Abstract—The paper deals with an open source portal framework developed and customized for university purposes, usefully applied in TRANSILVANIA University of Brasov by using the concept of single sign-on. The paper also presents an integration of the Moodle Learning System and DotNetNuke framework as an efficient opportunity for open standards; emphasis is given to the main aspects related to the implementation of student and teacher specific features in the Portal in combination with web-based specific applications used for communication, management and cost optimization.

Keywords: open source, portal, university management system

I. INTRODUCTION

It seems that making the argument for open-source software becomes easier every year. This is mainly due to open-source products such as Linux, MySQL, or the Apache Web Server (just to name a few), which have been very successful. Furthermore, quality software organizations such as the Apache Software Foundation boast a history of consistent, stable and successful software deliveries.

Many open-source projects are commercially developed code bases turned over to the open market as a business decision, because they no longer have to pay to maintain the code.

We cannot forget the best part about open-source software: it is free. Nevertheless, we should note that open source and free are not synonymous. Just because most open-source products are free, it does not mean that they are all free. Also, open source means that one can receive the source in order to identify bugs [1].

The greatest thing about capable open-source software is that it solves the classic “build versus buy” engineering dilemma. The software had become so complex that it was simply not practical to choose to build anything. Now, with reliable open-source alternatives, developers can start with the open-source products and build the features sets needed.

Building a portal is about integrating an organization with all its business processes, people and locations, this being a procedure that one cannot take lightly or accomplish easily. Business processes are complex because of the competitive forces and the ever-shifting variety of people associated with them. Building a portal enables the movement of information through the organization.

Therefore, it is better to invest in a person, rather than a product that will be outdated in a year.

According to Diamond Cluster [2], the main portal features are:

- **Portal Core Services** - Provides core portal application functions. These functions are the common services (such as user interface design and search) that are utilized by other components. These services can be supported using a 3rd party tool;

- **Content Services** - Content components are Information (knowledge) from internal sources, external feeds, databases etc. Third party tools or application service providers (ASPs) can provide this component;

- **Portal Interface Framework** - The interface framework provides a structure or container for other applications (or portlets/modules). The framework allows new applications to follow a consistent visual interface and programming standard;

- **Collaboration Services** – Collaboration components are those that facilitate employees working together. Many components facilitate virtual meetings through application sharing, community chat rooms or videoconferencing;

- **Integration Services** - Integration components are those that link existing services to the portal. These components provide application and transactional linkage to enterprise systems (such as email or corporate directories). Adapters can facilitate this integration, but integration is usually developed or customized on the basis of requirements.

A University portal requires at least three components: membership, proof of affiliation and proof of identity. A more comprehensive definition, offered by the University of Hull [3], is: “Campus Portal: A thin layer which aggregates, integrates, personalizes and presents information, transactions, and applications to the user seamlessly and securely, according to their role and preferences.”

The institutional information portal brings together a university’s two most valuable assets: its identity or brand, and its constituents—alumni, students, parents, staff and prospective students. With Web access to all university services, institutions are forced to rethink everything from institutional image, to systems architecture, to new business and instructional models, or portal strategy [4].
II. PORTAL ARCHITECTURE AND FEATURES

The work on the present system started in 2006, after several failures in implementing commercial applications. The last purchased student management application in Transilvania University was a commercial one implemented in 2004. One year later, as a consequence of the new requirements imposed by the implementation of the Bologna Process in Romania, an imperious need was felt to adapt the application with a view to solving the problem of missing features. Unfortunately, we met the vendor’s reluctance and consequently, the application had to be abandoned.

After an extensive search on commercial and open source Student Management Application - SMA applications and taking into account the requirements of the university management and the continuous changes, it was decided to develop our own applications.

We also used the experience we accumulated in building and evaluating open source web applications and portals in national funded projects like “Enhancing Communication in Public Administration from Brasov County by Means of a Communication Portal” [6]. In this project we analyzed portals/CMS like:

- PHP/Perl/Python: PHPNuke, PostNuke, Joomla, Drupal, Plone;
- Java: eXo, Liferay, Jetspeed, uPortal;

We found out that none of the portals/CMS was used as a basis for implementation and/or development of SMA. Thus, we have chosen the one that offered the greatest advantages (support, programming language(s), architecture etc.) for the development team.

