objectives.

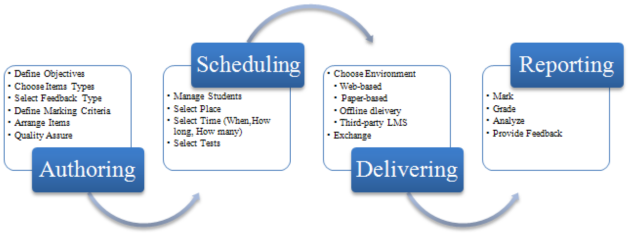

Figure 2: MASS main modules and their corresponding processes

3. REQUIREMENTS GATHERING

In order to define the interactional requirements of MASS users from different application domains, we used 'Personas'. Personas are used to model the users' interactions with the system interfaces, so that the development team will share specific, consistent information about users. They also guide the process of meeting the user requirements through a set of requirements priority levels [6].

Persona I, Tom:

Tom is a lecturer in a university. He is specialist in Mathematics Education and teaches Algebra to undergraduate students. One of his didactic objectives is to use computers to assist students during his courses as well as to deliver Algebraic assessments.

Scenario:

I need an e-assessment tool that can be used to deliver Algebraic assessments and exercises for the students. I believe that when students practice Algebra on computers and do more and more exercises they can easily learn and pass the course.

Persona II, Lora:

Lora is a lecturer in a department of computer science. She has a masters and computers science and she teaches undergraduate students programming courses for beginners.

Scenario:

I am looking for an e-assessment tool that I can use to assess the students programming assignments as well as to provide tests and quizzes for them.

Person III, Leila:

Leila is teaching English as second language training for beginners in a basic level.

Scenario:

I need an e-assessment tool that I can use to provide my students with variety of exercises ranging from multiple-choice to cross-word puzzles and video/Audio based exercises. By this tool the students can access and exercise from home as well as they can assess them self's using dedicated tests designed for such purposes.

Persona IV, Sami:

Sami is a teacher in a high school and he is interested in applying a set of online rubrics to assess the students' results according to a specific criteria.

Scenario:

I am looking for an assessment tool by which I can use rubrics to mark students' essays as well as to define the formal and content aspects of them.

4. GENARALITY AND FLEXIBILITY REQUIREMENTS

4.1 Standards Conformation

A standardized e-assessment system has been defined as the system that its components are designed and implemented according to specific standards and/or specifications. Furthermore, two levels of standardizations have been distinguished as an internal level and an external one. The internal one is related to the conformance of the internal system components and data to standards, where the external is more aware about the ability to exchange the internal components with external systems.

The use of standards facilitates the exchange between different systems as well as provides a common language and methodology of communication between them. Furthermore, standards-conform systems are able to be: *interoperable, manageable, reusable, accessible, durable, scalable, and affordable.* [1].

4.2 Framework

A *framework* represents a rich vocabulary that is used to support people in the domain as well as software developers to overcome the problems encountered through the software development. A framework is used to create a shared language that will describe the problems and their solutions. A framework also supports organizations to develop service-oriented architectures by identifying the main services that these organizations may need to develop their applications. The main aim of a framework is to have the ability to identify the services as well as to assign one or more open standards and specifications for each service. [2].

Service-oriented architectures support the development of modular and flexible systems [4], where the components of the system are flexible to be reused, added, replaced or removed. Furthermore, new systems can be composed from a collection of suitable services. SOA may also support e-assessment and e-learning applications to be more sharable and interoperable [2]. Based on a previous study [1], a Service-Oriented Framework for Assessment (SOFA) has been provided. In SOFA the assessment services (i.e. authoring, scheduling, delivering, and reporting) has been separated from the common services (i.e. user management services, authorization and authentication services, infrastructure services) which can be available in any LMS. This separation helps avoiding redundant system components implementation and encourages the integration with other LMSs. Assessment services and common services can be implemented as web services in a flexible way to be used by other authorized services or systems.

Figure 3, demonstrates how SOFA can be utilized to implement and develop the MASS main modules with reference to the application domain requirements and by using the available domain related web service. By utilizing this architectural framework, MASS can be:

- Generic: it can be used to assess different application domains.
- Flexible: its modules can be easily built from available web services with regards to the application-domain needs.
- Reusable: since its modular and based on service-oriented framework (SOFA). Its parts can be used to build different assessment systems are can be used with different systems and tools.

