DotTeach: A portal system for the academic enterprise

Rafael A. Calvo [(1)], Robert Ellis [(2)]
(1) Web Engineering Group – School of Electrical and Information Engineering
(2) Institute for Teaching and Learning
The University of Sydney

Abstract

This paper discusses the design, development and use of a portal system for the administration and integration of teaching activities to improve their effectiveness and coherence. The system provides 1) A knowledge repository of syllabus and instructional design of courses in a university 2) functionalities that enable staff to perform administrative activities online and 3) a collaboration environment supporting the knowledge repository. The three functionalities are integrated to benefit key stakeholders: students, who can benefit from a better understanding of the relationships between courses; teachers, who are able to see a reduction in their administration activities; and administrators who can use the system as a reporting tool. At the center of the system are the course descriptions that can be summarized and displayed to students and can be used for automatically processing some of the common administration activities that academics must perform when delivering a course.

1. Introduction

Most universities have several enterprise systems that provide solutions for managing staff, students, and tangible resources; and for their online delivery of course materials. Very few systems have been designed to support the university as a 'learning organization'. This term coined in 1990 [1] has generated a large number of interpretations depending on the context. More recently Laurillard [2] described what it would mean for Higher Education institutions to be learning organizations.

"A learning organization … is one that attempts to conduct an internal learning conversation that allows it to learn from experience, and adapt to its environment".

In an increasingly volatile environment, Higher Education institutions need organizational and technical infrastructures that support such learning and adaptation. This concept is related to what is known as "knowledge management" (KM) in the corporate world. It has become increasingly important due to new trends within institutions including: an increased rotation of teaching staff, a larger number of short-term contract academics, and increased workload demands on academic staff. Paradoxically, despite the perceived importance of knowledge management processes and systems, academics also perceive that their institutions are not implementing enough of these initiatives [3]. The institutional needs combined with the availability of web technologies that allow workflow, information organization and collaboration online, give us an opportunity to create software systems that help institutions build knowledge repositories based on the history of teaching activities, while reducing academic workload and improving the quality of teaching outcomes.

Information Systems research has shown that systems are successful only when users can clearly see their benefits. It is user commitment and user motivation that are the

main drivers of successful systems that transform organizations, not the systems themselves [4]. Universities have three main groups of stakeholders (students, academics and administrators), and organizational learning is about the development of a number of skills in each of these groups and in the institution as a whole.

In this paper we describe a system designed to achieve clear outcomes for each stakeholder group, including:

- providing students a complete and updated description of courses, concept maps and regulations related to courses they must take or choose from.
- providing academics the ability to manage and communicate more effectively with students.
- helping academics spend less time on administrative tasks.
- providing new and casual administrators with clear guidelines of the responsibilities and processes involved in teaching a course.
- providing administrators and academics the ability to easily collaborate with colleagues on the design of a course.

Students, of course, need to learn the domain specific knowledge for which they have enrolled and a number of generic attributes that will make them employable and valuable members of society. In order to achieve a deep learning experience students need to understand how different courses, and other university offerings, interrelate and support their learning experiences. Academics need to learn and be up to date not only within their knowledge domain but also in relation to teaching expetise within their discipline. Administrators, including support staff and university managers require an organizational infrastructure that enables them to engage in their common activities efficiently, learning from experience and adapting to the environment. Online collaboration functionalities can promote meaningful discussions between these three groups.

Generally speaking, a portal system provides an integrated set of views of one or more applications. In this project we describe a portal system that integrates the functionalities of a knowledge repository specially designed to gather information from academics who design and teach courses, and to deliver modified forms of this information to students and administrators, within a collaboration management system that supports the collaborative learning experience of all these stakeholders.

The knowledge management discipline focuses in studying best practices for this type of collaborative approaches, where the expert's knowledge is shared and managed in order to promote a "learning" organization. It has been shown that a key issue for the success of these initiatives is to go through a consultative process while designing the information systems that will support it [4].

Section 2 provides a brief review of the institutional context in which the system was built, the system requirements and the scope of the project based on this institutional context and an enquiry process and the functional design of the system. Section 3 describes the features implemented in our portal system and the dotTeach application. The system was built extending the functionalities of the dotLRN Learning and Collaboration Management Systems, and the system implementation is described in Section 4. Section 5 concludes.

