User Scenarios for the design and implementation of iLMS

Rafael A. Calvo
Web Engineering Group
University of Sydney

The aim of this article is to discuss possible user scenarios for "intelligent" Learning Management Systems (iLMS) and challenges for implementing them. We focus on those scenarios in which Machine Learning (ML) can be used to enhance general purpose web-based Learning Management Systems. We will propose a software engineering framework for the design and implementation of an iLMS.

1. Introduction

Before we can move towards creating "intelligent" Learning Management Systems (iLMS) we would need more detailed descriptions of user scenarios and the challenges for implementing them.

We can state that the basic goal of an iLMS is to improve the learning experience by personalizing and adapting the instruction, based on what the system knows about the student.

Because this goal is the foundation for Intelligent Tutoring System (ITS), it is appropriate to discuss previous work in this field to inform the research on iLMS. ITS are created for one specific course, based on domain-specific pedagogical research. For example, a researcher takes common problems that students encounter in a particular mathematics course and programs an ITS course that listens for these problems and responds with appropriate help. Intelligent Tutoring Systems require an excellent understanding of the types of problems students will encounter and the tools that instructors can use to help them (both of which are highly dependant on the particular subject area.) They also require a formal model of the student's knowledge, normally acquired using knowledge acquisition techniques that require an expert in the subject to write series rules describing the domain. The down side is that these requirements can be time consuming, expensive, and are unsuitable for certain knowledge domains.

Web-based teaching (such as through systems like ITS) is becoming a common part of educational, corporate and government organisations. Such teaching tools are used to deliver learning content online. The learning content may include multimedia such as video that students are able to access remotely, or PDF files of class handouts. Web-based teaching is not restricted to higher education environments; it is also common in corporation for training new employees and adapting the skills of existing staff according to the company's changing needs. Ideally, web-based teaching improves the learning experience of students and employees through features that are not available in face to face learning, and it allows students to access the learning materials and interact with the rest of the course at any time and from anywhere. Furthermore, E-learning has the capacity to personalize learning in a way that is not always possible in corporate training facilities or large lecture halls. Already, some general purpose Learning Management Systems (LMS) use student information to produce personalized calendars, send alerts about classes and deadlines, and check for overlaps in commitments. According to the e-Learning Guild (a leading community for e-learning professionals), educational, corporate and government organizations will increase e-learning expenditure by 85% in the next year. The National Institute of Standards and Technology estimates that instructional software sales will grow to $A300 billion by 2005.

This current interest in e-learning systems has been influenced by the need to train larger numbers of people, more quickly and more efficiently. In the late 1970s and early 1980s, researchers studied how to build software applications that adapted to what was known about the user [5,9,15]. Early research already showed that the systems would need to model (a) knowledge of the domain (called the expert model), (b) knowledge of the learner (student model), and (c) knowledge of teaching strategies (the tutor). There are two different ways these systems have been built: for individual courses and for general-purpose delivery of courses. The choice between these two approaches greatly affects the type of functionalities they can offer.

1.1 Course specialised systems

Student modelling [11] is employed in Intelligent Tutoring Systems [3,4] for specific knowledge domains such as training in algebra and computer programming [5,17], or in performance based tasks such as word processing. In this approach, ITS use a problem based approach to teaching. The system normally starts by assessing what the student already knows, creating the student model, and modelling what the student needs to know, based on the designed curriculum (known as the domain expert). Using these two models and following a teaching strategy, the system must decide what learning object (material such as a figure or a slide) ought to be presented next, creating a customized sequence [I][II][III]. From all of these considerations, the system selects or generates a problem, then either works out a solution (via the domain expert), or retrieves a prepared solution. The ITS then compares its solution, in real-time, to the one the student has prepared and performs a diagnosis based on differences between the two. In the next stage, the student receives feedback on the problem solved, the program updates the student model and increments the learning progress indicators. These updating activities modify the student model, and the entire cycle is repeated, starting with selecting or generating a new problem.

