

INTEGRATING INDUSTRIAL PARTNERS INTO E-TEACHING EFFORTS - A PORTAL TO SUPPORT THE INITIATION OF CO-OPERATIONS

Tobias Fries, Andreas Henrich
*University of Bamberg, Media Informatics
Feldkirchenstraße 21, D-96045 Bamberg
tobias.fries@uni-bamberg.de, andreas.henrich@uni-bamberg.de*

ABSTRACT

Learning Management Systems (LMS) form a powerful basis to integrate practitioners and industrial partners into teaching activities. Taking a university-wide LMS as the basis, one critical issue in this respect is to bring together lecturers and industrial partners willing to co-operate in teaching efforts addressing a mutual field of interest. In the present paper we introduce an approach to support the initiation phase of such co-operations between academic and industrial partners based on a web portal filled with content from the existing IT systems maintaining module descriptions, administrative data, and teaching content. We describe the requirements and basic as well as advanced use cases the portal should fulfill and explain the difficulties in integrating the portal into the heterogeneous system landscape at a university using the example of the University of Bamberg. We especially illustrate the different interfaces to be used to harvest information from the connected systems and to automate processes within a connected LMS to set up co-operations to be mediated by the LMS after being established. We also briefly describe the system architecture of the portal and how we tried to make it flexible and maintainable for the usage in different environments.

KEYWORDS

Learning management systems, IT infrastructure, integration of practitioners, practical projects

1. INTRODUCTION

For specific types of courses the involvement of practitioners or industrial partners into academic teaching can be beneficial for the lecturers, the students, and the practitioners, as well. In traditional teaching efforts limitations of time and space often prevented this desirable involvement. Nowadays university-wide Learning Management Systems (Henrich et al. 2007) form a powerful basis for this involvement and enable various forms of co-operations. For that we have developed different co-operation models, ranging from (1) just granting company members access to lecture notes, slides, and—if available—lecture recordings to (2) discussion forums and courses, where students use their academic knowledge in real-life company issues or to test, evaluate, or improve company products, to (3) courses, which are mutually planned, organized, executed, and evaluated by the lecturer and company representatives. The co-operation models have in common, that the company representatives are not just learners, but contribute knowledge, experience and real-life questions to the courses.

As university-wide LMS contain a huge amount of courses (the moodle-based LMS Virtual Campus at the University of Bamberg counts 1,210 courses¹ just for the current summer semester 2010) it is very difficult for interested industrial partners to find appropriate courses that are opened for practitioners and also provide adequate topics and co-operation types. Therefore, we implemented a co-operation portal to support the initiation phase of co-operations providing an easy to use possibility for the lecturers to offer their LMS courses for co-operations and a quick overview of available co-operation opportunities as well as more detailed information on co-operation offers for practitioners. The portal also provides a search feature and a direct connection to the LMS. In chapter 2 we define the requirements and use cases our portal should fulfill.

¹ Virtual Campus of the University of Bamberg (last checked May 5th 2010): <http://vc.uni-bamberg.de/moodle>

Chapter 3 deals with the system landscape at the University of Bamberg focusing on interfaces to harvest the information relevant for our portal and to automate processes within the LMS. Chapter 4 gives a brief overview of the architecture and prototypical implementation of our portal. In chapter 5 we draw conclusions and outline planned future work.

2. REQUIREMENTS AND USE CASES

The requirements for the portal can be divided into essential core features and advanced features. The entry co-operations are of the type *insight into courses with the possibility to use discussion forums*. If a lecturer wants to open his existing LMS course for industrial co-operation partners, he should just have to configure the course settings to enable access for industrial users. The industrial partners seeking for an appropriate co-operation offer should have a starting point, where they can get an overview of the available co-operations and a possibility to filter the supplied co-operations and to perform a keyword search for co-operations. When they have found interesting offers, they need provision of more detailed information about the topics of the course, the type of co-operation, and a description which role they can play in the co-operation. The industrial partners should also be able to find out more about the lecturer (including field of study and contact data) and to directly join the co-operation. The start page of the LMS usually displays the available courses hierarchically ordered by faculty, chair and semester. Whereas this navigation is intuitively operable for the students—as they know the university’s structure—it is inappropriate for industrial partners because of the following aspects: (1) Industrial partners are not familiar with the structure of the university and do not need to get familiar with it to find suitable co-operations. (2) They do not think in terms of university semesters when they look for co-operations. (3) Industrial partners need a quick overview of only the courses available for them and an adequate search and filter feature for their needs. (4) Students know that enrollment to the LMS course is necessary in order to download lecture notes and get access to other content important for their exams. People from industry might be discouraged when they have to spend much effort to find suitable courses. (5) Students have background information about the course and the lecturer because of their physical attendance at lectures and their everyday life as students of the university. Industrial partners on the other hand want more information about the course, the co-operation and the lecturer before they are willing to join.