2. Context, requirements and design.

The dotTeach system was built for the Faculty of Engineering at the University of Sydney. The faculty has 3,000 students and 280 academics (over 50% are adjunct or casual lecturers), organized in 4 schools.

The course syllabus contains the detailed descriptions of a course. They are distributed to prospective and enrolled students through different channels: hardcopies are distributed in class, the course description are available the web and summary copies are also published in the University Handbook. The faculty recognized that students were not getting a single version for a particular course creating a number of pedagogical and policy infringement issues. A common complaint was from students who argued that assessment policies for a particular course were not explained, or that the policies explained were not followed by the academic. A group within the faculty had also been studying how concept maps, created manually from the syllabus, helped students understand the relationship between courses and improve their learning experiences [5-7]. In addition, the faculty was committed to move this work towards a more sustainable technology and to improve the quality assurance processes. It was decided that the concept maps should be produced dynamically from the syllabus, but a simple content management solution did not seem enough. The original goal of the system was then to build a database with the syllabus of each course taught by the 4 Schools in the faculty.

Two additional forces affected the final design. Since inception the system was envisioned for a wider usage, possibly by the 40,000 students and 7,000 staff at the University of Sydney, so a broader set of requirements was taken into account. Also, an audit of the faculty's processes showed the need to improve communication between staff, particularly between Schools.

The original project aimed at building a web-based software application that would make the process of designing and managing a course more efficient and more effective. Early in the system design it became obvious that the same system could improve course management, facilitate the collaboration of academics designing new courses, build knowledge repositories of teaching strategies, automate what are now paper based administration tasks, and facilitate better communication between academics and tutors.

In summary, the additional goals were to:

- Build a knowledge repository of course descriptions.
- Maintain a course "history" that can form part of an institutional knowledge repository of teaching strategies used in each field of knowledge.
- Facilitate communication between academics and other teaching staff (i.e. tutors)
- Elicit and document common administration tasks in the design and running of a course, and implement web-based functionalities that automate these and improve the quality of their outcomes.
- Build the system in a way that can be easily customized for use by other institutions in Australia and worldwide.

The requirements could be divided into those related to a repository (content management) and those fulfilled by collaboration tools. The intranet portal produced is made of two components: a system to manage the teaching activities (dotTeach), the

syllabus management and delivery, and a collaboration framework (dotLRN). The same dotLRN system is also used as a Learning Management System to deliver courses in over 20 universities around the world.

dotTeach offers three sets of functionalities, all centered around the course syllabus and focused on each of the roles involved in this process (student, academic, administrator): student visualization tools, teaching functionalities, and management functionalities.

2.1 Student visualization tools

The original requirement for the system was to use information captured from the syllabus to automatically create dynamic concept maps that help students visualize and understand relationships between courses and degree structures. Students can use these graphical representations to 'navigate' a degree structure, find the relationships, requirements and learning goals of different courses [6, 7]. These visualization tools can also include other type of information such as the type of graduate attributes developed in each course.

When academics deliver the course syllabus information such as course content, course outcomes, assessment criteria and course relevance it is important to guarantee the quality and accuracy of the information provided to the students. A fully articulated course syllabus helps students to be well informed about their course, and encourage students to be engaged in the subject content rather than administrative details of the course or the degree program [5]. It can be used by students systematically throughout the semester as a point of reference in terms of their progress towards the learning outcomes of their subject. A poorly articulated course syllabus creates confusion and discourages deep learning strategies.

2.2 Teaching functionalities

The dotTeach application requires a single data model centered on the course syllabus and designed to allow a relational database to use the information in a number of ways. For example, instead of filling out many different administration forms at different times with repetitive information, an academic enters information once, and it can be used automatically to speed up other processes and even be used for new and innovative outcomes not possible before, including:

1) Administration, publication and versioning. Academics will be able to write the syllabus as they design the course.
2) Course 'requirements' and 'recommended electives' are used to produce concept maps that help students better understand relationships between courses.
3) Automatic tracking of graduate attributes developed in a course
4) Automatic timelines that detect schedule overlaps and improve coordination between courses
5) Efficient textbook requests. As soon as the academic publishes the list of books to be used, the library and local bookstore are informed.
6) Workflows for the budgeting and approval of tutor time.