The portal developed at TRANSILVANIA University of Brasov is based on the DotNetNuke (ASP.NET) open source enterprise framework.

The portal consists of modules that make available information for students, teachers and alumni, according to Figure 1.

![Figure 1. Portal architecture](image)

The portal integrates many applications, some of them not directly available to end users, but accessible from the portal by means of web services.

The portal and all the sub-applications are built using n-tier architecture, according to the DotNetNuke module design [5]:

- Presentation layer (main page – default.aspx and the user interface for all the modules – user controls);
- Business logic (with custom business objects, caching, personalization, search, exception management and event logging);
- Data access using an abstract data provider that is further implemented in a concrete data provider, specialized for each database (in our only SQL Server).

Also, from a developer perspective, the Portal and its sub-applications are available in a continuous integration environment in order to publish the modules to our staging sites, with obvious advantages (quick detection and fixes of defects, unit testing).

In the following parts of the paper, we will try to present the main features of the Portal, together with the connectivity options for data transfer.

A. Admission application

This is a web application used for gathering and managing admission data for students. This application was developed and first used in 2006 for pilot faculties. In the summer of 2007, the application was considered to be mature enough to be used in the whole university. Among the features, we may mention:

- Online pre-registration for candidates; the candidates have to choose, guided by a wizard-style interface, the faculty and the study program(s) they want to register to;
B. Detailed or summarized data introduction, including various dates and places (birthdate and birthplace, full address, date and place where the candidates graduated high school), all based on a complete database of regions and localities of Romania;

C. Finding and managing unique candidates: one candidate may register oneself to more than one faculty. One of the requested feature was this ability to see where one candidate has been registered, all based on Personal Number Code (CNP);

D. Data replication between various sites: for security and increased manageability reasons;

E. Education Fee payment for candidates;

F. Reports based on an online report-generation tool for displaying and printing reports. For faculty and university management, we included various reports based on all the data gathered in the first stages of admission, all these based on Analysis Services using a web-based interface for data and column manipulation.

On account of the fact that the admission process is accomplished mostly by teachers and the student management is carried out by faculty offices and not all the candidates are admitted, the data exchange between the two applications is made via web-services. After the admission process is finalized, the faculty offices may manage and import into the Student Management Application (SMA) only the admitted candidates for their faculty. Thus, for better manageability, the candidates are organized starting from fundamental domains/study programs according to the Bologna process. After a candidate is imported into the SMA, it cannot be further managed in the Admission application.

B. Student Management Application (SMA)

This is a web-based application used for managing student records.

The application has been in continuous development and improvement starting with the university year 2005-2006. It has features like:

- Collecting data about students: data is transferred from the admission application;
- Defining, assigning and collecting fees for students: we are able to gather and assign not only the education fee, but all the fees related to student activity;
- Campus Accommodation Management for students, featuring the ability to track student accommodation starting from the first year of study and the fees paid for the accommodation. It features several special reports (tax deductions, bad debts, student migration between halls);
- Grades and scholarship management: data are gathered for one university year. Using a trajectory-tree, we are able to find the corresponding student data from any university year and all related data;
- For the Accounting Department, Faculty Management and University Management, we developed a special portal that features only reporting tools, based on both Reporting Services and Analysis Services all accessed using a browser-based interface.

Using a DotNetNuke feature that permits a single code-base for many portals, each one with its users and security, we decided that it is best to implement such a portal for each faculty, each one having separate users, security and data. Also, campus accommodation management and scholarship management are available in their separate portals. For testing purposes, we also implemented a sandbox portal where each faculty staff may test the new features.

C. Portal for Students and Teachers

The application has been developed as a means for students to access and view their grades and fees and also for accessing their e-learning courses based on the Moodle platform. The e-learning and the portal platforms have been populated with data, in several steps, which included:

- generation of unique usernames and password for students and teachers in the Student Management application; regarding the information distribution, each student received a personalized paper which included information about one’s username and password for accessing the Portal, the e-learning platform and one’s personalized email in one of the University’s email domain names.
- importing these data into the Portal and, at the same time, into the Moodle platforms (one for each faculty) and the email platform;
- transforming data about the courses in a format that could be imported in Moodle.

