ED 454 816 IR 020 795

TITLE	ICCE/ICCAI 2000 Full & Short Papers (Artificial Intelligence in Education).
PUB DATE	2000-11-00
NOTE	229p.; In: Learning Societies in the New Millennium: Creativity, Caring & Commitments. International Conference on Computers in Education/International Conference on Computer-Assisted Instruction (Taipei, Taiwan, November 21-24, 2000); see IR 020 792. Some figures contain very small and illegible type.
PUB TYPE	Collected Works - General (020) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; *Computer Uses in Education; *Educational Technology; Elementary Secondary Education; Higher Education; Instructional Design; Internet; Systems Development
IDENTIFIERS	Learning Environments

ABSTRACT

This document contains the full and short papers on artificial intelligence in education from ICCE/ICCAI 2000 (International Conference on Computers in Education/International Conference on Computer-Assisted Instruction) covering the following topics: a computational model for learners' motivation states in individualized tutoring system; a fuzzy-based assessment for Perl tutoring system; a genetic approach to parallel test construction; a learning environment for problem posing in simple arithmetical word problems; a method of creating counterexamples by using error-based simulation; a study of a networked constructive CAI (Computer Assisted Instruction) system using multiplication; adaptive programming language tutoring system on the World Wide Web; an agent-based intelligent tutoring system; an educational system that can visualize behavior of programs on the domain world; an environment for learning by design; applicability of an educational system assisting teachers of novice programming to actual education; a case-based evaluating assistant of novice programs; development and evaluation of a call system for supporting the writing of technical Japanese texts on the Web; development and evaluation of a mental model framing support ITS (Intelligent Tutoring System); development of intelligent learning support system with a large knowledge base; educational agents and the social construction of knowledge; facilitating examples understanding through explicit questioning; generating interactive explanations by using both images and texts for Micro World; intelligent interactive learning environment design issues; Internet video on demand system of classroom teaching cases-building 'Rhapsody': an intelligent media-oriented remote educational system for self-learning support; learning protocols for knowledge discovery--a collaborative data-mining approach to creative science education; monitoring and verifying mathematical proofs formulated in a restricted natural language; natural language-like knowledge representation for multimedia educational systems; the application of uncertainty reasoning for an ITS; the design and implementation of automatic exercise generator with tagged documents based on the intelligence of students (AEGIS); the design of CAI with thinking activity to progress constructive teaching; the estimation of music genres using neural network and its educational use; the externalization support system of self-explanation for the learning problem-solving process; traversing the case graphs--a computer model for developing case-based learning systems; use of abstraction levels in the design of intelligent tutoring systems; and using decision networks for adaptive tutoring. (MES)

ICCE/ICCAI 2000 Full & Short Papers (Artificial Intelligence in Education)

ICCE / ICCAI 2000

Learning Societies in the New Millennium : Creativity, Caring & Commitments

HOT NEWS

Background
Committees
Call for Papers
Deadlines
Registration
Program
Paper Acceptance
Reviewers
Venue & Travel Info.
Contact Info.
Host & Sponsors
Related Websites
Discussion Board
Proceedings

HOME

Proceedings

Content

▶Full & Short Papers (Artifical Intelligence in Education)

▲ HOME

A Computational Model for Learner' s Motivation States in Individualized Tutoring System

Behrouz H. Far* and Anete H. Hashimoto**
** Faculty of Engineering, Saitama University*
*** Technical Development Department, IDO Corporation*
Shimo-okubo, Urawa 338-8570, Saitama, JAPAN
Tel: + - - - Fax: + - - -
E-mail: far@acm.org

A goal of the research is to develop an intelligent tutoring system (ITS) that adapts the delivery of instruction according to the learner' s needs by taking into account learner' s motivation states. Long-term and short-term parameters involved in the learning process are identified. We have found that learner' s motivation has strong influence on the learning achievement. A computational model to represent learner's motivational states, using Bayesian network, is proposed. This model is further used to plan the individualized tutoring actions. This probabilistic model is the key to represent both learner 's knowledge and motivational states.

Keywords: **ITS, Student Modeling, Motivation, Bayesian Network**

1 Introduction

When designing an ITS system, usually, the first consideration is the teaching side, that is, deciding what to teach, what teaching strategies to apply, and what sequence of instruction to follow to facilitate learning. Although all these tasks are of unquestionable importance, to whom to teach, that is, the learning side should not be ignored. Teaching involves knowing what the learner wants or needs to study and planning the teaching material that leads to the desired learning outcome. However, since learners come with different background knowledge and needs, planning the individualized tutoring is not a trivial task.

Both background knowledge and motivational states of the learner have strong influence on the learning outcomes. Educational psychologists have revealed that human's motivational states are the driving forces for learning. In other words, no matter how attractive the lecture is, the learner will not benefit from it if he/she does not have the willing to engage in the learning process. But since bandwidth between the teacher and learner in a conventional classroom environment is relatively unlimited, human teachers may have a chance to bring the unmotivated learner back to the class. In the virtual classroom, the virtual tutor must be equipped with a mechanism to increase learner's knowledge via diagnosis of learner's motivational states and plan the tutoring while keeping learner motivated. We will see that, although motivation cannot be transferred from person to person, there are some principles explaining the increase (or decrease) of motivation.

A goal of this research is to develop a framework of an intelligent tutoring system (ITS) that adjusts instructions to the individual learner's needs by taking into account the motivational states of the learner [1]. A key task is increasing the bandwidth between learner and the ITS system. In order to increase the bandwidth, one must find out the hidden relationship in the learner's behavior and observed learning outcomes. Usually, learning outcome is associated to the learner's knowledge level, only. In this research, we first observe what actions contribute to increase learner's motivation to engage in the tutoring and then to plan a course of actions.

2 The Nature of Human Learning

What makes the learning process easy for one and hard for others? Looking for the answer is the primary concern of educational psychologists. In this section, we look further into the parameters influencing the human learning process.

2.1 Human learning parameters

In educational psychology individual' s learning aptitude difference is explained in terms of several external and internal causes [2-5]. The external sources are usually associated to causes beyond the learner' s control like the type of media or the learning environment that affect the quality of learning outcome. On the other hand, internal sources are associated with the learner' s own parameters like abilities and motivations. In this work, we focus on the internal causes.

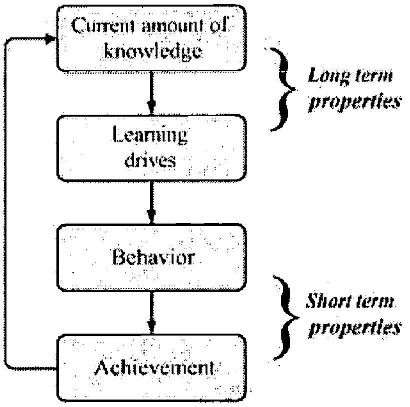


Figure 1 Human learning parameters

Figure 1, shows the set of parameters that influence the learning process. Each block in the model represents the set of parameters that describe the learning process, and the arrows indicate the direction of the influence. The first two parameters depict the learner' s intrinsic characteristics. It comprises the learner' s current amount of knowledge and aspects describing the learner' s unconscious learning drives, like the motivation to learn. The learner' s characteristics, in turn, are relevant to his/her behavior. It comprises the conscious learning drives used to measure how much effort his is putting to learn the new material. As the result of the learner' s behavior, his/her achievement can be measured by the learning outcomes. The achieved learning outcome, in turn, is fed back to the current amount of knowledge. In Section 3 we specify parameters compris ing each block.

Among the subject' s characteristics, one parameter that receives special attention ineducational psychology is the motivation that drives learning. Motivation can be classified in two types: extrinsic and intrinsic [4]. Extrinsic motivators comprise external driving forces like studying to pass an exam or to receive a reward. The intrinsic motivators, on the other hand, are internal forces inherent to the individual like the interest in the subject matter or the desire to be successful. The ideal is that both motivators influence learners, but the reality is different. Usually extrinsic motivators, like grades and prizes, become the objective in the classroom. Unfortunately, extrinsic motivators tend to have a short-term effect and affect the learning activity [5,4]. The intrinsic motivators are the parameters that generate learning results in long-term perspectives. The favor to intrinsic motivators can be observed in a study conducted in [6]. The explanation found is consistent with what is known about the relationship between extrinsic motivations (such as grades) and intrinsic motivation (such as challenging tasks): extrinsic motivators tend to inhibit intrinsic motivators. That is, if learners were given the choice, they would rather choose easier exams in order to get high grades than selecting more challenging tasks.

Based on this argument, the proposed tutoring system emphasizes learner' s intrinsic characteristics like abilities, progress, and confidence. It does not mean that extrinsic motivators are useless (test grades are not excluded in our system). Rather, the ideal is to balance both kinds of motivating drives. In the next subsection, the theories and principles that support our idea are explained.

2.2 The motivational and learning principles

We think that learning occurs only if the learner is motivated to learn. This desire to learn, whether intrinsic or extrinsic, is the driving force of how much effort the learner is willing to put in order to learn (see Figure 1). These efforts will be measured taking into account the learner' s observable behaviors such as the time spent to read a lesson or the frequency of visiting the same lesson to study. Herewith, we define the intrinsic characteristics that later will serve as the backbone of the student model.

1. ***Motivation:*** Motivational state is the force that drives the learner to engage in an activity because of a feeling of need or desire. Though motivation cannot be transferred, it may increase (or decrease)

depending on the situation that the learner is faced. One of the situations in which changes in motivational states may be observed is when the learner is presented tasks that fall in a range of challenge such that success is perceived but not certain [7]. Besides the perceived probability of success, others works [2,3] suggest also that the value of obtaining goal and acknowledge of progress are factors affecting motivation.

2. ***Learning:*** Learning is the ultimately desired change in behavior and knowledge to be achieved by the learner. Because of different background, motivational states and goals, learning results in different acquisition rate and outcomes. With regard to the factors influencing learning, readiness to understand the instruction is an essential requirement. Prerequisite knowledge is suggested as a measure of readiness. Anxiety and uncertainty of achieving goal have negative influences on learning.
3. ***Interest and progress:*** The acquisition of an ability or skill is a potential activator of interest since people tend to repeat things in which they are successful [4]. That is, when learners obtain evidence of their learning progress, not only interest tends to increase but also performance will be superior to what it would have been without such acknowledgment. Progress, may be thought of as the sum of learning achievements.
4. ***Retention:*** Retention is a measure of how well learners remember already acquired facts. The longer the time delay, the lower the retention factor. While time delay decreases retention, rehearsal strengths the ability to recall old information.
5. ***Ability degree:*** The learner' s ability degree is a measure of preparedness to learn academic material [3]. We define it as directly dependent on readiness, expertise level, and complexity of the topic. Expertise level, in turn, is measured by the amount of knowledge the learner has accumulated.
6. ***Attention:*** By attention we mean a measure of how the learner is directing his/her mind to the given task. We define it as the result of the positive influence of motivation and ability degree and the negative influence of distraction due to complexity of topic.
7. ***Effort:*** The effort tells us how the learner is behaving in order to achieve learning goals. Since it is not possible to observe it directly, we measure it by the frequency of dedicating to the study (frequency of use), the time delay between studies, the amount of time engaged in reading (time for reading), whether the learner performs the tasks (practice), and whether non-mastered topics are rehearsed (rehearsal).

It is obvious that intrinsic motivators are difficult to measure. Choosing challenging tasks neither brings immediate results nor it is easily measured. Marks and points, on the other hand, are concrete measures, easily interpreted and cause immediate satisfaction. The first task is to use intrinsic motivators in the student model such that they bring immediate and measurable results. The model presented in the next section covers this.

3 Student Modeling Task

In this section, we present the student model, using Bayesian network, based on the parameters mentioned in Section 2. The student model is divided in two parts: the motivational mo del and the knowledge model. The motivational model is generic, domain independent and applies to all learners. The knowledge model, on the other hand, is domain specific. The subject matter chosen for knowledge model is the concepts of the C programming language.

3.1 Modeling learner's motivational states

Tutoring based on the learner's motivation requires a mechanism to diagnose motivational states. Here, we take an approach that complements the limitation of existing proposals, such as [8]. However, it may introduce a new burden in creating motivational diagnosis. It is due to the modeling process and the task of estimating the probabilities for all variables in the network. On the other hand, the advantage is that it eliminates the learner' s burden because the diagnosis is running in background mode while the learner is using the system.

Building a student model based on Bayesian network requires two distinct tasks: the qualitative part that concerns the modeling of relevant variables involved in the domain, and the quantitative part that deals with the probabilities. As we are interested in representing the student motivational model, the qualitative modeling is concerned with the problems of identifying what information about the learner will be modeled and how that information will be modeled. In the quantitative modeling, we are concerned with the problem of specifying how the probabilities will be computed.

3.1.1 Qualitative analysis: encoding of dependence

The difficult part in the qualitative analysis is to find out how the variables influence each other. Our starting point was the learning parameters described in Figure 1: knowledge states, learning drives, learner's behavior, and learning achievements. These rough sets were further expanded based on the learning and motivational principles explained in Section 2. The refinement is done top-down: start from the first parameter down to the last one. The result is depicted in a network of Figure 2. The nodes in the network are divided into two types: directly observable nodes denoted by dashed-lines, and unobservable ones represented by solid-lines. The graph encodes the causal dependency among the motivational aspects relevant in the process. The common positioning of the variables is from cause to effect. An arrow from A to B is read as "A influences (or affects) B". For example, readiness is a factor that influences (or affects) motivation; ability influences attention.

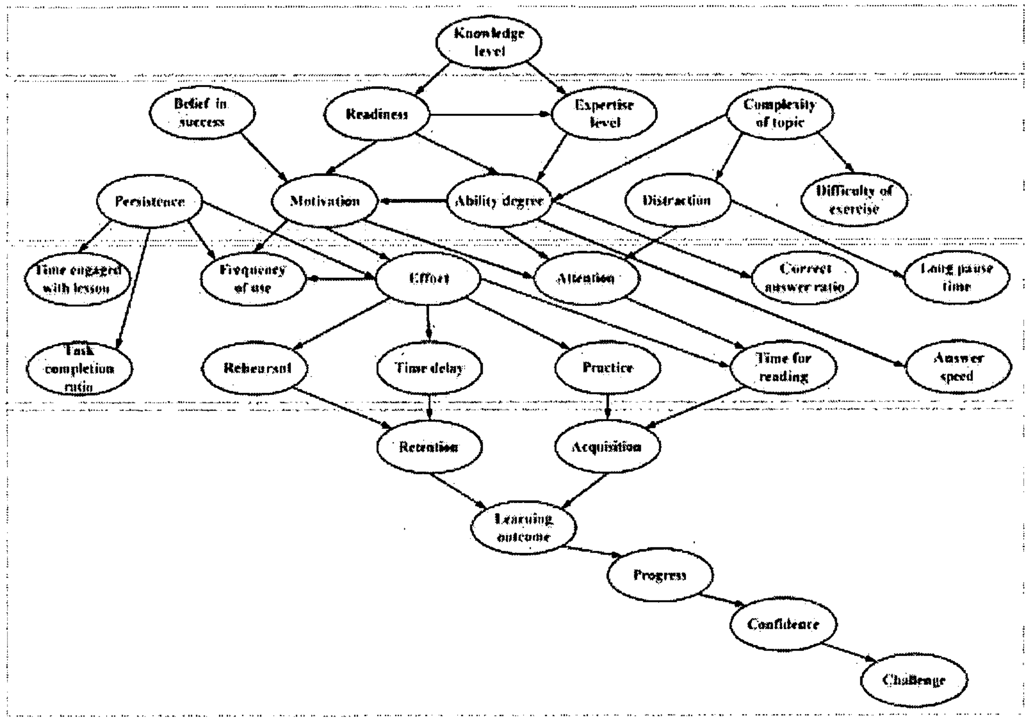


Figure 2 Student' s motivational states model

3.1.2 Quantitative analysis: expressing in numbers

The nodes probabilities may come from two different sources: probabilities set by experts and probabilities coming from repetitive calibration. It is worth mentioning that obtaining exact numbers is not really crucial since we are interested in the changes between the parameters rather than the values. In many cases, the advantages of Bayesian networks outweigh the load of eliciting the numbers. For example, locally encoding of information is an important aspect. Deleting or adding new information does not require the whole network be revised.

Initially, the probabilities in the student motivational model are rough estimations. The principles behind learning and motivation were translated to sentences like:

> *There is a high probability that motivated learners (motivation) works harder (effort)*

or

> *There is a low probability that the learner is persistent (persistence) if the task completion ratio is low (task completion ratio).*

We repeated this example for all variables in the network. Next, the qualitative terms like high and low are expressed in numbers. Finally, using a Bayesian network editor that we have built, those values are tested with repetitive calibration.

3.2 Modeling learner's knowledge states

Now, the qualitative and quantitative analyses for the student knowledge model are discussed.

3.2.1 Qualitative modeling: semantic of the network

The network depicted in Figure 3 represents the Bayesian network for the student knowledge models. Again there are two kinds of nodes: knowledge units and test nodes. Knowledge units represent relevant concepts comprising the domain to be taught. Test nodes represent problems that serve to verify the understanding level of each knowledge unit.

In order to build the Bayesian network of Figure 3 we start by eliciting the knowledge units comprising the domain, represented by a solid node, and ranking them according to the difficulty/complexity of the unit. For example, if a unit does not require mastery of other units, then it is a candidate to be in the easicst level. Another unit that requires just mastery of the easiest level unit is the candidate to be the second easiest level, and so on.

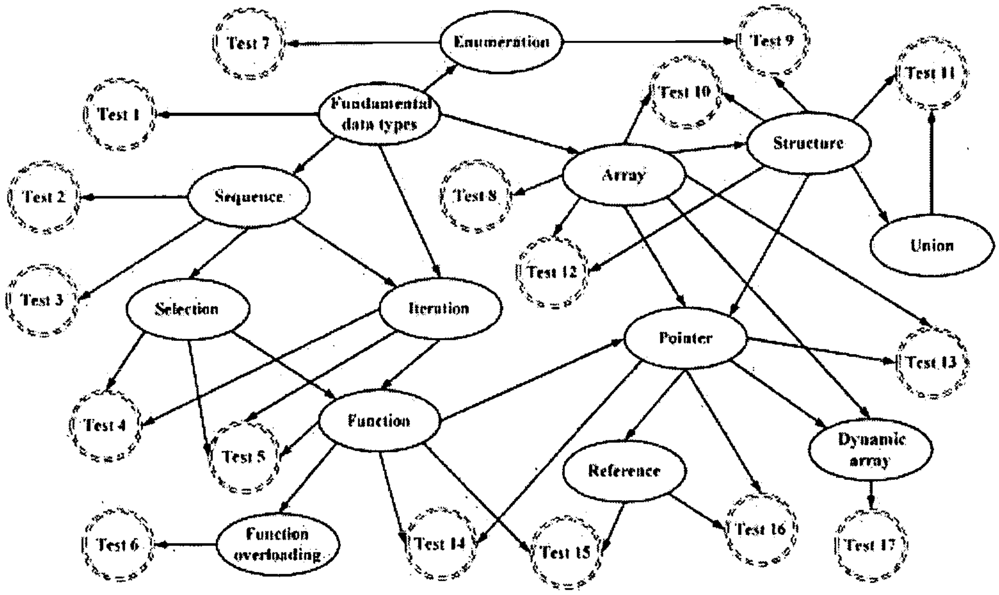


Figure 3 Student' s domain knowledge model

Besides this classification, we have to find out how to represent those knowledge units in the network. Usually, Bayesian network is modeled based on cause-effect relationship. Since this is not easily perceived in our case, we extracted the factors that describe the units such as description, usage, and limitation. This analysis helps us to understand the hidden relationship between apparently unrelated units. We observed that some units fulfill the limitation of other units: for example, array and structure. In other cases, units present similarity in usage: for example, pointers and references. A link is added between those units in order to depict the fact that knowing one unit makes the probability of understanding the related unit more likely. Depending on the relevance of the knowledge unit within the domain, we can add more test nodes to the unit. In this example, since "Function", "Array", and "Pointer" play an important role within the domain of programming language, we can elaborate several test nodes covering those concepts.

The line of reasoning is as follows: if the learner solves corrcctly a problcm associatcd to a knowledge unit, then the probability of knowing that unit increases. A link is added between knowledge unit and test node if it is required to know the unit in order to solve the problem. We add a link between knowledge units if exists a relevance relationship between them. Of course there is a tradeoff between compactness and preciseness. For example, learning about the "Fundamental data types" is essential for all remaining knowledge units, which we would have to add a link between that node and all other units. But, considering the precedence condition of the concepts, we were able to limit the links only to the directly relevant knowledge, such as

"Sequence expressions" and "Enumeration".

3.2.2 Quantitative modeling: dealing with probabilities

For each variable in the network, there is a conditional probability table (CPT) with respect to its parent nodes. For example, for the node "Test14", we have a CPT associating the "Test14" node to its parent nodes "Pointer" and "Function" knowledge units. That is, in order to answer the test correctly, the learner must understand both pointer and function. If the learner answers it correctly, it is inferred that he understands both units. If, however, the test was answered incorrectly, then, in the absence of other evidences, the associated units are considered not mastered yet. Suppose that we have already collected evidences that the learner knows about functions. In this case, rather than inferring both units as not mastered, it is more likely that only pointers have not been mastered yet. After including all the evidences and propagating the probabilities through the adjacent nodes, the network reaches an equilibrium state and we obtain the probability of the learner being in mastered level in each knowledge unit.

3.3 How the model works

Since learning occurs only if the learner has the desire or motivation to learn, the task we are concerned with is to keep the learner motivated to complete the tutoring. Consequently, the problems are: how to assess learner's motivational states and how to proceed tutoring in order to keep (or increase) motivation.

Let' s consider the following situations: a novice learner who spends a long time without accessing the tutorial comes back to continue the lessons. Because of the long time delay between lessons, it is likely that he/she forgot something about the past lessons and needs a review. But, at the same time, the novice learner would probably become more motivated if he/she made some progress. In another case, an intermediate learner is apparently loosing motivation because of repetitive unsuccessful response to exercises.

In each case, the system can infer different treatments for each learner needs and set appropriate courses of actions. Therefore, the model will be used to perform the following tasks:

1. ***Monitoring:*** observe the learner in a sequence of interactions to adjust prior beliefs about learner's knowledge and learning drives.
2. ***Inference:*** because only a limited number of events are observable, infer what these directly observable actions tell about the other parameters.
3. ***Prediction:*** predict learner's knowledge and motivational states in the next interaction given the information currently available.

To depict the evolution of the tutoring, we represent the learning cycle as a dynamic process, as shown in Figure 4. At each interaction, the learning achievement increases (or decreases) the amount of knowledge the learner possesses in the next interaction, which indirectly increases the motivational states. Including temporal characteristic is important because if episodic interactions were considered, the learner's motivation, for example, would be inferred based on the current situation without taking into account past failure or success in outcome.

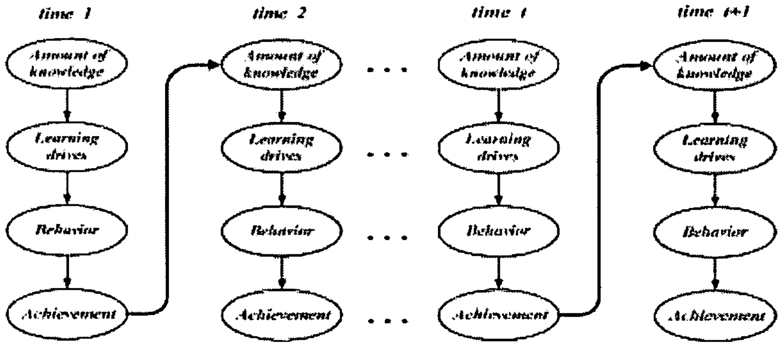


Figure 4 The dynamic process of tutoring

Dynamic Bayesian network [9] provides a mechanism to foresee the probability of interest in the next state with regard to the current beliefs. That mechanism is called probabilistic projection and can be performed by a three step updating cycle called *roll-up*, *estimation*, and *prediction* phases, as suggested in the dynamic model of Figure 4. Keeping at the most two time slices are sufficient to perform the inferential cycle. Figure 5 depicts the steps for updating a dynamic Bayesian network and below, a brief description of each step.

1. ***Prediction:*** suppose the network in Figure 5(a). Assuming that all the values have been calculated in time slice *t-1*, i.e., $Bel(X_{t-1})$, this probability should be incorporated in the next time slice by estimating $\hat{Bel}(X_t)$. In this step, the predicted probability distribution expected given the evidences known at time slice *t-1* is calculated.

   $$\hat{Bel}(X_t)=\sum_{X_{t-1}} P(X_t \,|\, X_{t-1}, E_{t-1}) Bel(X_{t-1})$$

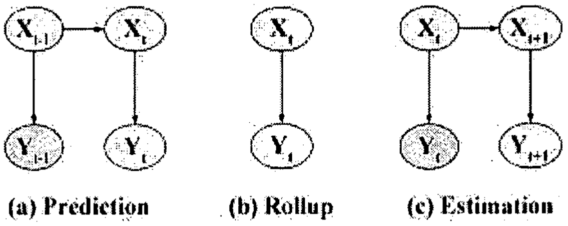
   Where E_{t-1} is all the evidence at time slice *t-1*; *P* is the probability and "^" denotes an estimation.

2. ***Roll-up:*** the roll-up is the process of removing the network on time slice *t-1* and assigning a prior probability table for the state variables at time *t*, which is the $\hat{Bel}(X_t)$ (Figure5(b)).

3. ***Estimation:*** now, using the standard probabilistic network updating, the probability distribution over the current time slice *t+1* is found and the steps for the next cycle can be repeated (Figure 5(c)).

   $$Bel(X_t)=\alpha P(E_t \,|\, X_t)\hat{Bel}(X_t)$$

   Where α is normalization constant.



(a) Prediction (b) Rollup (c) Estimation

Figure 5 The updating cycle

4 Planning Actions

Through educators frequently rely on experience and common sense to prepare a curriculum plan, there are some theories helping educators to organize the lessons and offer learners an easier way to assimilate new concepts. Following we describe the theories about the sequence of instruction and motivation strategies that we adopt in our project.

1. ***Theories about sequence of instruction:*** helps the instructor to select the next instruction when there are conflicting candidates.
   - *Simple-to-complex theory:* given two concepts A and B, if A is simpler than B, then choose A as the next candidate.
   - *Laws of organization:* if A and B are similar concepts and the learner knows A, then the probability of understanding B becomes higher.

2. ***Motivational Strategies:*** dictates the teaching strategies to apply given the learner' s motivational states and experience. Following, some examples of strategies:
   - Whenever a less motivated or confident learner does a task well, present similar tasks that are likely to be successful in order to increase his/her confidence and motivation.
   - If the learner presents high persistence or motivation, let him/her try again the task rather than promptly presenting the correct answer.
   - Show the learner his/her motivational and knowledge states in order to stimulate self-monitoring.

The way of actually planning actions and delivering instructions will be treated in the authors' another paper.

5 Discussions

In order to model motivational states, we need a formalism that simultaneously offers mechanisms to: (a) model the causality explaining the principles involved in the learning process, (b) reason under the uncertainties inherent to the effects of the process, and (c) represent the temporal changes observed due to learning. The framework we proposed in this paper can cover all these factors. It is suitable for handling problems that can be modeled according to certain relevance conditions. In our case, the learning principles are the conditions that enable us to model the learning parameters. Although it is impossible to identify and to model all the parameters involved in learning, but Bayesian network' s reasoning mechanism is capable of dealing with incomplete as well as limited amount of data. Moreover, the ability to reason about the problem without necessarily observing all the variables involved constitutes another advantage.