Additionally, collaboration functionalities should facilitate communities of practice around teaching areas. The system will support the creation, storage and sharing of

teaching journals. Tutors can keep a journal of experiences to share with course coordinators or other tutors. Academic staff can collect strategies to share with tutors or fellow academics. In this way knowledge can be built upon and reused in a collaborative environment. Online collaboration will support teaching and learning committees by giving them an online environment for asynchronous collaboration as well as a repository for storing information from meeting minutes, planning calendars, materials, etc. These same collaboration environments can support academics involved in collaborative course design.

2.3 Management functionalities

Academic managers (i.e. Heads, Deans, Directors of Teaching and Learning or Undergraduate Studies, etc.) often face difficulties with 1) managing quality assurance processes, 2) professional development efforts, 3) budgetary planning and 4) communication with academics. The dotTeach system described here is aimed at reducing the workload and improving the quality of outcomes. Specifically:

1) Quality assurance processes
- Managing student workload: The system will guarantee the number of daily/weekly deadlines does not exceed the value specified by each school. Where clashes occur a year coordinator can liase with academics.
- Managing graduate attributes: The system will assure that each course satisfies the minimum attributes requirement defined by the associated professional association (i.e. Engineers Australia) and facilitate the documentation of teaching strategies that prove successful for each of those attributes.
- Taking full advantage of student evaluations: Students' evaluations are extremely important to quality and the system would provide an automatic analysis tool to graph results and store them.
- Managing and analyzing teaching strategies, particularly tools that can report how specific strategies affect resource allocation like classrooms and ICT infrastructure.

2) Professional development
- Processing of scholarship index and Department of Education reporting.
- Recording of publications and other contributions useful to the teaching community.

3) Budgetary processes
- providing an interface for tutor allocation and management (that can automatically prevent tutor time going over-budget or time overlaps)
- Academic time allocation and loading: This feature would graphically represent data about time spent on different Course activities, so managers can quickly see how new management strategies affect demands on academic time for "presentation", "design", "tutoring", "marking", etc.

4) Communication
- The dotLRN system to be used is already a very mature collaboration tool. Collaboration features would allow a better communication of policies and requirements and more importantly improve conviviality within the institution.

2.4 Collaboration functionalities

Social researchers have shown that ICT has the potential of impacting on the social structure of a pedagogic activities [2], [8]. Communication, or the lack of it, will affect how academics see (and teach) their discipline. Within the University implementing the technology, this view was reflected by an internal audit of the University's Academic Board. The Academic Board recommended that the faculty implement a system to could improve the communication between academic staff and between staff and students that related key aspects of their teaching and learning activities more closely.

The consultative process for these requirements was led by two teaching and learning directors with support from the Faculty teaching and learning committee, so it was highly driven by its teaching and learning goals and not by the technology itself. The Faculty's teaching and learning Director (supported by a research assistant) is managing the relationship with academics, finding requirements, engaging them in the process and managing the operation of the system. The Faculty's Associate Dean for Information and Communication Technologies, defined the overall vision for the project, managed the requirements analysis and the implementation of the system. After a period of gathering information about how academics use the syllabus in their teaching, we engaged students and academics in the requirement elicitation and the implementation processes. After a prototype was developed the Dean of the Engineering Faculty and other management were given the opportunity to provide feedback. The system was released to academic staff and students in the first semester 2005.

3. System Features and Implementation.

The system was designed so it would build on the work done by other institutions and could be used by yet other ones. Several open source projects, with licenses that allow for this type of sharing, provide collaboration and content management components. The dotLRN Learning and Collaboration Management Systems was chosen based on previous development experience and the ample functionalities available.

Student visualization tools: dotTeach 1.0 provided all the visualization functionalities originally planned, regrettably the technology chosen (HTML) has not been able to provide good enough usability. The Concept and Graduate Attribute maps are displayed as HTML tables. Concept maps display the requirements for a particular course, and can be used by to students to understand the relationship between courses and plan which course to take in a given semester. Graduate attribute maps display how much emphasis a course gives to developing skills such as 'information literacy'.

The course map shows all courses associated with a chosen course-work program arranged in a hierarchical order of levels. Each box contains the name of the course, its code and the semester it is offered. By clicking on a particular course within the grid, all core pre-requisites and advisory pre-requisites for that course are illustrated with different colors.

As mentioned earlier HTML tables are not very flexible as a visualization tool so an SVG (Scalable Vector Graphic) 'composer' was built. This application would build SVG interactive images that students could interact with. Although it substantially improved usability not enough browsers have SVG plug-in and some like our library

systems do not allow a new plug-in to be installed. We are currently planning to re-implement this component to generate Flash images using Laszlo Systems tool.