So a landing page for interested company people is necessary, presenting interesting information harvested from the LMS and other available sources (cf. chapter 3) in a target group specific manner. It would be possible to implement this company members’ landing page as an extension to the LMS itself, but several reasons make an independent implementation more favorable: (1) An LMS extension would be implemented only for one LMS base system (e.g. moodle²) and could not be easily transferred to other LMS. (2) An independent solution could be integrated into the university web pages making it appear more familiar to company members than directly displaying it inside the LMS. (3) Extensions to an LMS bare the danger of being incompatible with future versions of the LMS base system. (4) As apart from the LMS other data sources are being used as well, a strong binding to the LMS is not useful.

Because of these reasons we decided to implement the initiation portal independent from the concrete LMS. As a prototypical implementation of our portal should be integrated into the website of the University of Bamberg³, we implemented it as an extension for the CMS Typo3⁴, which is used for our university website. Our approach is nevertheless generic, as Typo3 is wide spread among German universities⁵—which could easily add our extension to their existing website—and universities using another or no CMS could install a Typo3 system with our extension and include it as IFrame into their web pages. The advantages of a Typo3 extension to an implementation using PHP, JSP or ASP.NET and a database are that the configuration features, APIs (e.g. to the database and for AJAX support), and the security features of the CMS can be used. As Typo3 extensions are usually not hindered by updates of the Typo3 system, the portal can benefit from security updates. The architecture and interfaces to the surrounding systems of our prototypical implementation are described in chapters 3 and 4.

² LMS moodle: <http://moodle.org>

³ Co-operation portal of the University of Bamberg: <http://www.uni-bamberg.de/minf/lms4kmu-portal>

⁴ CMS Typo3: <http://typo3.org>

⁵ Usage of Typo3 as CMS at German Universities: <http://wiki.typo3.org/index.php/TYPO3-Hochschulen>