The results were a procedure that could be followed for any new student/teacher/course that would be imported manually. In addition to this procedure, there are three usable products: the Portal itself, populated with data about students, the e-learning platform populated with data about students, data about courses, data about teachers and data regarding the relationship between teachers, students and courses, and the e-mail platform populated with accounts for each student.

The Portal application has been accessed from its launch according to the Table 1.

<table>
<thead>
<tr>
<th>Month</th>
<th>Views</th>
<th>Visitors</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>932</td>
<td>164</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>56791</td>
<td>3479</td>
<td>1995</td>
</tr>
<tr>
<td>5</td>
<td>60621</td>
<td>3607</td>
<td>1960</td>
</tr>
<tr>
<td>6</td>
<td>180518</td>
<td>10080</td>
<td>4241</td>
</tr>
<tr>
<td>7</td>
<td>152957</td>
<td>11958</td>
<td>4205</td>
</tr>
<tr>
<td>8</td>
<td>52893</td>
<td>4224</td>
<td>2387</td>
</tr>
<tr>
<td>9</td>
<td>192732</td>
<td>12503</td>
<td>4490</td>
</tr>
<tr>
<td>10</td>
<td>153978</td>
<td>9904</td>
<td>4483</td>
</tr>
</tbody>
</table>

Regarding the functionality, we can separate the Portal features into features for students and features for teachers and researchers, according to the diagram in the figure above.
The first features implemented were related to the possibility for the students to access their grades and their assigned and paid fees. Both students and teachers had the possibility to access the e-learning application where the authentication into Moodle is accomplished through web-services [7]. In addition, they had another communication-enabled application, the email.

The Portal, the e-learning and the e-mail application being detached applications, had to handle the password change: the users may change the password in Moodle, e-mail and Portal in a separate manner, but when the password is changed in the Portal, it overwrites the passwords in Moodle (web service) and e-mail directory (via LDAP). A special case is represented by teachers who lecture at more than one faculty; in this case the password is changed in all the Moodle platforms where the teacher is found.

Another feature available was the possibility for students to see in the Portal their courses from both the course catalog (available via web services from the SMA) and the Moodle platform (information also available via web services), together with all the information related to those courses: number of classes per week, number of seminars /laboratories/ projects per week and the number of credits.

The course information is extracted for each student, from the Student Management Application and it is cached for a period of time in the Portal for quicker access.

The same happens for the Moodle courses that the student subscribed for; in this case the course is presented via a hyperlink that, when clicked, the student is automatically authenticated in Moodle (if this has not happened in the current work session) and transferred to the course itself (this was possible only with a small addition to the core Moodle source code). If one person attends more than one faculty, one is presented all the courses from all the study programs one registered to.

This last feature is also available for the teachers with the remark that one teacher may teach at several study programs from different faculties; in this case, the user has to be authenticated [8] in each Moodle application before the information is finally displayed.

A new feature available only for the students is the Electronic Registry Office. This enables students to obtain online, valid predefined attestations and certificates, even without going to the faculty office. The information flows in the following steps:

- the students fill in an online request for a certificate;
- if enabled by the faculty, the certificate is immediately available for downloading and printing by the students themselves;
- if the faculty does not choose to make available this features, the certificate is generated and printed at the faculty office and the students have to pick it up physically.

The certificates and attestations are secured using an encrypted bar-code consisting of 21 digits. This bar-code may be verified in any faculty office or by any teacher using a teacher-only Portal feature. Also, the certificates needed outside the university may be verified by anyone by entering the digits in a public form in the portal itself. If the certificate is valid the person verifying it is presented with the option to download and compare the physical certificate with the electronic one.

Another student-specific feature is represented by an interface to the Campus Accommodation Management where the students may fill-in online requests for accommodation for the following university year. The data flows, via web services, to a person in-charge with accommodation for one faculty who takes into account various information: distance from the residence, number of exams failed, student’s average grade etc. The students are notified via e-mail when and where or whether they are to be accommodated.

