The Sixth International Conference on e-Learning (eLearning-2015), 24-25 September 2015, Belgrade, Serbia

INTEGRATION OF INFORMATION AND E-LEARNING SYSTEMS IN HIGHER EDUCATION

KRISTIJAN ZIMMER
University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia
Kristijan.Zimmer@fer.hr

Abstract: *This paper covers general principles of IT integrations applied to typical information and e-learning systems found in the higher education sector as well as a project of creating integrated environment "E-Campus" at the Faculty of Electrical Engineering and Computing. Good practices of managing such projects and selecting optimal systems and protocols for integration are also covered from both technical and business sides.*

Keywords: *Information systems integration, e-learning, higher education*

1. INTRODUCTION

10 years ago, after further 4 years of development of the Content management system [1] *Quilt CMS* at the Faculty of Electrical Engineering and Computing, University of Zagreb, a request came from a group of professors and teaching assistants to include functionalities typical for Learning management systems (LMS's) [2].

Since in-house development of such functionalities seamed hardly possible due to the lack of resources and specific skills, the team responsible for development of centralised IT infrastructure and support searched for available alternatives. At that time, Croatian Academic and Research Network (CARNet) was offering commercial LMS *WebCT* [3] to higher education institutions as well as schools. Testing of the system a year earlier by the CARNet's *Referral centre for evaluation of courseware* proved that it was rich with useful functionalities as well having a good application programming interface (API) [4].

The first integration was developed between the CMS and the Student information system (SIS) [5]. Once a day, if a new course had appeared in the SIS, it would have automatically been created in the CMS, with all the students enrolled in it and teachers lecturing this course automatically assigned to it with appropriate user permissions. Similarly, the appearance of a new student in the SIS meant that his/hers user account would have automatically been created, if it hadn't existed already.

The second integration has been developed between the CMS and the LMS *WebCT*. Similarly, all the synchronisation of courses, users and their enrolments would have been "mirrored" from the SIS side, via CMS, to the LMS.

As needs for additional functionalities appeared, it seemed logical that the CMS should remain as a central integration point as well as a modular programming platform for modules that require usually smaller range of functionalities. A methodology of decision making concerning choosing optimal information systems for each set of new requirements, integration protocols, data sets and their formats has been introduced.

The main goal was that with each new system, the entire infrastructure should benefit from new data becoming available to all existing systems, making them more intelligent. The idea of an open, integrated, scalable and increasingly useful information and e-learning environment known as the *E-Campus* have been envisioned initially by Sinisa Tomic and Kristijan Zimmer, with many contributors in the development team in years to come. [6]

In those early years, openness was not a "standard". Most systems available on the market were so called "black boxes", with little or no possibility for exchanging any data. Even reading the data stored on their file systems or databases was often a challenge. 10 years later, this has radically changed for the better.

Later in 2005 *WebCT*'s licencing model changed and usage of the API had been conditioned by the upgrade to the newest version. Due to CARNet's lack of funding for such an expensive upgrade, the development team decided to implement the same functionality to *Moodle LMS* and integrate it to the CMS with the same set of functions.

2. THE E-BRICKS

In the *E-Campus* terminology, the terms *E-bricks* and *E-mortar* have been used by the development team for the information systems and open protocols, respectively.

As explained, the first three integrated systems included:

- **Content management system (CMS)** ***Quilt*** which still serves as a public web, intranet and teacher / student portal, as well as the development framework and integration layer for the entire *E-Campus*;
- **Student information system (SIS)** ***ISVU*** [7] which provides the CMS and the rest of the integrated systems with:
  - List of all courses for each study programme;
  - List of all students, with detailed information about their achievements and obligations;

- List of all teachers;
- Information about students' enrolment to the courses (student-course pairs);
- Information about teachers' assignments to the courses (teacher-course pairs).

- **Learning management system (LMS)** ***Moodle*** [8], which provides space for all course materials, quizzes, polls, grading, wikis, forums, chat and other important tools.

As the years passed, there has been an increasing need in the higher education for new information systems or smaller modules which could have been developed as the CMS modules.