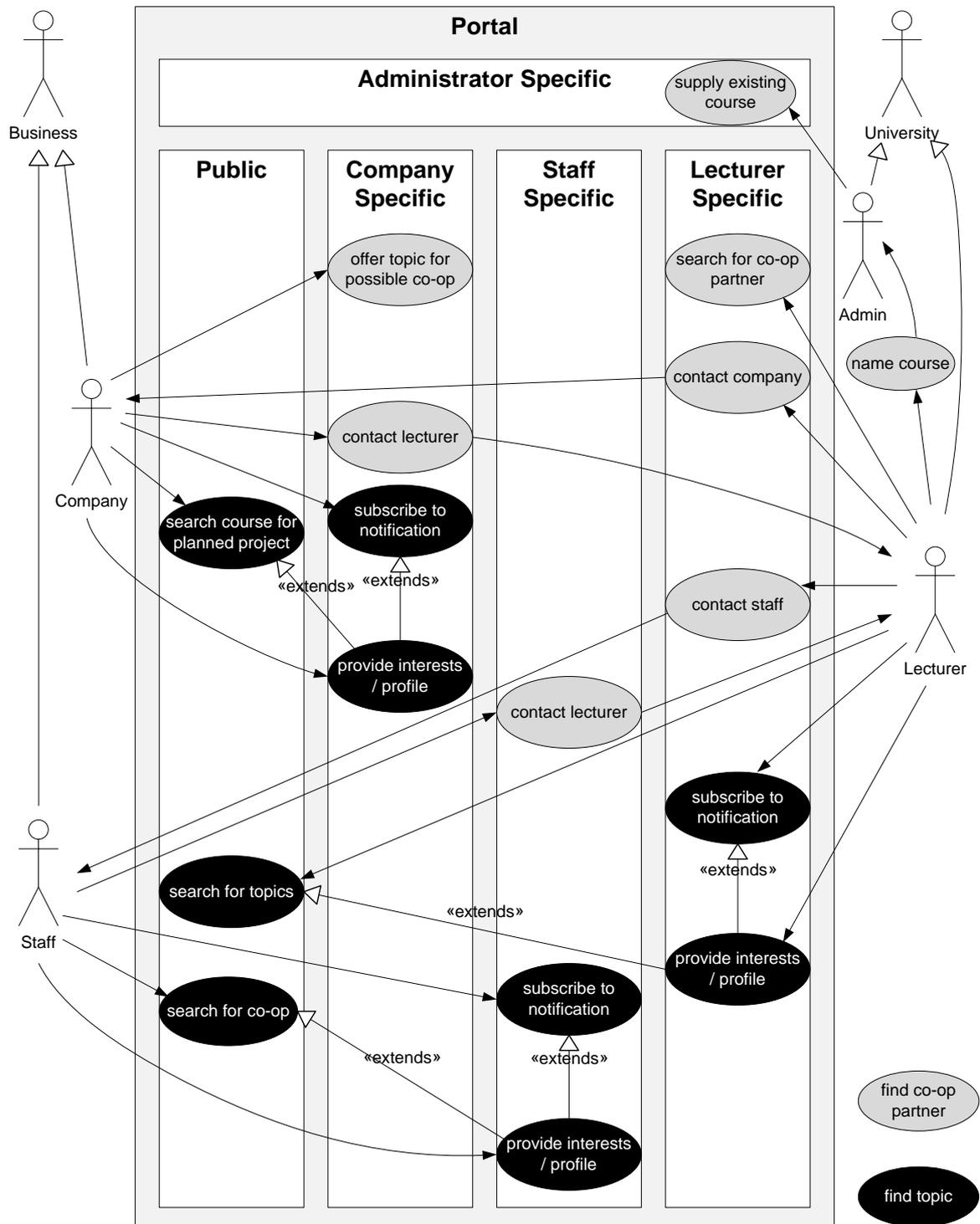


Figure 1. Advanced use cases for the co-operation portal. The black colored use cases meet the purpose to find a topic for a co-operation, the grey colored ones to find a co-operation partner. The use cases within the white box on the left are publicly available; the other use cases are only available for registered users of the specific user group

Our prototypical implementation delivers most of the mentioned essential features and is already in use for the pilot phase of our project. To ideally support the initiation phase of more sophisticated co-operation types than just *insight and discussion* and to enable individualized information e-mails to interested company members, the portal should be extended by several advanced features. To enable lecturers and companies to offer ideas or topics for future co-operations, the portal should provide a registration feature. Company users can define their interests in their user profile and this can be used as a filter or ordering criterion to display the co-operation offers. Other filter criteria should be the affordable intensiveness and activity the co-operation necessitates. The user should also be able to save a search or filter function (including his profile, keywords, preferred co-operation types etc.) to start it again in the future and to use it as a basis for an e-mail notification to be sent, when new co-operations meeting his search criteria become available. To allow the emergence of inter-organizational networks, users could make their profiles accessible to other users if they wish to. It is important, that these advanced features are implemented without an unnecessary increase in complexity. The essential features of the portal should be usable without the need for logging in. On the other hand, for using the advanced features, according to a single sign-on approach, the user account of the LMS should be valid for login to the portal, too. And after being logged in to the portal, the user should be able to enroll to an LMS course without having to log in to the LMS. Another feature to support the lecturers could be to automatically create a user group "Students" and a user group "Company Members", when a lecturer wants to open up a course for companies. Fig. 1 depicts these advanced use cases for the portal.

3. SYSTEM LANDSCAPE AT THE UNIVERSITY OF BAMBERG

Different systems for different purposes exist at the University of Bamberg: The public website of the university including public pages of the chairs and other facilities are managed in a Typo3 CMS. Organizational data like time tables, room reservation etc. are managed by an information system called UnivIS⁶. Lecture descriptions, examination applications and students' marks are managed by a system called FlexNow⁷. The Virtual Campus (based on the LMS moodle) is used for offering lecture notes and other content to the delimited circle of lecture attendees as well as for blended learning and e-learning courses (Henrich et al 2007). Other possible systems to connect with our portal are those providing general information about topics of chairs willing to co-operate with companies such as BayDat online⁸ and portals for further education studies like the Bavarian Virtual University. In the following we describe which data from the particular sources are useful to our portal, which APIs the source systems provide, and how we implement the data access. Table 1 lists 11 steps, in which our portal retrieves data from the connected surrounding IT systems.

