Using Domain Ontologies for Online Learning

Raymund Sison

College of Computer Studies
De La Salle University
2401 Taft Avenue
Manila, Philippines
raymund@ccs.dlsu.edu.ph

Abstract

Online learning environments are collections of tools that support the acquisition or construction of knowledge through Internet technologies and applications. This paper describes an online learning environment being developed at the College of Computer Studies of De La Salle University that supports a conversational model of the online learning process and a framework for Internet-based learning environments that utilizes reusable, hierarchically organized knowledge units. The knowledge hierarchy is actually a domain ontology with rich concept descriptions.

Introduction

Online learning, or *e-learning*, can be defined as the acquisition or construction of knowledge through Internet technologies and applications. The Internet in general, and the Web in particular, offer several advantages for learning. For instance, the Web is effectively the single largest, most up-to-date, and most accessible collection of digital information on earth. The Internet also enables agents, whether human or artificial, to communicate and collaborate with other Internet-connected agents at the least cost. Benefits such as these might have led John Chambers of Cisco to single out e-learning as the biggest growth area in the Internet, and the area that will be one of the biggest agents of change (Chambers 1999, cited in Rosenberg 2001 p.xiv).

In this paper, we describe an online learning environment that supports a conversational model of the online learning process (Sison, 2002; Sison, 2001) and a framework for Internet-based learning environments that utilizes reusable, hierarchically organized knowledge units.

Conversational Model For Online Learning

In (Sison, 2002), we have argued, mainly from (1) cognitive constructivist (Piaget, Vygotsky) theory and (2) essential features of the Internet and Web, that online learning entails conversations between learner and teacher, learner and peers, and learner and self. Forming the core of our conversational model, these conversations involve the construction, transmission, and interpretation of, and interaction with messages for the purpose of constructing or acquiring knowledge. Messages can be simple, e.g., a proposition, or complex, e.g., a hypermedia domain ontology[1].

Learner-self conversations, which are primarily for self-regulation (Schunk and Zimmerman 1998), proceed in three phases. The first phase involves the construction of a learning plan. This phase is quite critical, especially in self-directed learning (Candy, 1991), as it is in this phase that the learner will have to determine the best way to achieve her learning goal, given her current knowledge about the domain, about the world in general, about learning strategies, and about self. To support planning activities, an online learning environment must provide at least three types of tools: computer mediated communication (CMC) tools, search tools, and recording/scheduling tools. Analysis tools such as domain ontologies would also be helpful.

The second phase of L-S involves executing the plan developed in the previous phase, and monitoring the execution of this plan. Executing the plan for learning about a domain is, of course, tantamount to learning about the domain. From the constructivist perspective, this entails interpreting one's experiences about the domain, and

[1] Ontology is here used in the Artificial Intelligence (AI) sense, i.e., as a specification of a set of concepts, axioms, and relationships that describe a domain of interest. AI ontologies are developed mainly for automated reasoning and for achieving interoperability among various software and database applications (Niles & Pease, 2002). Though most published ontologies are organized as taxonomic hierarchies, ontologies need not be limited to this form. Moreover, ontologies do not necessarily run counter to constructivism, as ontologies can be viewed as interpretations about concepts and principles of a domain that are widely shared by the community of experts in that domain.

comparing one's interpretation with those of others (e.g., the teacher's, peers', an expert's). Interpreting experiences, in turn, entails the construction and deconstruction of structures that represent, in one's mind, the objects and actions in one's experiences, and plausible relationships among these and other objects and actions from past experiences.

Tools that would support knowledge construction are of three major types: content tools, recording tools, and analysis tools. What we call content tools actually include complex messages, from, say, the teacher, such as microworlds and hypermedia domain ontologies, which the learner will normally have to interact extensively with, possibly using tools that have been built into the complex message. Recording tools will include annotation tools and verbal protocol recorders. In addition, because "deep-level learning rests on the assumption that knowledge is hierarchical" (Candy 1991, p.295) and organizing information into a hierarchical framework facilitates recall (Hofer et al. 1998), the learning environment can also provide the learner not only with an ontology of the domain, but also with a facility to create and edit his or her own ontological interpretation of the domain. Analysis tools would include pattern recognition tools, declarative programming languages, and interpreters for "executing" one's ontology. In addition, all the specific tools for planning are also useful, though to a lesser extent, for learning.

Executing one's learning plan (i.e., engaging in learning) is one thing, monitoring the execution of the plan (i.e., monitoring one's learning) is another. When monitoring self, one basically takes note of whether learning is proceeding as planned or not, and if not, why. For instance, a learner modeling tool that overlaps the learner's ontology over the teacher-supplied domain ontology would enable both learner and teacher to quickly determine any discrepancies and act accordingly. Thus, there are at least three major types of self-monitoring tools: learner modeling tools, reminders, and plan revision tools.