Every time a new request appeared, the development team analysed one of three potential general approaches:

- Develop a new CMS module;
- Buy an *off the shelf* system and integrate it;
- Customise an open source system and integrate it.

From the business perspective, buying an *off the shelf* commercial system seemed problematic since most of the licensing models were *per-user* or *per-seat*. With more than 3.500 students and 350+ teachers, the majority of licencing models of commercial software were economically out of reach for the Faculty.

From a project management (especially resource) point of view, choosing an optimal open source system and customising it according to the Faculty's needs proved most often the be the best option. The analysis of the features always included the existence and quality of API's, structure and openness of the database, potential of successful development of modules of the chosen system (if needed in the future) and market potential for such newly integrated system, both to the existing and new institutional users.

As a result of such analyses, gradually new systems have been integrated to the already mentioned initial three.

The newly integrated systems included:

- **Integrated library system (ILS)** [9], also known as library management system, which covers wide arrange of functionalities such as acquisitions (ordering, receiving, and invoicing materials), cataloguing (classifying and indexing materials), circulation (lending materials to library users and receiving them back), etc. After an analysis, open source library system *Koha* [10] was chosen and integrated, as it best fitted Faculty's needs;
- **Enterprise resource planning system (ERP)** [11], with newly developed integration which enables users to see relevant financial information about their projects' balances, transactions and financial reporting. Integration has been done with one of the leading commercial ERP systems *SAP* [12], which has already been used by the Faculty's Financial department;
- **Customer relationship management (CRM)** [13] system, which in our setup enables users to report their needs and encountered problems. After an analysis of three systems, popular open source bug tracking system *Mantis BT* [14] has been chosen and adapted to address these non-software needs;
- **Authoring, communication and collaboration set of applications** During the project with the Faculty of Economics and Business, a single sign-on to institutionally adopted Google Apps [15] has been developed;
- **Timetabling system** that supports creation of course and exam timetables, managing changes to these timetables, sharing lecture rooms with other events, and scheduling students to individual classes has been implemented. After an analysis, open source system *UniTime* [16], developed at the Purdue University has been deployed and integrated with the CMS for optimal presentation of individual teachers' and students' timetables;
- **Learning object repository (LOR)** [17], once envisioned as a separate installation of the Ariadne LOR [18], integrated to provide search and retrieval functionalities of the GLOBE network to the LMS users, has been replaced by newly developed search and metadata harvesting modules for *Moodle* LMS [19], developed as part of Faculty's role in the European *Open Discovery Space* [20] project.

During the time of mentioned integrations of large systems, the team also frequently decided to develop other CMS modules to fulfil some requests for functionalities. This was due to the following reasons:

- The functionality was too specific to be found in an open source or commercial system;
- The functionality was small enough and introduction of another system would be an "overkill";
- It was wiser from the business or market perspective to raise the value of the core CMS system in such way.

In such manner, numerous small and mid-sized CMS modules have been developed:

- Organisation of all teaching activities (lectures, laboratories, examinations…), replacing two previously existing systems;
- Student group exchange;
- Automatic grading of paper exams (after OCR processing of paper submission forms);
- Lecture attendance monitoring;
- Student internships management;
- Graduate thesis collection and reviewing process management;
- Teachers' time burden calculation;
- Contract management;
- Management of departments' material needs;
- Material inventory management;
- Business trip management;
- Lecture room reservation;
- etc.

Currently, there is a growing need to develop several ERP-style modules, as the licencing of the currently used ERP *SAP* has proven to be too big financial burden for the Faculty.

