Emerging Internet Technologies for E-Learning

Qing Li, Rynson W.H. Lau, Elvis W.C. Leung, Frederick Li, and Victor Lee
City University of Hong Kong

Benjamin W. Wah
University of Illinois at Urbana-Champaign

Helen Ashman
University of South Australia

The electronic learning (or e-learning) industry, including Web-based learning, has been booming worldwide in recent years, due to factors such as reduced environmental costs and impact, quality education made affordable, and convenience and flexibility to learners. The Internet's ubiquity is promoting new sophisticated techniques and powerful applications for a new paradigm in e-learning that uses the Web as an interface for single-user as well as collaborative learning. Web-based learning and new concepts such as virtual classrooms, laboratories, and universities introduce many new issues. The articles in this special issue of *IEEE Internet Computing* highlight three efforts in creating Internet-based e-learning technologies that tackle challenges such as management of learning objects in an open and scalable architecture, incorporation of learners' pedagogical features in Web-based learning environments, and digital game-based learning.

Another challenge for e-learning is creating scalable technologies that support an arbitrary number of users while providing them with a personalized learning environment. Although e-learning systems can be diverse in their domains and implementations, common architectural motives have emerged. Figure 1 shows a reference architecture for e-learning system development.

As the figure shows, we can view a typical e-learning system on the Internet as a distributed system consisting of the *Internet infrastructure layer* (IIL), the *conceptual/modeling layer* (C/ML), and the *application layer* (AL). In the following, we examine some core issues of this architecture and look at applicable techniques and relevant work in each layer.

The Internet Infrastructure Layer

E-learning lets students and instructors participate in learning activities and access a wide range of learning resources, independent of place and time. It supplements traditional in-class learning activities, such as fact finding and experimentation, and supports additional ones, such as organizing personalized learning materials and providing in-

1089-7801/09/$25.00 © 2009 IEEE

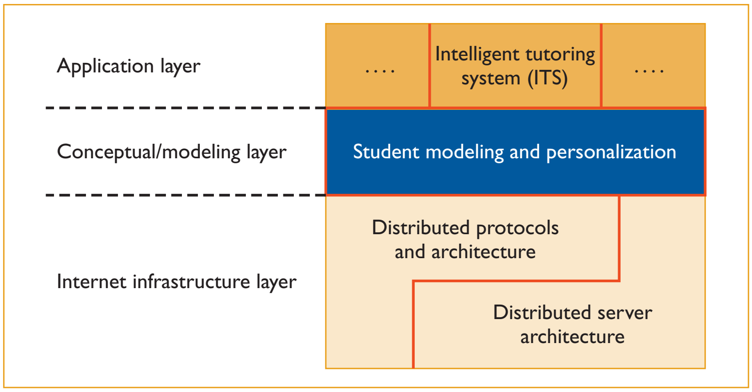

Figure 1. A reference architecture for e-learning system development. This architecture exhibits a layered structure, employing the principles of openness, separation of concern, and component-based development.

stant assessment through online tests. When a course has only a manageable size of e-learning users, a simple client-server architecture will suffice. As the number of users grows, however, the course might require a large-scale e-learning system to handle a potentially large number of concurrent, geographically distributed users and support a large database of e-learning materials. The IIL's main functionality is to address such scalability issues.

Distributed Server Architecture

As multimedia information becomes increasingly popular in e-learning, we find more learning materials composed of various media, from simple textual information and images to complicated and data-intensive video streams and 3D geometry. Such materials support a variety of e-learning applications, such as knowledge retrieval, simulations, and educational 3D games. Handling such diverse information and applications, as well as so many users, requires a multiserver system that can provide enough computing power to maintain system interactivity.

We can broadly classify multiserver systems into three types.[1] A *parallel architecture* comprises servers running in parallel to perform homogeneous tasks. An application that must process substantial data can thus partition that data among multiple servers. Alternatively, the designer might assign the same set of application data to more than one server to facilitate high system availability. A *distributed architecture* comprises servers running heterogeneous tasks and is best for applications that involve complicated logics or substantial operations. In this case, the system can partition the application into different subtasks and assign them to individual servers. Finally, a *hybrid architecture* adopts both the parallel and distributed architectures to form a hybrid that can provide both computational performance and reliability.