Teaching staff functionalities: dotTeach 1.0 already provides an efficient way to gather and use the course syllabus information, and manage administrative tasks. Figure 1 shows the interface for the current release. Only the tutor management functionalities (workflow and budgeting) have not yet been implemented. The implementation of some the functionalities is particularly interesting as it highlighted how different the internal processes of schools within a faculty can be, and how a common software system has to cover them all.

Course information is managed in dotTeach using a version control system that allows academics to work on the content, keeping versions of the work in progress and 'submitting' their changes when ready. The submitted version can then be approved by the director of undergraduate studies or the school's delegated authority (many people for some Schools). When the changes are approved they are published in the public website and become part of the 'stable' repository used by the Faculty to produce the handbook and other legal documents.

Course descriptions are divided in nine sections, each containing a different set of information fields:

1. Course Details: Describes basic information about the course: aims, learning outcomes, credit value, lecturers. Figure 2 shows a screenshot of this section.
2. Pre-requisite Details: Lists of pre-requisite, recommended courses and prohibitions.
3. Grade Descriptors: What do different grades mean in this course, how do students relate what they have learned to what grades they get?
4. Graduate Attributes: What 'generic' skills will students get in this course? The academic chooses from the group of 4 attributes defined by the University or creates his own. The university of Sydney has a standard set including communication skills, information literacy, personal and intellectual autonomy, research and inquiry.
5. Teaching/Learning Approach: Different learning situations and what students are expected to learn by: e-learning, lectures/seminars, tutorials and laboratories.
6. Workload Requirements. How much are students expected to work in the different learning situations, including after hours work. This section also includes a description on how academics will provide feedback to students' work.
7. Assessment Methods: Describes each of the assessment items: assignments, examinations, online activities, projects or other assessment tools the academic may use.
8. Course Schedule: The main activities and deadlines for the 13 weeks of the semester
9. Texts/References. Books and references used during the course. This information is automatically mailed to the library and bookstore, together with the number of students enrolled, so they can supply enough copies.

Figure 1: Teachers interface

Management functionalities: dotTeach 1.0 only fulfills some of the management functionalities described earlier. **Figure 2** shows the interface for administrators. Unexpectedly, our work for the current version improved quality assurance processes within the faculty just by forcing the discussion within the management of each school. DotTeach 1.0 produces reports for the status of each unit and student workload and assessment weighting schemes. For the next release we are including a section on the use of Information and Communication Technologies (ICT), textbooks, etc in each course, school or the overall faculty. The centralization of this information makes possible and efficient to investigate the effect of teaching strategies on student satisfaction, resourcing and workload.

Collaboration Functionalities: The dotLRN framework also provides tools to facilitate collaboration between colleagues and improve communication with tutors. Teaching communities such as '1st year experience' or 'Teaching and Learning committee' have an online environment that allows them to share files, discuss in virtual forums and keep others updated of their work through mailing lists and news postings.

This entire application (including dotLRN) is implemented using the OpenACS web application framework [9, 10]. The framework uses AOLServer and the Tcl programming language, and provides the option of using Oracle or PostgreSQL as a database (to this date DotTeach only runs on PostgreSQL). A datamodel with a number of tables is used*:*

uos: This table stores the unique 'Unit Code' and the 'Item ID' for every new course entered into the database.

uos_revisions: Provides the versioning functionality. The unique key, ‘revision_id’ identifies each of the revisions. Only one of the revisions of a syllabus is set as the live-revision. This table also utilizes the OpenACS standard tables, ‘cr_revisions’ and ‘cr_items’ which provide content repository support.
stream: Maintains a list of all the available degrees within a chosen faculty.
uos_stream_map: Maps courses to degrees.
uos_qualify: This table maps each course code with the code of its pre-requisites.
graduate_attributes: This table maintains a list of all available graduate attributes that may apply to any of the courses offered within the faculty.
uos_attribute_map: Each row in this table maps a course to an attribute.

OpenACS implements an object model on the relational database. Objects can be assigned attributes and operations. This OO model improves the reusability and quality of code since error checking, authentication and security in the user interaction, can be applied to content ‘types’. The content types (classes), ‘stream’ (the Australian equivalent of ‘degree’ in American English) and ‘uos_revisions’ are the only abstractions that were implemented for this package. They are sub-classes of the ‘acs_object’.