With respect to the computational advantages of Bayesian networks, the structure of the network allows the locally encoding of information rather than globally. That is, once the network is consistently built, each node interacts only with the directly connected nodes [9]. The gain with this property is that addition or deletion of nodes can be done locally without revising the whole network. Additionally, the computation can be performed with regard to the adjacent parameters only.

6 Conclusions

We proposed a framework for an intelligent tutoring system that adapts instruction based on the learners' needs by taking into account learner' s motivation states. Our main claim is that learner' s needs do not refer only to knowledge needs, but also to motivational needs. The bottleneck, however, is the limited bandwidth between human and machine. The first thought is, then, to direct the research in the latest technology in human-machine interface, like natural language understanding or eyes movement reading methods in order to increase the narrow communication channel. But is it really only the technological bottleneck that hinders the communication between human and machine? If so, then why human teachers have troubles with their pupils? This was the question that arose during the development of this work.

The bandwidth problem limits the communication channel, but providing the system all available information does not guarantee perfect communication. We realized that the cognitive and educational aspects come first. What is behind the human learning process? Why some students learn faster than the others? These questions, then, became the priorities in our work. After eliciting the parameters involved in learning, we faced with the problem of how to make best use of the limited source of information the system was capable of computing. The computational formalism that fulfilled our needs was Bayesian network. This probabilistic method not only reasons under limited source of information but also infers about yet unobserved variables. In this way, we could virtually increase the communication channel.

Planning based on motivational strategies is still in an immature stage and a subject of our forthcoming paper. For clarity, the student motivational model possesses a large number of parameters, which can be omitted according to the intended use. Since the model is modularized and domain-independent, it is also possible to reuse it to teach different domain application. The set of rules to execute motivational strategies we have defined is simply an adaptation of the motivational principles. Improvements are still needed in this direction.

References

[1] Hashimoto, A.H., Hajji, H., and Far, B.H., "A Probabilistic Approach for Student Modeling Task," 13th Annual Conference of Japanese Society for Artificial Intelligence, JSAI' 99, pp. 68-69, (1999).

[2] Eftekhar, N., Strong, D.R., and Hawaleshka, O., "Towards Dynamic Modeling of a Teaching-Learning System: The Unified Theory," International Journal of Engineering Education, vol. 12, no. 6, (1997). Also available at: http://www.ijee.dit.ie/articles/999981/article.htm (accessed on August 7, 2000).

[3] McIlrath, D.A., and Huitt, W.G., "The Teaching-Learning Process: A Discussion of Models," http://www.valdosta.edu/~whuitt/psy702/files/modeltch.html (accessed on August 7, 2000).

[4] Spencer, R., Ludington, J. and Graham, S., "Motivation and Learning: A Teacher' s Guide to Building Excitement for Learning and Igniting the Drive for Quality," Peak Learning Systems, (1999). Also partially available at: http://www.peaklearn.com/chap1.html (accessed on August 7, 2000).

[5] Lumsden, L.S., "Student Motivation to Learn," ERIC Digest 92, (1994).

[6] Litchfield, B.C., and Newman, E.J., "Differences In Student And Teacher Perceptions Of Motivating Factors In The Classroom Environment," National FORUM of Applied Educational Research, vol. 12E, no. 3, (1999). Also available at: http://www.nationalforum.com/16LITCHF.htm (accessed on August 7, 2000).

[7] Knight, J., "Educational Foundation," http://ag.arizona.edu/aed/aed695a/educatio.htm (accessed on August 7, 2000).

[8] de Vicent, A., and Pain, H., "Towards Motivating Instructional Systems," In G. Cumming, T. Okamoto, and L. Gomes, editors, Proceeding of the International Conference on Computers in Education (ICCE'99), vol. 1, pp. 744-751, IOS Press, (1999).

[9] Stuart, R., and Norvig, P., "Artificial Intelligence: A Modern Approach," Prentice Hall, (1995).

[10] Hirashima, T., "Student Modeling and Adaptive Interaction," Journal of Japanese Society for Artificial Intelligence, Special Issue on Student Modeling and Adaptive Interaction, vol. 14, no. 1, (1999).

A Fuzzy-Based Assessment for Perl Tutoring System[1]

Tang Ya*, Keith C.C. Chan*, Albert Wu and Pinata Winoto*****
*Department of Computing
Hong Kong Polytechnic University, Hong Kong
Email:{ csytang, cskcchan } @comp.polyu.edu.hk
**Faculty of Education
Chinese University of Hong Kong, Hong Kong
Email: albertwu@cuhk.edu.hk
***School of Business and Management
Hong Kong University of Science and Technology, Hong Kong
Email: pwinoto@ust.hk

In this paper, we present a fuzzy-based assessment for Perl Tutoring system. The Perl Tutor is implemented in a multi-domain framework so that it can teach target domain knowledge by giving supporting knowledge to reinforce the learning. In order to assess supporting knowledge, an assessment is performed *before* the tutoring begins. Its main purpose is to test student's previous declarative knowledge of computer programming. At the end of it, a directed tutoring graph will be generated to optimize the tutoring process.

Keywords: **fuzzy rule, assessment, student modeling, multi-domain tutoring**

1 Introduction

There exist many works on optimized assessment process concerned with the efficiency of testing and its completeness. *Granularity*, *prerequisite relationships*, *Bayesian propagation* and *neighborhood of knowledge states* are some of the successful attempts employed to increase the efficiency of testing [2,5,6,13,17]. Yet, even though they could increase the efficiency significantly, they still have too many burdens given the large knowledge spaces. Fortunately, not all the student models need to be precise to be useful [10]. To ease the burden to student modeling, a fuzzy approach has been used and has so far worked quite well [3,10,11].

The purpose of this paper is to present the fuzzy approach in the assessment of student's knowledge in the Perl Tutoring System [16], which teach programming language (Perl) by reinforcement from other supporting languages (C++ and/or Java). For the effectiveness of reinforcement, the system should quickly evaluate the student's knowledge of supporting languages. But the assessment needs not to be in high precision. Other works related to student modeling almost put their emphasis on the adaptive assessment *during* tutoring [14,15,17]. Yet due to the nature of our Perl tutor, we apply an assessment module *before* tutoring begins and it consists of two parts: questionnaire and testing. During the questionnaire part, students are asked to self-assess their knowledge by filling out a form provided by the system. In order to evaluate their statements, a testing part is given based on those statements. At the end of the assessment, the tutor will have a general picture of students' prior knowledge of supporting languages: with which part they are familiar etc. Since the goal of the assessment is only to get a rough knowledge states for supporting purpose, it should not take too long to complete. Thus, a coarse granularity with imprecise mastery level is appropriate.

In the next part of this paper, we briefly discuss the Perl tutoring system followed by the fuzzy logic. Then we will describe the questionnaire part and the testing part and end with discussion.

[1] The work related to this paper is funded under the Hong Kong Polytechnic University research grant No. PolyU5072/98E.

2 Overview of the Perl Tutoring System

Figure 1 illustrates the directed tutoring graph in the system [16]. The three pieces of knowledge items presented to students are: data type, logical operators and control structures. In the figure,

- Each vertex represents a sub-domain;
- Each pair of the sub-domain may be connected with a unidirectional or bi-directional arc.
- Each arc represents the relationship between two sub-domains.

Moreover, each sub-domain may consist of several vertices, which are the sub-sub-knowledge items of their parent domain. For example, under ' data type', we also have ' integer', ' float', ' boolean' etc.

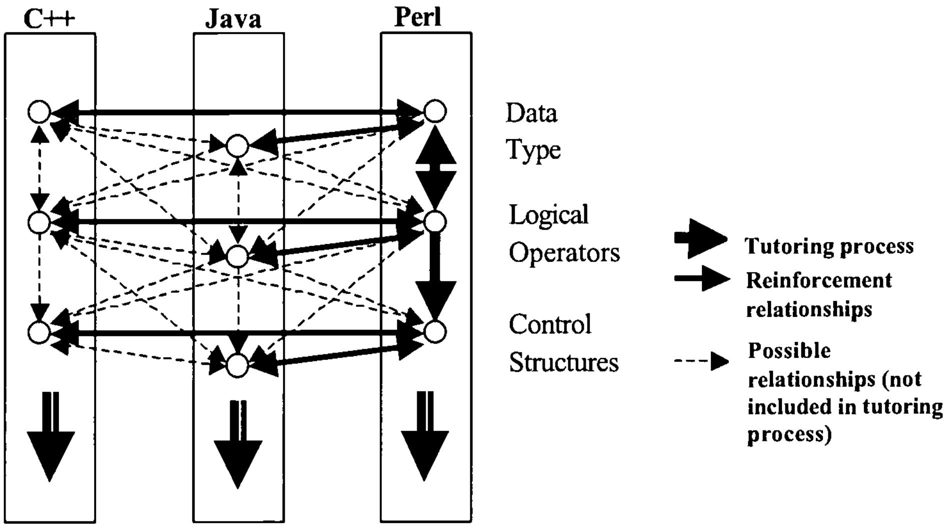

C++
Java
Perl
Data Type
Logical Operators
Control Structures
Tutoring process
Reinforcement relationships
Possible relationships (not included in tutoring process)

Figure 1 A Digraph of the Perl Tutoring System

C++ [1] and Java share many similarities with Perl, although they, of course, have their own features. See Table 1 for a comparison.

CDR terms (General)			Knowledge piece in PERL	Knowledge piece in C++	Knowledge piece in Java
Operators	Numeric operators		+,-,*,/, %, **	+,-,*,/, %,	+,-,*,/, %,
	Relational operators		<,<=,>,>=,<=>(**for numeric**) lt,le,gt,ge,cmp(**for string**)	<,<=,>,>=	<,<=,>,>=
	Equality operators		=,!=(**for numeric)** eq,ne(**for string)**	==,!=	==,!=
	Logical operators(binary)		&&,||	&&,||	&&,||
	Logical operators(unary)		!,	!,-	!,-
	Bit manipulation operators		&, |, ^, ~	&, |, ^, ~	&, |, ^, ~
	Bit shift operators		<<, >>	<<, >>	<<, >>, >>>
	Auto-increment & auto-decrement operators		++,--	++,--	++,--
	Speical operators	Conditional operators	?:	?:	?:
		Other operators	•,x(string operators) , >,\(**dereference/reference operator**	Sizeof ,	Instanceof
	Multiple operators			+=, -=, *=, /=, %=, &=, |=, <<=, >>=, ++=, --=	+=, -=, *=, /=, %=, &=, |=, <<=, >>=, >>>=, ++=, --=
Control structures			If,if/else, unless/else, While, do/while, for, continue, goto Notes:labled loops can be used within for, while, or do	If, if/else, while, do/while, for, continue, goto, switch, break	if, if/else, while, do/while, for, continue, break, switch Notes:labled loops can be used within for, while, or do
	Special structures		**Foreach** **Redo**	Exit	

Table 1 Similarities and differences in C++, Java and Perl

CDR represents ' cross-domain reference' which serves as a dictionary for the domains. It is composed of basic terms used across the computer language regardless of which language is being referred. If the student has learned computer language before, he will develop a clear picture of the terms or concepts used, which serves as a guide for the learning of Perl. Besides, he will also integrate his former learning into his current. Through this knowledge transfers, the time spent on learning Perl will be greatly reduced [8].

Before tutoring begins, a weight is assigned to every direction of arc that represents the easiness of the acquisition of one sub-domain (target) after acquiring another (source). Since different students have different knowledge levels, the weight assigned to the same arc may not be the same. Thus, the weight across domain is jointly determined by the student model and the characteristics of knowledge (for detailed explanation, refer to [16]), i.e.,

$$w_{ij} = \mathbf{f}\,(\mathbf{d}_{ij}, \mathbf{m}_{ij})$$

Where, w_{ij} is the weight of arc from i to j.

$\mathbf{f}: R^n \times R^m \rightarrow R_+$ is a non-decreasing function.

$\mathbf{d}_{ij}$ is an n-dimensional vector representing the similarity of i and j. $\mathbf{m}_{ij}$ is an m-dimensional vector representing the student model, i.e., the student' s knowledge level of i.
The dimension of $\mathbf{d}_{ij}$ and $\mathbf{m}_{ij}$ depends on the number of attributes considered. Moreover, the value of $\mathbf{d}_{ij}$ is predetermined and the value of $\mathbf{m}_{ij}$ is determined based on the student model. Thus, the system would carry an assessment module to test the knowledge of a student towards a specific supporting domain knowledge *before* tutoring begins. In this paper, we focus on the determination of $\mathbf{m}_{ij}$.

3 The Assessment Model—A Fuzzy Approach

Since the main purpose of the model is to test student' s overall abilities, it is not necessary for us to gain a very accurate picture of it (although it helps). And somehow we also cannot gain a clear picture of student history. Thus, we choose a fuzzy approach in analyzing the student' s performance, and we believe that the imprecise assessment of the student' s prior knowledge level is adequate.

3.1 The ' neighborhood of knowledge states'

The *knowledge state* has been defined as the subset of knowledge items from a large item pool that can be mastered by students [4]. Remember that knowledge items in different domains are identified by their names, which in turn are determined by a cross-domain vocabulary. Besides, each item is characterized by its relationship with other items. The neighborhood of a knowledge state was defined by Falmagne and Doignon [7] as all other states within a distance of at most one. It has been utilized for adaptive assessment by Dowling *et. al.* [6]. In our system, we will not measure the exact distance within knowledge items, but we adopt it from another perspective. We define the neighbors of a knowledge item as the possible knowledge items which could be mastered in association with it. Let us have a look at an example.

Example 1.

1. ' <' ,' <=' represent ' less than' and ' less than or equal to' respectively, and they are relational operators.
2. ' >' ,' >=' represent ' greater than' and ' greater than or equal to' respectively.
3. ' ==' ,' ! =' represent ' equal to' and ' not equal to' respectively, and they are equality operators.
4. ' <' ,' <=' can be used for both numeric and strings.
5. ' >' ,' >=' can be used for both numeric and strings.
6. ' ==' ,' ! =' can be used for both numeric and strings.
7. Numeric is data type.
8. Strings are data type.
9. The relational and equality operators can be used for all data types, numbers, expressions or their combinations.

Let $M_S(X)$ denotes the student is sure to have mastered X. And $M_l(Y)$ denotes the student is likely to have mastered Y. Where X,Y are sets of knowledge items. Then,

$M_S(X)$ □ $M_l(Y)$ can be interpreted as "if the student is sure to have mastered X, then he/she is likely to have mastered Y."

Then we will have:

1. $M_S(1)$□ $M_l\{2,4,7,8\}$
2. $M_S\{3\}$□ $M_l\{6,7,8\}$
3. $M_S\{4,5,6\}$□ $M_S\{7,8\}$
4. $M_S\{9\}$□ $M_S\{1,2,3,4,5,6,7,8\}$

For example, if the student knows well how to make comparisons for numeric and strings, then we assume that he/she is sure to have mastered: what is numeric, what is a string and the usage of the operators. Although we cannot determine that whether he masters other data types or not (that is, he is likely to have mastered other data types such as float etc), we can assess student' s knowledge state without having to extensively test his abilities of each knowledge item he/she may have learned. Therefore, test items in our model may test knowledge items in a wider ranger than similar work by Collins *et. al.* [2].

3.2 Fuzzy Logic

To express precisely the notion "sure", "likely" or "unlikely", we adopt fuzzy set methods and therefore using fuzzy rule for the inferences. For example, we define

Answer = {True, False}. And A1, A2 $\subset$ Answer, thus

$$Ai = \mu_{Ai}(T)/True + \mu_{Ai}(F)/False$$

Confidence = {unlikely, likely, sure}. And B1, B2 $\subset$ Confidence, thus

$$Bi = \mu_{Bi}(u)/unlikely + \mu_{Bi}(l)/likely + \mu_{Bi}(s)/sure$$

Assume we have two rules: R1: A1 $\rightarrow$ B1 and R2: A2 $\rightarrow$ B2

Then, by Mamdani' s direct methods:

$$B' = A' \circ R$$

Where, $R = R1 \cup R2$

$$Ri = \begin{pmatrix} \mu_{Ri}(T,u) & \mu_{Ri}(T,l) & \mu_{Ri}(T,s) \\ \mu_{Ri}(F,u) & \mu_{Ri}(F,l) & \mu_{Ri}(F,s) \end{pmatrix} \quad \text{and} \quad \mu_{Ri}(x,y) = \mu_{Ai}(x) \wedge \mu_{Bi}(y)$$

Note here that all operators used, such as: +, /, $\subset$, $\wedge$, $\cup$, and o, are defined in fuzzy domain.[2]

To illustrate it, let us assume that A1 is "doing well in bit shift operator", A2 is "doing bad in bit shift operator", B1 is "understand bit manipulation if doing well in bit shift operator", and B2 is "understand bit manipulation if doing bad in bit shift operator". Then, we can assign values such as:

A1 = 1.0/T A2 = 1.0/F

B1 = 0.5/l + 0.5/s B2 = 1.0/u + 0.1/l

And satisfied: R1: A1 $\rightarrow$ B1 and R2: A2 $\rightarrow$ B2. Thus,

$$R1 = \begin{array}{ll|ccc} & & \mu_{B1}(u) & \mu_{B1}(l) & \mu_{B1}(s) \\ & & 0 & 0.5 & 0.5 \\ \hline \mu_{A1}(T) & 1.0 & 0 & 0.5 & 0.5 \\ \mu_{A1}(F) & 0 & 0 & 0 & 0 \end{array}$$

$$R2 = \begin{array}{ll|ccc} & & \mu_{B2}(u) & \mu_{B2}(l) & \mu_{B2}(s) \\ & & 1.0 & 0.1 & 0 \\ \hline \mu_{A2}(T) & 0 & 0 & 0 & 0 \\ \mu_{A2}(F) & 1.0 & 1.0 & 0.1 & 0 \end{array}$$

[2] Many books [18,19,20] in fuzzy set theory provide good explanations on these operators. We are not going to explain it further in this paper due to limited space.

With two rules, the fuzzy relation Ri is made from the implication Ai → Bi (in this case, i=1,2). The compiled fuzzy relation R is given as Mamdani' s method:

$$R = R1 \cup R2, \text{ computed as:}$$

$$R = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 1.0 & 0.1 & 0 \end{bmatrix}$$

Now, assume after a series of testing, a student performance show A' = 0.9/T + 0.2/F in doing bit shift operator. Then, we can calculate his performance in bit manipulation as:

$$B' = A' \circ R$$

$$= \begin{array}{cc} T & F \\ {[0.9} & {0.2]} \end{array} \quad o \quad \begin{array}{c|ccc} & u & l & s \\ \hline u & 0 & 0.5 & 0.5 \\ 1 & 1.0 & 0.1 & 0 \end{array}$$

$$= [(0.9 \wedge 0) \vee (0.2 \wedge 1.0), \\ (0.9 \wedge 0.5) \vee (0.2 \wedge 0.1), \\ (0.9 \wedge 0.5) \vee (0.2 \wedge 0)]$$

$$= [0.2 \quad 0.5 \quad 0.5]$$

$$B' = A' \circ R = 0.2/u + 0.5/l + 0.5/s$$

Which shows 0.5 likely to understand, 0.5 surely to understand and only 0.2 unlikely to understand bit manipulation.

4 Questionnaire and Testing

The questionnaire part consists of a series of knowledge items to be checked by students. The knowledge items are grouped into several groups based on the their similarities and difficulties. Then, students are asked to fill the form about their mastery level in each group. Five grades are provided for each answer, i.e., very familiar, familiar, moderately familiar, not familiar, and never heard. After students provided their answers, the system retrieves a series of testing questions based on the difficulty (upper limit) of students' answers, especially for the items marked ' moderately familiar'. But it does not mean that the presumably mastered items are not tested at all. Even the items marked 'very familiar' will be tested, but with a very low probability. Testing could be in the forms of short program lists or short questions, which are made as short, clear, and simple as possible. The reason is to avoid noise or errors which do not come from student knowledge iteself. In order to avoid ambiguity in judging knowledge level when the question is not answered well, every question only consists few higher level concepts to be handled.

Moreover, an average of membership value is used if the same item occurs in several questions. (We can use Bayesian update but with higher cost, i.e., to set all the conditional probability among every question).
For example, if from question 1, 2 and 3, a student performance on ' bit manipulation' shows

0.8/T + 0.2/F, 0.9/T + 0.3/F, and 1.0/T + 0.1/F respectively,

then the overall performance is, simply, the average, i.e., 0.9/T + 0.2/F.

If the question needed does not exist in the database, then a similar question is retrieved. The measure of similarity is based on the maximum number of high level concept appeared.

Prerequisite relationship

In addition to the neighborhood relationships, prerequisite relationships are also applied. The prerequisite relationship provides not only test item ordering criteria in a "strong" sense, but also in a "weak" sense. In ordinary prerequisite criteria P(A, B) denotes "A is prerequisite of B". In our extended criteria, we introduce A' as:

If A' is closely related to A and $M_S(A')$□ $M_S(A)$

then we have P' (A' , B), that is, A' is weakly prerequisite of B.

So, if students have mastered item A' , we have: they are sure to have mastered B without testing whether they have mastered item A or not. By doing this, we can largely tighten the testing items and thus save more time.

5 Discussion

To know student' s learning history and his knowledge level, we cannot ask them too detailed questions in order to gain a more full picture of their knowledge state (although it helps) since it will make student modeling itself a kind of a complex system. But we need them to aid in the assessment, so how much trust should we have in the student's own assessment? This is the question we need answer before we proceed. In our system, we will not generate the tutoring graph solely based on their answers. Our solution is to test by giving them several pre-stored test items: if they can write out the outcome correctly, we assume that he has mastered the knowledge pieces and rules needed for this program.

Thus, the assessment will proceed. Test items need not to be like traditional testing questions in classrooms. They can be mini-programs or short questions provided that they can be used as a guide to assess students' mastery level of declarative knowledge.

Furthermore, we also should consider the nature of the language. For example, If the student has studied both Prolog and Java before, considering the respective relationship of them with Perl, we will still use Java as supporting knowledge because it is closer to Perl. This factor is called Knowledge Relation (K-R), and it will be assigned to $\mathbf{d}_{ij}$.

At the end of the self-assessment section, a directed tutoring graph is generated. And student will be tutored based on it.

Currently, we are constructing the fuzzy rules which are applied for the assessment module, followed by the implementation and evaluation of it.

References

1. Capper, D.M. "Introducing C++ for Scientists, Engineers and Mathematicians", U.K., Springer-Verlag, (1994).
2. Collins, J.A., Greer, J.E. & Huang, S.X. "Adaptive assessment using granularity hierarchies and Bayesian nets", Proceedings of the 3rd International Conference on Intelligent Tutoring Systems (ITS' 96), Montreal, Canada, LNCS 1086, (1996).
3. Derry, S.J. & Hawkes, L.W. "Toward fuzzy diagnostic assessment of metacognitive knowledge and growth", Proceedings of the Annual Meeting of the American Educational Research Association, San Francisco, CA, (1992).
4. Doignon, J.-P. & Falmagne, J.-C. "Spaces for the assessment of knowledge", International Journal of Man-Machine Studies, Vol 23, pp.175-196, (1985).
5. Dowling, C.E. & Kaluscha, R. "Prerequisite relationships for the adaptive assessment of knowledge", Artificial Intelligence in Education, pp. 43-50, AACE: Charlottesville, VA, (1995)
6. Dowling, C.E., Hockemeyer,C., Ludwig, A.H. "Adpative assessment and training using the neighbourhood of knowledge state", Proceedings of the 3rd International Conference on Intelligent Tutoring Systems (ITS' 96), Montreal, Canada, LNCS 1086, (1996).
7. Falmagne,J.-C. & Doignon,J.-P. "A Markovian procedure for assessing the state of a systcm", Journal of Mathematical Psychology, Vol. 32, pp. 232-258, (1988).
8. Harvey, L. & Anderson, J.R.. "Transfer of declarative knowledge in complex information processing domains", http://act.psy.cmu.edu/ACT/papers/Harvey_Anderson95.html, (accessed on August 1, 2000).
9. Juliano,B.A. & Bandler,W. "Tracing Chains-of-Thought ", Heidelbert: Physica-Verlag, (1996).
10. Katz, S., Lesgold, A., Eggan, G., & Gordin, M. "Approaches to student modeling in the Sherlock tutors", Proceedings of the Third International Workshop on User Modeling, Germany, pp. 205-230, (1992).

11. Lesgold, M., Eggan, G., Katz, S. & Rao, G. "Possibilities for assessment using computer-based apprenticeship environments", Cognitive Approaches to Automated, Hillsdale, Lawrence Erlbaum Associates, (1991).
12. Matsubara, Y., Nagamachi, M. "Motivation system and human model for intelligent tutoring", Proceedings of the 3rd International Conference on Intelligent Tutoring Systems (ITS' 96), Montreal, Canada. LNCS 1086, (1996).
13. McCalla, G. I., Greer, J. E. "Granularity-based reasoning and belief revision in student models", Student Modelling: The Key to Individualized Knowledge-Based Instruction, pp. 39-62. Springer-Verlag, Berlin, (1994).
14. Nguyen-Xuan, A., Joly, F., Nicaud, J.F., Gelis, J.M. "Automatic diagnosis of the student' s knowledge state in the learning of algebraic problem solving", Proceedings of the Conference on Artificial Intelligence in Education, Japan, (1993).
15. Sime, J.-A. "Modeling a learner' s multiple models with Bayesian belief networks", Proceedings of the Conference on Artificial Intelligence in Education, Edinburgh, Scotland, pp. 426-432, (1993).
16. Tang, Tiffany Ya. & Wu, Albert. "The implementation of a multi-agent intelligent tutoring system for the learning of computer programming", To appear in the 16th IFIP World Computer Congress— International Conference on Educational Uses of Communication and Information Technologies (ICEUT 2000) Beijing, China, (2000).
17. Villano, M. "Probabilistic student models: Bayesian belief networks and knowledge space theory", Proceedings of the 2nd International Conference on Intelligent Tutoring Systems (ITS' 92), Montreal, Canada, pp. 491-498, (1992).
18. Zadeh, L.A "Fuzzy sets", Information and Control, Vol. 8, pp. 338-353, (1972).
19. Zadeh, L.A. "A fuzzy-set-theoretic interpretation of linguistic hedges", Journal of Cybernetics, Vol. 2, pp. 4-34, (1973).
20. Zhu, Q. & Lee, E.S. "Fuzzy and Evidence Reasoning", Physica-Verlag, Germany, (1995).

A Genetic Approach to Parallel Test Construction

Koun-Tem Sun
Institute of Computer Science and Information Education
National Tainan Teachers College
Tainan 700, Taiwan
e-mail: ktsun@ipx.ntntc.edu.tw

The genetic algorithms have been widely applied to scientific and engineering problems in which a huge of computation is required for finding optimal solutions. In educational measurement, the parallel test forms' construction is a very important and difficult task. Since this problem is a combinatorial optimization problem that is required a large of time to select items for constructing tests with approximation a test information function. The genetic algorithm can code this parallel test construction problem as binary strings and solves them by using the genetic operations: reproduction, crossover, and mutation. Experimental results show that the genetic algorithm obtained very well results that are much better than those obtained by other methods. The average improvement ratio exceeds 98.26%. We successfully extend the applications of genetic algorithm to the educational measurement and this technique will be very useful to the test designers.