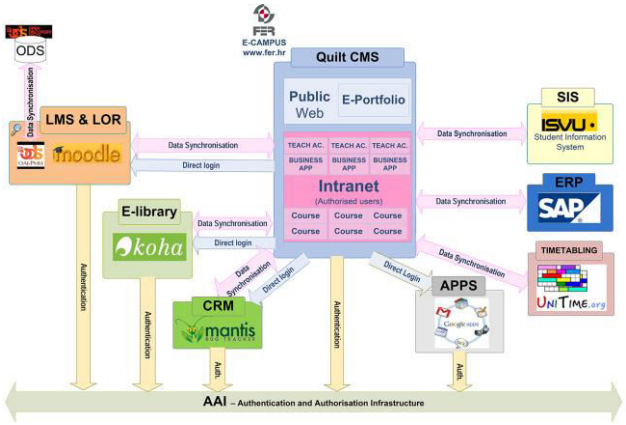

Figure 1: The *E-Campus – E-bricks* and *E-mortar*

3. THE E-MORTAR

To integrate *E-bricks* in the *E-Campus* many open protocols have been implemented.

In the beginning of the research 10 years ago, only a few existed. The implementations were far from widespread and often technically problematic.

Probably the biggest change that happened in 10 years has been a global acceptance of those open protocols, both by proprietary, commercial systems wanting to be more "pluggable" in customers' heterogeneous environments as well as by open source developers, often being first to implement an open standard. [21]

Today, there is an established set of open protocols used to exchange data between various systems. Usually, the integration project includes analysis of protocols supported by the new systems, ending with the decision which protocol will be used. In some situations, when there are no API's available, nightly (or hourly) batch jobs which include data transfer can accomplish certain tasks. [22]

In the implementation projects, the development team has implemented several protocols and batch scripts:

- Simple Object Access Protocol (SOAP) [23], used to retrieve data from the ERP system *SAP*;
- Representational State Transfer (REST API) [24], used to exchange data with SIS *ISVU*;
- The OAI-PMH protocol [25] used to enable ODS harvester to obtain metadata of learning objects stored in the *Moodle* LMS, which acts as a LOR system;
- XML-RPC protocol [26], used in the past to provide access to the CMS's data by other Faculty's systems, now largely replaced by the REST API;
- "Download XML" (*colloquial*) often via HTTPS, a batch method to retrieve dumps of data and process them, resulting in updated database tables. It is used to collect data from the University of Zagreb's information systems;
- "SELECT / INSERT" database-to-database batch script, where after a series of SELECTs from one database, a series of INSERTs is created and executed on the target database.

Concerning the authentication, the Authentication and Authorization Infrastructure (AAI) [27] is used by all *E-Campus*' *E-bricks* to ensure usage of a single username/password pair on all systems. The AAI is actually hierarchical LDAP [28] infrastructure in which each institution takes care of their own users' entries which include passwords. When local LDAP installation is asked by any other LDAP whether a provided username/password pair of a user is correct or not, it returns the true/false response with additional user information so that the other system can use it if needed.

The single sign-on is implemented between the CMS and several other systems to insure user authentication to other systems without a need to re-authenticate users.

4. SYNERGY OF DATA EXCHANGE

The most risky part of an increasing number of integrated information systems is that the entire *E-Campus* infrastructure may become too complex and too unstable to maintain.

By introduction of the methodology of analysing data needs of all integrated systems, it became obvious that addition of each new system brings new sets of data that have a potential to make existing systems more intelligent and useful.

When introducing a new information system to the *E-Campus* the development team tries to find answers to the five main questions:

- Which data will the new system bring?
- Which data from the existing systems could be beneficial to the new system?
- Which data from the new system could be beneficial to the existing systems?
- In which way and how frequently will the exchange take place?
- How would typical roles/types of users in higher education benefit from the introduction of new data, their processing and exchange?

5. BENEFITS FOR KEY USER ROLES

The further analysis concerns key user roles: teachers, students and technical administrators as well as their benefit from each integration. This is an important step, since it may reveal that a certain integration doesn't benefit any of the user roles, in which case is should be reconsidered and probably abandoned in the project, or may reveal that there is a potential for even more benefits.

In the last years, the benefits have become re-analysed [29] and they are described here in a short form, for all the systems exchanging data with the CMS.