The ‘stream’ object utilizes the permissions system which is part of the OpenACS framework so that a new stream can be added only by the system administrator or someone with equivalent permissions.

The ‘uos_revisions’ object uses the OpenACS permissions system to populate the table. The permission system and content revisions are implemented in the OpenACS Content Repository package. With this package, adding any new content updates the ‘cr_items’ table and any successive revisions are stored in the ‘cr_revisions’ table. Only one of these revisions is set as the ‘Live’ revision.

The administrator has permissions to edit any of the displayed content. By editing the existing content, a new version is created which can be made ‘live’ for students to view. The administrator has the option of viewing the entire ‘revision history’ of a course description.

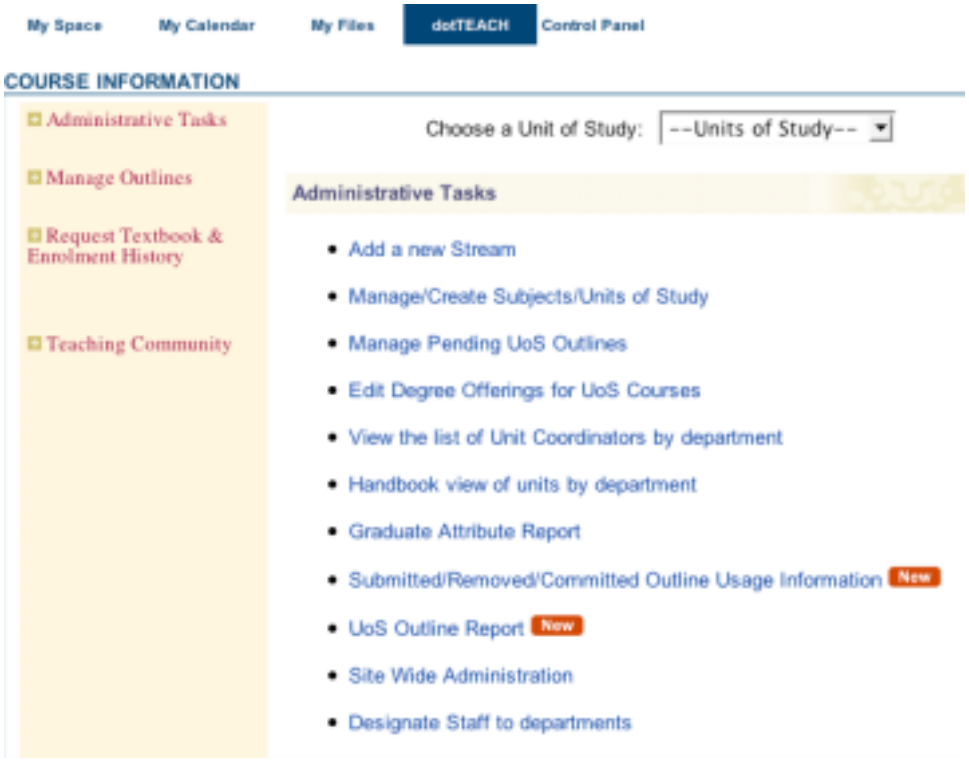

Figure 2: Administrators View of the Course Outline

4. User analysis

To date, system has predominately been used by teachers during its first implementation in 2005. A number of illuminative responses from staff have been captured which suggest a rigorous study would be useful in the coming months. For the purposes of this paper, the quotations below represent the extent of the variation in the users reponses to-date.

When asked what was the main purpose behind the use of the dotTeach system, one teacher reported:

> *I am using the system for basic 2005 course info to advise students. In other words, the subject descriptors and Course subject lists. I am using it because the 2005 Faculty handbook which has the official information does not yet exist so the intranet is the next best thing. As a dynamic source of information it will probably be more accurate than the handbook (if and when it comes out).*

The dynamic nature of the system, that information is more easily and readily available than the traditional mediums, was emphasized in this user's response. The same user was then asked how he was using the system.

> *I've used the subject maintenance components on two Units of study. I've done some editing on the aeromechanical web site to link to the intranet... I use my previous ITC experience to make educated guesses (about how to use the*

system) and then ask the people who deployed the system if this doesn't work. So far I've only needed to do this 2 or 3 times… I have used the on-line collaboration part to post 1 or 2 questions (about how to use the system).

