Information System Support for Quality Management Applying European Standards and Guidelines for Higher Education

Ulf Schreier, Marius Reusch, Marc Hüffmeyer and David Belzer
Furtwangen University, Robert-Gerwig-Platz 1, Furtwangen, Germany

Keywords: Quality Management, Document Management, Service Network, Organizational Networks, Service Description, Structured Documents, Higher Education, Accreditation.

Abstract: Higher education institutions strive for high quality of study courses and programs. One important tool is the introduction of a well-defined quality management system (QMS) supported by information systems. Editing service and review documents with office tools is not sufficient; a consistent and coherent management of all data is needed in an environment for authors. Data analysis, especially target-performance comparisons, and flexible generation of a variety of web and PDF documents are required tasks. This paper investigates the problems of simple file solutions in more detail and derives general requirements for better software support. Based on the requirements we propose an object-oriented framework that is able to handle core tasks around structured documents associated with organizational networks on top of a relational database. Document and organizational structures can be adapted to serve special needs of institutions. The system follows the European standards and guidelines for quality assurance.

1 INTRODUCTION

Quality assurance and improvement of educational services are essential tasks for universities and other higher education institutions. A standard procedure to accomplish quality management is accreditation of study programs carried out by external organizations (see for instance the description of the U.S. system (Eaton, 2012) or the standards and guidelines of the European system (ENQA, 2009)). Internal structures and procedures should support and extend this external review process. This combination of internal and external procedures leads to substantial quality management systems (QMS) inside universities. For instance, German universities can get a so-called system accreditation (Akkreditierungsrat, 2013), if they have implemented an internal QMS with specified characteristics. Quality management systems are understood as a bundle of business processes and associated information at the organizational level. Besides quality data itself detailed master data about the study programs like module descriptions or objectives of study programs are needed, as a QMS can check quality only, if the expected achievements of modules and programs are known.

Higher education institutions worldwide accept in general QMS as a tool for quality assurance and improvement, although sometimes concerns exist in the introduction phase. A description of common misunderstandings, viewed from the perspective of an accreditation organization, can be found in (Romero, 2008).

Many universities store the additional information in document files and continue to use classic database-oriented applications for course management (containing basic course information like name, extent or semester and mainly used for course registration of students) concurrently. But editing QMS document files is not enough. It might be quite easy for everyone involved to write the documents using standard office applications supported by document templates and to generate PDF files for publication. In addition, a centralized directory system could be used to group the files, but this simple solution comes along with some critical problems. However, internal discussions at our university, mainly at our quality management board with experts from university administration and all faculties, illustrated the problems of redundancy and subsequent inconsistency of too many files in too many versions. Moreover, experiences of board members with external accreditation show that other higher education institutions have the same

Schreier U., Reusch M., Hüffmeyer M. and Belzer D..
Information System Support for Quality Management Applying European Standards and Guidelines for Higher Education.
DOI: 10.5220/0004846403090316
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 309-316
ISBN: 978-989-758-020-8

problems.

An example of a number based quality data item with many implications is a credit point attribute of a course module (e.g. following the European Credit Transfer System (ECTS)). It needs a consistent representation (summations should be correct or breakdowns to specification of attendance and learning times should add up correctly). It is used at several positions (module description, study and examination regulations) and quality analysis compares it with actual student workload data. Those kinds of data items should not be hidden in document files. It should be under application control, but still be integrated with text-based information about modules.

For these reasons an adequate information system for QMS should help to manage, to integrate and to utilize all documents and data. Exploiting these data for analysis of quality (like target-performance comparisons) and for further processing (like web site publishing or PDF generation using the additional text based information) would be very valuable. Furthermore, an integrated information system facilitates the uniform handling of quality management data and documents at the entire university.

This paper outlines basic features of QMS at universities and the correspondence to general quality management. It is based on the European standards and guidelines for quality assurance (ENQA, 2009), but could be used elsewhere, if similar tenets are applied. It investigates the problems of simple file solutions in more detail and derives general requirements for better software support resulting in a design proposal. The consequences for adequate software support are extremely comprehensive. It turns out that the major challenge for information system support of quality management at higher education institutions is to find a proper mixture of features known from document management and data management. After introducing related work we propose an object-oriented framework based on structured documents with associated organizational networks.

FINQUAS is an on-going project developing an implementation in order to proof the proposed concepts, based on the experience of our institution with program and system accreditation, but it is adaptable to special document and organizational structures of other universities. A first release of the system is available at our university supporting peer reviews.