The increasing computing power of the personal computers has allowed researchers to build more elaborate models and improve the quality of these Intelligent Tutor Systems. In particular, they have created web interfaces [4,17] to their systems, allowing more students to access them. However, each of these web-based systems is still only delivering single courses, and maintaining independent models for a particular knowledge domain. This puts a number of limitations on the wide adoption of this technology:

- *Expensive*. Building the knowledge base for these systems is a time consuming and expensive task, making ITS unfeasible except for very well defined domains. One way of improving the usability of these systems is to use machine learning algorithms [7,17]. This releases the designer from having to write explicit rules and provides the system with novel ways of using content. Instead of using rules, these algorithms learn from the examples found in training corpora (the collection of data used to create the models). The algorithms can be trained to find similarities between documents [7], enabling us to build a system that can recommend content to the student, that can tag it and deliver it in a customized sequence.
- *Poor course reusability*: Intelligent Tutor Systems are not engineered in such a way that they can be used by many instructors (only the original development team would normally have access to changing the instructional design of the course). Appropriate software engineering practices must be followed in order to build a scalable and robust system that is usable by different instructors and that developers can maintain [6]. Object Oriented Application Frameworks [8] provide such methodology and can be applied to iLMS.

- *Poor object reusability*: The learning material produced for one course (often packaged as "learning objects") cannot be easily reused in other courses. For example, if an instructional designer produces a series of images for a course, it is not easy for another designer to take these images and reuse them in their own course.

1.2 General purpose systems

The second way in which e-learning systems have been built is as general-purpose systems that can be used to deliver learning content on any topic. These general-purpose LMS systems (i.e. WebCT, Aspen, Blackboard) manage large numbers of online courses, students, assessment, collaboration and teaching material. They have been widely studied and commercially used for the last 5 years. The commercial systems can be used to administer hundreds of courses and distribute them to thousands of students anywhere in the world. The information being learned and the people learning are often the two most valuable assets of an organization (this being especially true for higher education institutions). However, these systems do not have the adaptive features found in the web-based ITS described above [1,3,4].

In order to manage different types of content, from different knowledge domains, and to be able to exchange it between organizations (e.g. libraries), standards are being defined so content from one LMS can be easily ported to another. In these standards learning material is normally packaged as a number of "learning objects". The organization and management of these learning objects are well specified and can be extended. These standards also specify syntaxes for metadata descriptors and describe how content can be organized in well defined sequences [I],[II],[III]. Governments and large corporations are investing heavily in building learning object repositories. Work on iLMS could improve the reusability of this material, and create the metadata descriptors (a process called tagging) that is often as expensive as creating the learning objects themselves.

Machine learning can help tag these documents since the learning content can normally be converted to a textual representation and can be modelled statistically and learned by a machine learning algorithm [7,17]. Even when the learning content consists of images, the content is described by metadata that can be used to create the models. These models can be integrated with other applications [17] using the modular architectures becoming more common in LMS and intelligent user agents [2,16].

A number of theoretical and practical challenges have prevented machine learning models being used in these general-purpose systems, and this workshop aims to tackle these problems. The machine learning algorithms that have successfully been used in other areas can be applied to create e-learning systems that adapt to the student's model. We plan to study how these algorithms can be used for the management of learning objects, so they can be easily tagged, re-used and recommended to students and so they can be sequenced adaptively to the learner's model. We have named the LMS that use these techniques "Intelligent Learning Management Systems" (iLMS).

In section 2 of this presentation we discuss three possible use scenarios for the iLMS, in section 3 we analyse the challenges of building such applications, section 4 concludes.

2. Three Scenarios

User centred design requires that researchers and engineers study possible "user scenarios" and do a more formal use case analysis as is common in software engineering methodologies. These user scenarios are useful for defining and scoping systems and for communicating requirements to other developers.

In our approach to building iLMS, the student and domain models integrate a statistical representation of what the system knows with a knowledge base made of rules. The machine learning approach is used in two user scenarios: document tagging and recommendations systems, where training corpora already exist. The rule-based approach, as is often used in ITS, can be used for the sequencing of learning objects, where training data is harder to obtain.

We discuss here three possible scenarios where this combination of machine learning and rules can be used to improve the on-line learning experience.

2.1 Content Management and Tagging

In many commercial e-learning systems, content will be in the form of learning objects as defined in the Sharable Content Object Reference Model (SCORM). The management tasks include the difficult task of automatically tagging the objects in schemas as defined by IMS. An interesting application for ML methods is to use automatic document classification for tagging these objects. The classification could be based on pre-existing taxonomies, like the Open Directory Project (ODP) which provides a hierarchical description of web resources.