The third phase of L-S conversations involves reflecting on the execution of the learning plan. Here the learner examines the overall result of the learning process, and, depending on whether the overall result is viewed as positive or negative, assigns credit or blame to specific aspects of her plan or other aspects of the overall learning process (e.g., the teacher's style, the learner's mental ability). As a result of this reflection, the learner reconstructs or revises her knowledge about learning in general, as well as knowledge about self. While all the tools for phases 1 and 2 are also useful for this phase, the outputs of the performance trackers and the learner modeling tools of phase 2 are probably the most useful for overall performance analysis and reflection. These outputs, which can be viewed as models of the learner, can be examined against another tool called a *bug library* to determine whether one's mistakes (e.g., misconceptions) have also been made by other learners in the past, and how one can deal with these mistakes. In fact, one's mistakes can be incorporated automatically into the bug library (see, e.g., Sison et al. 2000a, where the bug library is implemented as an error hierarchy), for the benefit of future learners.

Framework For Internet-Based Learning Environments

In (Sison 2001), a framework for virtual learning environments that supports the conversational model was presented. The framework identified three distinct instructional activities, namely, knowledge or content creation, course design, and course delivery, illustrated in Figure 1.

Knowledge about a particular domain is organized in our framework as a hierarchy, or more specifically, a directed acyclic graph, of "knowledge units". Each knowledge unit contains a piece of domain knowledge, which can be a fact, procedure, concept, rule, principle, or strategy of the domain, and which can be expressed in text, utterances or other sounds, still or animated graphics, and/or video. Each knowledge unit will have a tag, which is a keyword or a phrase that concisely describes its contents. Each knowledge unit can also have several attachments. One class of attachments would involve question-answer pairs

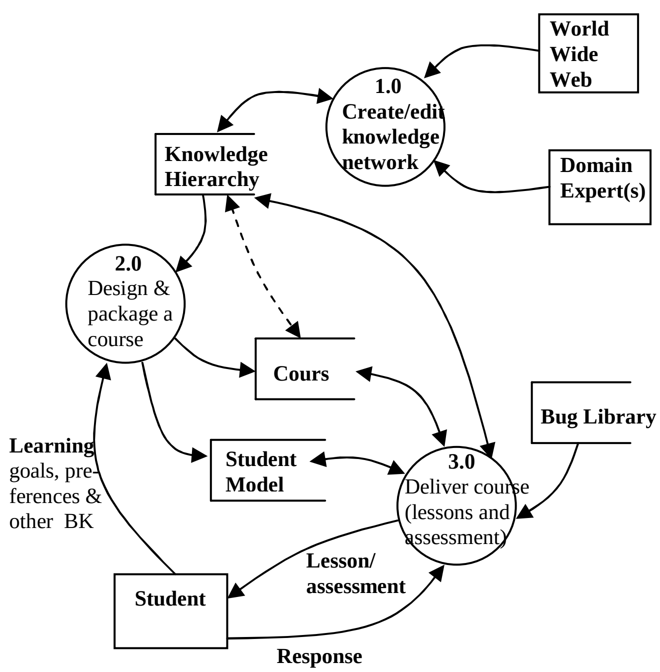

Figure 1. Framework for an Internet-based Learning Environment

related to a particular knowledge unit. Another class would involve pedagogical information on the knowledge unit, e.g., what would be a good way to present the knowledge unit, or what knowledge units would need to have been mastered before a particular knowledge unit is taught. A knowledge hierarchy can therefore be viewed as a domain ontology, with rich concept descriptions.

Creation of the knowledge hierarchy (process 1.0 in Figure 1) is normally done by a domain expert using a special editor that will support the creation and modification of knowledge units, and their relationships to the rest of the units in the hierarchy. Since knowledge units can be viewed as documents, tools for supporting document management (e.g., version control, locking when editing, spelling and grammar checking) will be useful. However, it should also be possible to automatically extend the knowledge hierarchy, using materials collected from Berners-Lee's semantic Web (Berners-Lee et al. 2001). Although an easy way to collect these materials would be through keyword search, more sophisticated techniques for document understanding would yield more trustworthy results.

Although the knowledge hierarchy of a domain can, by itself, be used by any person wishing to learn the domain, this might prove to be too complex for learners who are not adept at self-regulation. Thus, one can use the knowledge hierarchy to design a course (module 2.0 in Figure 1) tailored to the needs (e.g., learning goals) and characteristics (e.g., learning preferences) of individuals or groups.

We define a course as a sequence of presentations (of knowledge units) and exercises/exams, possibly interspersed with discussions and other pedagogical activities. Such a course can be packaged using an editor that will allow (1) selection, from the knowledge hierarchy, of knowledge units that will make up the course; (2) organization of these knowledge units into a taxonomy or sequence of activities; and (3) embellishment of the knowledge units for presentation so that they will capture and sustain the interest of learners. Alternatively, knowledge units can be automatically incorporated into a course, provided that the knowledge units are already in the preferred presentation format (what Merrill 1999 calls knowledge objects).

The last phase involves delivering the course (module 3.0 in Figure 1). Course delivery in our framework is simply a matter of making the course available to the learners. By clicking on elements of a course map or taxonomy, a student can view or listen to a presentation, do seatwork/homework, take an exam, or take part in a discussion. The delivery system must also provide the learner with the tools necessary to support all the self-regulated learning activities in Sison's conversational model.