2 OUTLINE OF UNIVERSITY QMS

In general, the established quality management practices at universities (for the European variant see (ENQA, 2009)) follow the basic scheme of PDCA (plan-do-check-act) cycles known from industrial management; see for instance (Deming, 2000). More sophisticated schemes are well known, too, and are applied as well. However, in the following we will only sketch and summarize major activities of quality management as preparation for the presentation of requirements and solution architecture. Specific institutions will vary appropriately these activity structures and associated information formats.

At the level of study programs the quality management activities can be summarized as follows:

- Planning defines the output by setting objectives like learning outcomes for study programs and key figures like a dropout quota of students (to take a simple example figure that does not take into account the influence of grades of incoming students) or professor/student ratio. Boards at institution or faculty level are usually responsible for setting the objectives.
- Doing refers to the implementation of the objectives. Higher education represents this as a program curriculum consisting of modules and their descriptions. In a broader sense it also comprehends the documentation about required technical (labs) and human resources with organizational structures. Precise descriptions are necessary as a base for quality assurance. The exact description structure can differ depending on the kind of study and university specialities. Besides the core teaching service, supporting processes and policies (examinations, notification of credit transfer, admissions, generation of certificates, course scheduling, etc.) have to be documented as well. A release process complements the development process of descriptions.
- Peer reviews are a standard practice for checking program quality (for instance as accreditation process). Based on documentation of the study program, on-site visitations and their domain knowledge, reviewers give a structured judgment. Checklists are a common way to support reviews. These lists are basically document templates filled out by reviewers. The written statements of reviewers can be

supplemented by comparisons of actual values with planned values for a set of key figures (for instance the dropout quota). The doing (implementation) should satisfy the objectives of the study program. Reviews and monitoring of key figures should be performed in periodic intervals. Before university boards approve reviews, usually answers on review conditions of persons in charge of a service (e.g. a program director) are considered in addition.

- Recommendations as review results lead to documented action plans and their execution in order to improve the quality of content and structure of study programs. Action plans have a common format as known from project management. Actions could be changes of program objectives or implementation. Traceability from review statements to actions and concerned learning services is an important demand.

PDCA cycles occur at other levels too. At a lower level quality assurance of modules is important as well. A module description has plan (objectives) and do (content, extent, examination) sections. Objectives of modules should be derived from objectives of the study program (which are derived from university or faculty objectives). Checking could be done by reviews or student evaluation. At a higher level the aggregation of programs to program families (at a department, faculty or school level) or of a whole university are under consideration. Objectives at a lower level should derive or extend from higher level.

3 REQUIREMENT ANALYSIS

The main subject matter of QMS is the quality of the educational services. An educational service can be divided into smaller units at the next lower level. Modules are usually the smallest considered unit. They are the basic building blocks of a study program, which can be perceived as a composed educational service. Closely related study programs build a service group indicated by organizational units like departments, faculties or schools. Finally, the entire university itself can be considered as an educational service as well. All educational services together build a service network connected by part-of relationships. The services have a common structure at all levels.

For each service a group of people is responsible, usually organized as a board (e.g. a departmental committee is responsible for a study program). Composition of services and responsible groups indicate the university organization from lecturers towards university executive board. Furthermore, all services have certain types of objectives and each service type has an individual set of attributes describing the special properties.

The checking activities of quality management (like peer reviews or monitoring reports) themselves have similar structures to educational services. Each activity has an assigned responsible group (e.g. a reviewer group is responsible for a peer review), has objectives and an individual set of attributes. Hence, these activities can be considered as services, too. The quality services are also part of the service network providing the interconnections with the educational services. The service network altogether describes the complete structural organization of a university. The two service types differentiate themselves by an emphasis on objectives (educational services) respectively review results (quality services).

Beyond these core attributes and relationships to supporting concepts, services have many individual attributes with textual and numerical descriptions as described above. The service description as a whole can be considered as a document, as a unit of work that is edited, printed, read, archived or moved to another point in the overall service network. Therefore a careful analysis of the functional requirements unveils many features known from document management.