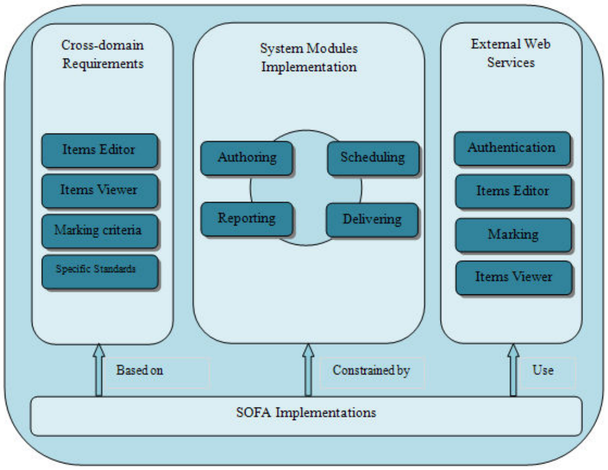

Figure 3: Architectural framework for SOFA

4.3 Cross-domain Requirements

Based on the personas in section 3 and analysis of the application domains specifications with the MASS main modules, the following high level requirements have been gathered:

1. Authoring Requirements:

a. Domain-based editor: this editor will be responsible for authoring test items related to a specific domain. For instance, in the domains of Algebra, physics and chemistry an equations editor is required by the item author where the algebraic formulas have special symbols and formulas that can't be entered by the keyboarded. The possible solution for that can be on the following:

i. Add-on to the MASS system that can be used for such editing.

ii. External web service for such specific editing.

iii. Web service from an algebraic assessment system.

b. Domain-based standards and specifications: for the sake of interoperability the test items must be standard-conform. In some special domains and subjects there are no specifications and standards. Therefore, we have to think in a way to represent such items as XML files based on web services provided by MASS system or the domain available system.

c. Domain-based Marking Criteria: for each authored item the author should also provide marking criteria (i.e. rubric) for handling the marking process. These marking criteria should be provided by the domain-related assessment system. Since the marking process is part of the item evaluation it should take place as part of the domain-related assessment system where the assessment engine should be configured to mark its related items.

d. Domain-based feedback: this depends on the type of the feedback. For instance, if the feedback is just either the answer is right or wrong, it can be prepared from the authoring step. But, if the feedback is some kind if online stepping evaluation of the answer, all the process should be handled on the domain-related system side or via web service.

2. Scheduling Requirements: There are no special requirements for this stage where managing users and selecting the place to deliver the assessment as well as the time for it can be handled by MASS.

3. Delivering Requirements:

a. Domain-based viewer: in specific domains such as algebra, mathematics, physics and chemistry there is a need for special items viewer where the items contains some formulas and symbols. In the scenario of programming code assessment if the tutor is interested in viewing the code in specific features such as structured or colored, then it is recommended to have a special viewer.

b. Domain-based editor: as in the authoring phase, it is also needed here because in this phase the students are asked to answer the questions which may be algebraic answers or special symbols and formulas.

4. Reporting Requirements:

a. Domain-based marking engine: this engine is part of the domain-related assessment system and it has the logic to mark and evaluate the domain related items. Therefore, such engine is required in the MASS system in order to evaluate the domain related answers. This problem can be solved as a web service provided by the target system. So the answers will be passed from MASS to this web service and the results are returned back to MASS so that they can be analyzed grade and reported to students.

4.4 Web Services

The domain-related systems (such as algebra assessment system, programming assignment assessment tool…) must provide their domain-related services (such as algebra marking, equation editor) as web services. The provided web services can be used by the framework to serve the assessment system modules for achieving the domain-related requirements.

5. SERVICE-ORIENTED ARCHETICTURE

Investigating the cross-domain requirements and the solutions that have been suggested for some of them, a service-oriented architecture for MASS has been provided. Figure 4 shows the abstract architecture for MASS where a web services layer has been added. This layer is supposed to handle the application domain requirements through a set of web services provided by domain related assessment systems. Once the domain related systems are standard-conform and they provide their services as web services assessment systems can use these services and serve more and more application domains. Therefore, such systems will be more independent from the application domain and more generic and flexible.