Integration of the CMS and the Learning Management System (LMS) Moodle

Roles	Benefits
Teachers	Their students are automatically enrolled to courses when teacher allows them to access the course. The accounts for all the students are automatically created. The teacher can retrieve all the grades from the LMS to the CMS automatically, check them and send them from the CMS to the SIS.
Students	Students can see relevant information stored in the LMS (grades, calendar events, announcements, forum posts, internal LMS's e-mail messages…) in the CMS. They have direct access (no need to re-authenticate) to all their courses in the LMS.
Admins	No administration of the LMS, besides biannual upgrades. Administrators only check reports to see if everything is working as expected.

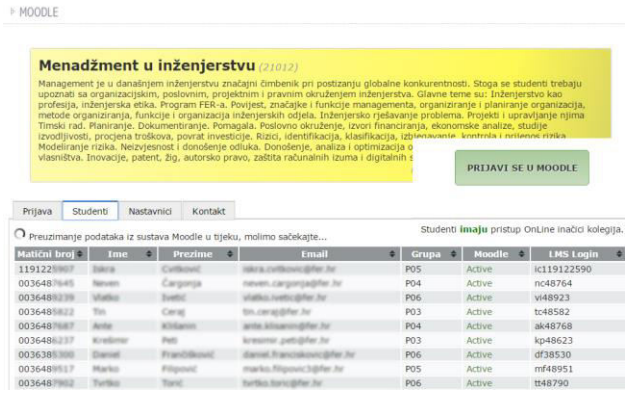

Figure 2: The LMS *Moodle* "seen" from the CMS administration panel. All LMS administration tasks are done automatically or manually from the CMS

Integration of the CMS and the Integrated library system (ILS) Koha

Roles	Benefits
Teachers	Teachers need to define keywords (or "tags") for new books and other materials in the ILS and when new books become available, teachers match them with their courses, and the materials are presented to students as a learning resource in the CMS (currently in a pilot implementation)
Students	Students (and other library users) can search books by several criteria, reserve them, get notified when the books become available and see which books they have landed.
Librarians	Librarians can make book acquisitions, inter-library exchanges, cataloguing, circulation, inventory of books and periodicals, as well as helping teachers with proposing which books cover parts of the curriculum they are teaching.
Admins	No administration of the ILS, besides annual upgrades. Administrators only check reports to see if everything is working as expected.

Integration of the CMS and the Customer Relationship Management system (CRM) Mantis

Roles	Benefits
Teachers	Teachers have fast and easy contact with the technical support and an online form to describe their needs or explain a problem they encountered.
Students	Students have fast and easy contact with technical support.
Admins	A standardised method of communication with users which tracks their requests and provide feedback when the task is completed. It enables several people to provide support for users by enabling insight into current and previous needs and ways how similar problems have been solved in the past, in the form of a knowledge base.

Integration of the CMS and the Student Information System (SIS) ISVU

Roles	Benefits
Teachers	Due to the synchronisation with the SIS, teachers are automatically assigned to their courses and they have access to the list of all enrolled students, prepared for them in the CMS, LMS, ILS and other systems.
Students	Students are automatically enrolled to their courses on all systems and are given appropriate user permissions.
Admins	No administration for the SIS integration. Administrators only check reports to see if everything is working as expected.

Integration of the CMS and the Enterprise Resource Planning system (ERP) SAP

Roles	Benefits
Teachers	Teachers who are managing scientific and professional projects have real-time information about the balance and transactions of their projects, as well as income and cost structures.
Students	Students have better information about their tuition payments and future financial dues (currently planned for implementation).
Admins	No administration for *SAP* as the hosting is outsourced.

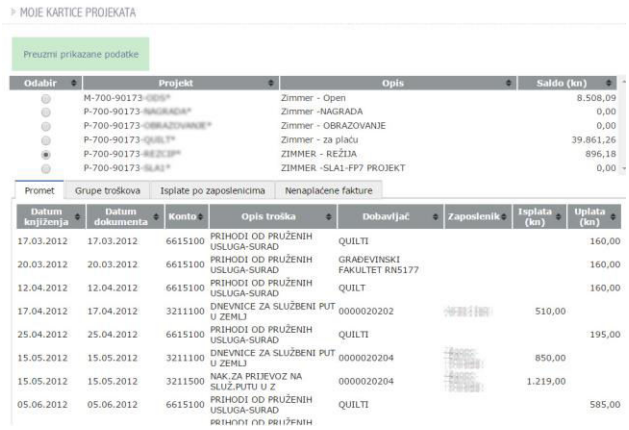

Figure 3: The ERP system *SAP* "seen" from the CMS in a form of an *e-banking* type of application.