- Document relationships: Documents (in the context of QMS actually service descriptions) have part-of relationships with other documents, for instance a module description could be part of one or more study programs. A review belongs to a study program.
- Version control: Study programs change over time and new improved versions which are only valid for a certain time period are continuously being released. New versions of documents could be valid only for new students, while older versions are still needed for current students. Versions no longer in use should be archived.
- Concurrent author access: Several authors might work concurrently on the same documents (e.g. a group of peer reviewers works on the same peer review at the same time). The time frame for work could be more than just minutes. It could be hours or longer.
- Flexible release workflow: New documents or documents with need for changes should be new versions in draft mode. Documents may only be

released after the approval of several boards or responsible individuals. Interim statements of reviewers and answers from document authors might be considered. Only after approval the documents can be applied. The workflow can differ from organization to organization. Therefore, an information system for quality management should be flexible and adjustable enough to satisfy the demands of various organizations.

- Flexible authorization: Some members of some boards are permitted to write on some documents, others not. Especially accessing of documents in draft mode should be disabled for not related groups or individuals.
- Auditing: It should be known who has changed what data, but not at tracing level (for instance at database operation level). The recording should be domain oriented and should show who called what application function and its parameter values.

Furthermore additional requirements exist which cannot be easily integrated into document management systems. These requirements can be achieved in a simpler fashion using solutions based on the standard relational database paradigm directly.

- Structured flat content: All documents regarding programs and their quality are highly structured and contain not only text but also numeric, enumeration, date and string data. The structure is usually simple, a linear sequence of sections and can contain lists of part information (e.g. a service with a list of objectives as part information).
- Extensible structure: It should be possible with small programming effort to add and to remove attributes from service documents in order to adapt to special needs.
- Data relationships: There are a lot of relationships between data at the detail level inside documents. For instance, learning outcomes of modules could be derived from learning outcomes of programs (sometimes documented additionally in matrix form and stored as a table sheet) or modules relate to lecturers.
- Data integrity: It is not possible to check data integrity with office applications for documents. While some input data are formatted text without any constraints, other data have number, date or enumeration types and require integrity checks, e.g. number of ECTS credits of a module, or the level of a learning outcome item described with help of the Bloom taxonomy (Kennedy et al., 2006). Constraint checking can get very complex, e.g. describing university boards responsible for study programs with a variety of roles (with minimal and maximal number of members) and member duration.
- Integration of other databases: Some information might be available in other databases or applications, for instance, master data from course management systems like weekly hours of a course, student progress or grading statistics.
- Data analysis: Obviously, quality management needs evaluation of data from status reports of documents for target-performance comparisons.
- Generation of varied mixed documents from partial files: It is quite easy to generate a PDF file from a text document, but sometime users like to generate a complete catalogue of module descriptions or a program curriculum (using partial data from module description).

4 RELATED WORK

An off-the-shelf document management system would support or could be customized to satisfy the first list of requirements (Päivärinta and Munkvold, 2005). But, they focus on documents as a whole (which could be extended only with attributes for meta information). The second list of requirements is specific to QMS and needs a special implementation. Furthermore it depends on the specific needs of each university. Data analysis could focus on just a few performance indicators, but could consist of a very detailed analysis. Convenient programming access to parts of documents is needed like query access to attributes in relational databases.

Content management systems are similar to document management systems and focus in addition on web publishing using XML and HTML (Päivärinta and Munkvold, 2005). Partial structure access is possible, but not fully integrated to persistence APIs of object-oriented programming languages.

Software support of QMS for industrial management is closely related to production planning and control (Gerber, 2008). For instance, test plans needs to be integrated into work plans or samples needs to be tested during production or at delivery of goods. Therefore, these systems cannot be easily reused for education services.

Management software for quality audits following ISO 19011 is another approach and part of the ISO 9000 (ISO, 2013) standard family of quality management. There are several software products for audit management available (easy to find by internet search with keywords like "quality audit software" or "QM software"; a scholarly overview is not yet available). They focus on any kind of audits (like reviews) on any kind of service activities. Adaption to university QMS would have to take into account the special data structures, processing and analysing at universities.

Although a lot of information about quality management at higher education institutions is available, there is only a small amount of scholarly literature about its software support. Reprotool (Pouyioutas et al., 2013) is a relational database application that manages course and program descriptions according to the European Credit Transfer System. The system facilitates the work of faculty members in a similar way as our system. An additional student module supports the recording and calculation of student workload. However, Reprotool is not focusing on quality audits and relationships to educational service descriptions.