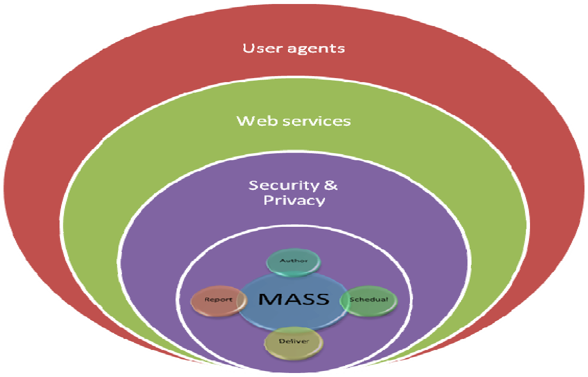

Figure 4: Abstract Architecture for MASS

As demonstrated in figure 5, a Multi-tier service-oriented architecture has been designed for MASS. This architecture consists of four Tiers: client application tier, business tier, service tier, and resource tier.

5.1 Client Application Tier

The client application tier forms the front tier where the user agents interact with the system. Matters such as security and privacy of data and processes are handled in this tier. Furthermore, unauthorized user agents are prevented through this tier from accessing the system resources.

The **controller** of each client application must be capable of controlling the whole actions in this tier. The application processes and requests are passed by this controller to a special **interaction services** layer that communicate with the business tier using communication standards and technologies (such as SOAP, or XML-RPC).

5.2 Business Tier

The whole business logic is taken place in this tier. In our case, the main MASS modules (Authoring, scheduling, delivering, and reporting) are part of this tier. For each module a set of **core services** are available in the service tier. Separating the business logic in a different tier fosters MASS to be more modular and flexible. Furthermore, this tier facilitates the update of the business logic in an easy and flexible manner. As in the client tier, this tier also has a **controller** and an **interaction services** to communicate with the service one.

5.3 Service Tier

In the server side there are two main operations:

- Service Invocation: this operation is mainly handled by the **service request layer** where the requested service form the client application through the business tier modules is searched in the **core services** repository. If the service is not available in the core services repository then it will be searched in the new registered services in the **services registry.**

- Service Registration: each service provider from the resources tier registers its web services through the **service provider interface** to the **service registry**. The service providers are normally domain related systems (such as algebra assessment system, programming assignment assessment tool…) that provide their business logic services (such as algebra marking, equation editor) as web services.

By using these two operations the requests from the client applications are answered. Once the business tier module (e.g. Authoring) uses a new registered web service from the **service registry** in the service tier, the service becomes available through that module **core services**. So the next time such web service is needed the module can directly use it from its **core services**.

5.4 Resource Tier

The resource tier contains the MASS infrastructure resources (such as MASS databases) as well as the domain related systems and tools. The domain related systems can be assessment related such as algebra assessment system or can be business related such as the Student Information System (SIS). For each of these systems and resources there are a set of web services they must provide. The more web services they provide the more flexible and generic MASS will be.

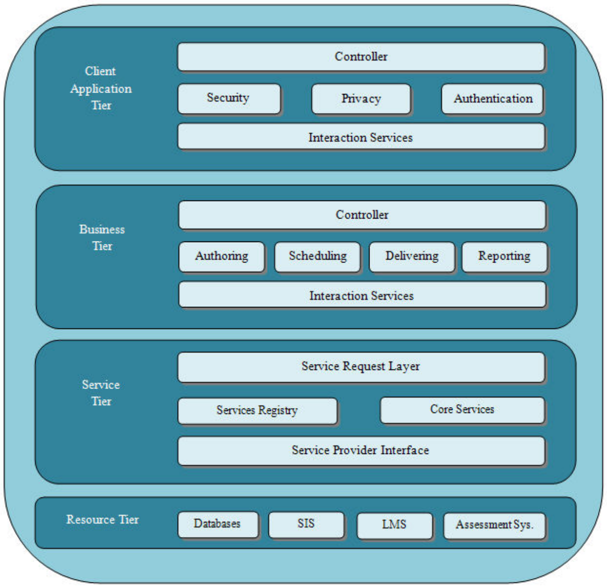

Figure 5: Service-oriented architecture for MASS

6. SHOW CASE

In this section we will discuss a show case to explain the flexibility of the suggested service-oriented architecture for MASS. Furthermore, to show how MASS can be utilized to serve as a generic and flexible assessment system.

Scenario:

A department of mathematics in a university is thinking of expanding their assessment system QAssess to assess algebra courses. QAssess is computer-assisted assessment system for multiple-choice questions.

What we have?

Let us assume we have the following:

- QAssess as a limited multiple-choice assessment tool.
- MASS as a flexible framework for assessment.
- QAlgebra as a tool that provide algebraic questions and exercises with the following features:
  - Symbolic Editor and Viewer: where it is possible to edit algebraic formulas and view them for the students.
  - Algebraic Engine: is capable to assess the students' answers and mark them.
  - Web Services for the symbolic editor and viewer as well as for the algebraic engine.

