

The Design of An Intelligent Tutoring System Based on the Ontology of Procedural Knowledge

Chun-Hung Lu, Shih-Hung Wu, LiongYu Tu, Wen-Lian Hsu
Institute of Information Science, Academia Sinica, Taiwan, R.O.C
Email: {enrico, shwu, masia, hsu}@iis.sinica.edu.tw

Abstract

This paper presents a new model to simulate procedural knowledge. This method divides procedural knowledge into two parts: process control and action performer. By adopting this method, we intend to help teacher construct curriculum and teaching strategies by capturing students' problem-solving process. Using the concept of procedural knowledge in intelligent tutoring systems, we can accumulate and duplicate the knowledge of teacher/curriculum manager and student model. Teacher/Curriculum manager can help the teacher create good learning maps for students. The student model can help the teacher collect students' error types and design more appropriate teaching strategies. The implementation of our system is near completion. We will design a user-friendly interface for the system and have students play with this software to collect feedback.

1. Introduction

Scientists and researchers have done extensive studies on mental representations. A person's mental knowledge generally begins with noticing and remembering. Mental representation is being called upon to provide knowledge. According to Anderson (Anderson, 1993), there are two essential components of spatial images: declarative knowledge and procedural knowledge. Declarative knowledge collects useful knowledge/information. It is the factual or conceptual knowledge that a person has. In designing a generic architecture to represent procedural knowledge, the actions defined by domain experts and the control of action flow are two important tasks. Unlike declarative knowledge, the meaning of procedural knowledge cannot be figured out until the whole process is finished. Processes are hard to describe but important in relation to problem solving. Self (1999) showed that focusing on the process by which

knowledge is constructed is more important than focusing on target knowledge. In this paper, we design an ontological representation scheme called process map (PM) to represent procedural knowledge. The main idea is to identify (1) the activity structure from given behavioral models of components; and (2) the connection relations of these components.

Most researchers use declarative knowledge as the sole basis for ontology simulation. However, in Sowa's opinion, a paradigm of declarative knowledge construction has largely failed to produce human-like cognitive processing in computers (Sowa, 1999). To cope with this situation, we develop an ontology, InfoMap, based on both declarative knowledge and procedural knowledge (see Figure 1).

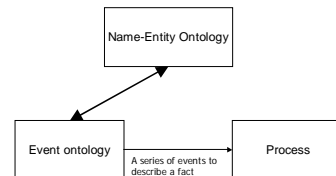


Figure 1. The conceptual architecture of our ontology

2. The Conceptual Architecture of Our Ontology

Our ontology is implemented based on InfoMap (Hsu, 1999), which was originally created as a named-entity ontology. We now have extended it to include event ontology and process map.

2.1 Named Entity Ontology of InfoMap

Knowledge representation has long been a bottleneck in simulating human understanding. Several strategies have previously been proposed for natural language understanding. But many have been confined to illustrations in textbooks rather than actually implemented in large-scale Natural Language Systems. The fact is that different representation schemes are appropriate under different situations.

We designed a knowledge representation scheme, InfoMap (Hsu, 2001), to facilitate both human browsing and computer processing of the domain ontology in the system. The domain ontology is constructed from structured concepts in each specific domain. Examples

of concept structures range from simple concepts, such as a word, a phrase, or an event, to more complex concepts, such as a sentence, a paragraph, a script (a collection of related events), a story, or the passive tense of English and so forth. Each concept is associated with a structure (a sub-map) describing the relationships of this concept to its related concepts. The system can store a large amount of events, syntactic or semantic structures and scripts. Given a natural language sentence, the system tries to match it to a sub-map or decompose it into several events in InfoMap.

We represent InfoMap as a tree hierarchy (Figure 2). There are two types of nodes: concept nodes and function nodes. The basic function nodes are: category, attribute, synonym and event. Function nodes are used to label the relationships between two concept nodes.