PROCON (Dosbergs, 2011) is another system specialised on managing curriculum descriptions. (Pah et al., 2008) describe the system eUniv that uses a general groupware software to manage QMS-related documents and projects without specifying details. e-EdU-Quality (Moisil et al., 2007) is an extension on top of eUniv providing document templates, workflow, integration of external data like student performance indicators and student questionaires. General groupware solutions have a generic user interface not tailored for the needs of quality management at higher education institutions. It is furthermore difficult to integrate application oriented data analysis, because groupware systems are based on a generic data model.

Tools supporting course evaluation by providing questionnaires, for instance (Mediero et al., 2010) should be also part of a QMS, but are complementary to our object of investigation.

In the next section we describe an object-oriented framework that can be used for quality management with structured documents with relational access to parts of documents.

5 A FRAMEWORK BASED ON STRUCTURED DOCUMENTS AND NETWORKS

FINQUAS provides an object-oriented solution for the requirements above. It is based on a relational database management system accessed by an object-oriented persistence layer. Consequently, features known from content/document management systems have to be created on top, but the implementation can be reused for any kind of service description. FINQUAS is a framework that implements standard tasks of quality management, though it can be customized by simple inheritance for data structure, changed user interface descriptions, changed workflow configuration or new data analysis functions.

The basic idea is a concept of an abstract service related to generic components providing the general functions required for handling services. The management of concrete service types deals only with the special data. An advantage of this approach is that generic functions (mainly content/document management features) can be reused for all service types. Another advantage is that extensions with general database query access to special attributes of services and their general processing in the application program are still possible. In this respect we get structured documents: document objects with relational persistence access to parts (attributes and partial objects).

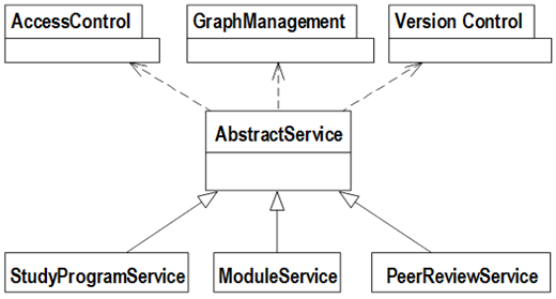

Figure 1: Related components of abstract and special service.

Figure 1 shows a simplified view of the abstract service concept. It explains how FINQUAS achieves its framework capability to adjust to diverse structure and organization details of higher education institutions. An `AbstractService` class provides the general features required in order to treat service documents. It represents the super class for concrete services like a study program, a

module or a peer review of a program. The concrete service classes contain the actual data, texts and numbers of the service description. `AbstractService` is related to a couple of generic components as shown in the next figures and explained in the following paragraphs.

As described before, services could have relationships to other services. A graph structure helps to represent these relationships in a flexible way. Several kinds of relationships have to be considered. One example could be curricular relationships. A single module for instance, might belong to two programs. It could be a mandatory or an elective module. Organizational relationships, for instance between program and university, may exist. Peer reviews and study programs have a special control relationship representing quality assurance. Figure 2 shows a sample object diagram with university u1, programs p1 and p2, a module m1 and a peer review r1 (simplified view without adding levels like faculty).

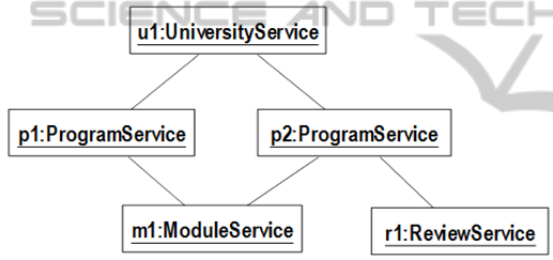

Figure 2: A sample service graph (or network).

To handle this task FINQUAS has a graph management component, illustrated in Figure 3. `AbstractService` inherits from a Node class. A `GraphProcessor` manages a set of nodes and edges (not shown in figure 3) between nodes.

A service has links to objectives that should be satisfied. Depending on the type of service, different types of objectives can be distinguished. FINQUAS uses subclasses of a general class for objectives to describe learning outcomes and planned values for key figures. Objectives should be coherent. This means in this context that objectives of superordinate services should be refined by services at lower levels. In order to comply with this basic principle of quality management the connections of objectives between higher and lower service level are captured at the definition time of services. These connections build a graph, too. Hence, graph management is reused for this task.

Each service has a group of people who are responsible for it. Figure 4 shows this relationship together with associated tasks. The configuration of a person group (at some university levels called board) could have to comply with complex rules. Group members might have a variety of roles. The number of instances for each role could be restricted to a specific range of values. The duration of memberships has to be considered, too. Additional representations are helpful for this kind of constraint processing. The group constraint processor in Figure 4 (class `GroupConstraintProcessor`) is a simplified depiction of this task.