Integration of the CMS and the university Timetabling system Unitime

Role	Benefits
Teachers	Teachers can see their timetable for the entire semester, shift students to another lecture halls and schedule them to another time periods, seeing how many of them would be available then.
Students	Students can see their individual timetables for the entire semester. Several adjoining CMS modules allow them to change groups in organised fashion, reserve lecture halls when they are empty and synchronise their timetable with their favourite calendar application (*iCal*).
Schedulers	The timetabling system allows schedulers to configure teachers' availability and requests, generates mathematically optimal schedule of lectures for the given space-time "continuum" and then enrols students to groups for each course in the way that the number of conflicts resulting in inability to attend lectures is minimal.
Admins	Currently, an automated data conversion between *Unitime* and the CMS is being developed, after which there would be no need for manual preparation of data.

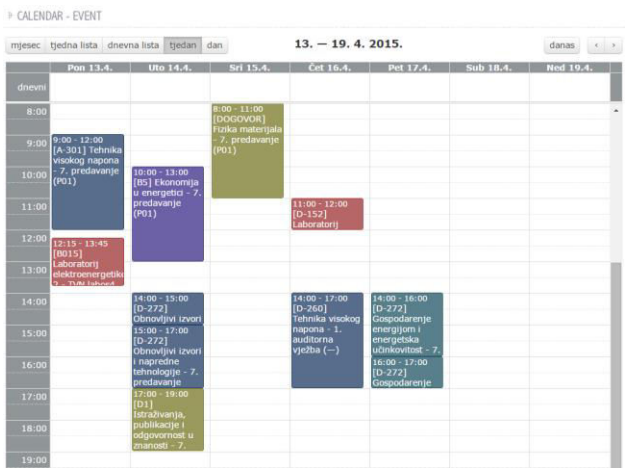

Figure 4: Student's timetable after it has been transferred from the timetabling system *Unitime* to the CMS.

Integration of the LMS and the Open Discovery Space LOR system

Role	Benefits
Teachers	Teachers who have their online and mixed-mode courses in the LMS can decide through the *Moodle* module *ODS OAI-PMH* that all their materials are visible and available to the European teacher community of the *Open Discovery Space* project. They can also search ODS's LOR through JavaScript interface available as *Moodle* block.
Students	Students can search ODS's LOR through JavaScript interface available as *Moodle* block.
Admins	After initial configuration, no administration is needed.

6. LMS-CENTRIC APPROACH

One of the questions often asked concerns the CMS (portal) – centric approach that the development team has chosen. [29] In some implementations of such integrated environments, the LMS-centric approach was believed to be more appropriate. This means that a LMS, such as *Moodle*, should be the centre for all integrations and data exchange.

LMS-centric approach has different issues, depending of what kind of system is supposed to serve as the "centre". If this is commercial LMS, the problem might be its integration capabilities and the fact that no changes of the code or database tables is usually allowed, due to the license that prohibits that. In the case of *Moodle* and some other open source LMS systems, there is a problem of frequent changes of versions. If some changes are made by the user institution to the database, user management or core modules and there is a need (due to the user demand, security vulnerability or end of support) to upgrade to the newest version of the LMS, that can be a complex and potentially extremely risky task.

Having a CMS with an open license as the portal that integrates all other systems shifts the integration development away from the core modules of the mentioned integrated systems, LMS included. This way there is not interference with the development lifecycles and business models of the external systems on which a user institution have no influence upon. It is much more logical and manageable approach to concentrate only on API's of those external systems, which handle data exchange, since changes of API's are usually announced in advance and well documented.