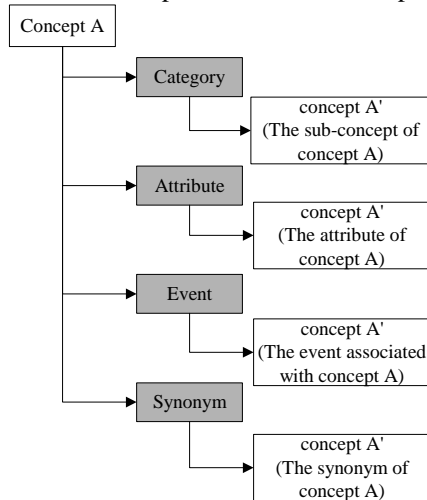


Figure 2. Ontology format of InfoMap

2.2 Event ontology

The name-entity ontology does not deal with the relations of events. To represent the relations in the sentence, “John saw Mary swimming in the river,” we need to apply event ontology. As an example of Hartley (Hartley, 1985), consider a simple change of state in that part of the world involving John and Mary. John has a book that he gives to Mary. The action of giving results in two state changes. First, John no longer has the book. Second, Mary has it now. We may choose to represent the sequence linguistically as the following events:

John has a book.
 John gives the book to Mary.
 Mary has the book now.

where the event “John gives the book to Mary” can deduce the precondition, “John has a book” and the

effect, “Mary has the book now”. In Section 2.2.1 we discuss in more detail the relations of events and illustrate these relations using the problem-solving process of subtraction.

2.2.1 Process Map

We call a “series of processes that indicate a fact” as a Process Map (PM). A Process Map is composed from a collection of processes that accomplish a specific goal. Each process can be regarded as a series of actions. Each action can be further decomposed into a series of frames. This idea is very similar to the decomposition of a large program into components. A process can be categorized into atomic and composite ones with different parameters in PM. An “atomic” process means a single process that has three properties: ID, processName and type, and five attributes: precondition, input, output, effect and action. A composite process is one composed from atomic processes.

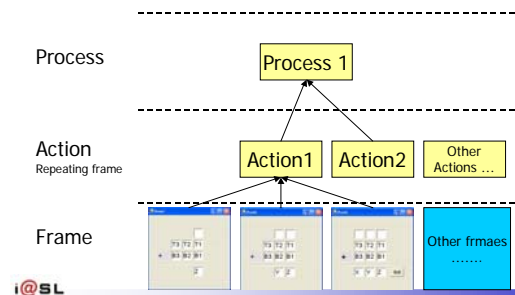


Figure 3 The relations of processes, actions & frames

The architecture of Process Map includes two parts: process engine and action engine. The function of process engine is to control processes, which includes junction control and direction control. An action is defined as the interpreted behavior of a component. An action engine is a mapping from actions to components. We need domain expert to analyze all required actions, and to identify exactly which actions can help learning.

To reduce the complexity of the learning problem, the task performance knowledge is decomposed into a hierarchy of operators, each of which encodes the strategies to execute a sub-task. The procedural knowledge can be treated as a series of processes^[1]. They are connected with junctions and links. The direction of the flow of each instance (a process or an action) is decided by the precondition of each process. In the following, we use a procedure of arithmetic subtraction in elementary school to illustrate our Process Map. The procedure is adapted from (Brown and Burton, 1978). Figure 4 gives a visualization of

^[1] i.e., production rules

our Process Map. Suppose we have a problem (p1) of subtraction in column:

$$\begin{array}{r} T3\ T2\ T1 \\ -\ B3\ B2\ B1 \\ \hline \end{array}$$

In the process of problem solving, we find that a person who finishes K-12 education regards subtraction as a reflex concept. But a novice needs to know a lot more detailed actions when they learn the concept of subtraction. Therefore, we use Process Map to represent the process of problem solving. We can compose and decompose an action based on users' need. Such a representation is very flexible and provides different views of the same procedure. As shown in figure 4, the gray boxes indicate composite processes, which can be further decomposed. For example, composite process C can be decomposed into process F and process G. Process G can be either regarded as an action or further decomposed into more detailed actions. The white boxes indicate atomic processes.