Several machine learning models have been applied to document classification: neural networks, support vector machines, k-Nearest Neighbours (kNN), Roccio, and Naïve Bayes. We have tested these algorithms for automatic classification tasks with data sets including news from the Reuters cable [7], questions and answers from a forum in teaching mathematics and web pages. From this experience we believe that we can classify documents efficiently for e-learning applications.

In order to use these machine learning algorithms we need a vector representation. The course materials are treated as text objects and these texts are then converted to very high dimensional vectors. We will do this using the Term Frequency (TF) and Inverse Document Frequency (IDF) definitions. Term Frequency (TF_{ij}) weighting is equivalent to the number of times the term *j* appears in document *i*. Inverse Document Frequency (IDF) weighting is defined as: $a_{ij} = IDF_{ij} = \log(\frac{N}{O_j}) + 1$, where N is the number of documents in the collection and O_j is the number of documents with term j. Normally, the products $f(TF_{ij}) \times g(IDF_{ij})$ are used. The dimension of these vector spaces is proportional to the number of terms remaining after stop-word removal and stemming. Even for moderate size text collections this can be tens or hundreds of thousands of terms. This is prohibitively high for some algorithms so dimensionality reduction techniques are needed. Normally, several techniques including document frequency and χ^2 are used.

The machine learning algorithms can be trained to find patterns in the texts (vectors). They can be used for automatic classification by using [**input**,**output**] pairs as examples. The **input** components are the vectors representing the texts, and the **output** are the classes that those texts belong to.

2.2 Recommendation systems

This feature would allow iLMS to recommend the most appropriate content to students. Recommendation systems are common in other web-based systems [14]. In the iLMS case the recommendation system may list the closest available learning objects to what the instructor describes as the module's content. Proximity is based on the vector representation of the texts as described above. Learning objects from the available repositories will be chosen and recommended to the student based on this proximity measure.

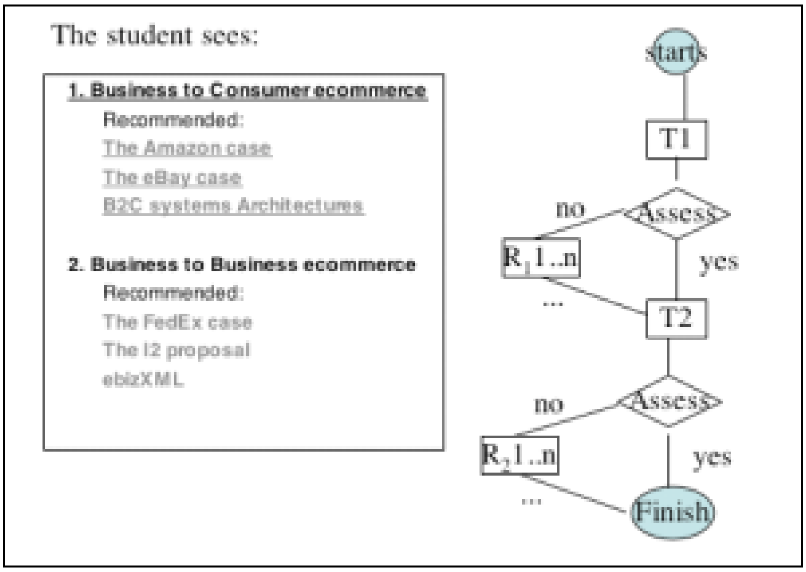

Figure 1: Interface and workflow for a possible scenario in an iLMS

In order to successfully use these techniques we need to relate them to a real user scenario. Figure 1 shows a possible scenario for an iLMS being used to teach e-commerce systems. In this scenario two topics have been identified by the instructor, and their titles are highlighted. Other documents from the available repositories have been recommended by the system. These documents are a subset of the nearest neighbours (in the vector space sense) to the instructor's document. The subset uses a filtering based on the student models and any constraints from the teaching strategy. The diagram on the right shows the workflow used for sequencing content: if the student fails, additional content is provided from the recommended ones or an explicit one by the instructor. If the student passes, the system directs him/her to the next reading (2 in this case). This content is not accessible until the sequencing engine says so (so the titles are not underlined links).