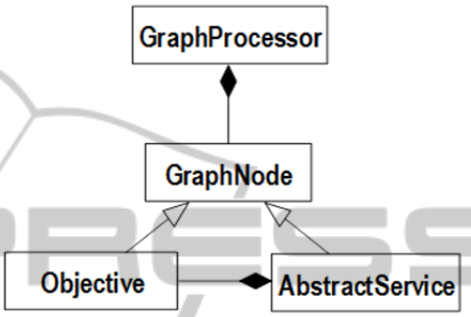

Figure 3: Graph management.

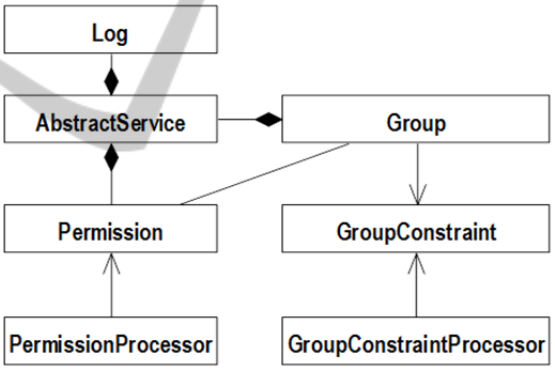

Figure 4: Authorization and group management.

A user who wants to perform an action on a service document needs an authorization from the system. The permissions depend mainly on some attributes of the service and of the user as well as on the group related to the considered service. In particular the role of the user, type of service, status (e.g. in draft mode only the authors have access) and action type are essential for permitting or denying access. Moreover, users belonging to a group associated with a superordinate service could have permissions on a service (depending on the organization service authors can have action rights for subservices, e.g. a member of the university executive board can have permissions to edit a subservice of the university like a program). Hence, a sophisticated and flexible processing of access

control is necessary, as indicated in Figure 4 by the class `PermissionProcessor`.

When groups of people collaborate on service documents, auditing information is needed in order to annotate who has changed what. The system does not log information about the domain operation at the database operation layer. For this reason an auditing processor is implemented, which is directly related to the `AbstractService` class. It is implemented as a listener, which is triggered as soon as a service has been created or removed or any data of a service has been changed. The listener stores the modification date as well as the originator of the modification in the database. Besides the control aspects, the tracked auditing information supports the collaborative work on service documents by providing the information via the user interfaces to the collaborators.

Furthermore, our framework provides for the first software version a locking concept in order to ensure a consistent access on service documents. The locking concept is implemented by an access controller, which also belongs to the generic functions, provided by the `AbstractService` entity (see Figure 1). As soon as a user with accordant permissions begins to edit a service, the service will be locked. Thus, other users can't edit this service anymore as long as the locking author is working on the document. However, it is possible to read the service and see the current changes, made by the lock owner. Future versions of our system shall contain more sophisticated mechanisms in order to support synchronous collaboration of several authors on the same document. For the beginning we preferred the more conservative and proven approach of locking.

The lifecycle of a service is represented by a workflow or at least a service is involved in workflows from other services. For instance the lifecycle of a peer review service begins with the workflow state "DRAFT" and after passing several states it results in the state "DETERMINED" as soon as the accordant committee has approved the preceding steps. In order to enable a simple and flexible workflow concept for all services and their related documents, a workflow processor implemented as a state machine is needed, (see Figure 5, class `WorkflowProcessor`).

FINQUAS handles versions in a simple way at the level of an entire service. Creating a new version of a document triggers the copying of all service information. This is based on the observation that curriculum descriptions are changed only on a yearly base in average.

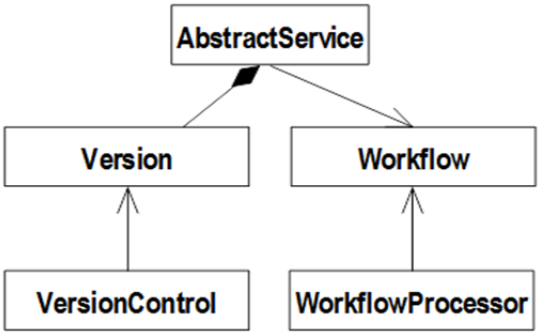

Figure 5: Version and workflow management.