Keywords: **genetic algorithm, parallel tests, test information function, combinatorial optimization.**

1 Introduction

Since John Holland introduced the Genetic Algorithm (GA) in 1975 [10], many applications, especially the optimization problems and the search problems, have been studied and solved by this technique [2, 5, 7, 8, 21]. However, many problems in education, such as the parallel tests' construction, are also very difficult to solve and urgently find a more effective method to generate better results. The design of parallel test forms can be performed by the *Item response theory* (IRT) [9, 11, 12, 13, 20, 22]. *Item response theory* has served as the basis for test design for a variety of measurement purposes [1, 3, 6, 18]. However, the test construction problem (or item selection problem) can be formulated as a zero-one combinatorial optimization problem [14] and therefore processing time increases exponentially with the number of items in the item bank. For this reasons, many mathematical programming methods and heuristic methods have been developed to facilitate test design. These methods commonly involve the construction of parallel test forms, in which the test information function varies as little as possible between forms. The test information function can be computed by calculating the sum of the item information function $I_i(\theta)$ for the items included on the test [4]:

$$I(\theta)=\sum_{i=1}^{m} I_i(\theta), \qquad (1)$$

where m is the number of items on the test. For example, if a test for screening out the middle ability students is required to be constructed (see Figure 1), then items with greater information at the middle ability levels would be selected from the item pool. Table 1 shows the three parameters and item information of items in the item pool. For constructing parallel tests, a test is dedicated as the target test and another test is designed to approximate the test information function of the target test. The less the deviation there is between the target test information function and the constructed test information function, the more satisfactory the test is. Therefore, a test designer selects items which allow the information function of the constructed test to most closely approach the target test information function. Since the item selection problem is a combinatory optimization problem, the number of combinations increases exponentially with the number of items in the item bank. For this reason, the designers must use the weak methods (heuristic algorithms), which are capable only of finding "good" solutions but not "optimal" solutions. For example, linear programming (LP) techniques are the most commonly used for the test construction. In linear programming techniques, items are selected in order to optimize objectives within the given constraints.

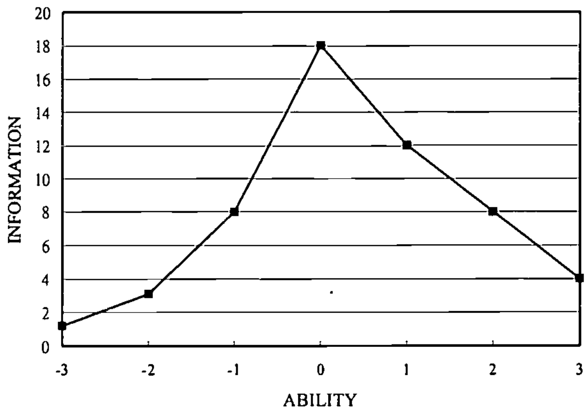


Figure 1. The curve of test information for screening out middle ability students.

Table 1. Three parameters and item information of items (the first 20 items of the item pool)

Item	Item Parameters			Ability Level						
	a	b	c	-3	-2	-1	0	1	2	3
1	0.763	-3.000	0.270	0.242	0.193	0.078	0.024	0.007	0.002	0.001
2	0.435	-0.546	0.200	0.024	0.052	0.083	0.093	0.075	0.049	0.027
3	0.803	0.537	0.310	0.000	0.003	0.035	0.171	0.250	0.130	0.041
4	1.187	0.080	0.350	0.000	0.002	0.059	0.466	0.293	0.052	0.007
5	0.844	-1.176	0.180	0.031	0.191	0.364	0.215	0.068	0.017	0.004
6	0.730	0.169	0.290	0.001	0.012	0.075	0.199	0.192	0.090	0.030
7	0.732	1.127	0.370	0.000	0.001	0.010	0.062	0.169	0.164	0.076
8	0.625	-1.650	0.260	0.062	0.147	0.164	0.101	0.044	0.017	0.006
9	1.428	2.831	0.370	0.000	0.000	0.000	0.000	0.001	0.102	0.715
10	0.541	0.063	0.280	0.006	0.024	0.069	0.117	0.114	0.072	0.035
11	0.983	-1.587	0.310	0.035	0.265	0.342	0.116	0.025	0.005	0.001
12	0.661	-1.707	0.250	0.070	0.171	0.182	0.102	0.041	0.014	0.005
13	0.538	-1.368	0.230	0.048	0.105	0.134	0.105	0.058	0.027	0.011
14	1.183	-0.378	0.290	0.000	0.012	0.247	0.549	0.156	0.024	0.003
15	0.400	-0.363	0.350	0.012	0.027	0.047	0.058	0.054	0.039	0.024
16	0.558	0.220	0.270	0.004	0.020	0.064	0.122	0.127	0.082	0.040
17	0.960	0.378	0.280	0.000	0.003	0.045	0.287	0.340	0.116	0.026
18	0.814	1.828	0.300	0.000	0.000	0.002	0.019	0.126	0.267	0.174
19	0.891	-0.490	0.310	0.002	0.033	0.205	0.301	0.132	0.035	0.008
20	1.083	-1.295	0.240	0.016	0.231	0.526	0.195	0.036	0.006	0.001

Good solutions can be produced by a variety of heuristic methods, such as the branch-and-bound method, the revised simplex method which use the relaxed 0-1 linear programming model, the weighted deviation model, thc ncural nctwork technique, and the greedy approach. Test construction problems commonly involve a list of objective functions with various purposes, but the test information function is the common objective of all test design problems. Therefore, in this paper, we will only consider how to select items in order to meet the requirements of the test information function. The difficulty of this problem, however, is not reduced by eliminating consideration of the content attributes. The test construction problem can be coded as binary strings, and then solved by the genetic operations: reproduction, crossover, and mutation. In order to evaluate the performance of this method, two hundred test information functions for parallel test forms of two types (one-peak and two-peak functions) were randomly generated. Experiments show that the GA technique generates much better results than those obtained by other methods with improvement ratios exceeding 98.26% (i.e., the errors of test information functions obtained by our method are better than others one order of magnitude). The GA method produces tests in which the test information functions very closely approach the target test information functions. It should be very useful to test designers for constructing parallel tests.

2 Test Construction by the Genetic Algorithm

The GA technique for the test construction problem is to code the state of the items as binary strings (i.e., chromosome strings), and then apply the genetic operations: reproduction, crossover, and mutation, to generation a set of new chromosome strings for next generation. The evolution of chromosome strings will proceed until the number of generations reaches the presetting value. Then, the most "fit" chromosome string is the result for the problem. So, a simple GA works as follows:

1. Randomly generate a population with ***P*** *n*-bit chromosome (candidate solutions to a problem).
2. Calculate the fitness (error), *f(x)*, of each chromosome *x* in the population.
3. Repeat the following steps until *k* offspring have been generated:

Cross over a pair of chromosomes with probability p_c (crossover probability) at a randomly chosen point to form two offspring (as shown in Figure 2).

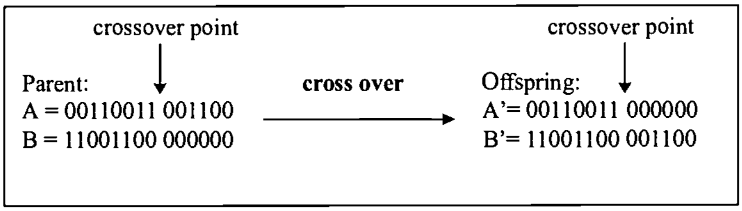


Figure 2. An illustration of 1-point crossover operation for a pair of chromosomes with 14-bit binary strings, and a crossover point is randomly selected between the 8th and the 9th bit positions.

(1) Mutate the state of gene in the chromosome with probability p_m (mutation probability), and place the resulting chromosomes in the new population (as shown in Figure 3).

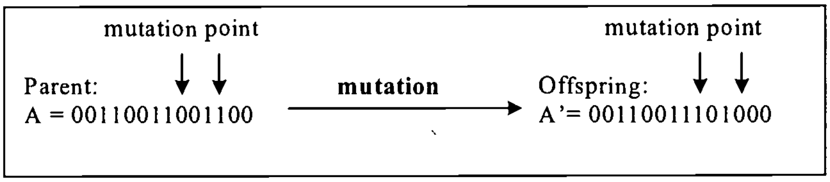


Figure 3. An illustration of mutation operation for a chromosome with the 14-bit binary string, and two mutation points are randomly selected at the 9th and the 12th bit positions.

(2) Select a number of chromosomes with the probability p_r from the current population, the probability of selection being an increasing function of fitness. The selected chromosomes are reproduced for the new population.

4. Replace the current population with the new population.
5. If the number of generations reaches the presetting value or obtain the required solution, then stop. Else go to step 2.

Each iteration of this process is called a generation, and the entire set of generations is called a run. For the test construction problem, the chromosome string and the fitness function are designed as follows. We use a binary string (i.e., chromosome), ***X***, to represent the state of all items in the item bank. Each element, x_i, i=1 ~ n, in ***X*** is used to represent the state of item *i* which is included in (x_i=1) or excluded from (x_i=0) the item bank. Then, the fitness function for evaluating the score of a chromosome is defined as Equation (2).

$$E_I = \sum_{j=1}^{s}(d_{j.} - O_j)^2 \cdot \qquad (2)$$

The less the value of the fitness function, the less the error of the solution. Based on the fitness function, the chromosome strings in new population can be generated by genetic operations and then find better solutions. This evolution process will continue until reaches the presetting generation number or the required quality of result. For all generations, the best chromosome string with the least fitness function will be the final solution which closely approximates the target test information function. The detailed operations of the proposed approach are described as follows.

The Genetic Algorithm for Item Selection

In the proposed genetic algorithm, each chromosome string, $\boldsymbol{X}$, contains n-bit, where n is the number of items in the item bank, in which m bits are "1" and else are "0" for constructing a test with m bits. The state of bit x_i in the chromosome represents the corresponding item i that is included in (x_i=1) the test or excluded from (x_i=0) the test. Then, for each chromosome, we can compute the error between the test information function of constructed test and the test information function of target test. This squared error (see Equation (2)) is then defined as the fitness function for evolution process. By applying the genetic operations: crossover, mutation, and reproduction, the population in the new generation can be obtained such that the better chromosome strings (less error) may be generated.

The detailed operations for this approach are stated as follows.

1. Set the initial population of chromosome strings and parameters of evolution process.
   An initial population of $\boldsymbol{P}$ chromosome strings is randomly generated, with each containing n binary bits in which m bits are "1" and else are "0". The crossover, mutation and reproduction probabilities are set to p_c, p_m and p_r respectively. The maximum number of generations is set to *gener_no*, and the initial generation number, t, (the iteration index) is set to zero.
2. Compute the values of fitness functions for all chromosome strings in population.

$$fitness(X^k) = \sum_{j=1}^{s}(d_j - O_j^k)^2, \tag{3}$$

   where O_j^k is the value of test information function at ability level j for chromosome k. It can be computed as:

$$O_j^k(t) = \sum_{i=1}^{n} w_{ij} x_i^k(t),\ \forall j = 1\sim s. \tag{4}$$

   where w_{ij} is the item information function for the item i at the ability level j, and x_i^k is the state of item i in the chromosome k.

3. For each chromosome k in the population, complete the following genetic operations for generating $\boldsymbol{P}$ offspring.
   3.1. Two-point crossover is used in our method. The offspring for each pair of parent is generated with the probability p_c, and then a period of chromosome string in the offspring is the same as one parent and else is the same as another parent (as shown in Figure 4).

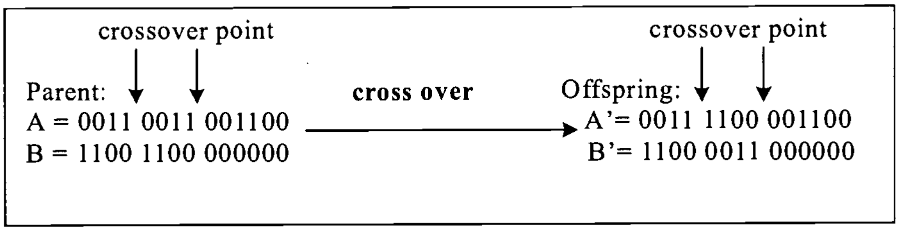


Figure 4. An illustration of the two-point crossover operation and 4 bits in the middle of chromosome strings are exchanged in offspring.

   3.2. For all chromosome strings, a mutation probability p_m is used to selected some chromosomes for changing the state of gene randomly.
   3.3. Find some chromosome strings with the probability p_r from the population, and then become the offspring for the new population.
   3.4. If the best chromosome in offspring is satisfied the requirement of test designer, or the generation number is reached the maximum, then the evolution process is stopped. Else, the generation number is increased by one, and goes to step 2.

4. Stop.

At the end of computation, the best chromosome string is the solution of the item selection problem and the genes with state $x_i = 1$, for all i, are the items selected for inclusion on the final version of the test.

In the proposed GA approach, all chromosome strings are selected by the genetic operations such that the new chromosome strings in the new population are generated. In this way, the offspring may be better than the parent, and obtain the better solution for the item selection problem. Experimental results show that our method is much better than other methods, as will be discussed in the next section.

3 Performance Evaluation

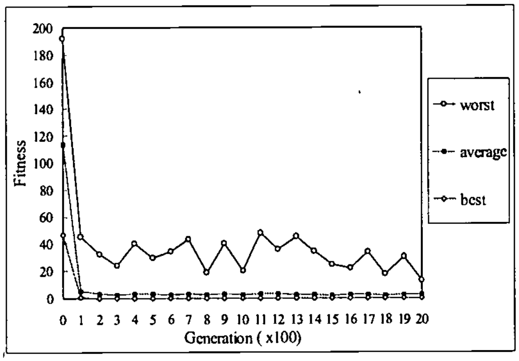
Several methods have recently been proposed for more efficient construction of parallel test forms; we will use four of them to compare the performance with our method. Those are the greedy approach [16], the neural network method [15], the Swanson and Stocking method [17], and the Wang and Ackerman method [19]. We used a real item bank with 320 items based on the three-parameter model of IRT to compare the performance of genetic algorithm with that of other methods. The amount of information on the target test varied within the ranges shown in Table 2 (for a one-peak shape test information curve) and Table 3 (for a two-peak shape test information curve). Following the limitations of the information quantities defined in these two tables, one hundred target test information functions were randomly generated for each of the two shapes. The parameters for evolution are defined as following: $\boldsymbol{P} = 150$, $n = 320$, $m = 40$, $p_c = 100\%$, $p_m = 0.4\%$, $p_r = 13.3\%$, and *gener_no* = 2000. Figures 5 and 6 show the evolution process for one-peak and two-peak test information functions, respectively. The test information functions constructed by different methods are shown in Figures 7 and 8. The average sum of the squared error between the information functions of the target tests and that of the constructed tests is shown in Table 4. We see that the proposed genetic approach greatly reduces error, with average improvement ratios greater than 98.26%. We see that the proposed item selection method can be applied to generate excellent results. It should prove be very useful to test designers who are constructing parallel test forms or desired tests for a variety of assessment purposes.

Table 2. The ranges of test information used to randomly generate 100 target test information functions (one-peak shape).

	Index of Ability Level				
	1	2	3	4	5
Ability Level	-2.0	-1.0	0.0	1.0	2.0
Test Information	4 ~ 5	6 ~ 8	18 ~ 21	6 ~ 8	4 ~ 5

Table 3. The ranges of test information used to randomly generate 100 target test information functions (two-peak shape).

	Index of Ability Level				
	1	2	3	4	5
Ability Level	-2.0	-1.0	0.0	1.0	2.0
Test Information	5 ~ 6	11 ~ 13	7 ~ 9	11 ~ 13	5 ~ 6



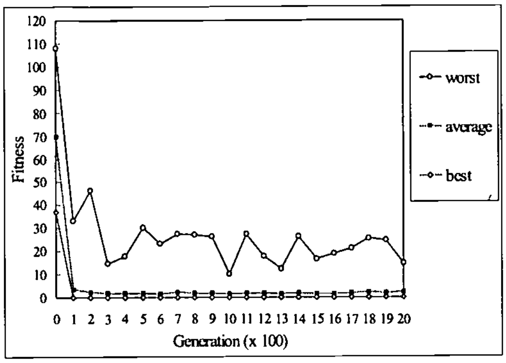


Figure 5. The worst, the average, and the best cases in the evolution for solving the item selection problem selection with one-peak shape test information function by using the genetic algorithm.

Figure 6. The worst, the average, and the best cases in the evolution for solving the item problem with two-peak shape test information function by using the genetic algorithm.

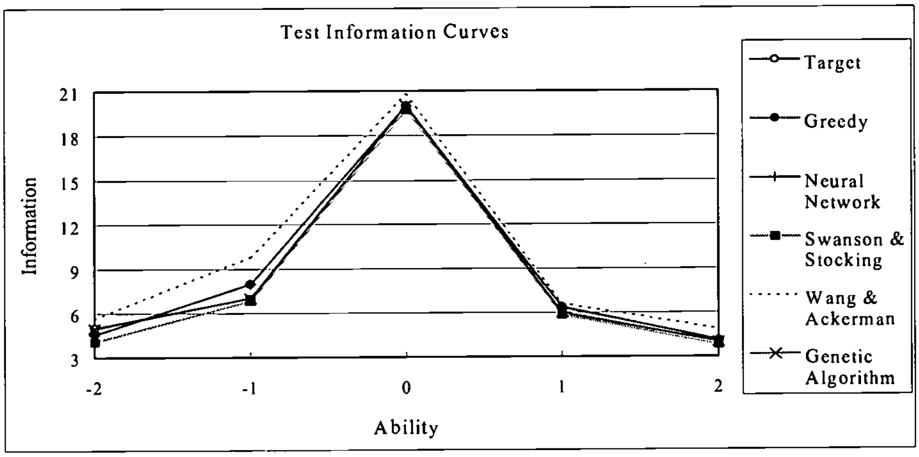


Figure 7. Test information curves (one-peak shape) for a target test, a test produced by the greedy approach, the neural network approach, Swanson & Stocking method, Wang & Ackerman method, and the genetic algorithm.

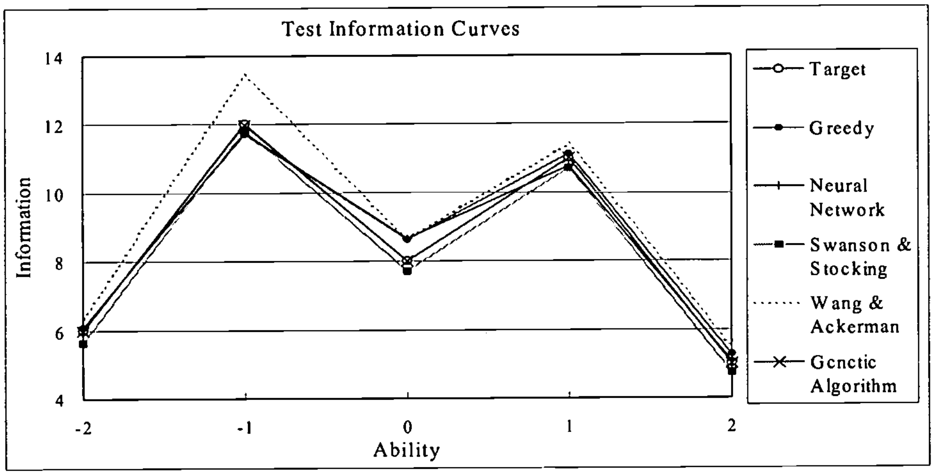


Figure 8. Test information curves (two-peak shape) for a target test, a test produced by the greedy approach, the neural network approach, Swanson & Stocking method, Wang & Ackerman method, and the genetic algorithm.

4 Conclusions

In this paper, a novel method, based on the genetic algorithm, is proposed to construct a desired test from an item bank. The proposed method can effectively construct parallel test forms or a test whose test information function closely approximates that of a target test. A real item pool was used to evaluate the performance of our method. The experimental results show that the proposed approach is able to obtain very well results that are much better than other methods. The average improvement ratio exceeds more than 98.26%. We successfully extend the applications of genetic algorithm to educational measurement and this technique will be very useful to the test designers..

Acknowledgments

This research was supported by the National Science Council of Taiwan, ROC, under the grant NSC 89-2520-S-024-001-.

Table 4. The average sum of squared error between the information function of the target test and the information function of the constructed test. The improvement ratios of the average error are also shown.

Methods / Conditions	Genetic Algorithm	Greedy Approach	Neural Network	Swanson & Stocking	Wang & Ackerman
Table 3	1.3925E-2	0.7251	0.6945	0.9715	12.8340
Table 4	5.6887E-3	0.6986	0.7416	0.2755	2.9536
Average Errors	9.8069E-3	0.7119	0.7181	0.6245	7.8938
Improvement Ratio* (%)	---	98.2664	98.6343	98.3976	99.8758

*Improvement Ratio (%) = ($error_X$− $error_G$) / $error_X$ × 100

$error_G$: the errors generated by the genetic algorithm.

$error_X$: the errors generated by one of the following methods: the greedy method, the neural network, the Swanson & Stocking method, or the Wang & Ackerman method.

References

[1] Ackerman, T., “An alternative methodology for creating parallel test forms using the IRT information function,” The Annual Meeting of the National Council for Measurement in Education, San Francisco, (1989).

[2] Altenberg, L., “The evolution of evolvability in genetic programming,” In K. E. Kinnear, Jr., ed., Advances in Genetic Programming, MIT Press, (1994).

[3] Baker, F. B., Cohen, A. S., & Barmish, B. R., “Item characteristics of tests constructed by linear programming,” Applied Psychological Measurement, Vol. 12, pp. 189-199, (1988).

[4] Birnbaum, A., “Some latent trait models and their use in inferring an examinee’s ability,” In F. M. Lord & M. R. Novick, Statistical theories of mental test scores (pp. 395-479). Reading, MA: Addison-Wesley, (1968).

[5] Bramlette, M. F., “Initialization, mutation and selection methods in genetic algorithms for function optimization,” In R. K. Belewand L. B. Booker, eds., Proceedings of the Fourth International Conference on Genetic Algorithm, Morgan Kaufmann, (1991).

[6] De Gruijter, D. N. M., “Test construction by means of linear programming,” Applied Psychological Measurement, Vol. 14, pp. 175-181, (1990).

[7] Davis, L., ed. Handbook of Genetic Algorithms, Van Nostrand Reinhold, (1991).

[8] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Mass, (1989).

[9] Hambleton, R. K., & Swaminathan, H., Item Response Theory-- Principles and Applications.

Netherlands: Kluwer Academic Publishers Group, (1985).

[10] Holland, J. H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Mich, (1975).

[11] Lord, F. M., "A theory of test scores," Psychometric Monograph, 7, (1952).

[12] Lord, F. M., & Novick, M. R., Statistical theories of mental test scores, Reading, MA: Addison-Wesley, (1968).

[13] Lord, F. M., Applications of item response theory to practical testing problems, Hillsdale, NJ: Erlbaum, (1980).

[14] Papadimitriou, C. H., & Steiglitz, K., Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Inc., NJ: Englewood Cliffs, (1982).

[15] Sun, K. T. & Chen, S. F., "A study of applying the artificial intelligent technique to select test items," Psychological Testing, Vol. 46, No. 1, pp. 75-88, (1999).

[16] Sun, K. T., "A greedy approach to test construction problems," Submitted to *The Proceedings of the National Science Council (Part D): Mathematics, Science, and Technology Education,* (2000).

[17] Swanson, L., & Stocking, M. L., "A model and heuristic for solving very large item selection problems," Applied Psychological Measurement, Vol. 17, No. 2, pp. 151-166, (1993).

[18] Theunissen, T. J. J. M., "Binary programming and test design," Psychometrika, Vol. 50, pp. 411-420, (1985).

[19] Wang, C. S., & Ackerman, T., "Two item selection algorithms for creating weakly parallel test forms using the IRT information functions," Psychological Testing, Vol. 44, No. 2, pp. 123-140, (1997).

[20] Weiss, D. J., "Improving measurement quality and efficiency with adaptive testing," Applied Psychological Measurement, Vol. 6, pp. 379-396, (1982).

[21] Wright, A. H., "Genetic algorithms for real parameter optimization," In G. Rawlins, ed., Foundations of Genetic Algorithms, Morgan Kaufmann, (1991).

[22] Wright, B. D., & Stone, M. H., Best test design, Chicago: MESA, (1979).

A Learning Environment for Problem Posing in Simple Arithmetical Word Problems

Akira Nakano, Naoki Murakami, Tsukasa Hirashima and Akira Takeuchi
Kyushu Institute of Technology Department of Artificial Intelligence
680-4 Kawazu, Iizuka 820-8502, Japan
E-mail : nakano@minnie.ai.kyutech.ac.jp

Several researchers indicate that to pose arithmetical word problems is an important way to learn arithmetic. However, the problem posing practice actually is not popular. In this paper, we describe an Intelligent Learning Environment which realizes the problem posing practice. In the problem posing practice, the learners pose problems by using the tools provided by the ILE. The ILE has a facility to diagnose the problems posed by the learners. By using the result of the diagnosis, the ILE indicates whether the problems are correct or not, helps the learner to-correct the wrong problems, and provides the next step of problem posing. We used the ILE in three different situations for evaluation. The subjects were elementary school teachers and elementary school students. We also report the results of the evaluation. In the ILE, the interface was implemented in Java, and the diagnosis module was implemented in Prolog. So it can be used on World Wide Web. The current environment deals with simple arithmetical word problems.

Keywords: **intelligent learning environment, problem posing, intelligent tutoring system, interactive education, World Wide Web**

1 Introduction

The main purpose of the practice to solve arithmetical word problems is to make learners recognize the relations between concepts and numerical relations, and master to use the relations. Although the problem solving practice is the most popular way, it is not the only way. Several researchers indicate that to pose arithmetical word problems is also effective. However, the problem posing practice actually is not popular.

The main reason is that the problem posing practice is strongly required teachers to deal with each learner individually in comparison with the practice of problem solving. We aim to realize computer-based learning environments for the problem posing practice [1]. For the problem solving practice, many ILEs are developed so far [2-6]. However, there are few ILEs for the problem posing practice until now.

This paper describes an Intelligent Learning Environment for the problem posing practice for simple arithmetical word problems that can be solved by the addition of one time or the subtraction of one time. The main characteristic of the ILE is the function to diagnose the posed problems. By using the results, the ILE indicates errors in the posed problems and suggests that the next step of problem posing.

Interface of the ILE was implemented in Java, and the diagnosis module was implemented in Prolog. Therefore, if only users have a computer connected to Internet with a popular internet browser, they can use thc ILE through WWW: E-mail: nakano@minnie.ai.kyutech.ac.jp http://www.minnie.ai.kyutech.ac.jp/~nakano/problem-posing.shtml (currently Japanese only).

In this paper, the first, the necessity of problem posing and an Intelligent Learning Environment for it are described. Then, interface and diagnosis module of the ILE are explained. The results of preliminary evaluation of the ILE are also reported.

2 Background

2.1 The necessity of an ILE for problem posing

Several researches about problem posing of arithmetical word problems suggested that problem posing was important to learn arithmetic, for example, analysis and investigation about the task of problem posing [7,8], investigation about effect of the problem posing practice [9], investigation in the problem posing practice at arithmetic class [10,11]. Besides, the Curriculum and Evaluation Standards for School Mathematics (in USA, 1989), and Professional Standards for Teaching Mathematics (in USA, 1991) also indicated that it was important for learners to experience to pose problems.

However, the practice actually is not popular in arithmetic class in comparison with problem solving practice. In the practice of problem solving, every problem has an answer and one or a few solution methods. Therefore, the teachers can easily judge the results of problem solving by learners. Then when the answer is wrong, to tell the correct answer or the solution method is not meaningless.