E-learning systems are often composed of several applications, including the aforementioned information retrieval applications and 3D games. The former emphasize extracting representative features from files in a potentially huge document database and then using those representative features to select the relevant information, such as course materials. Educational 3D games focus on interactive responses and convenience for learners accessing the games. Because these applications might have different performance requirements, the service provider might employ any number of multiserver architectures. A typical e-learning system might thus comprise one or more architectures, as we can see from the following examples:

- *Document retrieval.* A popular document retrieval system with a huge database is the Google search engine. To support efficient retrievals, each server uses a parallel multiserver architecture to maintain a subcollection of documents and a local vocabulary index. When retrieving documents, every server matches the user-supplied keywords with its local vocabulary index in parallel. The search engine then consolidates results from all servers to obtain a list of relevant documents. To support high availability, the system can deploy replicated parallel servers.
- *Multiplayer gaming.* Multiplayer online gaming lets remote users interact in a shared game. It can become computationally demanding when there are numerous users interacting with the game continuously. In such a situation, the service provider can use a parallel architecture to distribute the workload among multiple servers by dividing the users into groups or the game scene into regions for individual servers to handle.[2]
- *Multimedia applications and streaming.* Multimedia applications give users diverse, multimedia information. To enable interactive response, such applications often employ media streaming, which requires the server to spend a prolonged time period managing

each delivery task. For example, YouTube and Google Earth deliver on-demand videos and high-quality satellite images, which are partitioned among parallel servers, to users. Because multimedia applications might be composed of many subapplications, such as those that deliver satellite images, 3D building models, and hotel information in Google Earth, a hybrid multiserver architecture is useful. In this approach, a distributed architecture can serve the entire application, and a parallel architecture is assigned to serve each subapplication.

Because personal computers are becoming ever more powerful while their cost is very low, employing a distributed server architecture is believed to be a very effective solution.

Distributed Protocols and Component-Based Architectures

During the past decade, vendors have developed numerous commercial e-learning systems, such as WebCT and BlackBoard, which are now widely adopted in schools and universities. Although such systems support e-learning activities, we must address numerous system architectural issues for them to evolve into a truly and sustainable distributed e-learning environment.

Component-based architectures. Commercial e-learning systems are powerful, integrated systems that provide critical functions in a single package to satisfy e-learning stakeholders' requirements. However, such complex, monolithic systems have difficulty meeting requirements for every level of the educational hierarchy, and a change in the domain knowledge might require system-wide modifications.[3] Alternatively, component-based architectures might be a promising approach because they would replace such monolithic e-learning systems with a community of distributed communicating servers. Such architectures should provide one-stop comprehensive support for teachers and students and enable the e-learning community to reuse existing educational systems as components. The architectures' openness and flexibility would let educational systems developers compete by offering better or more innovative services.

Other alternatives come from the open source community, which has been developing options such as Moodle (www.moodle.com), in which educators can integrate new features into existing e-learning systems at minimal cost.

Distributed protocols. We can implement distributed e-learning systems using Web services technology that fundamentally supports a component-based architecture via standardized tools.[3] For example, the Web Services Description Language (WSDL; www.w3.org/TR/wsdl) supplies a common language that lets providers describe and publish their services, whereas SOAP (www.w3.org/TR/soap/) lets users invoke Web services that can communicate and move data among platforms. These Web services enable efficient reuse of services and content across e-learning systems. On the other hand, the dynamic and distributed nature of both servers and learning resources requires software-agent technology to provide adaptive and intelligent support to both learners and tutors. Agents are software programs that can autonomously cooperate with each other and interact with users. For example, we can use agents to develop, at the application layer, adaptive hypermedia (AH) systems that adapt the learning process for individual students on the basis of their preferences and knowledge.[4]

Open-access learning resources. To support users' (learners and software agents) open access to educational materials and facilitate interchangeable and reusable content, several standardization bodies, such as the Advanced Distributed Learning Initiative (www.adlnet.gov) have issued educational standards,[5] which are especially important for large-scale, distributed learning environments. Some of them focus on prescribing methods for storing learning content in a way that facilitates information exchange across learning systems. Standards for describing learning content, such as Learning Object Metadata (LOM), enable learning-object repositories to provide an efficient search for learning content. In contrast, other standards focus on interoperability among e-learning system components so those components are reusable and replaceable. The Shareable Content Object Reference Model (www.adlnet.gov/scorm/), for example, bundles a set of widely accepted standards and specifications that enables conforming e-learning systems to exchange and reuse learning materials.