In this response the user-teacher reports that he uses the system largely through intuition and his previous knowledge. He also reports that he is starting to use the functionalities of the software to make links between the courses in the students' course more explicit and linking them to discipline-specific content on the aeromechanical website in his faculty.

In contrast, another user did not find the experience of using the software as simple.

The interface is not intuitive and it is not at all clear what sort or information is expected in some of the fields. There are way too many mouse clicks - a single table format would be much easier that having to click on all those "Expand" buttons.

These responses by user-teachers of the dotTeach system provide some key directions for review of the software. While one found the system could be used through intuition with some help, the other did not. In addition, such responses indicated that only a small percentage of the systems functionalities were understood by the users. As a consequence, subsequent versions of the software functionalities will include greater description of the purpose and structure of the fields in each interface, with a more detailed overview of the relationship between the system and they key teaching and learning functions it is designed to support. This will be provided through pop-up windows and linked to the central help function.

5. Roadmap and Conclusions

This paper reports on version 1.0 of our system. Version 1.5 is expected in late April 2005 and it will focus on improving the usability of the system, it will include a section on the use of ICT, and some new reporting tools. Depending on funding, version 1.8 expected in December and will include better visualization tools for students, integration with our electronic portfolios ('eFolio') system and tools for keeping track of professional development activities and achievements such as research publications in Teaching and Learning.

One of the key outcomes for dotTeach 1.0 was to produce a complete course syllabus for students and to articulate the connectedness of individual courses. There is evidence that a better understanding of the course goals, and how they relate to the overall degree promotes a deeper approach to learning. The generated concept maps from the syllabus can communicate the various learning pathways available to students and therefore encourage students to further understand their course as part of whole rather than as individual concepts within their degree.

The portal application also makes it easier to perform administrative tasks, maintain and track changes in the syllabus, and improve the quality assurance processes in the faculty.

Through the collaboration functionalities, a conversation and exchange of ideas about courses and instructional design issues can take place between academics. The

system may improve the focus on teaching and learning within the faculty and will provide an opportunity for quality assurance to be carried out across all schools in the faculty to ensure consistency in the quality of information. Managers will be able to view the course information provided to students and assess the quality of the syllabus according to the standards set out by the faculty.

Improving usability is our main goal now, but a few additional functionalities will help improve the quality of the entire application and are part of our future roadmap:

- Multimedia (Flash) based concept maps
- Improved reporting tools.
- Search Engine.
- Improved usability/accessibility

Acknowledgements

The authors acknowledge Surath Uthayakumar, John Currie, Frances Zaballa and Jon Suen for the helpful discussions in writing the paper, and for their effort in the project described here. This project was possible thanks to a grant by Teaching Improvement Fund of the University of Sydney.

References

[1] P. M. Senge, *The Fifth Discipline: the art and practice of the learning organization*: Random House, 1990.

[2] D. Laurillard, *Rethinking University Teaching: A conversational framework for the effective use if learning technologies*, 2nd ed: Routledge Farmer, 2002.

[3] G. R. Oliver, M. Handziz, and C. Van Toorn, "Towards understanding KM practices in the academic environment: The shoemaler's paradox," *Electronic Journal of Knowledge Management*, vol. 1, 2003.

[4] Y. Malhotra and D. Galletta, "Building Systems that Users Want to Use," *Communications of ACM*, vol. 47, pp. 89-94, 2004.

[5] P. J. Cornwell, "Concept Maps and competence charts: a roadmap to understanding the global curriculum," presented at ASEE/IEEE Education Conference, 1996.

[6] J. Turns and C. J. Atman, "Concept maps for Engineering Education: A cognitively motivated tool for varied assessment functions," *IEEE Transactions on Education*, vol. 43, 2000.

[7] J. D. Jones, "Concept Maps and Competence Charts: A road to understanding the global curriculum," presented at ASEE/ IEEE Education Conference, 2000.

[8] L. C. Freeman, "The impact of computer based communication on the social structure of an emerging scientific specialty," *Social Networks*, pp. 201-221, 1984.

[9] R. Calvo, J. Sabino, and R. Ellis, "OpenACES: the open source solution to e-learning," presented at Moving Online II, Gold Coast, 2001.

[10] R. A. Calvo and D. Peterson, "The OACS application framework," presented at Ausweb, Sunshine Coast, Qld, 2002.