6 CONCLUSIONS

The European Association for Quality Assurance in Higher Education requires as a standard information system support for QMS and demands "Institutions should ensure that they collect, analyse and use relevant information for the effective management of their programmes of study and other activities." (ENQA, 2009, p.7). In order to satisfy this standard with a complete and integrated system much effort is necessary. Complex requirements have to be considered and implemented. The definition of a QMS that fulfils the system accreditation criteria is already a challenging task in itself. But, since a lot of documentation has to be managed in a consistent and clearly arranged manner, it is important to have information system support. Furthermore, there are chances to exploit the knowledge for analysis of quality status of the institution and to generate web and PDF based information (which in addition should no longer be written redundantly).

Our project is very useful for us in several ways. It helped us to understand the requirements needed in order to get a reasonable quality management system. First experiences with users show that it facilitates the uniform handling of quality management at the entire university. It allows us to treat the information needed for QMS with an understandable and quickly accessible structure without redundancy.

The first release of FINQUAS supports peer reviews. According to the user feedback the main benefit for this task is that the quality managers get a clearly represented list of peer reviews and their current states. Communication with peer review groups is eased, since email addresses and mail templates for standard information are known by the system and available by one user interface click.

Next goals are monitoring reports for study programs (which we internally refer to as quality

reports) and module descriptions. The developers could already proof the framework concept of the system. It is now simple to implement monitoring reports and module descriptions. Progress of programming is fast.

From a software-architecture point of view we found an interesting approach to work with extensible structured documents embedded into an organizational network of people. The network is quite flexible and can map any kind of hierarchical or matrix organization. Our first prototype implementation confirms our view of the architecture for a quality management information system. It is possible to extend the system to new service types with only a small effort.

REFERENCES

Akkreditierungsrat, 2013. *Systemakkreditierung*, www.akkreditierungsrat.de/index.php?id=systemakkreditierung&L=1%2Fsql%27, accessed October 2013.

Deming, W. E., 2000, *Out of the Crisis*, MIT Press, Cambrigde.

Dosbergs, D, 2011, PROCON – A Tool for Curricula Accreditation, *3rd International Conference on Computer Supported Education,* SCITEPRESS.

Eaton, J. S., 2012. *An Overview of U.S. Accreditation,* www.chea.org/pdf/Overview%20of%20US%20Accreditation%202012.pdf, accessed October 2013.

ENQA, 2009. *Standards and Guidelines for Quality Assurance in the European Higher Education Area,* www.enqa.eu/files/ESG_3edition%20(2).pdf, accessed October 2013.

Gerber, A., 2008. *Methode zur Konzeption eines unternehmensspezifischen Qualitätsinformationssystems für kleinere Unternehmen*, hwww.qucosa.de/fileadmin/data/qucosa/documents/5565/data/Diss_QIS_Gerber.pdf, accessed October 2013.

ISO, 2013. ISO 9000 – Quality Management, www.iso.org/iso/iso_9000, accessed October 2013.

Kennedy, D., Hyland, A., Ryan, N., 2006. *Writing and Using Learning Outcomes: a Practical Guide*, sss.dcu.ie/afi/docs/bologna/writing_and_using_learning_outcomes.pdf, accessed October 2013.

Mediero, L., Martin-Carrasco, F. J., Sanchez, A, 2010. The Internet as a Tool for Improving the Student Evaluation of Teaching, *2nd International Conference on Computer Supported Education,* SCITEPRESS.

Moisil, I., Oprean C., Lobont, L., Kifor, C., Ciprian, C., 2007. E-EdU-Quality – Software Tools for Quality Management in Higher Education, *6th International Conference on E-ACTIVITIES,* WSEAS.

Pah, I., Oprean, C., Moisil, I., Kifor, C., 2008. Technology to Support Education Software Solutions for Quality Assurance in e-learning, *International Conference of Computers, Communications & Control,* IJCCC.

Päivärinta, T., Munkvold, B., 2005. Enterprise Content Management: An Integrated Perspective on Information Management, *38th Hawaii International Conference on System Sciences,* HICCS Digital Library.

Pouyioutas, P., Gjermundrod, H., Dionysiou, I., 2013. The Pilot Use and Evaluation of Reprotool version 3.0, *International Conference on Education and New Learning Technologies*, IATED.

Romero, E. J., 2008. AACSB Accreditation: Addressing Faculty Concerns, *Academy of Management Learning & Education, Vol. 7, No. 2,* AOM.