The Conceptual/Modeling Layer

The C/ML provides high-level tools for supporting e-learning application development. Let's look at student modeling and personalization, an example research issue for the C/ML. User models are essential to e-learning systems, giving students learning continuity, tutors evidence of students' progress, and both a way to personalize students' learning materials to their abilities and preferences. Personalizing information has long been the motivation behind developing e-learning systems, giving rise to the AH systems mentioned earlier. Differences in student ability, background, learning preferences, and so on affect information delivery.

User models have been employed for a range of purposes. In the commercial world, user modeling forms the basis of Amazon's recommender systems, which promote items that sellers believe will interest buyers — this case is particularly interesting because a buyer's screen shows information motivated by many other buyers' purchasing or viewing habits. With this approach, e-learning systems can also incorporate other students' experiences into group learning activities. For example, a cohort of students can build up a rated readings collection that the system can recommend to other students in the same cohort.

User models can be short- or long-term. For example, Amazon implements short-term, pseudonymous user models by tracking buyers' browsing history within its Web site, recommending further items based on what the buyer has already viewed. Long-term user models generally store more personal information for subsequent visits. This latter, persistent user model is common in most e-learning systems because students return to the system regularly. In particular, it lets teachers observe students' interactions with the system to detect problem areas and generally follow students' progress over numerous visits.

We can create and modify user models in two different ways[4]: we can explicitly populate user-model variables (for example, with students' names or perhaps preferences for viewing materials), or we can implicitly update those variables by recording or inferring information from students' interactions with the e-learning system. An obvious interaction would be tests, which assess students' familiarity with the material before they can proceed to more advanced materials.

Modeling users and students, however, generates some issues.

First, getting user models to interoperate over numerous e-learning systems is in some respects more challenging than mere information interoperability because AH systems have adaptation rules as well as content and user models. This raises questions, such as what to do with unused variables when migrating a user model to another e-learning system (and back again); how to enable the receiving system to apply its adaptation rules to an imported user model if some information is missing; and how, or even if, the receiving system should update the user model before sending it back.

Second, user models don't necessarily demonstrate the value of personalization within an e-learning system. For example, some researchers promote learning styles as one aspect to personalizing information for students. They categorize students as *visual* versus *verbal* learners, or *global* versus *sequential*, among many other scales, and tailor the information students receive to what the teacher thinks is appropriate to that learning style. However, some evidence exists that tailoring a presentation to learning styles has no effect, at least among some cohort of university-level students studied.[6]

Finally, a more political issue is that some might construe presenting information differently to different students as unfair, with some students not receiving access to the same materials as others. A reasonable solution is to let students control both their user models and personalization processes.

Despite these issues, e-learning systems can still benefit from user models. Teachers can create a personalized learning plan within a short time, which is especially important with increasing class sizes and online courses. In addition, any personalized learning plan that responds to students' actions will optimize their learning achievements.

The Application Layer

Different e-learning applications are developed at the application layer. As an example, let's look at *intelligent tutoring systems* (ITSs), which exhibit many typical e-learning features.

Researchers have developed ITSs based on artificial intelligence and cognitive science, and they've accumulated considerable results in the past 20 years. The most generic ITS architectures suggest building a good student model that reflects systems' beliefs about learners' mastery level in certain concepts. Moreover, such an architecture enables systems to perform individualized tutoring for learners.[7] To further extend the generic ITS architecture, researchers have developed various types of ITSs based on different domains, pedagogical strategies, and other affecting factors.[5]

Early ITS systems emphasized duplicating the structure of student/human-tutor interactions. The first ITS, Scholar, was developed in the early 1970s to teach South American geography.[8] Success has been limited in such systems due to the technological challenges of making a computer system sufficiently intelligent to answer various questions in a pedagogically useful manner. The Web has changed pedagogical approaches to educational software by focusing more on simpler instructional approaches that are easily computerized. Other ITSs create complex instructional strategies requiring extensive expert-system representations customized for a given knowledge domain. Representative models include geometry and computer programming.