For organization and planning purposes the University of Bamberg uses a system called **UnivIS** developed by the company Config eG⁹, which is in use at many German universities. When the UnivIS id of a lecture is provided, one can get the name of the lecture, the URL of the corresponding page in the university website, and the UnivIS id of the lecturer from the system. With the UnivIS id of the lecturer, it is possible to retrieve his contact data and the URL of his home page in the university website.

Apart from an HTML user interface UnivIS also offers a publicly available so called PRG interface for which the output format can be set to XML¹⁰. When the UnivIS id of a lecture is known to the portal, a request to the PRG interface can be started using the PHP extension CURL and the resulting XML data can be parsed using the XML Unserializer of the PEAR PHP extension.

The **university website** was re-launched using a Typo3 CMS in 2006. Before that time, a kind of authoring environment based on Word macros was used and chairs had the possibility to restrict access to files via folder restriction with .htaccess files. With the usage of Typo3 the former heterogeneous website became more consistent and the adherence of the corporate identity could be achieved. As access restriction via .htaccess was no longer possible with the new system and a conceptual separation of public website and restricted areas for lecture related content were requested, shortly after the re-launch of the website a learning management system (the Virtual Campus, described on the next page) was installed (Henrich & Wolf 2006).

⁶ Implementation of UnivIS at the University of Bamberg: <http://univis.uni-bamberg.de>

⁷ Implementation of FlexNow at the University of Bamberg: <http://www.uni-bamberg.de/pruefungsamt/flexnow>

⁸ BayDat online: <http://www-futur.uni-regensburg.de/baydat/>

⁹ Config eG: <http://www.config.de>

¹⁰ User manual for UnivIS: <http://univis.uni-erlangen.de/userman.pdf>

The website of the University of Bamberg contains—amongst others—the publicly available pages for every chair, which are maintained by the chairs themselves. The content of the pages is not standardized, but the lecture pages should be independent of a concrete semester, as semester specific course information should be provided in the corresponding course on the Virtual Campus. Our portal retrieves the URLs of the web pages of the lecture and the lecturer from UnivIS. Both should be provided for the potential industrial partners as links to further information. The lecturer’s page usually contains a picture of the lecturer. The URL of the picture found in the HTML code of the lecturer’s page can be used to display the lecturer’s picture in the contact data page of our portal.

Table 1. 11 steps of data access from our portal to surrounding IT systems at the University of Bamberg

	Target	Source System	Interface	Data Input	Result
1	get VC names of participating courses	Virtual Campus (moodle, PHP)	HTML	use course search for “LMS4KMU”	names of all courses with “summary” containing “LMS4KMU”
2	get UnivIS id, VC id and possibilities for companies for a course	Virtual Campus	WSPP (PHP Objects)	call WSPP function get_courses (course name)	UnivIS id in field “idnumber”, VC id in field “id”, possibilities for companies part of field “summary”
3	create groupings for companies and students	Virtual Campus	WSPP	call function add_grouping (grouping name, course id)	grouping created, grouping id returned
4	create groups for companies and students	Virtual Campus	WSPP	call function add_group (group name, course id, enrolment key)	group created, group id returned
5	assign group to grouping	Virtual Campus	WSPP	call function affect_group_to_grouping (group id, grouping id)	group assigned to grouping
6	get link to webpage, FlexNow id of lecture, name of lecture and UnivIS id of lecturer	UnivIS	PRG (XML)	use course search with UnivIS id of lecture	lecture name in <name>, FlexNow id in <origin>, URL of webpage in <url_description>, lecturer id in <Person><id>
7	get course content	FlexNow (JSP)	XML	FlexNow id	description from module manual
8	get contact data and link to webpage	UnivIS	PRG (XML)	UnivIS id of lecturer	tags within <location> tag
9	get picture of lecturer	Website (Typo3, PHP)	HTML	URL of webpage	webpage of the lecturer usually containing a photo
10	get further info about research and co-operations of lecturer	BayDat online (PHP)	HTML	navigate from webpage of lecturers at University of Bamberg to webpage of current lecturer	webpage with research topics, general co-op interests and successful co-ops
11	get links to similar courses at VHB	VHB website (JSP)	HTML	use course search with search keywords	webpage listing VHB courses with links to VHB course pages

The **Virtual Campus** is the LMS that contains the courses to be opened and used for the co-operations. As our portal is thought to be an entrance point to the courses in the Virtual Campus, it should amongst others enable the user to get access to the LMS, display the course names of the available courses, and provide a possibility to enroll into a course as company member and also a link to enter the course.