For reducing the workload on the faculty office, teachers may, if enabled by the faculty, enter grades directly into the portal and, via web services, into the Student Management Application. This teacher-specific feature is available in conjunction with the information about students, study-programs, and teachers and their lectures, so that a teacher may not enter grades for another lecture. A special case is represented by the foreign languages courses, where students may choose the language they want to study. Because of this and of the fact that the curriculum does not always differentiate between languages, representing them generically, e.g. “Language 1”, where student A may study English and student B French, a security matter occurred.

For solving the above mentioned problem and other security problems, the grades are neither deleted, nor replaced, but only added in the system. In the case of grades added by teachers directly into the Portal, a grade (G) for a student (S) – course (C) is always associated with the teacher/user (T) and date (D), resulting the following secure combination:

\[(S) + (C) + (G) \rightarrow \text{Grade Position (GP)} \leftarrow T + D\]

Each teacher or the faculty office may generate a special report where each grade is associated with the person who entered it into the Student Management application or into the Portal. This report also depicts all the grades and (GP) obtained by a student for a certain course. Relating this feature and the possibility to secure the data entered in the database we currently research digital signatures for teachers and faculty office staff – e.g. when a teacher enters grades through the portal into the SMA, the record in the database should be digitally signed.

A Portal-specific feature for our teachers and university [9] is represented by the web-based research management and validation modules, used for collecting and tracking research information for teachers and researchers. The data collection comprises the following categories of research items:

- Collecting the data for published books and articles at national or international conferences;
- Collecting the data for research grants;
- Collecting the data for doctoral activity.

Data validation consists of the possibility of checking and validating the data entered by teachers and researchers at the specialized departments within our university. Therefore,
the University Library validates the data entered for articles, books and conferences; the data for research grants are validated by the Economic and Research Projects Department, whereas the Doctoral Department validates the data for the exams, papers and doctoral theses.

The system allows teachers/researchers to enter data and information for Research Criteria directly in their Curriculum Vitae. The Research Criteria are divided into two large categories (international and national, and University - local) with the possibility to add other criteria. It is possible for each category to be further divided into sub-categories:

- Didactic: manuals, textbooks etc;
- Research: national and international grants and projects etc;
- Professional: PhD thesis etc.;
- Institutional: other courses, research materials etc.

The reports are available for each Research Criteria with the possibility to divide and create reports for faculty, department, research direction and teacher/researcher level, all browser-based.

D. Didactic Activity Management

It represents a specific application in our university, used for recording, managing and mostly optimizing information for teacher-course management. This feature evolved from an Excel-based application to a complex, multi-layer, web-based application, integrated into the Student Management application. Figure 2 depicts this complex design that helps faculty and university management to reduce and optimize costs according to the number of students, number of hours, teachers and teacher positions.

Thus, for cost reductions purposes, more courses from one faculty may be coupled if the syllabus is the same (e.g. the subject “Mathematics for Economists” has the same syllabus for all the study programs at the Faculty of Economic Sciences, therefore the courses may be coupled into a single large course with all the students from the respective year of study. Further on, if the syllabus is the same at more than one faculty, those courses might be coupled (e.g. if the subject “Mathematics for engineers” would have the same syllabus for more than one study program from more than one faculty, it may be coupled). The reporting features for this application is available to a small number of managers (department, faculty and university) while the entered data are automatically available to other components of the application.

![Figure 2. Information flow into the Didactic Activity Management application.](image)

III. CONCLUSIONS AND FURTHER WORK

DotNetNuke framework is a feasible and flexible open source tool for the integration of a large variety of university applications, like student admission, student management, research management, didactic activity management, digital library, eLearning.

The Moodle learning system can be properly customized for university purposes, both in ODL and full time learning systems.

Using the concept of single sign-on, integration of the Moodle Learning System and DotNetNuke framework is an example of efficient opportunity for open standards to contribute to the development of complex integrated university applications.

The Portal application helps students and teachers to keep a better connection with the university, to be better informed and, at the same time, to provide feedback to the university, with a view to improving its services for both students and teachers.

The availability of optimization tools for faculty and university managers makes it possible for cost reductions by better understanding and viewing relationships between
courses/seminars/laboratories from fundamental domains/study programs and even between faculties.

Among new features that could be included into the Portal, we envisage a job bank for final year students, which could be developed with the support of alumni.

Furthermore, focus is going to be given in the future to the development of a collaborative feature that should enhance student-teacher cooperation outside the academic environment.