ITSs can also assess students' performance. Two major assessment methods exist: *summative assessments* formally assess whether students have achieved learning targets, whereas *formative assessments* help improve students' achievement of learning targets. Debate exists about whether to computerize the latter in ITSs because some argue that learners must receive feedback at an appropriate point in the learning process. ITSs must gather computerized formative assessments in an efficient and quantifiable manner for them to be useful in providing immediate feedback.

With the Internet's evolution, researchers have attempted to deploy ITSs on the Web. These Web-based systems retain most generic ITS architecture features, such as AH, which generates content with different levels of detail according to users' knowledge. This is known as *adaptive presentation*. In addition, an AH system can offer *adaptive navigation* by giving users directional assistance in selecting the most relevant link. Such adaptive methods' main purpose in an ITS context is to support students in hyperspace orientation and navigation, improving users' performance on a particular task such as "how to learn a topic." Two well-known examples are ISIS-Tutor (cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html) and Hypadapter (www.springerlink.com/content/jkh0k872vu318135/).

In addition, Web-based *adaptive and intelligent educational systems* (AIESs) have begun adopting and benefiting from AH technologies. Examples include Web-administered multiple-choice tutors.[9] The three most popular techniques used in Web-based AIES are direct guidance, adaptive link annotation, and adaptive link hiding, some of which are quite similar to ITS-adaptive sequencing (or dynamic navigation). Indeed, the difference between direct guidance and ITS-adaptive sequencing disappears gradually in a Web context. As long as some educational material, such as presentations and questions, are represented as a set of nodes in hyperspace, the sequencing becomes indistinguishable from direct guidance. Representative examples reflecting these include ELM Adaptive Remote Tutor (ELM-ART), Adaptive Statistics Tutor (AST), and InterBook.

In this Issue

Our reference architecture supports e-learning system development in a world of growing e-learning technologies. This special issue looks at three of those emerging technologies and applications, though selecting just these few was a difficult task.

The first article, "The Ariadne Infrastructure for Managing and Storing Metadata," by Stefaan Ternier and his colleagues, advocates supporting learning object integration in multiple distributed repository networks. It presents and analyzes the standards-based Ariadne infrastructure, which manages learning objects in an open and scalable architecture. Indeed, going for an open source, standards-based architecture is an effective way to move forward in managing ever-increasing digital resources for e-learning systems.

As the e-learning field keeps evolving and expanding, game-based learning represents a promising future learning style. "Game-Based Learning with Ubiquitous Technologies," by Wen-Chih Chang, Te-Hua Wang, Freya H. Lin, and Hsuan-Che Yang, proposes a ubiquitous game-based learning model and examines technical and experimental considerations. The authors' integrated learning environment uses advanced ubiquitous technologies to construct a location-aware, digital game-based learning environment for e-learning users and application systems.

Finally, Tiffany Y. Tang and Gordon McCalla present "A Multidimensional Paper Recommender: Experiments and Evaluations," which highlights the importance of incorporating learners' pedagogical features in making paper recommendations and proposes a pedagogical-oriented paper recommender. The article reports on studies in designing and evaluating a six-dimensional paper recommender that's especially useful to Web-based learning environments.

As an emerging field, e-learning in the broad sense (including distance learning, Web-based learning, and digital game-based learning) has attracted increasing attention from both industry and academic sectors. To facilitate development of successful e-learning systems on open Internet platforms, we need scalable technologies that support an arbitrary number of users while providing them with a good learning environment. Besides the technical problems addressed by the three articles in this special issue, more issues (both technological and pedagogical ones) in developing and deploying e-learning systems will emerge, especially in view of the field's diversity and interdisciplinary nature. Expectedly or unexpectedly, electronic learning might entail life-long research as much as it facilitates life-long learning!

Acknowledgments

We thank the many individuals who helped make this special issue possible, especially editor in chief Fred Douglis and associate EIC Michael Rabinovich, for their strong support and valuable comments and suggestions at various stages of the process. We're indebted to publications coordinator Hazel Kosky for her frequent kind reminders to get this special issue ready on schedule. Last but not least, we thank all the authors who submitted their manuscripts for consideration and the many dedicated anonymous reviewers who helped us arrive at our final decisions.