To access the Virtual Campus, the portal navigates through the HTML web pages via CURL PHP. Nevertheless it would be neater to have a special interface to the Virtual Campus just for data retrieving purposes. Unfortunately the current version of moodle (v1.9.7) does not offer an appropriate interface. There is only a Web Service Interface (XML-RPC)¹¹ that can be used for connection and interaction of two separate moodle instances, but is not very handy to use for our purposes. Maybe the interface will become more useful

¹¹ Documentation of moodle’s current Web Service Interface: http://docs.moodle.org/en/Development:Web_services_API

in the upcoming major update to moodle 2.0¹², but at present for moodle v1.9.7 the extension WSPP seems more operational. It was developed by OKTech¹³ and has been further developed until present by Patrick Pollet of INSA de Lyon¹⁴. It can be downloaded as extension from the moodle repository¹⁵. The client component of the WSPP interface retrieves data as PHP objects (also client implementations for Java and Python exist), which can be used without any need for parsing. As the WSPP interface does not provide access to all of the possibilities the HTML interface does, we use the WSPP interface whenever possible and the PHP CURL alternative just in exceptional cases.

As stated in chapter 2, the lecturer should be able to allow access for industrial users in the course settings. We could have modified the source code of our moodle system to display a checkbox for this purpose and save its status in the database, but these modifications often get lost when updating to a newer version of moodle. We have decided that it is more sustainable, if the lecturer includes the sentence “This course is open for industrial partners of the project LMS4KMU” in the description field of the course. The portal can then use moodle’s course search to find all courses containing this sentence. As the course search is not available over WSPP, the portal starts it via PHP CURL. Using the course names, it is possible to get other data about the courses via the WSPP interface, including the course id, the short name, the description, and the course id number. The lecturer can describe the participation possibilities for industrial partners in the course description (headed by a key sentence like “Possibilities for industrial partners:”) to be displayed in the portal. To get more information about the course, the portal also needs the UnivIS id of the corresponding lecture. The lecturer could again add it to the course description, but this may look odd as the course description is visible for all users. It is better to use the field “internal id number” for this purpose: it is designed for arbitrary internal usage, not visible to the students and not used at the University of Bamberg so far. When the portal retrieves a new participating course for the first time, it can create a grouping for students and one for industrial partners in the VC course via WSPP and set automatic group allocation via access keys in the course settings.

The portal presents the user a link to the registration page of the Virtual Campus. It would be possible to create a user account for the user of the portal via the WSPP interface, but in this case the user would have to insert all necessary registration data into the portal which seems to have no advantages to directing the user to the registration page of the moodle system. The portal also presents the access key for industrial partners to enroll to the selected course and a link to the course page. If the user wants to participate in the co-operation, he can follow the link, log in to the Virtual Campus and enter the access key to enroll to the course. To simplify this procedure, the user can enter his VC user name and password in the portal, which verifies these via the WSPP interface and automatically enrolls the user to the VC course without the need for him to enter the access key.

The examination management system **FlexNow** has been developed by the Chair for Business Informatics especially System Development and Database Application¹⁶ of the University of Bamberg since 1994 and the Academic Institute for University Software in Bamberg (ihb)¹⁷ founded in 2003. Apart from the University of Bamberg, it is used at many other German universities. It has been developed in Java and has an HTML user interface based on Java Server Pages. Apart from this, an interface providing XML output is available. The FlexNow system at the University of Bamberg contains the descriptions of the general contents of the continuously provided lectures. At the moment a function is being implemented in FlexNow that returns the lecture description for a given FlexNow course id (which can be obtained from UnivIS). As soon as it is available, our portal can use it via accessing the XML interface of FlexNow.

BayDat online is a website developed at the University of Regensburg providing information (contact data, fields of study and former successful co-operations) about chairs at Bavarian universities as publicly available uniformly formatted HTML pages. As many lecturers of the University of Bamberg are represented at BayDat online, it would be useful to display general co-operation offers and fields of study in our portal. Unfortunately, no appropriate interface exists to easily integrate this data. We could only access the website of BayDat online and parse the HTML pages, but as the system is not being maintained any more, a connection to BayDat online does not seem to be sustainable.