In contrast with problem solving practice, to prepare every correct problem in the problem posing practice is very difficult. Besides, the correct problem that a learner is trying to pose, after depends on the wrong problem posed by the learner. Therefore, the teachers have to examine each problem whether the problem is correct or not, and where of the problem is wrong.

Based on this consideration, we believe that to realize an ILE for problem posing with problem diagnosis function is the promising way to make learning by problem posing popular.

2.2 The problem posing dealt in the ILE

Silver has noted that the term "problem posing" is generally applied to three quite distinct forms of mathematical cognitive activity [12]. They classified three types of problem posing: (1) presolution posing, in which one generates original problems from a presented stimulus situation, (2) within-solution posing, in which one reformulates a problem as it is being solved, and (3) postsolution posing, in which one modifies the goals or conditions of an already solved problem to generate new problems. The problem posing deal in our ILE is (2) within-solution posing. In the ILE, because, in the ILE, first, a learner decides a calculation formula to solve the problem, and next, he/she is trying to pose problem solved by the calculation.

Currently, the LIE can deal with only Change-Problem[13]. In Change-Problem, the quantity in the initial situation is changed to the quantity in the final situation by the change action. The Change-Problem usually consists of three sentences: the first sentence describes the initial situation, the second sentence describes the change action, and the third sentence describes the final situation. Therefore, we prepare a "problem template" that composed of three single sentence templates. By filling in the blanks of three single sentence templates, the problem is completed.

In the ILE, the template of Chang-Problem is composed of the tree single sentence templates that describe: initial situation, change action, and final situation, respectively. The initial situation has the four information: "owner", "object", "number", and "unit". This means that "owner" has "object" and the number of "object" is "number", then, the unit of the number is "unit". The change action has the five information: "actor", "object", "number", "unit", "action". Several actions, for example, "take" has two more information: "from" and "to". The final situation has the four information: "owner", "object", "number", and "unit".

3 ILE for problem posing

3.1 Configuration of the ILE

The current version of the ILE consists of clicnts and server shown in Figure 1. A client is an interface of the ILE. The interface provides learners the tools to pose problems and gives them guidance to promote problem posing. Inter face is explained in more details in this section 3.2.

The server has two modules: the one is Problem Diagnosis Module and the other is Advice Generator. First, the ILE receives the posed problem, and diagnoses it in Problem Diagnosis Module. Next, in Advice Generator, the ILE generates advice for each learner by using the result of diagnosis. These are explained in more details in this section 3.3.

Because the ILE deals with several learners by one server, the ILE manages ID, PW, and Learner Model in Private Information Manager.

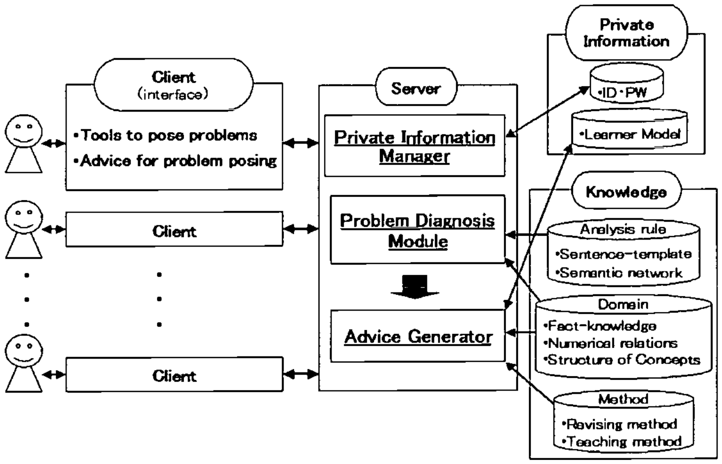


Figure 1: The frame of the ILE

3.2 Interface

Figure 2 shows the interface of the ILE. Current interface deals with only Japanese. In Figure 2, Japanese was translated into English for this paper. The parts of the interface are expressed as follows.

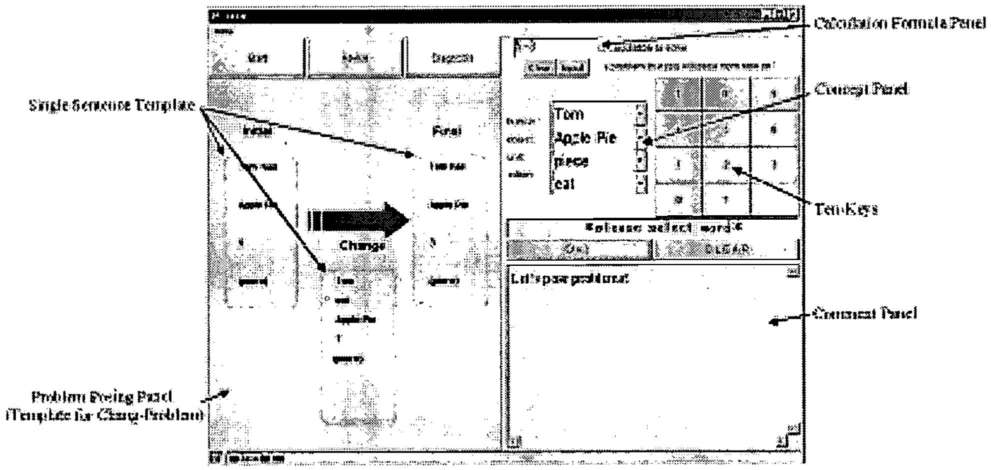


Figure 2: the prototype Interface (English version)

- Calculation Formula Panel

In this panel, a learner gives calculation formula. The learner poses problems which can be solved by this calculation formula.

- Concept Panel

This panel provides concepts to fill in blanks of sentences (three single sentences templates). The concepts that are provided in the Concept Panel are classified in the five categories: “human”, “object”, “unit”, “action”, “number”.

- Ten-key

Numerical values are put into blanks of sentences with Ten-key.

- Problem Posing Panel

In the current version, this panel provides the template of Change-Problem. The ILE asks a learner to fill in blanks of sentences. In the order of the blanks, the ILE gives questions. By answering the questions, the blanks are filled in. Here, the learner has to select concepts from Concept Panel. By using Figure 3, posing a sentence of initial situation in Chang-Problem is explained. The left side of the figure shows questions.

For example, the initial situation in Chang-Problem is composed of four elements: "owner", "object", "number" and "unit". So, the ILE asks the learner "Who has?", "What the person has?", "How many?", "What is unit?". The learner also should decide what number is the answer by selecting the question mark in Ten-key.

The right side of the figure shows an example which the learner answered the questions. The result shows "Tom has 5 pieces of Apple Pies".

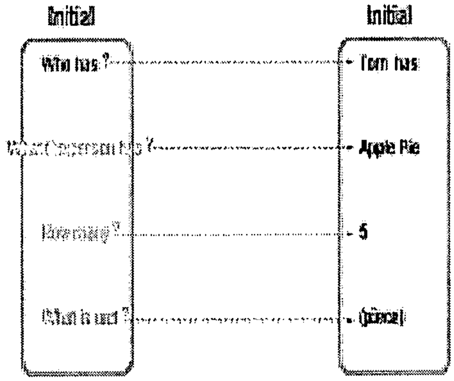


Figure 3: An example of posing sentence by using Single Sentence Template

By answering the all questions, learners pose problems For example, Figure 2 shows the correct problem in Problem Posing Panel: the initial situation is "Tom has 5 pieces of Apple Pies", the change action is "Tom eats the 3 pieces of Apple Pies", and the final situation is "How many pieces of Apple Pies does Tom have?".

- Comment Panel

This panel shows advice and suggestion massages that are generated based on the diagnosis of the posed problems.

3.3 Problem posing in the ILE

A learner poses a problem by the following process.

(1). Giving a calculation formula

First, the learner gives a calculation formula. The calculation formula consists in three elements. That is, two operands and an operator. Because the calculation formula is the way to get the answer of the problem, we call it solution.

The solution can be applied to several numerical relations. For examples, if the learner assigned "5-3" to the solution, the solution can be applied to the following four numerical relations: (a) "5-3=X", (b) "5-X=3", (c) "3+X=5", (d) "X+3=5" (the current version of the ILE only handles natural numbers). Here, numerical relation (a) means the answer is the number in the final situation, numerical relation (b) and (c) mean the answer is the number in the change action, and numerical relation (d) means the answer is the number in the Initial situation.

(2). Selecting concepts from Concept Panel and combining them with the template of Change-Problem

The template has several blanks, and the ILE asks the learner to pose a problem by filling the blanks with the concepts. Then, if the learner selected a concept from the set of wrong concepts, the ILE can give the learner feedback, which suggested that the concept is wrong.

(3). Request to diagnose a problem

When the learner clicks the "diagnosis button", the problem is sent to the server and is diagnosed.

(4). Revising the wrong problem by using the suggestion given in the Comment Panel

When the posed problem is wrong, the learner receives feedback that indicates an error at Comment Panel.

The ILE generates the message by using the result of the diagnosis.

(5). Posing the new problem by using the suggestion

When the learner posed the correct problem, the learner receives feedback which is suggests to pose the new type of problems.

3.4 Problem Diagnosis Module and Advice Generator

Problem Diagnosis Module and Advice Generator are functions of the server in the ILE. Problem Diagnosis Module diagnoses problems sent by the client, and Advice Generator generates messages that are provided for each learner.

The ILE, first, diagnoses a single sentence and then diagnoses the problem composed of three sentences, and compares the solution given by a learner with the problem posed by the learner. In the first step, the module has knowledge about acceptable sentences (initial situation, change action, final situation). We call each sentence "basic relation", and the knowledge "single sentence schema". The single sentence schema checks each basic relation to find the errors in a sentence.

In the second step, the relation among the sentences is diagnosed. The module has the knowledge about acceptable relations among basic relations. We call the knowledge "problem schema". The problem schema checks the numerical relation between the sentences to find the wrong sentence in the problem.

In the third step, the relation between the solution and the problem is diagnosed.

In the following section, the diagnosis process is explained. Then, the feedback made by the diagnosis result is presented.

3.4.1 Diagnosis of the posed problems

Diagnosis of the posed problems is carried out in three steps: the first step is the diagnosis of a single sentence. The second step is the diagnosis of the problem composed of three sentences. The third step is the diagnosis of the relation between the problem and the solution.

(1). Diagnosis of a single sentence

In this diagnosis, two types of errors are detected: (1-a) errors in the relation between object and action, and (1-b) errors in the relation between object and number. Here, Mismatch of blanks (that is, object blank or action blank and so on) and concepts is already checked in the interface.

An example of (1-a) is a sentence that "Tom eats his 2 sheets of postcards." "Tom has 5 cups of apple pies" is an example of (1-b). These errors are detected by checking with sentence schema in that the acceptable relations between object and action or object and number are described.

(2). Diagnosis of problem

In this diagnosis, three types of errors are detected: (2-a) errors in the final situation, (2-b) errors in the change action and (2-c) no relation errors. (2-a) means that the initial situation can be changed by the change action, but cannot be changed to the final situation. (2-b) means that the initial situation can be changed to the final situation, but cannot be changed by the change action. (2-c) means that the initial situation cannot be changed by the change action and to the final situation. These errors are detected by comparing by problem schema in that the acceptable relations among the situations and the change action are described.

An example of (2-a) is the problem composed of the following three sentences: "Tom has 5 pieces of apple pies", "Nancy eats Tom's 3 pieces of apple pies" and "how many pieces of lemon pies does Tom have?" An example of (2-b) is the problem composed of the following three sentences: "Tom has 5 pieces of apple pies", "Nancy eats her 3 pieces of apple pies, and "how many pieces of apple pies does Tom have?"

(3). Diagnosis of the relation between the problem and the solution

The diagnosis module can generate an equation from the problem. In this diagnosis, first, the module solves the equation. Then the calculation to derive the answer is compared with the calculation posed by the learner as the solution. When the two calculations do not correspond, an error in the relation between the problem and the solution is detected.

3.4.2 Feedback for the client

(1). Indication of an error

If the diagnosis module finds an error, the ILE indicates it. Even if the problem includes several errors, the

ILE indicates the error detected first.

(2). Suggestion of the next step of problem posing

The ILE suggests the next step of problem posing when the posed problem is the correct one. In the diagnosis, the module diagnoses not only whether the problem is correct or not, but also what concepts, actions or equations are used in the problem. Based on the results, the ILE can suggest more difficult problem posing by specifying concepts or an equation type to be allowed to use in problem posing.

4 Preliminary evaluations

A prototype of the ILE has been already developed. We used it in three different situation for evaluation, as follows: (1) Use by teachers of the elementary school, (2) Use by students of elementary school in arithmetic classes, (3) Use by students of elementary school outside the class.

In (1), we asked the teachers to evaluate the ILE from the viewpoint of teaching. Then, two of them permitted us to use the ILE in their arithmetic class. So, we had two opportunities to evaluate the ILE in the second situation. In (2), we asked the students of elementary school to pose arithmetical word problems with the ILE in two arithmetic classes. In the trial, although we collected the answers for our questionnaires, we failed to record logs of problem posing. Therefore, we could not get the data about the number of posed problems, and the students behave for feedback from the ILE. In (3), we gathered several students again, and asked them to use the ILE out of class. Here, the students used the ILE for the first time.

In this section, we report these results.

4.1 Use by the teachers of the elementary school

To evaluate a learning environment, the evaluation by teachers is important. We asked five teachers of elementary school to use the ILE. After they posed several problems by using this ILE, we asked them several questions. The questions are as follows: (1) How do you evaluate the effect of problem posing to learn arithmetic? (2) How do you evaluate the way of problem posing used in the ILE? (3) How do you evaluate the interface? (4) How do you evaluate the indications for the errors in posed problems? (5) How do you evaluate the advises to suggest the next step of problem posing? Table 1 shows the results.

Table 1: Evaluation of the teachers

	Good	So-so	Bad
(1)	5	0	0
(2)	3	2	0
(3)	2	3	0
(4)	4	1	0
(5)	5	0	0

Table 1-(1) means that all teachers think to learn arithmetic by using problem posing is effective. Table 1-(2) suggests that the ILE realizes an useful environment for learning by problem posing. Two teachers out of three teachers who answered "Good" to the question (2), gave us opportunities to use the ILE in classes. A few teachers also indicated that the limitation of concepts that were allowed to use in problem posing should be revised. This is one of our future works. In Table 1-(3), three teachers answered "So-so". The result means that the interface is not always easy to use. In Table 1-(4), four teachers answered "Good". The result suggests that the indications for the errors in the posed problem are acceptable. However, several teachers also indicated that the sentences of the indications may be difficult for elementary students. In Table 1-(5), the all teachers answered "Good". This result means that the teachers think the suggestions to make learners progress the next step of problem posing adequately support learning by problem posing.

4.2 Use by the students of elementary school in arithmetic classes

We used the ILE in two classes: the one was composed of 25 students in third grade and the other was composed of 30 students in fifth grade. In each class, 15 minutes were used to explain the use of the ILE, and 20 minutes were used for the problem posing practice with the ILE. In this problem posing practice, students were two people one set, and they operated one personal computer with two. Then two assistants assisted them to operate the ILE in the experiments.

Table 2: Evaluation of the

		Yes	No Answer	No
Third-grad	(1)	22	0	2
	(2)	22	0	2
Fifth-grad	(1)	27	2	1
	(2)	25	4	1

We asked two questions after the problem posing practice: (1) Are you interested in problem posing by using this ILE? (2) Do you want to pose more problems by using this ILE? The result is shown in Table 2.

The results suggested that most students were interested in problem posing with the ILE. But we were not able to get enough data to confirm that the students pose problem well.

4.3 Use by the students of elementary school outside the class

Subjects were one student of fourth-grade, and threes students of sixth-grade in elementary school. In the experiment, we used 15 minutes in the demonstration of this ILE, and 25 minutes in the problem posing practice. The results were as follows. In Table 3, *Diagnosis* indicates the number of time of request to diagnose.

Table 3: Logs of the problem posing

Diagnosis	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
i	**α-C**	**β-C**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-C**
ii	**α-C**	**α-C**	**β-W**	**β-C**	**γ-W**	**γ-W**	**γ-W**	**γ-W**		
iii	**α-C**	**α-C**	**β-C**							
iv	**α-C**	**β-C**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-W**	**γ-C**	

α: A±B=X, **β**: A±X=C, γ: X±B=C (A,B,C are numerical values. X is a variable)

C: Correct, **W**: Wrong

In Table 3, equations named by Greek (α, β, γ) specify the type of problem posed by subject. "A" is the number in the initial situation, "B" is the number of the change action, and "C" is the number in the final situation. "X" is the number that is derived by the solution. In the α type, the answer is in the final situation. So this type of problem is the easiest one. In the β type, the answer is in the change action. In the γ type, the answer is in the initial situation. In this order, problems become difficult. The ILE can judge not only "C (correct)" or "W (wrong)", but also the type of problem whenever the student requests the diagnosis.

In Table 3, three subjects (i, ii, iv) tried to pose the problems of the all types, and subject-iii tried to pose the two types of the problems. The subject-i posed the wrong problem of the γ type on the 3rd request to diagnose in the practice. And the subject was repeating to revise it in seven times. As a result, the subject posed the correct problem of the γ type in the 10th trial. And the subject-ii posed the wrong problem of the β type on the 3rd request to diagnose in the practice, then the subject posed the correct problem of the β type in the 4th trial. And the subject-iv posed the wrong problem of the γ type on the 3rd request to diagnose, then the subject posed the correct problem of the γ type in the 9th trial, too. But, the subject-ii gave up to correct the wrong problem of the γ type, although s/he was repeating to revise the wrong problem in three times. The results suggest that the feedback is effective to forward the learner to revise the wrong problem.

In the current ILE, if a learner corrected the problem, the ILE suggests the next step of problem posing. The first step is problem of the α type. the second step is problem of the β type. and the third step is problem of the γ type. In Table 3, all subjects follow the suggestion. In the results, when a learner posed a correct problem, the learner can not poses only the same type of problem again, but also other types of problem by using the feedback. This suggests that the feedback is also effective to advance the next step of problem posing.
Conclusions

5 Conclusion

In this paper, we described ILE for problem posing in simple arithmetical word problems. The ILE provides the template to pose Change-Problem in current version. And the ILE can diagnose the problem that learners fill blanks of the template with several concepts, values, and question mark. Besides, the ILE can support each learners by using the results of diagnosis. We used the ILE in three different situations for evaluation. In the results, we consider that this research provides basis functions to realize the problem posing practice by ILE about simple arithmetical word problems.

In future work, we will refine functions in the ILE. For example, in the ILE, we will deal with not only Change-Problem, but also the other types of problems. And we will develop a function in which teachers can customize concepts provided for their students in their problem posing practice, because teachers hope to use concepts which are popular in their classroom. Then, we will evaluate the ILE again in order to investigate about the effect to learn arithmetic.

References

[1] Nakano，A.，T.Hirashima，A.Takeuchi，"Problem-Making Practice to Master Solution-Methods in Intelligent Learning Environment，" Proc of ICCE'99，pp891-898，1999.

[2] Barr,A., Beard, M., Atkinson, R.C., "The Computer as a Tutorial Laboratory: the Stanford BIP Project". Int.J. Man-Machine Studies, Vol.8, pp.567-596, 1976

[3] MaCalla, G.I. "The Search for Adaptability, Flexibility, and Individualization:Approaches to Curriculum in Intelligent Tutoring ILEs", In Jones, M. and Winne P.H.(Eds.), Adaptive Learning Environment: Foundation and Frontiers, Springer-Verlag, pp.91-122, 1990

[4] Hirashima, T., A.Kashihara, J.Toyoda : "Toward a Learning Environment Allowing Learner-Directed Problem Practice". Proc. of ITS'96 Lecture Notes in Computer Science 1086, pp.466-474,1996

[5] Hirashima, T., A.Kashihara, J.Toyoda : "Providing Problem Explanation for ITS", Proc. of ITS'92 Lecture Notes in Computer Science 608, pp.76-83, 1992

[6] Hirashima, T., T.Kohno, A.Kashihara, J.Toyoda : "Development of Support Facilities for Arithmetical Word Problem Exercises in Intelligent Tutoring ILE". Electronics and Communications in Japan, pp.22-33, 1992

[7] Silver，E.A.，Mamona,J.，"Posing Mathematical Problems:An Exploratory Study，" Journal of Research in Mathematics Education，vol.27，No.3, pp293-309，1996.

[8] Lyn D. English，"Children's Problem Posing Within Formal and Informal Contexts，" Journal of Research in Mathematics Education，vol.29，No.1, pp83-106，1998.

[9] Ellerton, N. F., "Children's made up mathematics problems: A new perspective on talented mathematicians," Educational Studies in Mathematics, 17, pp261-271, 1986.

[10] Silverman, F. L., Winograd, K., Strohauer, D. "Student-generated story problems. Arithmetic Teacher," 39, pp6-12, 19992.

[11] Silver，E.A.，Cai,J.，"An Analysis of Arithmetic Problem Posing by Middle School Students，" Journal of Research in Mathematics Education，vol.27，No.5, pp521-539，1996.

[12] Silver, E.A, "On mathematical problem posing. For the Learning of Mathematics," 14(1), 19-28

[13] Kintsch, W., Greeno, J.G., "Understanding and Solving Word Arithmetic Problems," Psychological Review, vol.92, No.1, pp109-129, 1985.

A Method of Creating Counterexamples by Using Error-Based Simulation

Tomoya Horiguchi* and Tsukasa Hirashima**

**Faculty of Information Engineering, Kobe University of Mercantile Marine*
5-1-1, Fukaeminami, Higashinada, Kobe 658-0022 JAPAN
Phone: +81-78-431-6263, E-mail: horiguti@ti.kshosen.ac.jp
***Department of Artificial Intelligence, Kyushu Institute of Technology*
680-4, Kawatsu, Iizuka, Fukuoka 820-0067 JAPAN
Phone: +81-948-29-7618, E-mail: tsukasa@ai.kyutech.ac.jp

The method of creating counterexample by using educational simulation is proposed. Error-Based Simulation (EBS) is used for this purpose, which simulates a learner's erroneous equation in mechanics problem. A learner's error is visualized as unnatural motion of a physical object. In order for EBS to be effective as counterexample, the followings are essential: (1) A learner can recognize the difference of unnatural motion in EBS from natural one in correct simulation, and (2) EBS must provide a learner sufficient information to understand the cause of error and to reach correct understanding. The former has been studied in the authers' previous works. In this paper, the latter is discussed. To identify a learner's error, misconceptions are classified based on problem-solving model, and are linked to their appearance on a learner's answer (error-identification rules). Then, to indicate the cause of error by EBS, unnatural motions in EBS are classified and linked to the misconceptions which they suggest (error-visualization rules). These functions are realized as rule-base systems. The architecture of EBS management system, which judges a learner's error and generates the suitable EBS using these functions, is proposed.

Keywords: **counterexample, simulation, mechanics, error, student model, motion perception**

1 Introduction

It is well known that cognitive conflict promotes learning process. It often occurs when a learner encounters the fact which is contradictory to her/his idea. Cognitive conflict motivates a learner to reconsider her/his idea, and often causes conceptual change [Gagne 85, Fujii 97].

Counterexample is useful for creating cognitive conflict. It provides a case in which a learner's idea doesn't account for the fact, or her/his procedure doesn't produce the correct solution.

However, one must be careful in using counterexample, because a learner often ignores or refuses it. Even when she/he accepts the counterexample, she/he needs some kinds of help to reach correct understanding. Without any assistance, a learner often comes to an impasse, or makes ad hoc rules which explain the exception only. Therefore, in using counterexample, the followings are essential [Fukuoka & Suzuki 94, Nakajima 97]:

(1) Counterexample must be recognized to be meaningful and acceptable. When the difference is clear and reliable between counterexample and a learner's expectation, she/he easily accepts it and reconsiders her/his idea.

(2) Appropriate assistance must be provided to lead a learner to correct understanding. Counterexample must include sufficient information for this. It will be helpful to explicitly describe the distinguishing attributes of counterexample.

Error-based Simulation (EBS) is an educational simulation which provides a learner counterexample. It simulates an erroneous equation made by a learner in solving mechanics problem. In EBS, a learner's error often appears as unnatural motion of a physical object, which differs from her/his prediction (She/he can usually predict the correct motion).

The authors have developed the method of generating effective EBS mainly from the above viewpoint (1) [Hirashima et al. 98, Horiguchi et al. 99]. The condition on which a learner can recognize the difference between EBS and correct simulation was formulated (Criteria for Error-Visualization: CEV), and the mechanism to estimate the quality of difference was proposed, which considers both clarity and reliability of the difference.

However, though such an EBS motivated a learner by indicating the existence of errors, it was not sufficient to lead her/him to correct understanding. It didn't provide sufficient information for this.

Therefore, this paper proposes the method of managing EBS from the above viewpoint (2). EBS must justly indicate the cause of a learner's error, and suggest how to correct it. The followings are the requirements and approaches for this purpose.

(a) The function which identifies the cause of error behind a learner's erroneous equation or her/his handwriting diagram.
Approach: First, construct a problem-solving model of mechanics. Secondly, based on the model, classify the misconceptions which occur in problem-solving as causes of errors. Thirdly, classify the appearances of the misconceptions on a learner's equation or handwriting diagram. Lastly, appearances and causes of errors are linked together correspondingly. These are called *Error-Identification Rules*.

(b) The function which generates the EBS indicating the identified cause of error by unnatural motion of a physical object.
Approach: First, classify the unnatural motions in EBS. Then, link them to the corresponding causes of errors, considering what kind of unnaturalness suggests what kind of misconception. These are called *Criteria for Cause-of-Error-Visualization*. With these criteria, EBSs are estimated their effectiveness. When there is no EBS which is judged effective, other teaching methods will be considered.

2 Previous works in Error-Based Simulation

Before proceeding to the main topic of this paper, we outline the stream of study in Error-based Simulation, which may be helpful to clarify the present problem and the position of this paper.

Stage 0 **[Hirashima et al. 98 for summary]**

The fundamental idea of EBS is very simple. In mechanics problem, many learners feel difficulty in thinking by equations, so EBS maps their equations from mathematical world to physical world. It embodies a learner's error as unnatural motion of a physical object, which makes it much easier to recognize the error. Here, we assume that unnatural motion in EBS is differ from a learner's prediction, that is:

Precondition-1: A learner can predict the correct motion (in spite of her/his erroneous equation).
This precondition is set through all stages of the research of EBS.

Stage 1 **[Hirashima et al. 98]**

Apparently, the key of this method is how a learner sees the difference between the unnatural motion in EBS and the predicted natural motion. At least, the difference must be noticed by a learner. When the difference of two motions is small, she/he may not notice it, or cannot judge which motion is correct (unfortunately, the ability of human vision is not so sensible). Therefore, we set the following assumption:

Assumption-1: EBS must satisfy CEV-1 and/or CEV-2 below to indicate the existence of error.
Condition for Error-Visualization 1 (CEV-1): There is a qualitative difference between the motion in EBS and the one in correct simulation, that is, the qualitative values of a physical object's velocity are different between them.
Condition for Error-Visualization 2 (CEV-2): There is a qualitative difference between the change of motion in EBS and the correct simulation, that is, the qualitative values of the derivative

of a physical object's velocity are different between them.

Stage 2 **[Horiguchi et al. 99]**

When regarding EBS as counterexample, the viewpoints (1) and (2) in chapter 1 are important. We previously worked out how to estimate the effectiveness of EBS from the viewpoint (1). It is subdivided into two viewpoints: (1-1) how clear the error appears in EBS, and (1-2) how reliable the EBS is as counterexample.