LK: A Learning Environment Using Knowledge Units

A system called LK (Sison et al. 2001) has been developed that partially implements the above framework. As in the framework, the system has three modules. However, the current version does not yet perform automatic extension of the knowledge hierarchy using the semantic Web. Neither does it perform automatic course creation at this point. However, it enables a course designer to rapidly "initialize" a course by copying individual as well as entire groups (subtrees) of knowledge units from the knowledge hierarchy into the course. These copies are linked to their originals in the knowledge hierarchy so that any changes made to the originals can be reflected in the copies. Since knowledge units (in the knowledge hierarchy) are not meant for presentation, LK provides the course designer with facilities for embellishing the knowledge units in her course (but not the "originals" in the knowledge hierarchy), as well as for reorganizing them.

The LK system is developed as part of the E-College project (Sison et al. 2000b) of the College of Computer Studies of De La Salle University. E-College is a portfolio of online information systems and learning support systems for higher education. LK and E-College are mainly written in Java. Other tools used were Macromedia Dreamweaver UltraDev, Java Server Pages (JSP), and Oracle *x*i.

Summary and Future Work

In this paper, we described a conversational model for online learning and a framework for Internet-based learning environments that uses reusable knowledge units. The framework is implemented in a system called LK, written in Java.

The knowledge hierarchy provides the key to the integration of the conversational model of online learning and the Internet-based learning framework. The knowledge hierarchy not only enables the rapid and possibly automatic construction of customized courses from reusable knowledge units; the knowledge hierarchy, and tools for the construction and reconstruction of one, can also assist learners as they engage in conversations with teachers, peers, and self for the joint construction of meaning and for reflection on their own knowledge constructs and knowledge construction processes.

Future work on LK would be along two lines. The first involves developing and delivering more online courses using LK. The second involves adding more intelligence to LK. For example, intelligent support for knowledge unit acquisition can come in the form of automatic ontology creation (or automatic initialization of the knowledge hierarchy given an ontology of the domain), automatic search of the Web for useful resources if not knowledge units (and the automatic incorporation of these web-sourced knowledge units into the knowledge hierarchy), or automatic knowledge unit construction from digitized material. Additional intelligent support for course creation can come in the form of automatic course and exam creation. Finally, intelligent support for course delivery can come in the form of automatic grading, automatic revision of the course ontology and contents (as a result of unexpected performance variances and revised learning goals), intelligent tutoring, i.e., using a bug library to understand causes of behavioral errors of an individual and to remediate to eliminate the causes of his/her behavioral errors, automatic construction of the bug library, or the automatic but controlled update of the knowledge hierarchy to incorporate useful pedagogical information gleaned from interaction with the students.

References

Berners-Lee, T., Hendler, J., and Lassila, O. 2001. The Semantic Web. *Scientific American*, (May 2001), 28-37.

Candy, P. 1991. *Self-Direction For Lifelong Learning: A Comprehensive Guide To Theory And Practice*. San Francisco, CA: Jossey-Bass.

Chambers, J. 1999. Keynote speech to the Fall 1999 Comdex Trade Show.

Hofer, B., Yu, S. & Pintrich, P. 1998. Teaching College Students to be Self-Regulated Learners. In D. Schunk & B. Zimmerman (eds.), *Self-Regulated Learning: From Teaching To Self-Reflective Practice*. New York: Guilford.

Merrill, M. 1999. Instructional Design Theory (ITT): Instructional Design Based on Knowledge Objects. In Reigeluth, C. ed., *Instructional Design Theories and Models: A New Paradigm of Instructional Theory*. Malwah, NJ: Erlbaum.

Niles, I. & Pease, A. 2002. IEEE Standard Upper Ontology Draft Proposal.

Rosenberg, M. 2001. *E-Learning: Strategies For Delivering Knowledge In The Digital Age*. New York: McGraw-Hill.

Schunk, D. & Zimmerman, B. (Eds.). 1998. *Self-Regulated Learning: From Teaching To Self-Reflective Practice*. New York: Guilford.

Sison, R. 2001. Framework for Intelligent Internet-based Learning Support. Proceedings of the *International Conference on Computers in Education 2001 (ICCE 2001)*, Seoul, Korea.

Sison, R. 2002. A Conversational Model of Online Learning. Forthcoming.

Sison, R., Numao, M. and Shimura, M. 2000a. Multistrategy Discovery and Detection of Novice Programmer Errors. *Machine Learning*, 38, 157-180.

Sison, R., Pablo, A., and the E-College Team. 2000b. Value Chain Framework and Support System for Higher Education. Proceedings of the *Southeast Asian Regional Computer Conference 2000 (SEARCC 2000)*, Manila, Philippines.

Sison, R. and the LK Team. 2001. Learning Environments Using Knowledge Units. Proceedings of the *Philippine Computing Science Congress (PCSC 2001)*, Manila, Philippines.