¹² Documentation of moodle’s upcoming Web Service Interface (v 2.0): http://docs.moodle.org/en/Development:Web_services

¹³ OKTech: <http://www.oktech.ca>

¹⁴ Patrick Pollet at INSA de Lyon : <http://prope.insa-lyon.fr/~ppollet/pp.ssi>

¹⁵ WSPP extension in moodle’s repository: <http://cvs.moodle.org/contrib/patches/ws/wspp>

¹⁶ Chair for Business Informatics especially System Development and Database Application at the University of Bamberg: <http://www.uni-bamberg.de/seda>

¹⁷ Academic Institute for University Software in Bamberg: <http://www.ihb.uni-bamberg.de>

The **Bavarian Virtual University**¹⁸ (Virtuelle Hochschule Bayern, vhb) is an alliance of all Bavarian Universities and Universities of Applied Sciences to provide e-learning courses to students. All students matriculated at a Bavarian university or university of applied sciences can participate in any course of the Bavarian Virtual University exempt from charges no matter which field of study they are matriculated in. People who want to participate in e-learning courses of the vhb e.g. for the purpose of further education can do so as long as they are willing to pay the necessary fee. If they succeed in the exam they can also get a certificate.

It should be clear from the description given in chapter 1 that our approach explicitly differs from further education. Nevertheless, we have experienced that some company members are interested in getting certificates for their participation. So whenever a company member searches for a co-operation in our portal, we want to display further education offers of the vhb as an additional service for the user. If he is interested in any of the vhb courses, he can follow the link provided in our portal to the corresponding vhb course page. The HTML user interface of the vhb (developed as Java Server Pages) provides a course search functionality which we can access for our purpose.

4. ARCHITECTURE OF THE PORTAL

As we have explained in chapter 2, our web portal has been planned as a Typo3 extension which makes PHP the appropriate programming language. For persistence our portal uses the Typo3 database API and for the user interface JavaScript, which communicates with the server via AJAX. Typo3 provides an AJAX interface and the framework Scriptaculous¹⁹ based on the Prototype²⁰ framework. So far our implementation of the portal uses only features of the Prototype framework, but none of Scriptaculous' graphical widgets and features. To display an orderable list of the available courses, we use the TableKit²¹ framework.

As many of the data available in the portal come from other sources (cf. chapter 3), it would have been possible to do without storing these data locally in the portal, harvesting always the current versions from the sources when the data are needed. This approach would have had the advantages of no needed disk space for storage and no needed persistence program code and would have assured that the portal always works with the latest version of the data. On the other hand when one of the sources is temporarily unavailable, it is beneficial to have a local copy of the data. The most important argument for storing the data in a local database is, that the planned search and filter functions for co-operation offers do not deliver satisfying results in a reasonable time when based upon contacting several sources. We believe that it is the best compromise to use locally stored data for the co-operations overview and for searching whereas to contact the sources for getting latest data when course or lecturer details are to be displayed. We designed the architecture of our portal in a way that the local data are being updated by a PHP Cron Job executed in an interval the system administrator can define. An "External Sources Manager" Component of the portal guarantees, that whenever newer versions of data are harvested from any source, they are updated in the local database and whenever an external data source is unavailable the stored data are used instead. The described default behavior of using local data for overview and search and harvesting external data for course and lecturer details can be changed by the administrator in the portal's system settings. It would be also possible to force the system to always use local data when a threshold of active portal users or response time of external data sources is exceeded.

The Structured Entity Relationship Model (SERM) of the basic version of our portal is shown in fig. 2. It assumes that every course has only one lecturer but different co-operation types (CoopModality) are possible within the same course. The co-operation types differ in activity and involvement of the industrial partners.