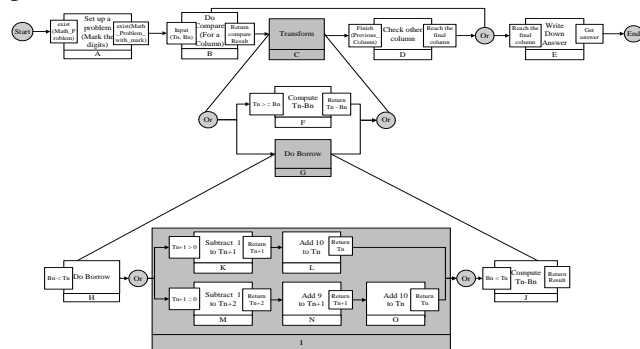


Figure 4. The visualization of the Process Map

3. An Intelligent Tutoring System Developed on InfoMap

An Intelligent Tutoring System is a computer program that uses artificial intelligence techniques to teach a person. Researchers use ITS to find out how people learn. There are four software components that emerge from the literature as part of an ITS (Burns and Capps, 1988). They are the **Expert Model**, **Student Model**, **Teacher/Curriculum Manager model**, and **Instructional Environment**. Based on InfoMap, we have implemented the first three models..

3.1 The Expert Model

We can see an expert model as a repository for storing and organizing information. It should include knowledge that a teacher wants students to learn.

With sensitivity analysis, a component or system can be examined to see how responsive its behavior is to differences in the information given to it (Gaschnig

et al, 1983). This could be particularly relevant when evaluating ITSs that offer individualized instruction. The sensitivity of an ITS towards different learner characteristics might indicate whether additional teaching expertise needs to be incorporated into the system.

In the initial phase the domain expert determines what actions will be used. The system designer develops actions and registers them in a repository. At the same time, the expert constructs an exemplification to help teachers use these actions. After a complete course of interaction, the expert collects the teachers' comments to revise the actions.

3.2 The Teacher/Curriculum Manager Model

The teacher uses curriculum manager to arrange learning modules (lesson plan), where each module may include one or more learning objects to help students learn. Each learning object has its teaching strategy. Teaching strategy and curriculum can all be represented in InfoMap.

The curriculum map gives teachers more comprehensive understanding of what they are prepared to teach. It can eliminate sequencing errors, and enable teachers to develop lessons that are truly interdisciplinary (Martin, 1994). Similar to an outline or a flowchart, we describe curriculum map by PM. Every element in the curriculum map can be regarded as a composite process that can be further divided into more detailed processes. Finally, we represent the procedural network of subtraction (Brown and Burton, 1978) by Process Map (as shown in Figure 5).

Not only teaching strategies can be represented in curriculum manager by Process Map, it can also be arranged as post-conditions with error types in Process Map. After teachers have collected students' problem-solving procedures and error types, they can update new learning map for students.

Curriculum manager creates a sustained cycle: "curriculum design, teaching strategies design, recording (student's behavior), error analyze and feedback to teaching strategies," which can help other teachers create good learning maps for their students.

the goal of the task: The more detailed information one needs, the more detailed conditions are required (or more depths in the Process Map).