The "Intelligent" layer of an iLMS will provide efficient use of all available learning objects. By automatically tagging and organizing documents in a structured taxonomy, they can be more easily found by instructional designers and students. They will also add a level of abstraction to concrete learning objects. An instructional designer could set a sequence where the student needs to read about a topic after showing proficiency in another topic. These "topics" may be abstract, broadly defined concepts, probably described by a standard piece of text, and not a concrete learning object (e.g. slide, video). This way, when a better learning object is made available it can be immediately recommended to the student.

2.3 Sequencing of learning objects.

Machine learning could be used to improve iLMS in two ways: to recognise similarities between documents, and to allow learners and instructors to use them for reinforcing concepts. A set of rules will be defined by the instructor as learning strategies. These sequences will integrate a workflow approach and the SCORM v1.3 - IMS Simple Sequencing Specification. This application would be a breakthrough in the way LMS work today, and it will change how learning objects are used in tomorrow's e-learning systems. Novel approaches could be used for firing transitions and workflow events such as test results, or the inclusion of a document with certain characteristics.

The sequencing engine could be based on Petri Nets modelling of a workflow system. If the LMS is built using an existing web application framework [6], a pre-existing workflow system may be adapted for such a purpose.

Several engineering standards are being developed that will influence the way the student models are represented and used in iLMS. The IMS Learner information package (http://www.imsproject.org/profiles/) defines a way to represent student information. The SCORM 1.3 standard defined by the Advanced Distributed Learning Consortium (http://www.adlnet.org) will influence the way content is sequenced.

3. iLMS Challenges

In order to evaluate the most important challenges in building an iLMS, we first need to look at some activities that will be required. Implementing an iLMS requires a series of steps, including:

- Use case analysis.
- Theoretical and architectural analysis of the iLMS.
- Theoretical and performance analysis of ML algorithms seen as potential candidates for e-learning applications, including neural networks, support vector machines, Naïve Bayes, K-Nearest Neighbours.
- Integration of an LMS with a ML framework.
- Analysis for sequencing standards and implementation in the LMS.
- The intelligent sequencing engine is trained and tested using available corpora. A complete analysis of the engine would provide a better understanding of how an instructional designer may create a sequence of abstract learning objects that suits those learning needs, and how the concrete objects can be identified and/or assembled.
- The algorithms and the architecture are re-evaluated based on analytical performance results.
- Usability tests that can provide usability and design guidelines.
- The theoretical framework is re-evaluated to include results and feedback related to general purpose LMS.

The central challenges for building iLMS are to (1) improve the learner and knowledge domain models; (2) improve the engineering of e-learning systems. These issues will need to be studied in the (3) context of a system design that is centred on the student. We discuss here the challenges related to the engineering of such systems, the ML models to be created and how to design a system centred on the user/learner.

3.1 User and knowledge domain models

Non-adaptive instruction hinders learning as it does not adjust to the student's preferred learning tendencies, cultural background or previous learning experiences. Learners sometimes prefer a structured, linear or hierarchical approach to learning, or sometimes a less structured approach. Some learners find it easier to learn from visual representations while others are more comfortable with text-based presentation. Learners have different learning histories and may require different emphases in different learning contents. Computer systems are needed that adapt to these constraints and deliver appropriate content to each student. Phenomenographic research (e.g., Laurillard [13]) focuses on how adaptive tutoring systems could be able to diagnose "what a student needs to be taught". This research shows that the student model should enable the system to know the student in the same way he/she knows the domain, and both models should use comparable knowledge representations in order to identify what the student needs to be taught. These models of the domain can be used to process data automatically—recommending, filtering and sequencing learning content adaptively to the students. The challenge of such an approach is to obtain appropriate precision in these models and make them scalable for large amounts of data.

The standard approach to building a student model is to adjust an initial model to the knowledge made explicit by the student [11]. A developed web-based system can be based not only on assessment results as in most ITS, but also on reading/navigation behaviours [12]. The system would then produce a modified syllabus based on the individual student model. A complementary approach involves assessing incoming knowledge and skills, either instead of, or in addition to, emerging knowledge and skills. The approach discussed here is novel as it models student behaviour in the collaboration environment surrounding a course, this includes all the student's contributions to discussion forums, file sharing, etc. Using machine-learning techniques for pattern recognition [1,2,7], we plan to build models of the students' behaviours, the knowledge domain and integrate them into the Learning Management System (LMS). To our knowledge, these type of models have never been integrated into a general-purpose system.