The portal architecture follows the Model View Controller Pattern (MVC, cf. fig. 3). The model manifests in the tables of the MySQL database belonging to the Typo3 installation. The controller consists of the PHP files of the Typo3 Extension and the JavaScript files are the View component of the portal. The package structure of the PHP and JavaScript files again follows the MVC pattern. As well as the tables in the database, the PHP objects are derived from the SERM. The mapping between both is mediated by the Typo3

¹⁸ Bavarian Virtual University: <http://www.vhb.org>

¹⁹ JavaScript framework Scriptaculous: <http://script.aculo.us>

²⁰ JavaScript framework Prototype: <http://prototypejs.org>

²¹ JavaScript framework TableKit: <http://millstream.com.au/view/code/tablekit>

built-in database API. The PHP controller classes process the incoming AJAX requests and provide the appropriate responses (both mediated by the Typo3 AJAX API). On JavaScript side, the connection to the server is mediated by the Prototype AJAX framework, the incoming and outgoing data are represented by JavaScript Objects (JSON) processed by a JavaScript controller and displayed by JavaScript objects for DOM manipulation. This architecture enables high flexibility of the solution and supports maintenance of the portal. Being a Typo3 extension it can be easily integrated into an existing website managed by Typo3 and configured from a usual Typo3 backend page. The standardized External Sources Modules can easily be adapted to other data sources and the External Sources Manager and Local Persistence Manager guarantee standardized access and synchronization between local and external data.

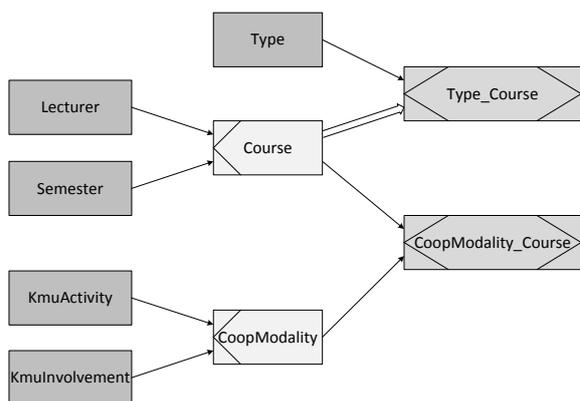


Figure 2. SERM of the portal (basic use cases)

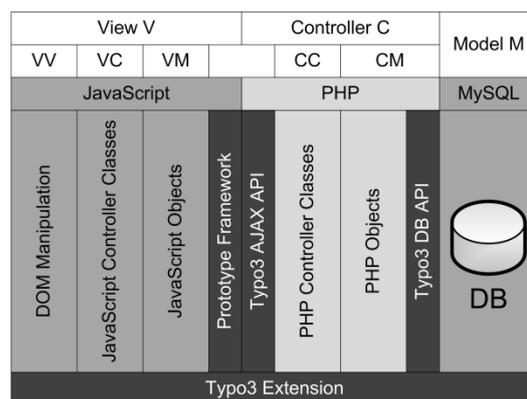


Figure 3. MVC architecture of the portal

5. CONCLUSION AND OUTLOOK

We have developed and prototypically implemented a web portal as entrance point for industrial partners to join co-operations with lecturers mediated by a university-wide LMS. We have considered easy usage, simple provision of important background information to the industrial partners, and an intuitive search function as well as reducing the effort for lecturers and system administrators. The prototype of our portal is a specific implementation for the existing moodle LMS and other systems at the University of Bamberg, but we tried to build it as flexible as possible to make it adaptable to different system environments. We have experienced difficulties because of the heterogeneous IT systems environment consisting of a lot of different systems with different data export interfaces (or none at all). Data quality is also an important issue as some systems provide optional data fields with ambiguous meaning and therefore unequal usage by the lecturers. We will extend the pilot phase of our project and use our portal to initiate co-operations and evaluate the usage of the portal. In the future we want to implement the advanced features (cf. chapter 2) and adapt the portal to other source systems and LMS at other universities. We will also evaluate the possibilities of implementing different user interfaces, such as applications for mobile phones or social communities.

REFERENCES

- Henrich, A. et al, 2007, Integration eines hochschulweiten LMS in die Systemlandschaft einer Hochschule – ein pragmatischer Ansatz. *Flexibel integrierbares e-Learning – Nahe Zukunft oder Utopie?: Proc. Workshop on e-Learning*. Leipzig, Germany, pp. 57–76.
- Henrich, A. and Wolf, S., 2006, Virtueller Campus der Uni Bamberg: ein übergreifendes eLearning-System auf Basis von Moodle. *e-Learning: Weg von der lokalen Insel zur globalen Community; Proceedings of the Workshop on e-Learning*. Leipzig, Germany.



LMS4KMU is co-financed by the European Social Fund.
“ESF in Bavaria – We invest in people”