From the viewpoint (1-1), the more CEVs the EBS satisfies, the more effective it is. In general, changing parameters of the mechanical system makes EBS satisfy more CEVs. For example, in Figure 2, the EBS based on erroneous equation m2a = T + μm2g (Figure 2d) satisfies CEV-1. (The qualitative value of relative velocity between two blocks is [+], while it is [0] in normal case.) But, when the mass of m2 increases, the EBS becomes to satisfy CEV-2 besides CEV-1. (The velocity of m2 increases, while it decsreses in normal case.) We categorized the methods of parameter-change and their influence on the clarity of errors.

However, from the viewpoint (1-2), such parameter-change harms the reliability of EBS, because a learner feels it factitious to change parameters too largely. The smaller parameter-change the EBS has (no change is the best), the more reliable it is. This discussion is summarized as follows.

Assumption-2: From the viewpoint of clarity, EBS should satisfy more CEVs.

Assumption-3: From the viewpoint of reliability, EBS should have less parameter-changes.

Stage 3 **[just this paper]**

In estimating the effectiveness of EBS, there is another, and important viewpoint: whether the EBS provides appropriate information for correcting the error, that is, the viewpoint (2) in chapter 1. Stage 0-2 have been mainly concerned with how to make a learner notice the error, while at this stage, our concern is how to make him correct the error.

For example, consider the erroneous equation m2a = T + μm2g (Figure 2d). From the viewpoint of reliability (1-2), the EBS shown in Figure 2c is generated. But it shows the string between two blocks shrinking, which may suggest something is wrong about tension of the string. It is misleading because the real cause of error is the friction of m2. In this case, the EBS in Figure 2e should be generated to indicate the cause of error. (It is generated when taking the viewpoint of reliability (1-2), but by accident.)

Of course, the viewpoints (1-1) and (1-2) are useful to impress on a learner the existence of error. However, in considering the error-correction, to generate EBS from the viewpoint (2) becomes necessary. It is the very topic of this paper.

3 Mechanism for Identifying the Cause of Errors

Now, we'll explain how to realize the functions described in chapter 1. The mechanism for identifying cause of errors is realized as follows:

1. to generate the correct solution by problem-solving model.
2. to specify the erroneous part of a learner's solution by comparing with the correct solution.
3. to identify the cause of error by applying the Error-Identification Rules, which link the appearance of erroneous part to its cause.

Here, a learner's solution means the equation and handwriting diagram made by her/him, from both of which the information about her/his problem-solving process is derived.

3.1 Problem-Solving Model

We deal with the mechanics problems of high school level, which ask a learner to set up equation of motion by using Newton's second law. The problem-solving process is divided into three steps [Robertson 90, Plötzner 94]:

step-1 to predict the motion of physical objects in the mechanical system qualitatively.
step-2 to enumerate the forces acting on each object.
step-3 to compose the enumerated forces and substitute them for the left side of formula F = ma.

Table 1.Force-Enumerating Rules (FERs) (abstract)

force	Rules for enumerating forces		
gravity	R0:	r0-c1 Object-1 has mass m > 0	
		→	
		r0-a1 Gravity F to Object-1	Qualitative
		r0-a2 Direction: vertically downward	Qualitative
		r0-a3 Magnitude: F = mg	Quantitative
friction	R3:	r3-c1 Object-1 and Object-2 are touching together ∧	
		r3-c2 coefficient of friction of touching surface μ > 0∧	
		r3-c3 normal force N acting on touching surface ∧	
		r3-c4 Object-1 and Object-2 are moving oppositely along the tangent	
		→	
		r3-a1 friction Ff1 to Object-1	Qualitative
		r3-a2 friction Ff2 to Object-2	Qualitative
		r3-a3 Direction(Ff1): opposite to the velocity of Object-1	Qualitative
		r3-a4 Direction(Ff2): opposite to the velocity of Object-2	Qualitative
		r3-a5 Magnitude: Ff1 = Ff2 = μN	Quantitative

Set up the equation of the block of mass m, which pulled by external force F, moving on the floor of coefficient of friction μ.

erroneous equation: ma = F

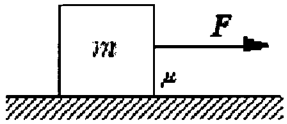


Figure 1. Example Problem-1

In step-1, a learner predicts the motion of objects in the system, and gives each object acceleration vector. Approproate axes are also set up. In step-2, she/he enumerates the forces which aren't given in problem description. Both qualitative knowledge (what kind of force acts in which direction?) and quantitative one (algebraic description of the magnitude of force) are used. In step-3, she/he decomposes/composes the enumerated forces along the axes, and substitute them for the formula F = ma.

In this paper, we don't model the error-occurring process in step-1, because it is presupposed that a learner correctly predicts the qualitative motion of objects in using EBS (Precondition-1 in chpter 2). We also omit the occurrence of error in step-3, which mostly concerns the knowledge of vector calculation.

Therefore, modeling step-2 is our central issue. Takeuchi and Otsuki (1997) considered that a learner constructs a model of causal structure of mechanical system, with which she/he infers the occurrence and propagation of forces. They formulated this process as a set of production rules. We modify them considering their qualitative/quantitative characteristics. A part of our model is shown in Table 1. The rules are called Force-Enumerating Rules (FERs).

3.2 Error-Identification Rules

In our model, a learner's errors are considered as the ones of FERs. The errors of FERs themselves and the ones in their application are included. In fact, these errors appear as the missing/extra/errors of the term of force in equation, or of the arrow of force in handwriting diagram. They are also linked to the strategies for correction.

For example, in Figure 1, the term of friction (-μmg) is missing in the erroneous equation. The cause of this error and its instruction are considered as follows:

1) A learner doesn't know the concept of friction itself, that is, doesn't know the rule R3 (Table 1).
Instruction: Re-teach the concept/definition of friction.
2) A learner is overlooking the preconditions of R3, that is, overlooking the fact that the block is touching the floor (r3-c1), or the fact the coefficient of friction is nonzero (r3-c2).
Instruction: Re-show the problem and indicate the corresponding part of the diagram.
3) A learner is missing the force which causes the friction, that is, missing the normal force (r3-c3).
Instruction: Proceed to the correcting strategy of normal force.
4) A learner doesn't think the block moves along the floor, that is, missing the relative velocity of them (r3-c4).
Instruction: This is the erorr of prediction of movement. So, out of the range of this paper. But, it may be useful to indicate the force which causes the block's motion.

Through such a consideration, the appearances of errors and their causes are classified as shown in Table 2. These are the Error-Identification Rules (EIRs), which are applied to the erroneous part of a learner's answer (specified by comparing with the correct solution), to identify the cause of error.

In Table 2, each error has its strategy for correction. Note that, it is not necessary to use EBS for every case. Of course, when other instruction method is more appropriate, it should be used. However, the aim of this paper is to clarify what kind of errors EBS is effective for, and how to estimate its effectiveness. For this purpose, we need to study the unnaturalness of physical objects' motion in simulation.

4 Criteria for Cause-of-Error Visualization

The identified error must be corrected. In this chapter, we formulate the criteria for judging whether an EBS is effective for the error. It means that EBS rightly indicates the cause of error and suggests the way of correction.

4.1 Motion and Forces

In EBS, it is the motion of physical objects (or their relationships) to be observed. Therefore, we classify the motions and connect them to the mechanical concepts they suggest.

Table 2. Error-Identification Rules (EIRs)

force	appearance	cause of errors	correcting strategy
external force/ gravity	missing extra error	missing knowledge of gravity (R0)	re-teach the concept/definition
		misunderstanding the problem (r0-c1)	re-show the problem and indicate the corresponding part
tension	missing	missing knowledge of tension (R1)	re-teach the concept/definition
		overlooking the string (r1-c1)	re-show the problem and indicate the corresponding part
		missing the force which causes tension (r1-c2)	proceed to the correcting strategy of that force
	extra	belief that string propagates tension (r1-c2)	re-show the problem and indicate the corresponding part
		overlooking that motion looses string (r1-c2)	indicate that tension is extra
	error	error of the force which causes tension (r1-c2)	proceed to the correcting strategy of that force
		error of direction/magnitude (r1-a2/3)	indicate that direction/magnitude is erroneous
normal force	missing	missing knowledge of normal force (R2)	re-teach the concept/definition
		overlooking the touch/attachment (r2-c1)	re-show the problem and indicate the corresponding part
		missing the force which causes normal force (r2-c2)	proceed to the correcting strategy of that force
	extra	belief that normal force works (r2-c2)	indicate that normal force is extra
		extra of the force which causes normal force (r2-c2)	proceed to the correcting strategy of that force
	error	error of the force which causes normal force (r2-c2)	proceed to the correcting strategy of that force
		error of direction/magnitude (r2-a2/3)	indicate that direction/magnitude is erroneous
friction	missing	missing knowledge of friction (R3)	re-teach the concept/definition
		overlooking the touching together (r3-c1)	re-show the problem and indicate the corresponding part
		overlooking that coefficient of friction $\mu > 0$ (r3-c2)	re-show the problem and indicate the corresponding part
		missing normal force (r3-c3)	proceed to the correcting strategy of normal force
		belief that normal force doesn't work (r3-c4)	indicate that friction is missing
	extra	missing that coefficient of friction $\mu = 0$ (r3-c2)	re-show the problem and indicate the corresponding part
		extra of normal force (r3-c3)	proceed to the correcting strategy of normal force
		belief that normal force works (r3-c4)	indicate that friction is extra
		extra of the force which causes friction (r3-c4)	proceed to the correcting strategy of that force
	error	error of normal force (r3-c3)	proceed to the correcting strategy of normal force
		error of the force which causes friction (r3-c4)	proceed to the correcting strategy of that force
		error of direction/magnitude (r3-a3/4/6)	indicate that direction/magnitude is erroneous
propagating force	missing	missing knowledge of force propagation (R4/5)	re-teach the concept/definition
		overlooking the touching together (r4/5-c1)	re-show the problem and indicate the corresponding part
		missing of the force which causes force propagation (r4/5-c2)	proceed to the correcting strategy of that force
	extra	belief that force propagates (r4/5-c2)	indicate that propagating force is extra
		extra of the force which causes force propagation (r4/5-c2)	proceed to the correcting strategy of that force
	error	error of the force which causes force propagation (r4/5-c2)	proceed to the correcting strategy of that force
		error of direction/magnitude (r4/5-a2/3)	indicate that direction/magnitude is erroneous
other	extra	impetus	indicate that impetus is extra

How does a human perceive and recognize moving objects? Though it is well known that their figurative characteristics (figure, size, texture, etc.) and composition (position, direction, symmetry, etc.) have great influence on the arising images, it is difficult to generalize them because they much depend on the cultural factors. Therefore, we limit our target to the physical world of simulation, in which things are thought in the sense of mechanics.

When observing an object to move, a human feels its motive 'force' working. Of course, this kind of 'force' is of naive impression and doesn't always correspond to the real force. But it appeals to human's intuition so much more. Bliss & Ogborn (1992) classified such naive concepts of force according to the stages of child development. Based on their findings, we consider the relations between the motions in EBS and the forces they suggest.

4.2 Motion of a single object

A moving object arises the feeling of force working. (e.g. A falling down ball suggests gravity.) Therefore, the object moving unnaturally in EBS is supposed to suggest the erroneous force acting on it. (e.g. gravity, friction etc.) Unnatural motions of a single object are classified as follows:

(a) Directions of both velocity and acceleration are opposite to the ones of correct motion.
(b) Direction of only velocity is opposite to the one of correct motion.
(c) Direction of only acceleration is opposite to the ones of correct motion.
(d) Directions of both velocity and acceleration are same as the ones of correct motion.

Here, it is assumued that human can distinguish at most the qualitative difference of velosity or acceleration of an object in motion [Hirashima et al. 98, Horiguchi et al. 99].

For example, in case (a), when a learner observes an object moving in the opposite direction to her/his prediction (which is correct), she/he will recognize that the force is missing which acts in the predicted direction, or that the force is extra which acts in the present direction.

Table 3 shows the relations between unnatural motions and the errors they suggest. They are called Criteria for Cause-of-Error Visualization (CCEVs).

4.3 Relative Motion of two objects

Moving plural objects also arises the feeling of force working. We limit to two objects. When observing two objects moving together, the force maintaining their relative motion is felt. (e.g. A moving dolly pulling another one connected by string suggests tension.) Therefore, two objects relatively moving in unnatural manner in EBS are supposed to suggest the erroneous force interacting between them. (e.g. tention, normal force etc.) Unnatural relative motions of two objects are classified as follows:

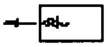
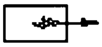
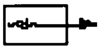
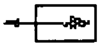
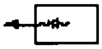
(e) Two objects are closing with each other, which are connected by string. (String shrinks.)
(f) Two objects are going away from each other, which are connected by string. (String stretchs.)
(g) Two objects are overlapping each other.
(h) Two objects are parting from each other, which are attached together.

For example, in case (g), when a learner observes such unnatural relative motion, she/he will recognize that the normal force is missing or too small which interacts between two objects.

Table 4 shows the relations between unnatural relative motions and their suggesting errors. They are also called Criteria for Cause-of-Error Visualization (CCEVs).

Note that, all of the motions in Table 3 and 4 have at least some kinds of qualitative difference from the correct motions. This is because, however precisely an EBS indicates the error, it isn't effective unless a learner recognizes it as 'unnatural.' The difference is judged with Criteria for Error-Visualization (CEVs) [Hirashima et al. 98, Horiguchi et al. 99].

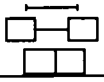
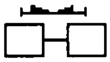
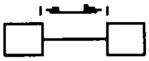
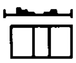
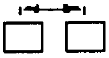
Table 3. Criteria for Cause-of-Error Visualization (CCEVs) (for single object)

motion		difference	suggesting errors
correct motion		–	–
motion (a)		velocity. opposite acceleration. opposite	missing of the force opposite to moving direction ⊙ extra of the force same as moving direction⊙ larger of the force same as moving direction○ smaller of the force opposite to moving direction○
motion (b)		velocity. opposite acceleration. same	missing of the force same as moving direction△ missing of the force opposite to moving direction ⊙ extra of the force same as moving direction⊙ extra of the force opposite to moving direction△ larger of the force same as moving direction○ smaller of the force opposite to moving direction○
motion (c)		velocity. same acceleration. opposite	missing of the force same as moving direction⊙ extra of the force opposite to moving direction⊙ smaller of the force same as moving direction○ larger of the force opposite to moving direction○
motion (d)		velocity. same acceleration. same	missing of the force opposite to moving direction △ extra of the force same as moving direction△

note 1. ⊙ . able to suggest the error by itself with great effect
○ . able to suggest the error by itself with small effect
△ . need to be modified some parameter(s) to suggest the error

note 2. The error of force in direction is divided into the missing of the force of correct direction and the extra of the force of incorrect direction.

Table 4. Criteria for Cause-of-Error Visualization (CCEVs) (for two objects)

motion		unnaturalness	suggesting errors
correct motion		constant distance	–
motion (e)		closing spring shrinks	extra/larger of the tension⊙ extra/larger of the propagating forc○
motion (f)		going away spring stretches	missing/smaller of the tension ⊙ missing/smaller of the propagating force○
motion (g)		overlapping	missing/smaller of the normal force⊙ extra/larger of the normal forc○ missing/smaller of the propagating force○ extra/larger of the propagating forc○
motion (h)		parting from each other	missing/smaller of the normal force○ extra/larger of the normal forc⊙ missing/smaller of the propagating force○ extra/larger of the propagating forc○

note 1. ⊙ . able to suggest the error by itself with great effect
○ . able to suggest the error by itself with small effect

note 2. The error of force in direction is divided into the missing of the force of correct direction and the extra of the force of incorrect direction.

5 Examples

In this chapter, we illustrate the process of identifying the cause of error and generating the EBS which indicates the error. The example problem is shown in Figure 2.

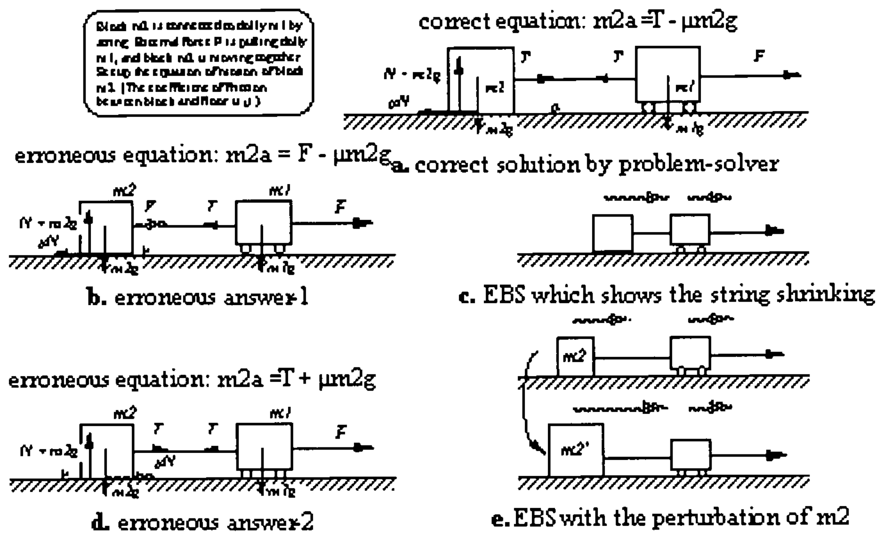
5.1 A Simple Case

First, the solution (correct equation and diagram: Figure 2a) is generated by problem-solver. Then, it is compared with a learner's answer (Figure 2b) to specify the erroneous part. In this case, it is the erroneous value (too large) of tension beside block m2. Secondly, EIRs (in Table 2) are applied to identify the cause of error. It is identified as the error of magnitude of tension. According to Table 2, the correcting strategy of this error is to indicate the fact. Then, CCEVs (in Table 3 and 4) are applied, to find that the motion (g) satisfies this demand.

Based on the erroneous equation of Figure 2b, the EBS shown in Figure 2c can be generated, in which block m2 moves faster than its normal case, consequently the string shrinks. This unnaturalness is equal to the one of motion (g). Therefore, this EBS is judged to satisfy the instructional demand, and shown to the learner.

5.2 A Complicated Case

Consider the erroneous answer of a learner in Figure 2d. In this case, the erroneous part is the erroneous direction of friction acting on block m2. By EIRs, the cause of error is identified as the error of direction of friction, and the correcting strategy is to indicate the fact. Since the error of force in direction is divided into the missing of the force of correct direction and the extra of the force of incorrect direction (see note 2 of Table 3), the motions (a), (b), (d) satisfy this demand.


Block m2 is connected to dolly m1 by string. External force F is pulling dolly m1, and block m2 is moving together. Set up the equation of motion of block m2. (The coefficient of friction between block and floor is μ.)
correct equation: m2a =T - μm2g
N - m2g
μN
m2
T
T
m1
F
m2g
m1g
erroneous equation: m2a = F - μm2g
m2
m1
N - m2g
μN
F
T
F
m2g
m1g
erroneous equation: m2a =T + μm2g
m2
m1
N - m2g
T
T
μN
F
m2g
m1g
m2
m2'

a. correct solution by problem-solver

b. erroneous answer1

c. EBS which shows the string shrinking

d. erroneous answer2

e. EBS with the perturbation of m2

Figure 2. Example Problem-2

Based on the erroneous equation of Figure 2d, however, it is impossible to generate the EBS which contains the motion (a) or (b). In addition, even when the EBS containing the motion (d) is generated (it is possible), it causes the unnatural relative motion (e), which indicates another error. In fact, the EBS, in which block m2 is closing to dolly m1 (the same as Figure 2c), strongly suggests the error of tension. This misleads a learner.

Therefore, in this case, the EBS must be modified to precisely indicate the identified error. Perturbing the mass of block m2 is a promising method. When the mass m2 increases, in EBS, the velocity of the block increases (Figure 2e). This is a strange change of motion. Observing this, a learner may think some physical amount is wrong which concerns the mass m2. She/he may notice the erroneous friction acting on block m2.

As for the EBS of Figure 2e, the difference from the correct simulation is not so much clear and reliable as the EBS of Figure 2c. Instead, it provides precise information for correcting the error, while the EBS of Figure 2c doesn't. In general, plural EBSs can be generated from one erroneous equation. The best should be chosen according to the purpose.

6 Concluding Remarks

In this paper, we proposed a method of creating effective counterexamples by using Error-Based Simulation. The effectiveness of EBS is judged mainly from the viewpoint whether it provides sufficient information to recognize the cause of error and correct it. The mechanism for identifying the cause of error and for

generating the EBS which satisfies the instructional demand was also proposed. We are now implementing the mechanism. The experiment to evaluate our method is planned.
Our future works are as follows:

1. *Cooperation with other instructional tools:* Of course, EBS isn't sufficient for all of the error correction in Table 2. It must be studied to use other instructional tools (textbook, normal simulator etc.), and to coordinate them with EBS.
2. *Refinement of the problem-solving model:* Our model for problem-solving is very simple, so the range of the error it covers is limited. We are going to refine the model, especially considering the process in which a learner qualitatively predicts the motion of mechanical system.
3. Consideration of conflict among CCEVs: As is noted in section 5.2, the effects of plural unnatural motions sometimes conflicts each other. One unnatural motion may invalidate the effect of other unnatural motion. Therefore, it is necessary to set some kind of preferences to CCEVs.

References

[1] Gagne, E.D. (1985). The Cognitive Psychology of School Learning, Little Brown and Company.

[2] Fujii, T. (1997). Mathematical Learning and Cognitive Conflict, In Japan Society of Mathematical Education (Ed.), Rethinking Lesson Organization in School Mathematics, Sangyotosho, Tokyo (in Japanese).

[3] Fukuoka, T. & Suzuki, K. (1994). A Case Study on Lower Secondary School Pupils' Conceptual Changes through Demonstration of Counterexamples: The Concept of Dynamics in the Case of Roller Coaster Models, Bulletin of Society of Japan Science Teaching, Vol.35, No.2, pp.21-32 (in Japanese).

[4] Nakajima, N. (1997). The Role of Counterevidence in Rule Revision: The effects of instructing metaknowledge concerning non adhocness of theory, Japanese Jour. of Educational Psychology, 45, pp.263-273 (in Japanese).

[5] Hirashima, T., Horiguchi, T., Kashihara, A. & Toyoda, J. (1998). Error-Visualization by Error-Based Simulation and Its Management, Int. Jour. of Artificial Intelligence in Education, Vol.9, pp.17-31.

[6] Horiguchi, T., Hirashima, T., Kashihara, A. & Toyoda, J. (1999). Error-Visualization by Error-Based Simulation Considering Its Effectiveness: Introducing Two Viewpoints, Proc. of AI-ED99, pp.421-428.

[7] Robertson, W.C. (1990). Detection of Cognitive Structure with Protocol Data: Predicting Performance on Physics Transfer Problems, Cognitive Science, 14, pp.253-280.

[8] Plötzner, R. (1994). The Integrative Use of Qualitative and Quantitative Knowledge in Physics Problem Solving, Frankfurt: Peter Lang Verlag.

[9] Takeuchi, A. & Otsuki, S. (1997). An Intelligent Tutor for Kinetic System Modeling, Proc. of AI-ED97, pp.340-346.

[10] Bliss, J. & Ogborn, J. (1992). Steps Towards the Formalisation of a Psycho-logic of Motion, In A. Tiberghien & H. Mandl (Eds.), Intelligent Learning Environments and Knowledge Acquisition in Physics, Heidelberg: Springer-Verlag.

A Study of Networked Constructive CAI System Using Multiplication-Concept of "Transformation of Unity Quantity" on Elementary School

Pou-Yah Wu*, Shu-Fen Tsai, Yao-Quan Li, Jian-Zhi Lin, Jun-Rong Chen, and Xiang -Ting Huang

Department of Mathematics Education, National Tainan Teachers' College
33, Sec. 2, Su-Lin Street, Tainan, Taiwan 700
**Pou-Yah Wu's email:pywu@ipx.ntntc.edu.tw*

The feature of networked constructive CAI system lies in shaping the computer environment in which students clarify and construct the concept by ways of communication, discussion, and dialectics, utilizing the practical pedagogic content edited by the spirit of new curriculum in Taiwan at 1993. Because we stress the concept of "transformation of unity quantity" as main activity in teaching multiplication, students' comprehension of "unity quantity", "unity number", and "combined numbers" plays an important role in establishing networked constructive CAI system. We consider that the greatest difference between the networked learning environment and that of the general classroom pedagogy is the deficiency of interaction. Thus, analyzing the strategy of students' solving problems to establish the effective tool table of operation and judging the mode of the students' thought by checking the tools which students use will strengthen the interactive relationship of the system and the learners. Then, use the networked technology and the principle of the expert system to set up the CAI of constructive pedagogy, so that the learners can communicate with each other and the system can conduct dynamically which formally construct a wholly co-operative learning environment and will help the learners to form the whole mathematics concepts.

Keywords: **Constructive pedagogy, Elementary School, Multiplication of Mathematics, Networked CAI.**

1 Introdution

The characteristic of implementing new curriculum of mathematics at elementary schools in Taiwan now lies in the addition of spirit of constructivism, aiming at expecting students to construct knowledge positively. Thus, the teachers' role, in the process of pedagogy in new curriculums, changes into "problem poser", whereas students' learning activities in class attain socialized mutual sense, chiefly by communication, construct their own mathematical knowledge by way of mutual dialectics [5,6]. However, it takes pedagogy of construction longer than traditional pedagogy. The atmosphere and skill as to how the teacher directs students to construct knowledge and how the students discuss influence the effect of implementing new curriculum. In the light of the fact the trend of the times facilitates pedagogy of network to become widespread, the future construction of leased network lets us expect the popularity of "learning at home" and "long distance pedagogy". Therefore, the possibility of displaying a really approximate leaning setting of constructing pedagogy in the network environment becomes much higher. The establishment of the network system of pedagogy of construction, owing to the trend of current situation, is becoming imminent.

The aim of this study consists in designing a learning environment of network suitable for "multiplication concept in elementary school" . The greatest difference between the learning environment of network and that of the general classroom is the deficiency of mutual response [22,26]. And the pedagogy of construction hopes the communication and dialectics to bring about reflection, inspiring students to construct mathematics self-concept. Therefore, how to promote the mutual relationships between the system and the user is one of the considered points about constructing system in this study. Furthermore, how to develop the characteristic of pedagogy of construction in the system and how to make the pedagogic contents of the new curriculum manifested in the system wholly and fluently is the second chief point taken into account. Aimed at the above two points, that we use network technology, letting the real-time communication proceeded between the learners, or between the learning and system make up a wholly cooperative learning environment. Furthermore, making use of the principles of the expert system to deal with the learning strategy of the problem solver, through the concepts manifested by the problem solver, the system will feedback suitably, and will communicate with the students properly, which can make the pedagogic activity proceed dynamically [19,25]. The design of the pedagogic content, expect considering the sprit of the new curriculums, the students' learning state, after the teachers' real pedagogy, is mainly considered about designing pedagogy. Hence, this system is much closer to the real situation of pedagogy them CAI sold in the market. And the activities of problem solving given to the learner by system would be more congenial to the learner's mode of thought.