Methodology:

Based on the service-oriented architecture of MASS, QAssess will interact with the client application tier. Once QAssess is authorized as a user agent of MASS it can use the business tier modules. Let us assume that QAssess is trying to author algebraic items. As MASS provides web services for its main modules (Authoring, scheduling, Delivering, and Reporting) the authoring web service will be used. The request for a web service that is capable to author algebraic items will be passed from the authoring module in the business tier to its core services repository in the service tier. For each main module of MASS there are a set of core web service in the core services repository. If the requested service is in the core services repository, it will be provided to QAssess through the authoring module web services. In the other hand, If the service is not in the core services repository the request will be redirected to search for the service in the services registry as a new registered services. Since QAlgebra is providing a web service for editing algebraic formulas, MASS will deal with it as one of its resources and QAlgebra's web services can be registered to the services registry in the service tier. Once the requested service is available in the service registry it will be moved to the core services repository as a web service related to the initiator module. And then it will be provided to QAssess as an authoring module web service.

7. CONCLUSION

In this paper, the possibility to have a generic and flexible assessment system has been discussed. A set of personas has been provided to stress the variance in the application domains of e-assessment. This variance has caused some universities to have more than one e-assessment system in their different departments. Therefore, such universities have an extra required resources and budgeting plans as well as, dedicated staff to develop and maintenance those assessment systems. Therefore, a Modular Assessment System for modern learning Settings (MASS) has been suggested to solve this problem.

In the context of MASS, the interactions between MASS and its environment have been discussed in this paper. Furthermore, the assessment process represented by a set of MASS modules and their corresponding sub-processes has been presented. As MASS is supposed to be generic and flexible, this paper has investigated the main requirements for that:

- Standards conformation
- Framework for Assessment
- Cross-domain requirements
- Web services

Following standards in the process of designing and developing MASS, facilitates the futuristic integration and interaction between MASS and other e-assessment and e-learning systems. Furthermore, MASS will be capable to be expanded and scaled achieving more and more application-domain requirements. A framework for assessment is required to form the vocabulary that defines all the required services and their corresponding processes. Based on the special domain-based requirements as well as by using the available web services provided by different assessment systems and tools for these requirements, a framework can be utilized to implement MASS main modules and processes.

A service-oriented architecture for MASS has been suggested. In this architecture, four main tiers have been discussed:

- Client Application Tier
- Business Tier

- Service Tier
- Resource Tier

As MASS suggested architecture is service-oriented, MASS can have a modular design and is capable to be flexible. This flexibility supports MASS to cover more than one application domain in a way towards a generic assessment system for modern learning settings.

For future work, a set of application domains systems will be selected. For integrating with those application domains systems, detailed requirements will be investigated. These requirements will be used to update the MASS suggested architecture. Based on this architecture and those requirements MASS first prototype will be developed. This first prototype will be used to evaluate the eligibility of the suggested architecture by using it with the pre-selected application domains systems.

8. REFERENCES

[1] AL-Smadi, M., Guetl, C., Helic, D. 2009. Towards a Standardized e-Assessment System: Motivations, Challenges and First Findings, International Journal of Emerging Technologies in Learning (iJET), Volume 4, Special Issue 2: "IMCL2009".

[2] Davies, W. M. and , Davis, H. C. 2005. Designing Assessment Tools in a Service Oriented Architecture, In Proceedings of 1st International ELeGI Conference on Advanced Technology for Enhanced Learning, Napoli, Italy.

[3] ISAC, http://www.ist.tugraz.at/projects/isac/

[4] Milligan, C. 2003. Question and Test Interoperability (QTI): Extending the specification for Mathematics and Numerical Disciplines, Maths CAA Series, The Maths, Stats & OR Network, University of Birmingham, UK, Nov. 2003.

[5] Myers, R. (1986). Computerized Grading of Freshman Chemistry Laboratory Experiments, Journal of Chemical Education, Volume 63, Pages 507-509.

[6] Pruitt, John & Adlin, Tamara. The Persona Lifecycle : Keeping People in Mind Throughout Product Design. Morgan Kaufmann, 2006. ISBN 0-12-566251-3

[7] Rottmann R. M. and Hudson H. T. (1983). Computer Grading As an Instructional Tool, Journal of College Science Teaching, Volume 12, Pages 152-156.

[8] Safran, C. 2008. Collaborative Feedback: Code Peer Review in Higher Education. In Proceeding of ICL2008, Villach , Austria, 2008.