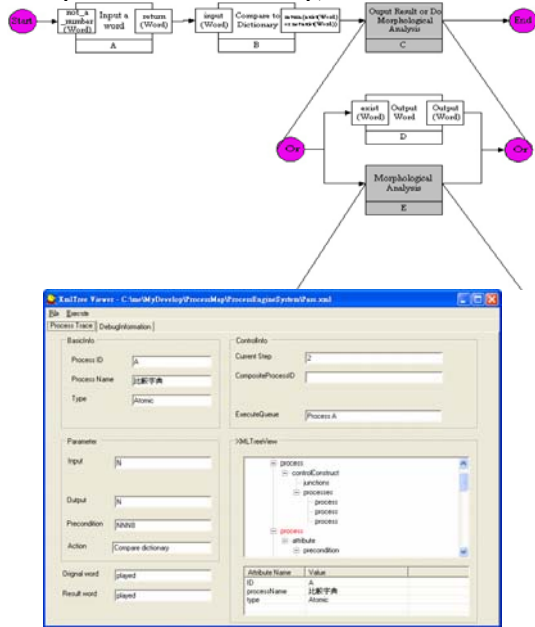


Figure 7. A Process Map of Morphological Errors

5. Conclusions

This paper describes domain knowledge and student knowledge representation in our ITS based on InfoMap. By using Process Map in ITS, we intend to help the teacher accomplish the following tasks:

Accumulate teaching experiences by observing the experts or other teachers' teaching strategies.

1. Develop teaching strategies with personal style.
2. Observe students' learning map.
3. By detecting and classifying students' error types, design appropriate teaching strategies.

Our ITS implemented by InfoMap can also help students in the following way. If the student has any systematic and predictable misconceptions, the system determines the underlying reasons for such errors by expert's arrangement. The Process Map will record students' behavior, which can provide more feedback to teachers.

6. References

- [1] Anderson, J. R., Rules of the Mind. Hillsdale, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1993
- [2] Brown, J.S., & Burton, R.R. "Diagnostic models for procedural bugs in basic mathematical skills.", In Cognitive Science, 1978, pp. 155-192.
- [3] Burns, H. L. and Capps, C. G., "Foundations of Intelligent Tutoring Systems: An Introduction. Foundations of Intelligent Tutoring Systems." , Lawrence Erlbaum Associates, Hillsdale, NJ., 1988.
- [4] Carpenter, T. P., "Learning to add and subtract: An exercise in problem solving. In E. A. Silver (Ed.)", Teaching and learning mathematical problem solving: Multiple research perspectives, Hillsdale, NJ: Erlbaum., 1985, pp. 17-40
- [5] H.L. Wang, W.K. Shih, C.N. Hsu, Y.S. Chen, Y.L. Wang, and W.L. Hsu, "Personal Navigating Agent", Proceedings of AGENT'99, 1999.
- [6] H.L. Wang, S.H. Wu, I.C. Wang, C.L. Sung, W.L. Hsu and W.K. Shih, "Semantic Search on Internet Tabular Information Extraction for Answering Queries", Proceedings of the Ninth International Conference on Information and Knowledge Management, Washington DC, 2000.
- [7] Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E. & Terry, A., "Evaluation of expert systems: Issues and case studies." In F. Hayes-Roth, D.A. Waterman, & D.B. Lenat (Eds.), Building expert systems. Reading, Massachusetts: Addison-Wesley, 1983.
- [8] L.Y., Tu, W. L. Hsu and S. H., Wu, "A Cognitive Student Model – An Ontological Approach," ICCE'02, Auckland, 2002.
- [9] Martin, D. J., "Concept Mapping as an aid to lesson planning: A longitudinal study.", Journal of Elementary Science Education, 6(2), 1994., pp. 11-30
- [10] Roger. T. Hartley., "Representation of Procedural Knowledge.", Department of Computer Science Technical Report, New Mexico State University, February, 1998.
- [11] Self, J. The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. International Journal of Artificial Intelligence in Education(10), 1999.
- [12] Sowa, J.F., "Knowledge Representation: Logical, Philosophical, and Computational Foundations." , Brooks Cole Publishing, Pacific Grove, CA, 1999
- [13] W.L Hsu, "Elementary school Math. tutoring agent", Proceedings of Agent Technology Workshop, 1997.
- [14] W.L Hsu, Y.S. Chen and Y.K. Wang, "Natural language agents – An agent society on the Internet" , Proceedings of PRIMA'99, 1999.
- [15] W. L. Hsu, Y. S. Chen and S. H. Wu, "Event Identification Based on the Information Map - INFOMAP," NLPKE'01, 2001, Tucson