2 Principles of system constructing

2.1 Base of learning theory

"Knowledge is positively constructed by the learner rather than being inculcated passively from outside," which is the fundamental proposition of constructing pedagogic paradigm. The students, with acquired knowledge, enter another stage as an active subject of recognition, with good theory by themselves, instead of ignorance and irrationality [16]. But pedagogy of construction does not mean the teacher's role is unnecessary. On the contrary, we realize the aim of pedagogy is to make children construct the activity types of solving problems. In the light of this, the teachers' role becomes "problem poser" rather than "problem solver" in the process of pedagogy. By way of the teachers' posing problems, children undertake the activity of solving problems by themselves; or children become "imitators" through the activity of solving problem provided by the teacher [4]. By these processes, students are provided sufficient experience of solving problems, and then construct the correct mathematics conceptions. Besides, what we must also pay attention to is the teacher and the learner grasp the intentions of each other aiming at the proceeding actives of each other, through trial and dialectics, until both of them relieve the pressure aroused by the interchange actives. The relief of pressure is limited by the fact if the problem is solved according to the activity, and is also influenced by the affectionate expression of them both present of them both present [24]. Therefore, in pedagogy of construction, socialized communication is an important feature [3].

2.2 Base of system establishing

This system is a learning environment constructed in the network, adopting three-tier client/server system architecture: that is, adding another service server on the original framework of the two-tier client/server system in this three-tier client/server system architecture, the management of Database Server charges learning data. Web Server is responsible for teaching, whereas the user of Client precedes all kinds of learning activities ivies through browser machine.

3 Pedagogic design of Multiplication using transformation of unity quantity

3.1 Concept of multiplication

Multiplication referred to by Davydov (1991) is the problem of transformation of unity quantity, that is, the transformation from composite unit to that of the single item [20]. And Clark and Kamii (1997) think that if children own the multiplicative thinking, they will simultaneously deal with lower level unit such as unit of one and the higher level unit different from unit of one [18]. Tzyh-Chiang Ning (1994) mentions that the so-called multiplication operation contains at least two kinds of relationship: (1)the coordinating relationship of two levels,(2)the part-whole relationship of two levels. The problem of multiplication is in reality that of the transformation of unity quantity, namely, the problem of transforming quantity from higher level unit to lower level unit [7,8,9].

Adaptive Programming Language Tutoring System on the Web

Jin-woo Choi and Chong-woo Woo
Department of Computer Science, Kookmin University
861-1 Chongnung-Dong, Sungbuk-ku, Seoul, Korea
Tel: +82-2-910-4795
Fax: +82-2-910-4799
Email: cwwoo@kmu.kookmin.ac.kr

Many of the web-based educational systems could not provide an individualized instruction or an interactive problem solving, since they are mostly built upon static hypertext. One possible approach to solve these problems could be adapting the existing proven techniques from the stand-alone Intelligent Tutoring System(ITS). Some recent web-based ITS researches show this efforts by employing the techniques selectively, and this needs to be studied further to support more effective web-based instruction. In this paper, we describe the design and the development of a Web based Adaptive programming Language Tutoring System(WALTS). The system is designed based on the ITS structure primarily, and it is adapting previous ITS techniques into the system successfully. Especially our focus is on the three levels of the instructional planning mechanism, which can generate lesson contents dynamically whenever it is requested. This way we do not need to crate all the lesson contents in HTML forms which must reside in the system in advance. In addition, the system has adapted CORBA structure to support the user more consistent and reliable performance. Together, the system behaves more adaptive and interactive, than the existing non-ITS based web systems. The test domain of the system is learning C programming language for the first year computer science student.

Keywords: **Web-based learning system, Intelligent Tutoring System, Instructional planning**

1 Introduction

Many recent web-based educational systems could not provide an individualized instruction or an interactive problem solving, since they are mostly built upon static hypertext. One possible approach to solve these problems could be adapting the existing techniques from the stand-alone ITS. Brusilovsky[2] states that some ITS techniques can be adapted into a web-based educational system, and divides the techniques into three distinctive parts, such as, automatic creation of curriculum, dynamic problem solving, and intelligent analysis of student model. However, most of the recent web-based ITS research show the efforts by employing the techniques selectively[2][3][4][7], such as adapting student modeling or problem solving capability at some level. Therefore, this needs to be studied further to enhance the overall capabilities of the system at the previous stand-alone ITS level. For instance, automatic generation of curriculum or lesson plan is necessary to provide a flexible instruction for each individual user.

In this paper, we describe the design and development of Web-based Adaptive programming Language Tutoring System(WALTS). The system is designed based on the ITS structure primarily, and it is adapting many techniques from the stand-alone ITS into thc wcb-bascd systems. First, we designed the knowledge base using the object-oriented method in order to handle flexible management of object inheritance and tutorial strategies. Second, the student modeler can avoid the network traffic by designing the modeler stays in the server-side at the beginning of the session for maintaining necessary administration duties, and creates an individual student model in the client side. And the third important approach is having the instructional planning mechanism, which generates lesson contents dynamically for each individual user. This is important feature for moving towards the web-based system, because most web-based educational systems creates all the lesson contents in HTML forms in advance, and they must reside in the system physically. And then the user navigates the system for learning, such as in ELM-ART[2] or CALAT[4]. Intelligent navigation strategy might be one of the intelligent way of guiding the user to learn the material, but rather inefficient compare to the dynamic generation of lesson contents. WALTS only generates the necessary lesson contents whenever it is requested by the system, which can be another advantage. In addition, we have approached distributed architecture by employing CORBA(Common Object Request Broker Architecture) structure to support the user more consistent and reliable performance while the user using the system. The initial web-based educational systems are mostly developed by using the CGI(Common Gateway Interface) techniques, which often results in bottleneck problem when many users access the system at the same time. In this sense, our structure might avoid such a problem, and the system could also be easily updated when we need to revise some part of the system. Together, the system behaves more adaptive and interactive, than the existing non-ITS web-based educational systems. The test domain of the system is learning C programming language for the first year computer science student.

The rest of the paper is organized as follows. In section 2 we described a distributed infrastructure of WALTS system. Section 3 presents each components of the system and also some intelligent aspects of the system. We conclude the paper in section 4.

2 Distributed infrastructure

The previous web-based educational systems have been built as either a server-based architecture or a client-based architecture[6]. Each of them has some advantages and disadvantages. The server-side architecture mostly rely on CGI techniques, which has shown some problems of handling complex client/server communication because of its connectionless feature. Also client-side architecture needs to have all the plug-ins installed on client computer before using the system. Therefore the recent web-based applications tend to adapt CORBA or Java based distributed infrastructure. That is free from the connectionless or stateless problem, and also has some advantages of distributed system technology, such as message passing, RPC(Remote Procedure Call), and proprietary communication protocol. The client connects to the server using the HTTP protocol only for the initial connection, and after the downloading the specific mobile code application(for example, client side application, JavaScript, Java Applet, and etc), the client use the proprietary protocol(non-HTTP), so it does not communicates with web server, but communicates with proprietary server(non-Web server).

WALTS employed CORBA to adapt this kind of distributed infrastructure. The system is designed by HTTP server which takes care of user requests and responses, and CORBA-based server which performs the capabilities of the ITS. Also the system could be easily re-organized if we want to modify the structure later on [see figure 1]. In short, one of the major advantages of WALTS is that it can easily avoid the bottle-neck problem of CGI techniques, and also we believe that this style of architecture might be another best solution for building web-based client/server educational system.

3 Basic architecture of the system

The basic architecture of WALTS is designed by typical ITS structure primarily, including expert module, the student modeler module, and the instructional planning module.

3.1 The expert module

The expert module of the system consists of the object-oriented knowledge base, and the problem solver.

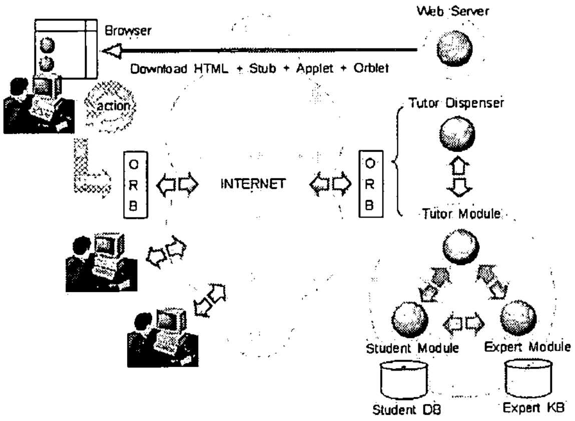


Figure1 The main architecture of the WALTS

Object Oriented-based knowledge base. First, we employed the frame knowledge representation techniques for the main knowledge base. Because the domain knowledge does not require any complex causal relationships, but rather it is consists of simple C language concepts. The object-oriented approach make it easy to modify the data type, can reduce the knowledge base reference by having slot values as member data, and can provide more flexibility for updating or manipulating tutoring strategy [5].

In this system we designed a frame with several meaningful slots, and each frame does not have to have the same number of slots, since the inference engine can get all the necessary informations due to the inheritance feature of the system. The 'type' slot can possess a concept, example, or quiz. The 'source' slot points to its superior frames. The 'Pfame' and 'CFrame' slot is necessary when we need to show the related nodes in linked list structure. The 'reference' slot may contain all the necessary frame names that are related to the current frame. This kind of slot structure is very common in every frame structure, and also important in object-oriented structure, because each frame can have common attributes and can generate an object of having its own attribute. Also, the system allows an abstract class, which plays the backbone of the system, and supports a hierarchical structure, and the definition of the method can be done only in the lower class [figure 2].

```
Frame Variable Declaration Quiz
                [Source] Chapter1-3-1-1
                [Type] Quiz
                [Title] Variable declaration Quiz
                [Template] Data Type | Variable | General Grammar
                        1 : Select the correct %type variable declaration
                        2 : ...
                [PFrame] Variable Declaration
                [CFrame] Null
```

Figure2 Variable declaration quiz frame

The Problem Solver. WALTS can generate a problem dynamically depending on the current topic. Since the planner knows what is being taught at the moment by generating a lesson unit, the tutor can decides whether it is 'teaching concept' or 'show example' or 'quiz'. At the moment, we have only three styles of lesson unit. If it is a 'teaching concept', the planner sends the lesson unit to the user in HTML form. If it is a 'quiz' type, then planner requests the problem solver to generate a question. The problem solver first creates a problem table by referring to the current lesson unit. The generating and solving a problem occurs at the same time, and the solver stores the correct answer. And then, it presents the generated questions to the user in appropriate HTML form through the HTML generator. This method can provide different styles of questions for different users even though they are accessing the same lesson unit, which can be another advantage of WALTS. Since the column name of the table is object's name, the planner can reply to the user's request, such as hint or help, by referencing this table.

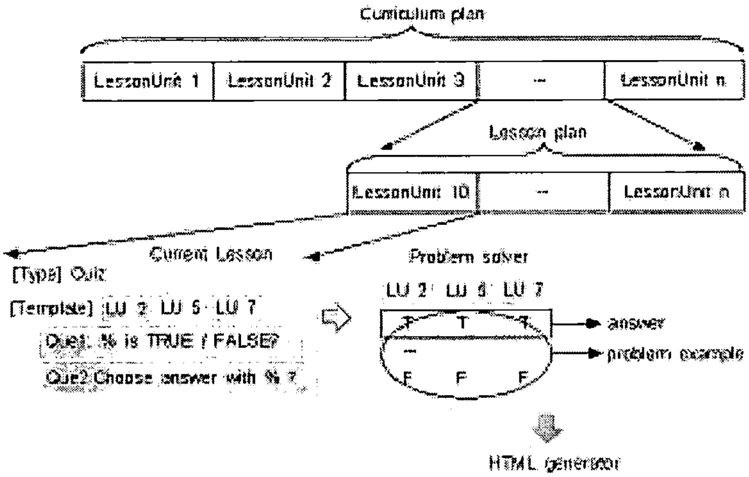


Figure3 Creation of a quiz for 'variable declaration'

The strategy of asking user for answering quiz is multiple choices. So that we need to generate problems along with the appropriate multiple choice answers also. For instance, let us think about a simple quiz about asking user 'a data type'. A typical 'data type' is consists of three parts, for example, 'int x ;'.The 'int' is a data type integer, 'x' is a user-defined variable name, and ';' is needed for ending a sentence in C language. We are trying to generate this simple data type declaration statement sentence as follows. First, the data type 'template' slot consists of three parts as in [Figure 3]. Then we can generate eight different answers as in [Figure 5], since each one part of a statement can be correct or incorrect. And we can select some of them randomly including correct answer; the numbered answers are selected ones in the figure. And also we can obtain designated unit object's content as in [Figure 4]. The generated correct answer is stored in memory, and then later it is compared with the user's answer. For example, if the user selected number 2 as in [figure 5], we can analyze that the user does not know about reserved word. And the planner needs to revise the lesson plan to correct the misconceptions by giving special messages, such as hint or help, and then the planner re-organizes the lesson plan including 'reserved word' lesson unit. The [figure 6] shows a sample session of solving a generated quiz.

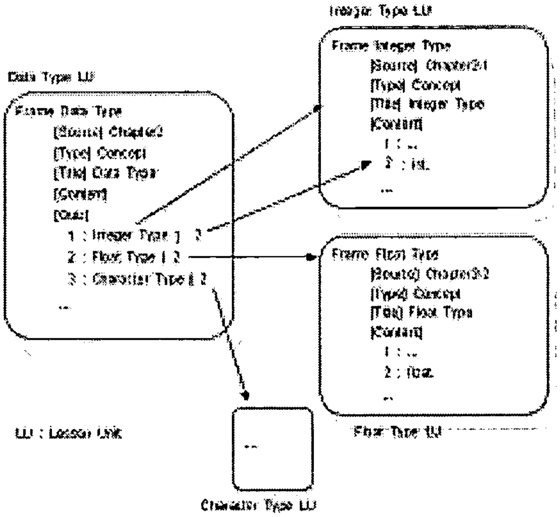


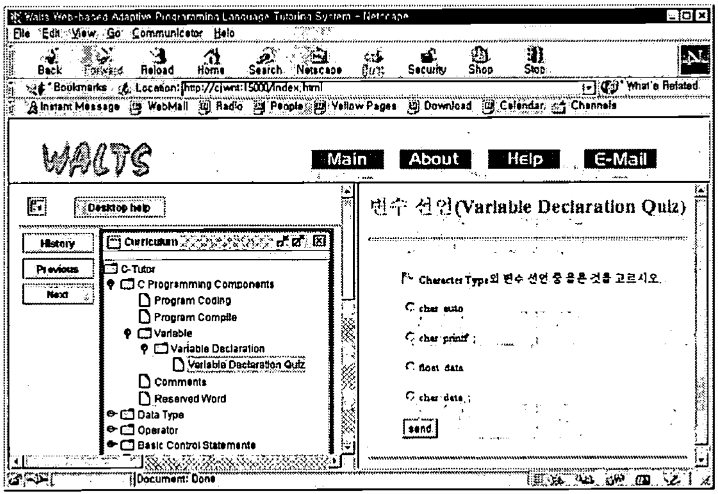
Figure4 Data type unit frame

Data Type	Variable	General Grammar	
T	T	T	④
T	T	F	
T	F	T	③
T	F	F	①
F	T	T	
F	T	F	⑤
F	F	T	
F	F	F	

Figure5 Table for possible answers for data type quiz

3.2 The Instructional Planner

The most web-based educational systems built upon hypertext, which is hard to make hyperlink in every HTML pages, and also needs to have carefully designed navigation strategy[2]. And also all the lesson contents are built as HTML pages in advance, and must reside in the system physically. We believe that generating a lesson plan dynamically, for each individual user, is more efficient than the above approaches. Therefore, we adapted the traditional ITS instructional planning mechanism into the system. The instructional planning of the WALTS can be further divided into 3 steps, a curriculum planning, a lesson planning, and a delivery planning. The curriculum planning of WALTS generates a curriculum in tree structure; the curriculum planner extracts information from the knowledge base and creates a curriculum hierarchically in the order of prerequisites. Then the lesson planning sets up the lesson sequence within a single lesson unit. The role of delivery planning is limited to presenting the selected lesson content to the user.


Walts Web-based Adaptive Programming Language Tutoring System - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://cjwnt:15000/index.html
What's Related
Instant Message WebMail Radio People Yellow Pages Download Calendar Channels
WALTS
Main
About
Help
E-Mail
Desktop help
History
Previous
Next
Curriculum
C-Tutor
C Programming Components
Program Coding
Program Compile
Variable
Variable Declaration
Variable Declaration Quiz
Comments
Reserved Word
Data Type
Operator
Basic Control Statements
send
Document: Done

변수 선언(Variable Declaration Quiz)

Character Type의 변수 선언 중 옳은 것을 고르시오.

○ char auto

○ char printf ;

○ float data

○ char data ;

Figure6 An example of solving a generated quiz

Curriculum planning. The purpose of the curriculum planning is to provide a curriculum to the user, in other words, to provide an individualized optimal learning path to the user[1]. The generated curriculum is in the form of a tree structure. It is constructed by creating an initial node by referencing the value of the attributes in the lesson unit slot, and further expands the structure in the order of the way the student must learn, which will be accessed as linked list structure. The lesson unit of the system is organized according to some basic rule, such as the student must learn prerequisite concept first and the move to the next topic. So the curriculum is set up in the form of hierarchical and linear sequence.

Lesson Planning. The lesson planner generates a lesson plan by referencing the curriculum and the student model. The information from the student model shows the results from single lesson unit and based on this record, the planner sets up appropriate lesson plan for the student. When the student selects other learning path on purpose before the current lesson plan is finished, the system must decide what to do next, such as whether to store the current lesson plan and execute the user's request, and then resume the current plan or destroy the current plan and re-plan the whole sequence all over again. In that sense, WALTS uses re-planning strategy when the user wants to quit the current topic, and move to another learning path. Another case of re-planning occurs when the student made an error on the selected quiz lesson unit. If the student made a mistake on this, the current lesson plan is suspended, and another new lesson plan is created to correct the student's error. After the remediation process is finished, the suspended plan will be resumed.

Delivery Planning. The lesson unit has been generated by lesson planner and needs to be delivered to the user. The possible delivery tactic in this domain could be "present concept, show example, give exercise, and etc". Of course if the system allows mixed-initiative control, the delivery planning needs to be more sophisticated in order to handle all the user request or questions. The delivery planning part of the WALTS is made of simple structure, and will be enhanced further in the next research.

The HTML generator. The very distinctive feature of the system is the HTML generator. This feature can be regarded as the interface part of the system. When the delivery planner decides the immediate unit lesson, the content of the lesson is converted into HTML form by the HTML generator. The HTML generator generates HTML pages according to the HTML2.0 protocol and inserts "next" or "previous" button in order to navigate adaptive learning path. But if the lesson unit contains some applet, the system directly searches the physical location and sends the URL to the student's browser without consulting HTML Generator. The [figure 7] describes the HTML generator sends two different results to two different users, since their learning background is different.

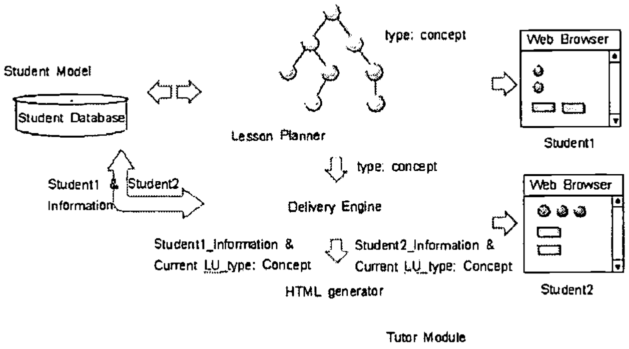
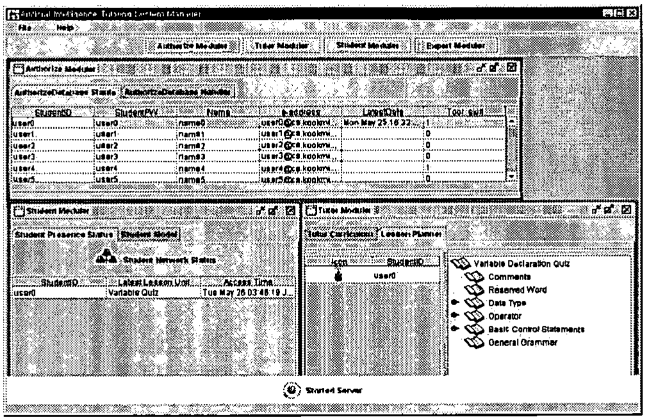


Figure7 The HTML generator

3.3 The student modeler

The strategy for building the student modeler is the simple overlay, which simply reflects user's learning process about current topic. And this should be enhanced by including the buggy information later on. But an important enhancement is that the student modeler of WALTS can avoid the unnecessary network traffic. For instance, if the system maintains the student model in the server-side, then whenever the user accesses the system the server needs to update the user's student model in the server. This may cause another bottleneck problem, and the most CGI-based systems still have this problem. Our approach on the student modeler is as follows. The server-side student modeler crcatcs a table, and keeps all the necessary administrative informations on the server-side, such as initial student's ID, password, e-mail address, the access time[figure 8], which can be used for various administrator purposes. And the information regarding the student's learning process is stored in the student model[figure9], which is created in the client-side machine for each individual user whenever they logged on. The student model has several parameters that reflect the student's learning history, and each parameter has unique meanings. For example, the 'HelpCount', means how many times the user has been helped, and 'HintCount' means how many times the user has requested hints, and they can be updated only when the 'unit lesson' is quiz. The 'ReferenceCount' means the user is weak at the current unit lesson since the specific lesson has been accessed more often than other frames. The 'LessonLevel' stores information about how the level of the current topic, and the 'LessonType' means whether the current unit lesson is concept, cxample, or quiz, and so on.

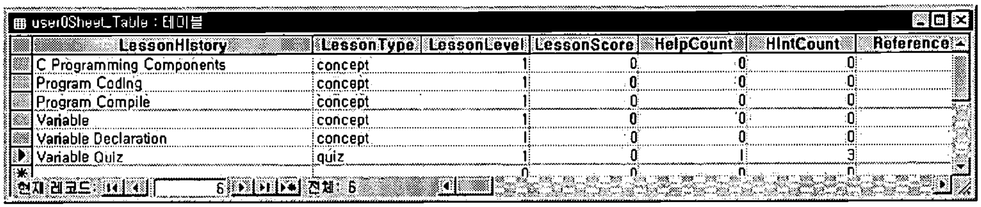


StudentID	StudentPW	Name	e-address	LatestDate	Tool skill
user0	user0	name0	user0@cs.kookmi...	Mon May 25 16:22:...	1
user1	user1	name1	user1@cs.kookmi...		0
user2	user2	name2	user2@cs.kookmi...		0
user3	user3	name3	user3@cs.kookmi...		0
user4	user4	name4	user4@cs.kookmi...		1
user5	user5	name5	user5@cs.kookmi...		0

StudentID	Latest Lesson Unit	Access Time
user0	Variable Quiz	Tue May 26 03:46:19 J...

Icon	StudentID
	user0

Figure8 The Server-side student modeler table



LessonHistory	LessonType	LessonLevel	LessonScore	HelpCount	HintCount	Reference
C Programming Components	concept	1	0	0	0	
Program Coding	concept	1	0	0	0	
Program Compile	concept	1	0	0	0	
Variable	concept	1	0	0	0	
Variable Declaration	concept	1	0	0	0	
Variable Quiz	quiz	1	0	1	3	
		0	0	0	0	

Figure9 The student model

4 Conclusion

We have designed and implemented a web-based ITS, WALTS, which is a learning C programming language tutor aiming for the first year computer science students. The main goal of this paper is, first, the adaptation of the existing ITS techniques into the web platform. Therefore, we have designed and implemented the system based on the major ITS architecture, and this brings us several advantages over traditional HTML-based educational systems. First, the main knowledge base is created as an object-oriented concept, which can provide more flexibility for manipulating frame objects and tutoring strategy also. Second, we have generated a quiz dynamically by the problem solver and also can solve the problem. Third, we designed a student modeler that can avoid the network traffic in the minimal, by having the modeler in the server-side, and creates an individual student model in the client-side. Fourth, the instructional planner can generate an instructional plan dynamically, and this is another advancement of building web-based ITS, since the current web-based ITS research shows further work on this subject. Additional issue of the paper is that we designed the system as the distributed infrastructure using CORBA as backbone of the system. This structure solves the bottleneck problem of previous CGI dependent systems, and also gives some benefits of better performance and also gives flexibility in the case of further enhancement of the system.

References

[1] P. Brusilovsky. "Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies" http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/brusilovsky.html

[2] P. Brusilovsky, E. Schwarz, and G. Weber. "ELM-ART: An Intelligent Tutoring System on the WWW." The Proceedings of ITS'96, pp261-269, Springer, (1996).

[3] J. Kay and R.J. Kummerfeld "An Individualized Course for the C Programming Language." The Proceedings of 2nd International WWW Conference, Chicago, IL, pp17-20, (1994).

[4] K. Nakabayashi, M. Maruyama, Y. Koike, Y. Kato, H. Touhei, and Y. Fukuhara. "Architecture of an Intelligent Tutoring System on the WWW." The Proceedings of AI-ED'97, pp39-46, (1997).

[5] K. Nakabayashi, H. Touhei, S. Hitoshi, and Y. Fukuhara. “An Object-Oriented Architecture for a Web-based CAI System”, The Proceedings of EDMEDIA'98, (1998).

[6] A. Silva, M. Silva, and J. Delgado. “A Survey of Web information Systems.” The Proceedings of WebNet’97, pp520-530, (1997).

[7] M. Stern, B. Woolf, and J. Kurose, "Intelligence on the Web?," The Proceedings of AI-ED'97, pp490-497, (1997).

An Agent-Based Intelligent Tutoring System

C.M. Bruff and M-A. Williams
University of Newcastle,
NSW 2308, Australia
Email: cgcmb@cc.newcastle.edu.au *and* maryanne@eBusiness.newcastle.edu.au

In this paper we describe the architecture of an agent-based intelligent tutoring system. The agent architecture is based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using an agent oriented programming language called JACK [1]. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to embed previously developed Java modules in an agent environment. In essence our intelligent tutoring system builds and maintains a student model in a dynamic learning environment where new, possibly inconsistent or uncertain, information is obtained through interactions with the student, and where the system may not have complete knowledge when deciding on the next instructional step. Our architecture supports the development of highly individualised student models using techniques in belief revision, nonmonotonic reasoning and possibility theory.

Keywords: **Educational Agent, Intelligent Tutoring Systems, Artificial Intelligence in Education, Belief Revision**

1 Introduction

In this paper we describe the architecture of an agent-based intelligent tutoring system. The agent architecture is based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using in an agent-oriented programming language called JACK. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to embed previously developed Java modules in an agent environment. In essence our intelligent tutoring system builds and maintains a student model in a dynamic learning environment where new, possibly inconsistent or uncertain, information is obtained through interactions with the student, and where the system may not have complete knowledge when deciding on the next instructional step. Our architecture supports the development of highly individualised student models using techniques in belief revision, nonmonotonic reasoning and possibility theory. This architecture was described in a previous paper [5].

2 Using Intelligent Agents

Agents vary in capability from procedural wizards to information agents which are used for information filtering and retrieval. Herein the term agent is taken to mean: *an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives,* as suggested by Wooldridge [29]. By autonomous we mean agents have control over both their internal state and their behaviour, in other words they can make choices regarding their actions depending on their internal state and the goal they seek to achieve. Objects in the object-

[1] JACK was developed at Agent-Oriented Software (http://agent-oriented.com)

oriented paradigm do not have this capability. Agents can be both reactive or proactive, in other words, they can react to external events and they can pursue goals. Agents can make run-time decisions that were not foreseen at design time.