3.2 Software Engineering Issues

The goals of adaptive learning are to: "develop computer systems that provide or support effective learning experiences for a wide range of learners across a broad spectrum of knowledge domains" [10, p. 395]. Web based systems that provide such support need well engineered architectural designs. One of the software engineering techniques used to achieve this is the division of problems and solution into smaller parts called software components. This divide and conquer approach extends the capabilities of ITS by breaking down the problem (and the systems) into smaller parts [3,16]. Component based architectures have since been applied to the development of several ITS as well as in general purpose iLMS. Other techniques such as application frameworks extend this approach but have yet to be exploited in building e-learning systems. They are challenging to build and maintain because they make the design guidelines and the source code of one project applicable to another, so the big picture of building complex e-learning software systems, and the minutiae of their implementation must both be addressed in detail [8].

LMS are also large software systems that require innovative approaches in design and implementation. Because they are so expensive to produce and have such a big impact in an organization, they need to be robust, maintainable and reusable in different contexts. We are designing an iLMS by using a new software engineering methodology known as Object Oriented Application Frameworks. Application Frameworks [8] are software artefacts designed to increase reusability of design and implementation. They are a combination of software components (implementation code) and conceptual solutions that have stood the test of time (design patterns). Design pattens have been widely used to communicate ideas in the software engineering community, describing a problem to be solved, a solution, and the context and tradeoffs of that solution. Finding the design patterns in e-learning applications is a difficult problem that has not been addressed; it requires looking at architectural problems and solutions across several software systems and specifying them formally.

3.3 Student Centred approach

A student centred approach is important for this research project, and is equivalent to what software engineers call user centred design. A student centred approach is based on an understanding of the student needs and is exactly what pedagogical researchers and practitioners do normally. A student centred approach improves the impact that these software systems have on education, but it requires a multidisciplinary collaboration. Several student centred issues have already been addressed in LMS and ITS research and will be considered in our iLMS project. For example, some researchers have argued that ITS are weak in inception as they are based on an instructionist approach that emphasizes the transfer of tutor's domain knowledge to the student (as opposed to a holistic approach where learning is seen as a two way process). Also, some authors question the notion of domain knowledge and our ability to represent it. We believe that recent developments in ML and language technologies improve the way we represent and acquire knowledge. Assuming that the knowledge is contained in the learning materials (documents) used by the instructors, we can represent it as a collection of large dimensional vectors, and machine learning algorithms can be used to find patterns and commonalities in them. This approach has been used in other fields such as financial markets [17] and the semantic web where novel technologies have been employed to improve the way we use information.

4. Conclusions

In this paper we bring together ML and general purpose learning management systems. Previously, learning management systems have been simple in their management of learning content. Even the simple sequencing of learning objects, that allows an LMS to present learning material in a sequence, has not been fully developed [I]. As a bootstrap definition, we will say here that Intelligent Learning Management Systems are LMS that have adaptive behaviour, can assist in the management of learning objects using machine learning techniques, and can learn from a student's performance and sequence the delivery of learning content according to that individual's learning needs. This novel way of managing content improves the reusability of the learning objects, and reduces their production and maintainance costs.

Acknowledgements

The author would like to acknowledge Elizabeth Wilson, Dorian Peters and Ernie Ghiglione for their assistance in the writing of this article.

References

1. Boticario, J.G. and Gaudioso E. "Towards Personalized Distance Learning on the Web". The 5TH International Work-Conference on Artificial and Natural Neural Networks (IWANN'99). Alicante, Spain, June 2-4. 1999. In: IWANN'99 Proceedings (Volume II). Eds. J. Mira and J.V. Sánchez-Andrés. Lecture Notes in Computer Science 1607 (pp. 740-749).
2. Boticario, J.G., Gaudioso E. "A Multiagent Architecture to Support Distance Learning Personalization on the Web" . The 13Th FLAIRS Conference (with AAAI), In J. Etheredge and B. Manaris, editors, Proceedings of the 13th International Florida Artificial Intelligence Research Society Conference, pp. 124-127, AAAI Press, Orlando, Florida, USA, May 22-25. 2000.
3. Brusilovsky, P. (1995) Intelligent tutoring systems for World-Wide Web. In R. Holzapfel (Eds.), Poster proceedings of Third International WWW Conference. Darmstadt, April 10-14, 1995 - pp. 42-45
4. Brusilovsky, P. (1999) Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, 19-25.
5. Burton and Brown. 1982. An investigation of computing coaching for informal learning activities. Sleeman D. and Brown J. (eds) Intelligent Tutoring Systems. Orlando Florida. Academic Press.
6. Calvo R. & D. Peterson "The OACS application framework". Ausweb 2002, Sunshine Coast, Australia.
7. Calvo R. A. and HA Ceccatto. Intelligent Document Classification. *Intelligent Data Analysis*, 4(5), 2000.