References

1. M. Norman and P. Thanisch, "Models of Machines and Computation for Mapping in Multicomputers," *ACM*

Computing Surveys, vol. 25, no. 3, 1993, pp. 263–302.
2. B. Ng et al., "A Multi-Server Architecture for Distributed Virtual Walkthrough," *Proc. ACM Symp. Virtual Reality Software and Technology*, ACM Press, 2002, pp. 163–170.
3. F. Lin, L. Esmahi, and L. Poon, "Integrating Agents and Web Services into Adaptive Distributed Learning Environments," *Designing Distributed Learning Environments with Intelligent Software Agents*, Lin Fuhua, ed., Idea Group (IGI), 2004, pp. 184–217.
4. P. Brusilovsky, "Adaptive Hypermedia," *User Modeling and User-Adapted Interaction, Ten Year Anniversary Issue*, vol. 11, 2001, pp. 87-110.
5. B. Woolf, *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning*, Morgan-Kaufmann, 2009.
6. E. Brown et al., "Reappraising Cognitive Styles in Adaptive Web Applications," *Proc. World Wide Web Conf.*, ACM Press, 2006, pp. 327–335.
7. S. Lajoie and S. Derry, "A Middle Camp for (Un)Intelligent Instructional Computing: An Introduction," *Computers as Cognitive Tools*, Erlbaum, 1993, pp. 1–11.
8. M. Grignetti, C. Hausman, and L. Gould, "An Intelligent On-line Assistant and Tutor: NLS-SCHOLAR," *Proc. Nat'l Computer Conf.*, AFIPS Press, 1975, pp. 775–781.
9. D. Hoole et al., "A Bank of Chemistry Questions on an On-Line Server," *J. Science Educational and Technology*, vol. 11, no. 1, 2002, pp. 9–13.

Qing Li is a professor at City University of Hong Kong. His research interests include object modeling, multimedia databases, Web services, and e-learning systems. Li has a PhD in computer science from the University of Southern California. He's a senior member of the IEEE. Contact him at itqli@cityu.edu.hk.

Rynson W.H. Lau is an associate professor at City University of Hong Kong. His research interests include distributed virtual environments, multimedia systems, and computer graphics. Lau has a PhD in computer science from the University of Cambridge. He has co-guest-edited a number of special issues in ACM/IEEE transactions and served as conference cochair and program cochair of several conferences. Contact him at rynson@cs.cityu.edu.hk.

Benjamin W. Wah is the Franklin W. Woeltge Endowed Professor of Electrical and Computer Engineering and professor of the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. His research interests are in the areas of nonlinear search and optimization, multimedia signal processing, and computer networks. Wah has a PhD in computer science from the University of California, Berkeley. He's an IEEE fellow, a fellow of the American Association for the Advancement of Science, and an ACM fellow. Contact him at wah@uiuc.edu.

Helen Ashman is an academic at the University of South Australia. Her research interests include personalization, Web search and log analysis, and hypertext. Ashman has a doctorate from the Royal Melbourne Institute of Technology. She is also joint editor in chief of ACM Transactions on the Web. Contact her at helen.ashman@unisa.edu.au.

Elvis W.C. Leung is a module leader at the School of Continuing and Professional Education, City University of Hong Kong. His research interests include personalization and adaptive e-learning systems. Leung has a PhD in Computer Science from City University of Hong Kong. Contact him at wcleung@cityu.edu.hk.

Frederick Li is a lecturer (assistant professor) at Durham University. His research interests include computer graphics, virtual reality, surface modeling, distributed virtual environments, and multimedia systems. Li has a PhD in computer science from City University of Hong Kong. He has served as program cochair for numerous conferences, including the International Workshop on Digital Entertainment Technology (IDET 08) and the International Conference on Web-Based Learning (ICWL 07). Contact him at frederick.li@durham.ac.uk.

Victor Lee is an assistant professor at City University of Hong Kong. His research interests include mobile data management, transaction processing, real-time database systems, modeling, and performance evaluation. Lee has a PhD in computer science from City University of Hong Kong. Contact him at csvlee@cityu.edu.hk.