It has been consistently argued, for example by Wooldridge [29], that an agent-oriented approach to problem solving and software engineering offers substantial benefits for complex systems. In particular, Jennings [14] has argued that the usual tools identified by Booch [4] of problem decomposition, abstraction and organisation acquire more power if an agent-oriented approach is adopted, because the agent-oriented approach supports distributed processing and at the same time reduces the system' s control complexity. Decisions about the next action to be performed are devolved to the agents, and this obviates the need for a global controlling module and as a consequence gives rise to more flexibility and better performance.

3 Implementing Intelligent Agents in JACK

JACK is an agent programming language, which in essence provides agent infrastructure to the Java programming language. Agents are designed in JACK and compiled into standard Java code before being executed. Agent-oriented programming is a highly sophisticated paradigm which is highly suited to intelligent tutoring in a real-time environment.

JACK agents are based on the BDI Framework of Rao and Georgeff [20]. They are autonomous software components that execute plans (intentions) to achieve their goals (desires). The plan chosen at any given time depends on the current set of beliefs. JACK agents can also respond to events as well as striving for goals, in other words they exhibit both proactive (goal-driven) and reactive (event-driven) behaviour. Each agent possesses: (i) a **database** (set of beliefs), (ii) a set of **events** it will respond to, (iii) a set of **goals** that it wishes to achieve, and (iv) a set of **plans** that describe the appropriate responses to events or ways to achieve goals.

A JACK agent remains idle until it is given a goal to pursue, or until it is has to respond to an event. The agents are autonomous and they must determine the appropriate response to goals and events, i.e. the appropriate plan to be executed. A JACK Agent is able to exhibit the following behaviours [JACK Manual]:

- A goal-directed focus. The agent focuses on the objective and not the method chosen to achieve it.
- Real-time context sensitivity. The agent will keep track of which options are applicable at each given moment, and makes decisions about what to try and retry based on the present conditions.
- Real-time validation of approach. The agent will ensure that a chosen course of action is pursued only for as long as certain maintenance conditions continue to be true.
- Concurrency. The agent system is multithreaded. If new goals and events arise, the agent will be able to prioritise.

The JACK Agent Language extends Java in the following ways:

- It defines new base classes, interfaces and methods.
- It provides extensions to the Java syntax to support new agent-oriented classes, definitions and statements.
- It provides semantic extensions (runtime differences) to support the execution model required by an agent-oriented software system.

The JACK Agent Language provides the following five main class-level constructs:

- **Agent**: The agent construct is used to define the behaviour of an intelligent software agent. This includes capabilities an agent has, what type of messages and events it responds to and which plans it will use to achieve its goals.
- **Capability**: The capability construct allows the functional components that make up an agent to be aggregated and reused. A capability can be made up of plans, events, databases and other capabilities.

- **Database**: The database construct provides a generic relational database. It has been designed specifically so that it can be queried using logical members. Logical members are like normal data members, except that they follow the rules of logic programming. Agents can also use regular Java data structures for storing information, but the built -in database can generate events when particular changes occur.
- **Event**: The event construct describes an occurrence that the agent must take an action in response to.
- **Plan**: An agent's plans are analogous to functions. They are the instructions the agent follows to try to achieve its goals and handle its designated events.

4 Agent Communication

Our system is a multi-agent system. Agents need to interact to build and manage the student model. As a consequence our agents require the ability to communicate to one another. JACK provides the infrastructure of the communication but does not specify a particular language or protocols, hence designers can choose the most appropriate for their application. We have chosen KQML as our protocol for exchanging information and knowledge. It is based on speech acts theory as described by Searle [23]. One of the main reasons for our choice is that all the information for understanding the content of the message is included in the communication itself. It is defined by the following protocol structure, as outlined by Huhns and Stephens [13]:

```
(KQML-performative
                          :sender          <word>
                          :receiver        <word>
                          :language        <word>
                          :ontology        <word>
                          :content         <expression>
                          ..)
```

The performatives in our systems are: evaluate, achieve, monitor, revise, extract, tell, and ask. In JACK the agents must know one another's name, and when agents that are communicating are running in separate processes, then the JACK network communications layer needs to be used to allow these processes to communicate. KQML-speaking agents behave as clients and servers, and communication can be synchronous or asynchronous.

5 The Architecture

The ability of an Intelligent Tutoring System to deliver appropriate individualised instruction to a student depends heavily on the type and calibre of the information held about the student in the student model. This in turn depends on the type and level of sophistication of the knowledge representation used in the system and on the effectiveness of the methods used to elicit new information about the student and to incorporate the new information into the student model. Problems arise when new information conflicts with information already in the student model; when the student model contains insufficient information for the tutor to decide on the next instructional step; or when there is uncertainty associated with some of the information about the student - for example, there may be more than one way of interpreting an error made by a student in terms of what the student knows or does not know. There have been many approaches to dealing with these problems.

A number of studies (e.g. Mizoguchi et al [19], Kono et al [17], Giangrandi and Tasso [9]) have applied Truth (or Reason) Maintenance Systems (Doyle [7], DeKleer [6]) to overcome the problem of new information conflicting with old. The TMS identifies the conflicts, which must then be resolved by some domain specific reasoning system. A TMS must maintain not only the beliefs of the student, but also the justifications for them, and therefore use of a TMS is computationally very intensive. Huang and McCalla [12], and Huang [11] have developed a "Logic of Attention", a modification of the TMS which overcomes the problem of efficiency by focusing only on the parts of the student model and instructional planner that are relevant to the current sub-goals. Jones and Poole [15] examined how Reiter' s default logic [21] could be used to build expert diagnostic systems.

One general approach to coping with uncertain or incomplete information is to assume that student models do not need to be completely accurate and absolutely precise to be successful. In granularity-based recognition of students' problem solving strategies the philosophy is that student behaviour can be recognised at some level of detail, even if this is very coarse (McCalla and Greer [18]). In the "fuzzy" student model approach (eg Hawkes et al [10] and Katz et al [16]), which is grounded in fuzzy set theory, a student might have partial membership in the set of students who are expert in a particular skill, and partial membership in the set of students who are less expert in that skill. Alternatively, application of Bayesian belief networks (e.g. Villano [25], Shute [24], and Reye [22]) deals with the problem of uncertain information and also facilitates prediction of student knowledge and performance, but most likely at the cost of extensive knowledge engineering and programming.

Intelligent tutoring systems will, in general, have to provide mechanisms to deal with four interrelated information modeling problems:

- Uncertainty of information,
- Incompleteness of information, ie all relevant information may not be known.
- Fusion of information, where information is merged from different sources, and
- Revision of an existing knowledge base when new information is obtained. This new information may be inconsistent with the knowledge base.

Information that is uncertain or incomplete may need to be revised as it is refined over time. Hence, revision of a knowledge base is closely related to modelling both the uncertainty and the incompleteness of information.

In a previous paper [5] we proposed an architecture in which the problem of conflicting information is resolved using methods recently developed by Williams [26, 27, 28] based on the AGM paradigm for belief revision (Alchourron et al [1]). We used possibility theory (see Dubois and Prade [8]) to take care of uncertain information, nonmonotonic reasoning, in particular Reiter's default logic [21] and the formalism of Antoniou and Williams [2], to deal with missing information, and Theory Extraction for fusion.

Our new proposed system architecture is illustrated in Figure 1 below. It has been modified to take advantage of the agent-based architecture, and consists of the three component agents: the knowledge management agent, the student agent, and the inference agent.

6 The Agents

The **Knowledge Management Agent**: This agent mainly responds to events which take the form of requests from other agents. Agents request information regarding such things as domain knowledge, typical errors and misconceptions, suggestions for the next task to present the student. The domain knowledge is structured to suit the application at hand and managed by an agent. We are currently exploring two applications; one based on a mathematical dictionary for schools, The MathProbe[2], and a second focusing on database design for our own courses at the University of Newcastle. An agent that knows about common student errors and misconceptions has a better chance of diagnosing problems than a system without this knowledge. The quality of this knowledge is often what sets a good teacher apart. In both the domains that we have selected this knowledge is well known and widely accepted. For example, *students who consistently place foreign keys in the wrong database table normally do not understand the concept of cardinality of relationships between entities.*

The notion of knowledge granularity has been widely used in the literature (eg. McCalla and Greer [18]). In our architecture granularity is used in both the domain knowledge and in the set of common errors and misconceptions. Levels of granularity fit naturally into the agent architecture and can be used to help the agent choose an appropriate plan.

The **Student Agent**: Each student is assigned an individual agent instantiation. The main objective of this agent is to manage the evolution of the student model, i.e. a representation of the student' s knowledge about

[2] See http://mathresources.com/mathbrow.html

the domain knowledge and the student' s personal goals and preferences. This model is described using the following components:

- The Student' s Goals using JACK Goals
- The Student' s Preferences using the JACK Database,
- Explicit Knowledge about the Student based on their performance so far using the JACK Database.

The student agent is autonomous and responds to input from the student.
The **Inference Agent**: The inference agent manages a team of agents that provide several forms of useful inference mechanisms and sophisticated reasoning operators. It is not necessary for the agents requesting knowledge to know how the Inference Agent generates that knowledge. The Inference agent uses slave agents for deduction[3], abduction and induction. The belief revision agent[4], possibilistic reasoning agent[4], nonmonotonic reasoning agent[4] and theory extraction agent[4] rely exclusively on the slave agents.

The student agent' s goals will typically vary from student session to student session, and can be customised by a third party such as a human tutor. These goals determine the learning strategies and tasks to be used during a given learning session. The learning strategies together with the database describing the current state of the agent and its knowledge about the student' s capabilities will largely control the agent' s behaviour. These strategies are ultimately implemented via an agent that constructs a learning task hierarchy. This learning task hierarchy is constructed at run time. It can be viewed as a sub-hierarchy of the global task hierarchy customised using the current student profile. This sub-hierarchy is designed to provide feedback about the student that can be used to build and manage the student model during a learning session. In addition, it is used to diagnose student problems and subsequently offer remedial action.

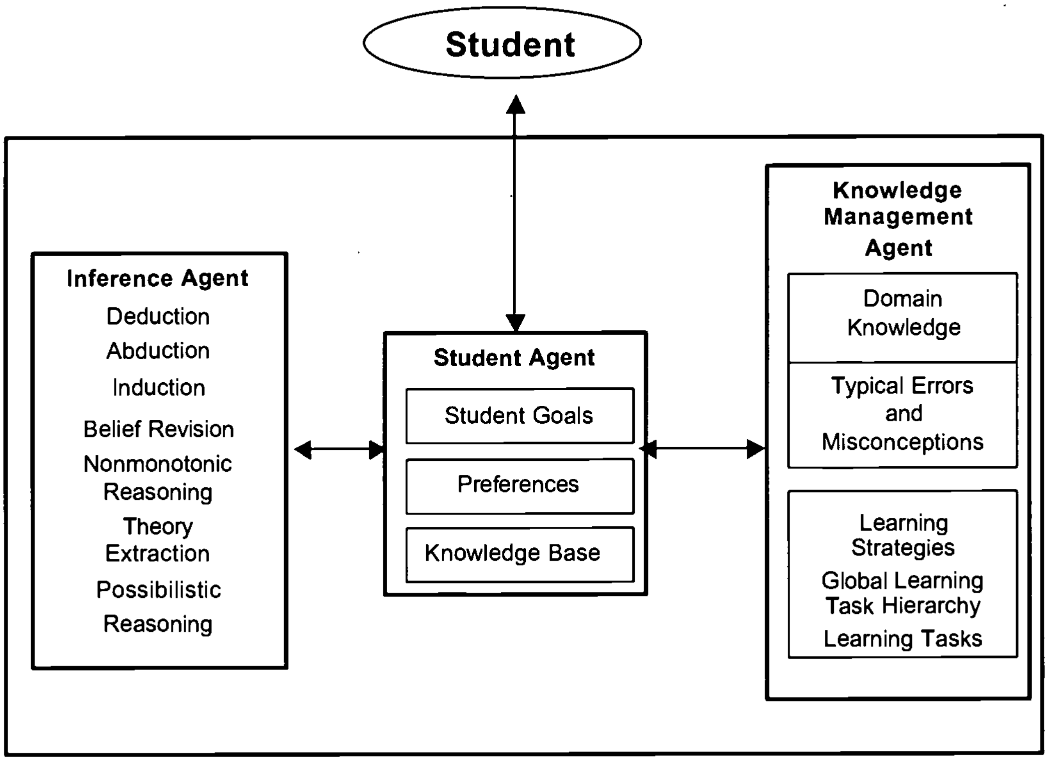


Figure 1: An Agent-Based Intelligent Tutoring System Architecture

[3] See http://ebusiness.newcastle.edu.au/vader

[4] See http://ebusiness.newcastle.edu.au/saten

7 Conclusions

In a previous work we identified several information modeling problems that arise in Intelligent Tutoring Systems: change, incompleteness, information integration, and uncertainty. We described an architecture for an intelligent tutoring system that addressed these problems based on recent developments in knowledge representation and reasoning in the areas of belief revision, possibilistic reasoning, nonmonotonic reasoning and theory extraction.

In this paper we described an agent-based design of the architecture based on the BDI framework. The BDI framework is based on the use of Beliefs, Desires and Intentions. Our architecture is under construction using an agent oriented programming language called JACK. JACK provides an agent-oriented development environment. It supports the BDI framework and is built on top of the Java development environment. It not only provides all the necessary agent infrastructure for our architecture, but it also allows us to use previously developed Java modules in an agent environment.

The main advantage of using an agent-based approach is that the control module has been eliminated. Control has successfully been devolved to the agents, i.e. there is no need for a superagent to oversee the communication and interaction. This leads to better performance and a more customised student learning session.

References

[1] Alchourrón, C., Makinson, D. and Gärdenfors, P. (1985), On the Logic of Theory Change: Partial Meet Functions for Contraction and Revision, *Journal of Symbolic Logic* **50**: 510 - 530.

[2] Antoniou, G. and Williams, M-A., Reasoning with Incomplete and Changing Information, *Journal of Information Science*, 99, **1 & 2**: 83 – 99, (1996).

[3] Antoniou, G. and Williams, M-A., Teaching Intelligent Information Management, in the *Proceedings of the First International Conference on Computational Intelligence and Multimedia Applications*, 380 – 384, (1997).

[4] Booch, G., Object-oriented Analysis and Design with Applications, Addison-Wesley, Reading, MA, 1994.

[5] Bruff, C.M. and Williams, M-A, An Intelligent Tutoring System Architecture based on Knowledge Representation and Reasoning, in *Proceedings of the 7th International Conference on Computers in Education*, IOS Press, 736-743, 1999.

[6] De Kleer, J., An Assumption-based TMS, *Artificial Intelligence*, **28**, 127 – 162 (1986)

[7] Doyle, J., A Truth Maintenance System, *Artificial Intelligence*, **12**, 231 – 272 (1979)

[8] Dubois, D. and Prade, H., Epistemic Entrenchment and Possibilistic Logic, *Artificial Intelligence*, **50**, 223–239, (1992).

[9] Giangrandi, P. and Tasso, C., Truth Maintenance Techniques for Modelling Students' Behaviour, *J. Artificial Intelligence in Education*, **6**, 153 – 202, (1995).

[10] Hawkes, L.W., Derry, S.J. and Rudensteiner, E.A., Individualized Tutoring using an Intelligent Fuzzy Temporal Relational Database, *International Journal of Man-Machine Studies*, **33**, 409 – 429, (1990).

[11] Huang, X., Modelling a Student's Inconsistent Beliefs and Attention, *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, eds. J.E. Greer and G.I. McCalla, 267 – 280, (1994).

[12] Huang, X. and McCalla, G., Instructional Planning using Focus of Attention, *2nd International Conference on Intelligent Tutoring Systems* eds Frasson, C., Gauthier, G. and McCalla, G., Montreal, 443 – 450, (1992).

[13] Huhns, M.N. and Stephens, L.M., Multiagent systems and societies of agents, in *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*, ed. G. Weiss, MIT, 79 – 120 (1999).

[14] Jennings, N.R., On agent-based software engineering, Artificial Intelligence **117**, 277 – 296, (2000).

[15] Jones, M. and Poole, D., An Expert System for Educational Diagnosis based on Default Logic, *Proceedings of the 5th International Workshop on Expert Systems and their Applications*, Vol. **2**, 673 – 683, (1985).

[16] Katz, S., Lesgold, A., Eggan, G. and Gordin, M., Modelling the Student in Sherlock II, *J. Artificial Intelligence in Education*, **3**, p. 495, (1993).

An Educational System that can Visualize Behavior of Programs on the Domain World

Taketoshi Ando*, Yoshinobu Kawada*[(†)], Ryoji Itoh*,
Tatsuhiro Konishi*, Yukihiro Itoh*
* *Department of Computer Science, Shizuoka University.*
Johoku3-5-1, Hamamatsu, Shizuoka, 432-8011, Japan.
Tel: +81 - 53 - 478 – 1474• • Fax: +81 - 53 - 478 – 1499
E-mail: s5005@cs.inf.shizuoka.ac.jp

In this paper, we discuss extension of our educational system that gives domain-oriented-explanations of programs. A programmer solves problems on a world where elements necessary to describe the problems and the solving processes of them (objects in the problems, relation among the objects, and so on) are represented. We call such a world 'the domain world' of the problem. Our system has a domain world model and simulates a target program on the model, to understand behavior of the program. By analyzing the result of understanding, it generates an explanation. Outputs of our original system are only verbal explanations. However, when the system explains by using only sentences, some learners cannot get a concrete image of behavior of the program. Therefore, we are trying to add a facility of generating explanations by using animations (visual explanations) to the system. Our extended system can generate both visual and verbal explanations (bimodal-explanations) in various abstraction levels. We discuss the method of generating bimodal-explanations from the result of simulation.

Keywords: **Intelligent Tutoring System, Programming Education, Algorithm Animation System, Bimodal-explanation**

1 Introduction

The purpose of our research is to construct an educational system that helps novice programming learners by explaining domain-oriented-functions of programs. We take Pascal as our target programming language.

Programming is generally carried out in the following process.

Step1. A programmer understands a problem that must be solved.
Step2. He considers the solving process of the problem on a world where the problem is present. We call such a world 'the domain world' of the problem. For example, when he considers a solving process of sorting, he imagines a world in which he pays attention to numerical order such as greater and lesser (we call this world the world of greater and lesser).
Step3. He implements the algorithm: selects data structures suitable to represent the domain world and translates the algorithm into a programming language.

Usually, relatively simple problems are set in novice class of programming. So it is rare that learners fail in the step 1. But they tend to confuse because they cannot distinguish between step 2 and 3. So many novice programmers cannot find whether the causes of bugs are hidden in the algorithm or in their implementation. On the basis of this idea, we proposed an educational system explaining programs using vocabularies on a domain world[2][5][6][7]. Difference of our system from existing educational systems of programming [1][8] is that the purpose of our system isn't pointing out bugs in learner's programs, but rather helping learners find bugs by themselves. Our system helps learners in the following way:

- To help learners to understand sample programs given by a teacher by explaining them.
- To help learners to find and fix bugs in their own programs by explaining the faulty behavior of them.

Our previous system outputs sentences using vocabularies on a domain world as the explanation. However, when the system explains by using only sentences, some learners cannot get a concrete image of behavior of the

[(†)] *Presently with System Integration Group, VICTOKAI, LTD.*

program. If animations of the behavior of programs are shown with the sentences, learners can easily understand their algorithms. Therefore, we realize the ability to generate animations (visual explanations) that show behaviors of the target programs. In this paper, we discuss the way to generate visual explanations for programs in the domain world of greater and lesser.

Existing algorithm animation systems can be classified into two types: The first one is a system such as courseware editors embody particular commands to target programs in order to generate visual explanations, like Zeus[3] and TANGO[9] system. So, this type of systems can generate visual explanations of high quality by using concrete objects on the domain world. For example, a length of bar is used to concrete values of variables on the visual explanation of XTANGO system. The second type of system doesn't need embodying particular command to target programs, like UWPI[4] and tracers. However, this type of systems cannot generate any visual explanation using concrete objects on the domain world. They can only generate visual explanations showing structure of data and changes of contents of variables. Our system can generate a visual explanation using concrete objects on the domain world without embodying any special commands to programs. It generates visual explanations on the basis of the result of "simulation based program understanding[5]". So it can accept buggy programs and generate visual explanations of buggy behaviors of the programs. Moreover, it can also generate verbal explanations on the basis of the result of program understanding.

In the next section, we illustrate an overview of our previous system. In section 3, we point out some functions necessary to generate an effective explanation by using both verbal explanation and visual explanation (a bimodal-explanation). In section 4, we describe the method of constructing the bimodal-explanation system. Then, we show examples of bimodal-explanations by our system.

2 Our Previous Work

2.1 Overview of our previous system

Our previous system is composed of the static analyzer, the simulation based analytical unit and the explanation unit (Figure 1). In this paper, we omit detail of the system (For further details, please see our previous papers[2][5][6][7]). The static analyzer parses target programs and analyzes information necessary for the simulation such as data flow. The simulator simulates target programs, and the observer observes the world model while simulation, and recognizes some important characteristics of data or patterns of structured data. The explanation generator generates verbal explanations of target programs.

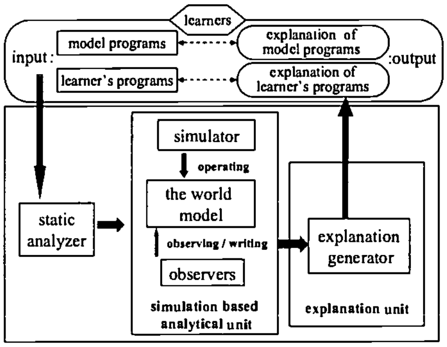


Figure 1: Configuration of our system

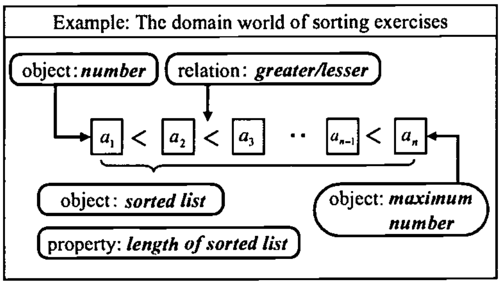

Example: The domain world of sorting exercises
object: number
relation: greater/lesser
a_1 < a_2 < a_3 ·· a_{n-1} < a_n
object: sorted list
object: maximum number
property: length of sorted list

Figure 2: An example of domain world

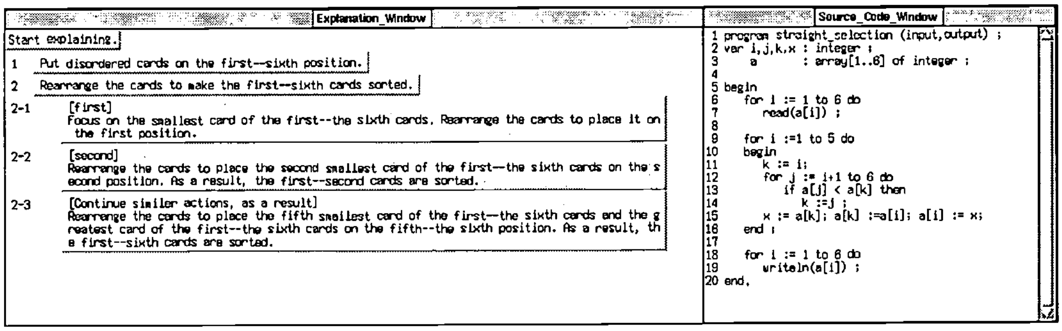
2.2 Domain world models

We examine programming exercises and classify them into 15 types. We prepare domain world models designed for each type of exercises [2].

A domain world model consists of four types of elements called 'object', 'property', 'relation among objects' and 'change'. For example, Figure 2 shows the domain world model of greater and lesser as an example. In order to recognize specified characteristic or patterns in the domain world, our system has daemon units called "observer" which are burnt when they are observed. In the Figure 2, the object *'maximum number'*, *'sorted list'* and property *'length of sorted list'* are recognized by observers. There are some cases that some observers take outputs of the other observers as their inputs. Then the outputs of observers make hierarchy. When a result of observation is output on the basis of a result of another observer, the former has larger grain-size than the latter and implies the fact corresponding to the latter.

2.3 Generation of a verbal explanation

The explanation unit generates verbal explanations of the target program by using results of simulation and outputs of observers. The results of observations have a hierarchical structure, as mentioned above. The system generates a hierarchical verbal explanation by using the hierarchical structure (it also uses syntactical structures of programs). In other words, the system notices the largest grain-sized result of the observation firstly, in order to generate the verbal explanation. Secondly, if learners request the detailed verbal explanations, the system generates the explanation using results of observation having smaller grain size. Figure 3 shows the example of verbal explanations generated by our system. It illustrates the verbal explanation of behavior of a sorting program on the domain world of greater and lesser. The indentation in the figure means that *behavior 1* and *behavior 2* are executed sequentially and that *behavior 2* is equivalent to the sequence of *behavior 2-1, behavior 2-2,* and *behavior 2-3*. Each Behavior is implemented by a single statement or a sequence of statements. When a verbal explanation for a behavior implemented by a sequence of statements is clicked, more detailed verbal explanations showing the way to implement the behavior are displayed.



```
Start explaining.
1    Put disordered cards on the first--sixth position.
2    Rearrange the cards to make the first--sixth cards sorted.
2-1      [first]
         Focus on the smallest card of the first--the sixth cards. Rearrange the cards to place it on
          the first position.
2-2      [second]
         Rearrange the cards to place the second smallest card of the first--the sixth cards on the s
         econd position. As a result, the first--second cards are sorted.
2-3      [Continue similer actions, as a result]
         Rearrange the cards to place the fifth smallest card of the first--the sixth cards and the g
         reatest card of the first--the sixth cards on the fifth--the sixth position. As a result, th
         e first--sixth cards are sorted.
```

```
program straight_selection (input,output) ;
var i,j,k,x : integer ;
     a      : array[1..6] of integer ;

begin
   for i := 1 to 6 do
      read(a[i]) ;

   for i :=1 to 5 do
   begin
      k := i;
      for j := i+1 to 6 do
         if a[j] < a[k] then
            k :=j ;
      x := a[k]; a[k] :=a[i]; a[i] := x;
   end ;

   for i := 1 to 6 do
      writeln(a[i]) ;
end.
```

Figure 3: An example of the verbal explanation of a target program

Next, we show procedures for generating the hierarchical verbal explanation like Figure 3. An input is a result of simulation of some statements (For further details, please see our previous paper[7]).

(1). The case that a certain behavior is implemented by a sequence of statements.
The system observes differences of the states of the domain world model before and after execution of the sequence of statements. According to these differences, the explanation unit selects a template and generates verbal explanations for the statement. Now we show an example of a template.

- The case that extension of the sorted list is observed.

The differences are composed of the following three elements.

Object(s) recognized at the state before execution of some statements	: a sorted list
Object(s) recognized after execution of the statements	: an extended sorted list
Recognized changes of states of objects	: an extension of the sorted list

A template for the extension of the length of a sorted list is applied (Please see Figure 3).

Template: "Rearrange the [Type of added object] to place [An added object] on [The position of the insertion] position. As a result, [A sorted list at the after state] [Type of inserted objects] are sorted."

"[]" means a procedure which generates a certain pattern of string.••

[An added object]
: A procedure that generates a noun phrase expressing the new object added to the sorted list.

[A sorted list at the after state]
: A procedure that generates a noun phrase expressing the range of the sorted list at the after states.

(2). The case that a certain behavior is implemented by a single statement.
The explanation unit calls each procedure corresponding to types of the statement. The procedures are defined for each structure of the program like sequential structures, selective structures, iterative structures, an assignment statement, a statement for input, and a statement for output. Same as the case (1), templates are prepared for each structure of the program. For example, we show a template of 'if' statement.