8. Fayad, M., D. Schmidt, and Johnson, R Eds., Building Application Frameworks : Object-oriented foundations of framework design, John Wiley and Sons 1999.
9. Jones, M., & Winne, P. (eds.) (1992). Adaptive learning environments: Foundations and frontiers. Berlin: Springer-Verlag.
10. Jones, M., Greer, J., Mandinach, E., du Boulay, B. & Goodyear, P. (1992). Synthesizing instructional and computational science. In M. Jones, & P. Winne (eds.) Adaptive learning environments: Foundations and frontiers (pp.383-401). Berlin: Springer-Verlag.
11. Kay J. and Kummerfeld R. K., 'User models for Customized Hypertext in Advances in Hypertext for the World Wide Web' in J. Mayfield and E. C. Nicholas (eds), Springer Verlag, 1997, pp. 47-69.
12. Kay J., 'The um Toolkit for Cooperative User Modelling',User Modelling and User-Adapted Interaction, Kluwer, 4:3, 1995, pp.149-196.
13. Laurillard, D. (1992). Phenomemographic research and the design of diagnostic strategies for adaptive tutoring systems. In M. Jones, & P. Winne (eds.) Adaptive learning environments: Foundations and frontiers (pp. 233-248). Berlin: Springer-Verlag.
14. Manber U. Patel A. and Robison J. Experience with personalization on Yahoo!. Communications of ACM. August 2000, vol 43, No 8, 35-39.
15. McCalla, G. (1992). The search for adaptability, flexibility, and individualization: Approaches to curriculum in intelligent tutoring systems. In M. Jones, & P. Winne (eds.) Adaptive learning environments: Foundations and frontiers (pp.91-122). Berlin: Springer-Verlag.
16. Ritter, S. and Koedinger, K. R.: An architecture for plug-in tutor agents. Journal of Artificial Intelligence in Education 7, 3/4 (1996) 315-347
17. Rosatelli M, J. Self & M. Thiry. LeCS: a collaborative case study system. In: G. Gauthier, C. Frasson and K. VanLehn (eds.) Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1839, (Proceedings of 5th International Conference on Intelligent Tutoring Systems,ITS 2000, Montreal, Canada, June 19-23, 2000) Berlin: Springer Verlag
18. Williams K, and RA Calvo. A framework for text categorization. 7th Australasian Document Computing Symposium. December 2002, Sydney Australia.

Web references:
[I] ADL, 2002. Advanced Distributed Learning Initiative. [http://www.adlnet.org]
[II] IMS 2002 Global Learning Consortium Inc. [http://www.imsproject.org]
[III] LOM (2000). *LOM working draft v4.1* [http://ltsc.ieee.org/doc/wg12/LOMv4.1.htm]

Rafael A. Calvo

Rafael received his PhD in Physics (Machine Learning applied to document classification) from the Universidad Nacional de Rosario, Argentina in 2000. He is now Lecturer in Software Engineering at The University of Sydney.

He has worked at Carnegie Mellon University (USA) and Universidad Nacional de Rosario (Argentina), and as an Internet consultant for projects in Australia, Brasil, USA and Argentina.

His overriding interest has been bringing machine learning to information systems to improve their productivity for both government and commercial organizations. These experiences were the source of his book “Introduction to e-commerce systems” published by Pearson Education in 2002. Rafael is author of 1 book and of about 20 contributions (book chapters, journal papers and conferences) in the area of e-business, software engineering and machine learning. He is co-chair of the Intelligent Learning Management Systems Workshop, 2003