7. CONCLUSION

The ongoing research project *E-Campus* continues to integrate new systems and develop new modules. Most importantly it is a good example that it is possible to build complex IT environments when systems are open, meaning that they are able to exchange large sets of data in a secure way.

With popular LMS's, ILS's, ERP's and other systems having hundreds, sometimes thousands of developers and huge resources for further development, it is important to use this awareness as an advantage to institutional projects of building integrated IT environments and not to make mistake by trying to compete with those niche winners. The tagline could be: "Don't annihilate – integrate!"

Well-chosen systems with open, useful and well-documented API's, implementing internationally accepted open standards and protocols for data exchange are the key building blocks of such advanced environments.

The growing market demand for open systems in the last 5-10 years forced commercial vendors and open source developers to make their systems open, so that they can better fit in the user institutions' heterogeneous environments.

With such positive trend, projects such as the *E-Campus* will continue to become simpler and easier to scale, if managed well.

LITERATURE

[1] Content management system (CMS): https://en.wikipedia.org/wiki/Content_management_system

[2] Learning management system (LMS): https://en.wikipedia.org/wiki/Learning_management_system

[3] WebCT: https://en.wikipedia.org/wiki/WebCT

[4] Application programming interface (API): https://en.wikipedia.org/wiki/Application_programming_interface

[5] Student information system (SIS): https://en.wikipedia.org/wiki/Student_information_system

[6] Sinisa Tomic; Kristijan Zimmer, Mario Zagar; Vlatka Paunovic; Ivan Voras: Living the E-Campus Dream, EDEN 2007, Proceedings, Vienna, Austria, 2007

[7] Information system of higher education institutions - ISVU: http://www.isvu.hr/javno/hr/index.shtml

[8] Moodle LMS: https://moodle.org/

[9] Integrated library system (ILS): https://en.wikipedia.org/wiki/Integrated_library_system

[10] Koha ILS: http://www.koha.org/

[11] Enterprise resource planning (ERP): https://en.wikipedia.org/wiki/Enterprise_resource_planning

[12] SAP: http://go.sap.com/index.html

[13] Customer relationship management (CRM): https://en.wikipedia.org/wiki/Customer_relationship_management

[14] MantisBT: https://www.mantisbt.org/

[15] Google Apps: https://www.google.com/edu/products/productivity-tools/

[16] University Timetabling - UniTime: http://www.unitime.org/

[17] Learning object repository (LOR): http://edutechwiki.unige.ch/en/Learning_object_repository

[18] Ariadne LOR: http://www.ariadne-eu.org/

[19] Sinisa Tomic, Vlatka Paunovic, Kristijan Zimmer: Searching and Exposing Learning Objects from Moodle: The ODS experience, IEEE Bulletin of the Technical Committee on Learning Technology, Volume 17, Issue 1/2, April 2015 (http://www.ieeetclt.org/issues/april2015/5-Tomic.pdf)

[20] Open Discovery Space project: http://opendiscoveryspace.eu/

[21] William A. Ruh, Francis X. Maginnis, William J. Brown: Enterprise Application Integration: A Wiley Tech Brief, 2001

[22] Chris Britton, Peter Bye: IT Architectures and Middleware: Strategies for Building Large, Integrated Systems, Unisys Series, 2004

[23] Simple Object Access Protocol (SOAP): https://en.wikipedia.org/wiki/SOAP

[24] Representational State Transfer (REST): https://en.wikipedia.org/wiki/Representational_state_transfer

[25] Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH): https://en.wikipedia.org/wiki/Open_Archives_Initiative_Protocol_for_Metadata_Harvesting

[26] XML - Remote procedure call (XML-RPC): https://en.wikipedia.org/wiki/XML-RPC

[27] Authentication and Autorisation Infrastructure (AAI): http://www.aaiedu.hr/

[28] Lightweight Directory Access Protocol (LDAP): https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[29] Sinisa Tomic, Ene Tammeoru, Kristijan Zimmer, Ivan Voras, Vlatka Paunovic, Mario Zagar: Benefits and synergies of using integrated technological solutions for e-learning, Online Educa Berlin 2008 Conference proceedings, Berlin, Germany, 2008