Template: "if [explanations of the conditional clause], [explanations of the 'then' clause]
(otherwise [explanations of the 'else' clause]) "

[explanations of the conditional clause]
: The procedure that explains the conditional statement of 'if' statement.

[explanations of the 'then' clause]
: Apply the procedure for generating the verbal explanation to the clause recursively.

[explanations of the 'else' clause]

: Apply the procedure for generating the verbal explanation to the clause recursively.

Thus, the system can generate hierarchical verbal explanations. When a verbal explanation generated by the procedure (1) is shown and a learner requests more detailed explanation, the system tries to apply the procedure (1) recursively to make such an explanation. If it cannot generate any explanation, it applies the procedure (2).

3 Functions necessary to generate an effective bimodal-explanation

In order to construct a system generating effective visual explanations, we have to consider what visual explanation is effective for learners to understand an algorithm or behavior of a target program. By designing mock up visual explanations repeatedly, we find that the effective visual explanation has following three facilities.

(1) The facility to generate visual explanations with various grain-sizes.
When learners learn programming by using a system explaining behaviors of programs, they need various grain-sized explanations. For example, when a learner wants to grasp algorithm roughly, a large grain-sized explanation would be effective. On the other hand, when he wants to understand a precise method of implementation, smaller grain-sized explanations are effective. Moreover, when he wants to diagnose his own program at a glance, he needs the largest grain-sized explanation. When he wants to find buggy codes, he needs smaller ones. In order to generate such various grain-sized visual explanations, the system should be able to :

- regard a sequence of statements as a blackbox and generate a visual explanation showing its function.
- generate a visual explanation showing a function of each statement sequentially.

(2) The facility to explain a function of a program by using both animations and verbal texts.
If a system shows only visual explanations, learners sometimes cannot understand behavior of target programs clearly, because such learners cannot understand what phenomena are essential. Thus, it is necessary for our system to have the facility to generate verbal explanations showing a major phenomenon of each step of visual explanations. Thus our system should have a facility of generating combination of verbal explanations and visual ones (bimodal-explanations).

(3) The facility to generate explanations on the total effect of a sequence of statements.
Generally, a task is achieved by a sequence of statements, and each sub-task is achieved by each sub-sequence of the statements. When the system shows a sequence of explanations each of which has a certain grain-size corresponding to a sub-task, a learner sometimes cannot find the fact that the task has been achieved. In order to prevent learners from such misunderstanding, the system should show them a verbal explanation remarking the fact.

4 Methods to realize the functions to generate bimodal-explanations

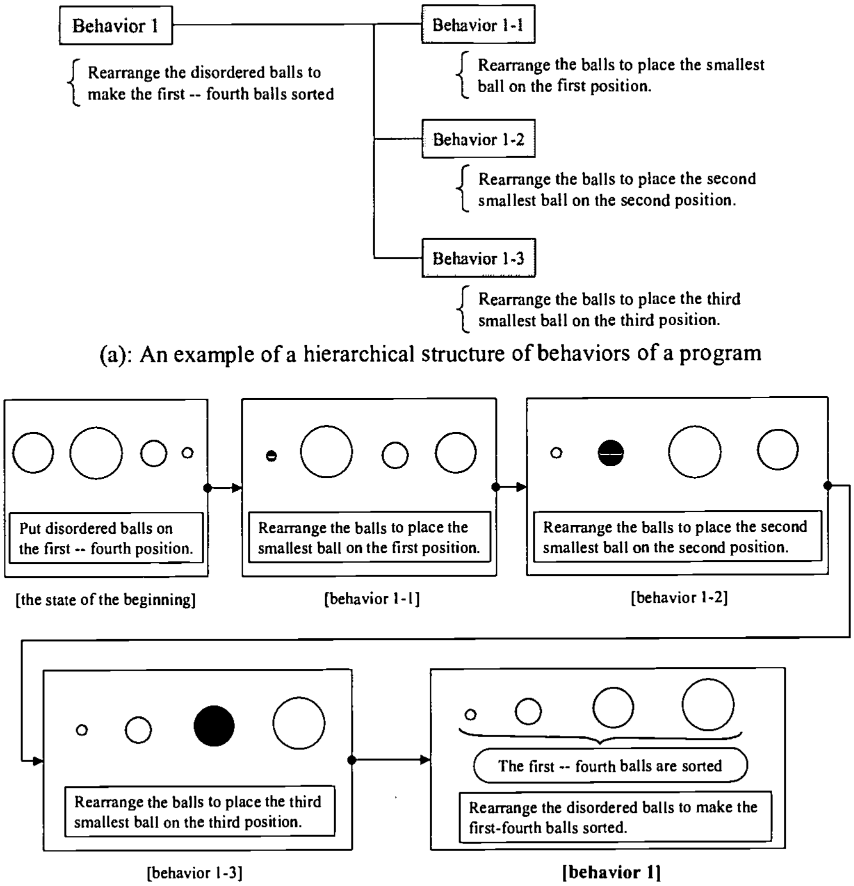
4.1 Basic ideas

(1) The method of generating visual explanations on various grain-size.
As we describe in section 2, our system can generate hierarchical verbal explanations. In other words, it can understand behavior of a target program on various grain-size. And the system holds the result of understanding as hierarchical data. Therefore we can realize a system generating visual explanations on various grain-size, by developing a method to generate a visual explanation from a result of understanding.

(2) The method of generating combination of verbal explanations and visual explanations.
Our program understanding mechanism can recognize the major phenomena in the domain world. And we have already developed a method to generate verbal explanations from the result of program understanding. Thus, if the system can generate a visual explanation from the result of it by the method (1), it becomes to be able to generate both visual explanations and verbal explanations remarking major phenomena from common data.

(3) The method of generating explanations on the total effect of a sequence of statements.
By generating an explanation remarking that a task is achieved just after explanations of sub-tasks are finished, the system can generate explanations on the total effect of the task. The explanations of the task and the sub-task can also be generated by the method (1) and (2). For example in Figure 4, just after the explanation corresponding to the behavior 1-3 is finished, the system generates the explanation corresponding to the behavior 1 as the explanation of the total effect. As a result, the explanation shown in Figure 4 is generated.

In consequence, if we can realize the method (1), the method (2) and (3) can also be realized. Therefore we discuss the detail of the method (1) in the next section.


Behavior 1
Rearrange the disordered balls to make the first -- fourth balls sorted
Behavior 1-1
Rearrange the balls to place the smallest ball on the first position.
Behavior 1-2
Rearrange the balls to place the second smallest ball on the second position.
Behavior 1-3
Rearrange the balls to place the third smallest ball on the third position.
Put disordered balls on the first -- fourth position.
[the state of the beginning]
Rearrange the balls to place the smallest ball on the first position.
[behavior 1-1]
Rearrange the balls to place the second smallest ball on the second position.
[behavior 1-2]
Rearrange the balls to place the third smallest ball on the third position.
[behavior 1-3]
The first -- fourth balls are sorted
Rearrange the disordered balls to make the first-fourth balls sorted.
[behavior 1]

(a): An example of a hierarchical structure of behaviors of a program

(b): Generated explanations

Figure 4: An example of a bimodal-explanation of a total effect

4.2 Generating visual explanations

The system visualizes behavior of the target program in various grain-size. The generated animations are shown with verbal explanations. The detail of our method to generate verbal explanation is seen in [6], so we omit it in this paper.

At first, the system starts explaining with the largest grain-size, then shows more detailed explanation on an action of which detail a learner wants to see.

The methods to draw a step of animation are classified into thc following two types:
1) The method of visualization for a function implemented by a single statement.
2) The method of visualization for a function implemented by a sequence of statements.
The detailed process of 1) and 2) is discussed in 4.2.1 and 4.2.2 respectively.

4.2.1 How to generate a visual explanation of a function implemented by a single statement

In order to generate a visual explanation on a statement, we prepare specific procedures for each type of a statement. The statements of inputting, assignment, selection, and iteration have their individual procedures.

Procedures for inputting statement should be classified into several types in order to generate effective explanations. For example, the basic function of inputting statcment "read (A);" must be "a datum is input to the variable A". However, showing only the basic function is not always a good explanation. If a meaningful datum has been stored in the variable "A" before inputting, the system should also explain that the datum is deleted by the inputting. Therefore, the procedures for inputting statement are classified according to some conditions on the role of the statement in the target program and the domain world: for example, the condition whether the datum stored in the destination variable of inputting has been referred before the input sentence or not (if it has been referred, it must be meaningful).

Similarly, procedures for assignment statement should also be classified. For example, the basic function of the statement "A:=B;" is "the datum in the variable B is copied and the copy is written on the variable A". But, if the datum in B will never be referred after the assignment, the explanation " the datum in B is moved to A" must be better because it represents the role of the statements more directly. We show an example of a condition to classify procedures for assignment statement and a procedure corresponding to the condition, by using the statements illustrated in Figure 5.

The condition and procedure for assignment statement meaning the copying process of objects in a sorting program is as follows.

Condition: [the datum in B represents an object in the world of greater and lesser]
and [B is referred during *2] and [A is also referred during *1]

Procedure: Seen in Table 1. And the visual explanation generated by the procedure is seen in Figure 6.
Table 1 also shows the templates for generating verbal explanations in this condition.

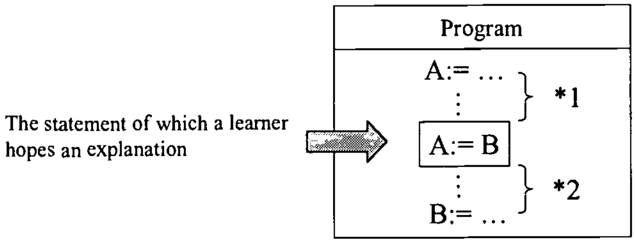


Figure 5: An example of an assignment statement in a program

The system doesn't generate visual explanations corresponding to statements of selection and iteration. For example on "if-then-else" statement, it generates visual explanations corresponding to 'then' block or 'else' block, while it generates verbal explanations whether the condition part of the statement is true or false. By the verbal explanation, a learner can understand why the 'then' block or the 'else' block is executed. Templates used to generate such verbal explanations are not the ones mentioned in 2.3 because of the following reason:

- In general, verbal explanations can be abstract. For example, we can explain a sorting process of N pieces of balls.
- Visual explanations must be concrete. For example, the system has to decide how many balls exist in the domain world in order to draw sorting process.
- Therefore, verbal explanations corresponding to visual explanations have to be generated by templates designed for bimodal-explanation.

Table 1: Examples of procedures and templates for generating explanations of an assignment statement

order	Procedures for generating visual explanations.	Templates of verbal explanations
1	Show a ball having an assigned value with a black color.	Focus on [a ball holding an assigned value].
2	Show a copy of the ball having the assigned value.	Prepare a copy of this ball.
3	Show a ball assigned the value with a gray color.	Remove a ball on [an assigned position].
4	Draw an arrow from the copy of the ball having the assigned value toward the ball assigned the value.	Move the copy of this ball to [an assigned position].
5	Show the copy of the ball having the assigned value at an assigned place.	Move this ball to [an assigned position].
6	Show a state of end.	

As mentioned above, "[]" in template means a procedure by which a certain pattern of string is generated. The bimodal-explanation corresponding to a single statement is composed of 3 ~ 5 scenes.

4.2.2 How to generate an explanation of a function implemented by a sequence of statements

In order to generate such a visual explanation, the system needs to display the states before/after the function has been applied. In order to decide designs of both states, the system generates a sequence of visual explanations on the statements by which the function is implemented. The generated explanations are only stored in a database without being displayed to a learner. The system picks up the initial state and the final state from the database and displays them one after another.

In addition, the system also needs to explain an effect of the sequence of statements directly. In order to generate an explanation of a total effect of statements, the system should generate a verbal explanation remarking the total effect and a step of animation showing the effect directly. The system can generate the verbal explanation by applying templates to the result of simulation. In order to generate a step of animation, we prepare procedures for visualizing a concept recognized as a result of the simulation. For example, brace and the words attached to it in Figure.4 (b) are drawn by such procedures. The number of such procedures is nearly equal to the number of template for the result of simulation (illustrated in 2.3.(1)).

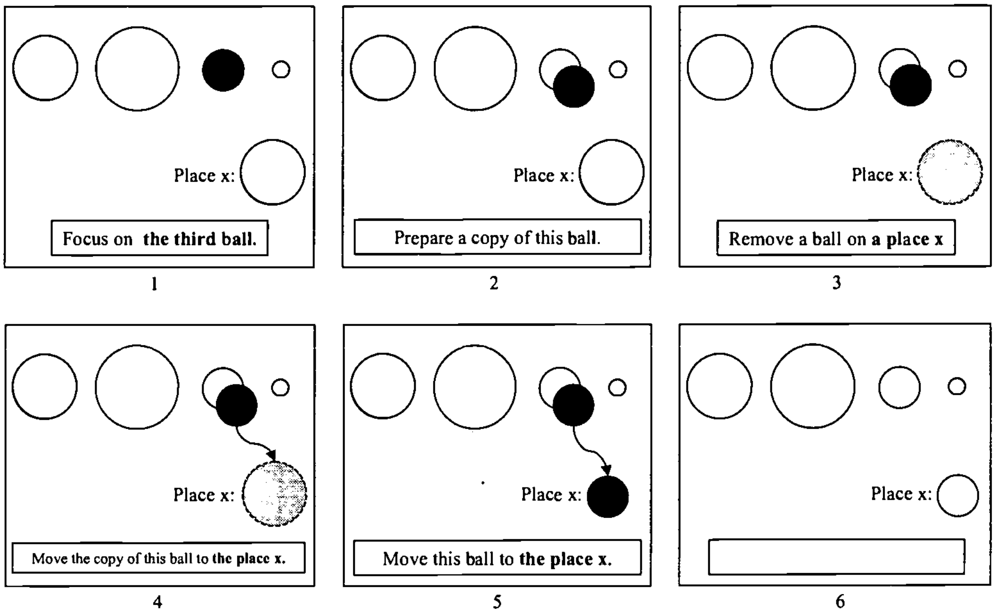

Place x:
Focus on the third ball.
1
Place x:
Prepare a copy of this ball.
2
Place x:
Remove a ball on a place x
3
Place x:
Move the copy of this ball to the place x.
4
Place x:
Move this ball to the place x.
5
Place x:
6

Figure 6: The visual explanation corresponding to each statement

5 Implementation

The system is developed on Unix workstations. The unit of generating bimodal-explanation is implemented by Tcl/Tk. Now we have finished implementation of procedures generating bimodal-explanations of statements of input, assignment, selection and iteration. Figure 7 shows an example of bimodal-explanations generated by our system. The target program is sorting by straight insertion. The system explains the process of sorting five balls according to their sizes. Figure7-1 shows the state just after that the smallest ball (in this figure, it is the 4th ball from the left end) has been found, the ball has been copied to the 'place x', and it has been removed. After that, the following processes are explained one after another.

- The copy of the first ball is moved to the fourth position (Figure7-2, 7-3).
- The ball on the 'place x' is focused (Figure7-4).
- The ball on the first place is removed (Figure7-5).
- The ball on the 'place x' is moved to the first position (Figure7-6, 7-7).

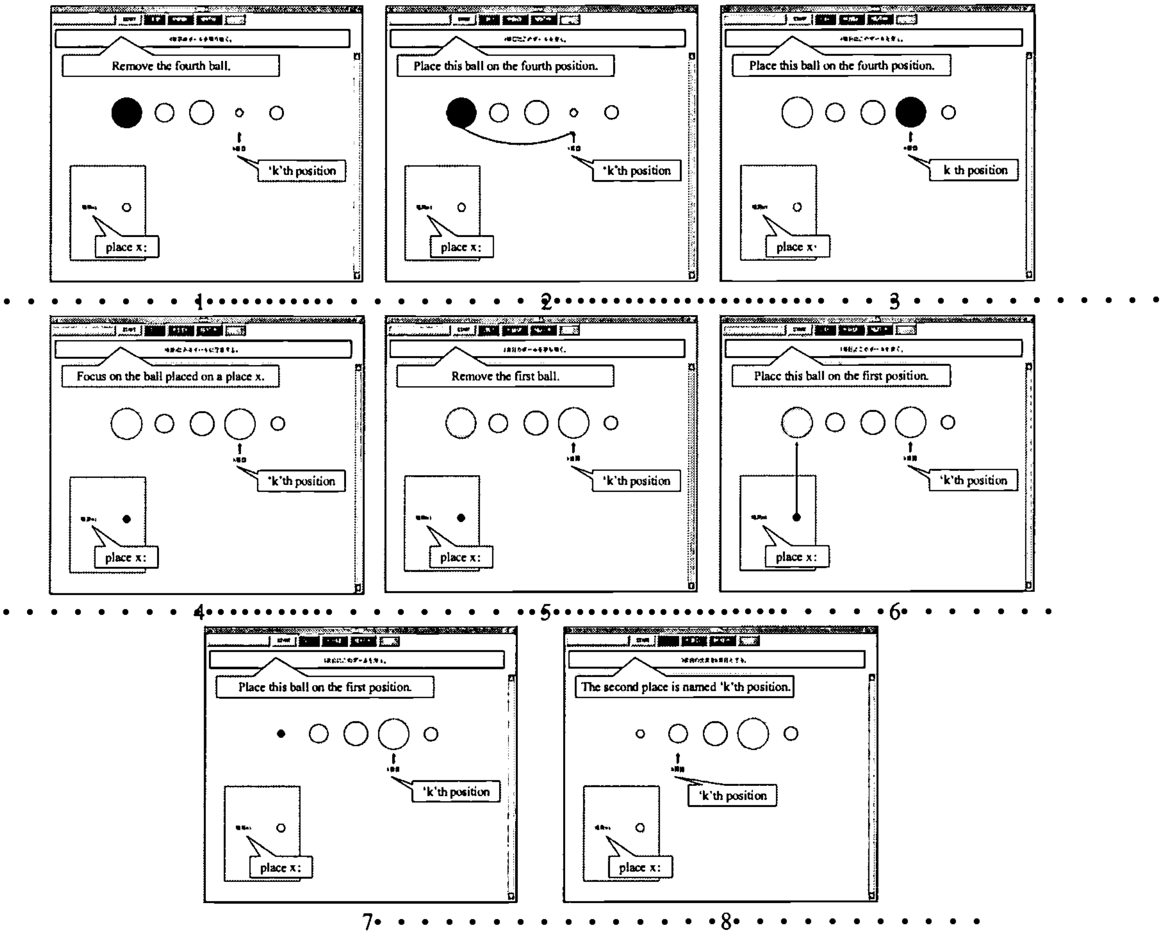
By these explanations, learners can imagine the process that the smallest ball is moved to the first place. The explanation after Figure7-8 continues in a similar way. The system also shows the process that the second - fifth smallest ball is moved to the second - fifth place respectively. Thus, the whole process of the sorting is illustrated. The original messages generated by our system are Japanese, but we add corresponding English messages to this figure.

6 Conclusion

In this paper, we proposed a method of generating a bimodal-explanation. Our system accepts the result of simulation and generates bimodal-explanations. Our current system can deal with only on the domain world of greater and lesser. Constructing procedures for the remaining types of a statement and applying this system to other domain worlds will be our future work.

Acknowledgments

This research was supported in part by Grant-in-Aid for Scientific Research (c)(11680216) from Japan Society for the Promotion of Science.


1 Remove the fourth ball.

2 Place this ball on the fourth position.

3 Place this ball on the fourth position.

4 Focus on the ball placed on a place x.

5 Remove the first ball.

6 Place this ball on the first position.

7 Place this ball on the first position.

8 The second place is named 'k'th position.

Figure 7: Outputs of our bimodal-explanation system

References

[1] Adam, A., Laurent, J.P.: "LAURA, A system to debug student programs", Artificial Intelligence, Vol.15, pp.75-122, (1980)

[2] Ando, T., Itoh, R., Konishi, T., Itoh, Y.: “Extension of the Educational System that Generates Domain-Oriented-Explanations of Programs – On Methods for Designing Domain World Models and for Simulation”, Proc. of ICCE'99, Chiba, Japan, Vol.1, pp.856-860, (1999)

[3] Brown, M. H.: “Zeus: A system for algorithm animation and multi-view editing (research report No.75)”, DEC systems research center, (1992).

[4] Henry, R. R., Whaley, K. M., Forstall, B.: “The university of Washington Illustrating Compiler ”, Proc. of The ACM SIGPLAN'90 conference on programming language design and implementation, pp.223-233, (1990).

[5] Itoh, R., Nishizawa, M., Konishi, T., Itoh, Y.: “On analysis of novice PASCAL programs based on simulation to extract Information for education”, Proc. of ICCE'97, Kuching, Sarawak, Malaysia, pp.485-492, (1997)

[6] Itoh, R., Nishizawa, M., Konishi, T., Itoh, Y.: “A programming learning system which makes Domain-Oriented-Explanations”, Proc. of ICCE'98, Beijing, China, Vol.1, pp.432-440, (1998)

[7] Itoh, R., Konishi, T., Itoh, Y.: “On Constructing a programming Education System that makes domain-oriented-explanations”, Journal of JSAI, Vol.15 No.2, pp.362-375, (2000), in Japanese.

[8] Johnson, W.L., Soloway, E.: "PROUST:knowledge-based program understanding", IEEE Software, vol.11, no.3, pp. 267-275, (1985)

[9] Stasko, J. T.: “Tango: A framework and system for algorithm animation", IEEE computer, Vol.23, No.9, pp27-39, (1990).

An Environment for Learning by Design
- In the Case of Learning of Search Algorithm -

Tsukasa HIRASHIMA, Hidenori IKEMATSU, Mayt Kalayar, Akira TAKEUCHI
Kyushu Institute of Technology, Department of Artificial Intelligence
680-4, Kawazu, Iizuka, 820-8502, JAPAN
TEL: +81-948-29-7618, FAX: +81-948-29-7601
tsukasa@ai.kyutech.ac.jp

This paper described a learning environment for search algorithms. In the learning environment, learners can build search algorithms by combining several parts by direct manipulation. Then, the environment diagnoses the algorithms in order to give feedback about the algorithms. First, the environment judges whether or not the algorithms are adequate. When the algorithms aren' t adequate, they are diagnosed using heuristics rules. In the diagnosis, errors in the algorithms are detected. By using the results of this diagnosis, the environment can give messages to help the learners revise their algorithms or to motivate them to build the next type of algorithms. We have already implemented the learning environment. As a preliminary evaluation of the environment, we asked 13 students to use the environment, and gathered several types of data. As a result, the experiment suggests that the learning environment is promising.

Keywords: **Learning by design, Error diagnosis, Search algorithm**

1 Introduction

An effective way to learn procedural knowledge in depth is to make learners apply it to various cases. However, although the learners may master how to use the procedure through the experience, it is not enough to answer the question "what the procedure is". Several investigations [1-4] suggested that "learning by design" is a promising way to promote the learner's understanding about "what that is". For example, in order to understand a machine in depth, assembling it from its smaller parts is the best way. In the case of the understanding of procedure, to build up the procedure by trial and error is useful in order to understand it.

This paper reports about a learning environment for learning by design, targeting basic search algorithms taught in an introduction to artificial intelligence lecture, that is, depth-first search, breadth-first search and three heuristics searches (best-first, minimum consuming cost, and A algorithm). In the lecture, usually, the procedure of each search algorithm is taught and learners carry out the searches following the procedures by hand. Some of them understand the meaning of the algorithms through the practice, but some of them only memorize the procedures. Our environment provides several parts of the search algorithm as icons. Learners can assemble them by direct manipulation in the environment. The environment interprets the assembled parts as a search algorithm and diagnoses it, for example, as to whether it falls into the loop or not. Then, the environment gives feedback for the algorithm to revise it or to try to build the next one. The tree structures that are generated as the results of the searches following the algorithms are also presented to the learners. These feedbacks are necessary to realize learning by design effectively.

In this paper, first, the model of the search algorithm that is the basis of the design of this environment is described. Then, the configuration and functions of the learning environment are explained. The preliminary evaluation of the environment is also reported.

2 Interactive Learning Environment of Search Algorithms

Figure 1 shows the configuration of the ILE. It is composed of the interface and reasoning module. In the interface, learners design and build search algorithms, and receive feedback from the system. In the reasoning module, the algorithms are diagnosed and feedback messages for them are generated. The interface is implemented in Java as a client and the reasoning module is implemented in Prolog as a server. Therefore, the ILE can be used on the Internet.

In this section, first, the model of search algorithms used in the ILE is described. The modeling is indispensable for designing the interface for algorithm building and in order to diagnose algorithms. Then, the interface where learners can build the search algorithms by direct manipulation is presented. The diagnosis of the algorithms and the feedback generated based on the results of the diagnosis are also explained.

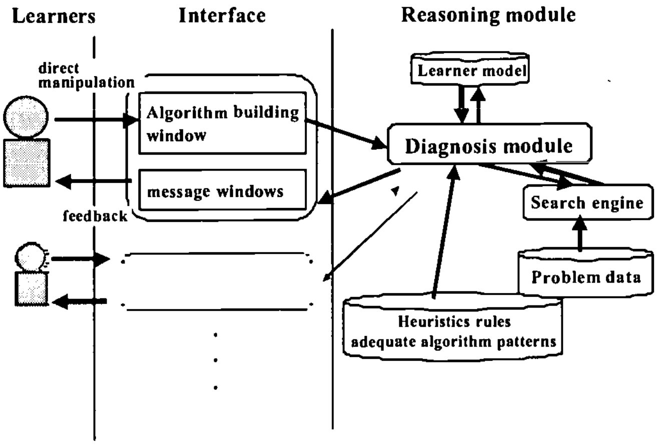


Figure 1. Configuration of the ILE

2.1 A Model of Search Algorithm

Search algorithms taught in the introductory lecture of artificial intelligence share the same procedure as follows. Here, both "Open" and "Closed" are lists composed of search nodes.
1) The start node is put into Open.
2) If "Open == []" then "the end of the search is in failure".
3) Pick up one node at the head of Open (the node is called *n*)
4) If "*n* == goal" then "the end of the search is in successful".
5) Generate child nodes from *n*.
6) Put the child nodes into Open.
7) Put *n* into Closed.
8) Return to Step-2.

The differences between the search algorithms are characterized by the operation of Step 6. For example, depth-first search is characterized as the algorithm in which the child nodes are put into the head of Open in Step 6. Breadth-first search is characterized as the algorithm in which the child nodes are put into the tail of Open in Step 6. In heuristics searches, the way to sort Open is an essential characteristic. In addition, for every algorithm, the method of selection of child nodes to put into Open is also an element that characterizes the search algorithms.

In our system, search algorithms are characterized by the combination of the following three list operations used in Step 6: "selection," "connection" and "sort." There are two types of selection operations: the first is "to select nodes that are not included in a list," and the other is "to select nodes that are not included in a list or are lower in cost than the same node in the list." Connection also has two types. The first is "to put nodes into the head of a list" and the other is "to put nodes into the tail of a list." The referred list is usually Open. We prepared three types of sorts: "to sort in the order of the consumed cost (minimum consumed cost search)," "to sort in the order of predicted cost (best-first search) " and "to sort in the order of the total of the consumed and predicted cost (A algorithm)."

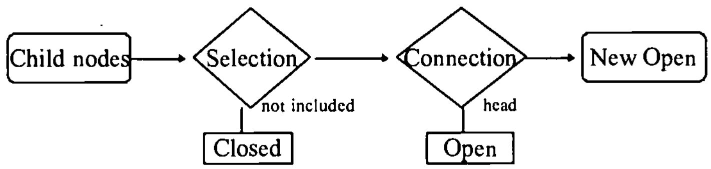


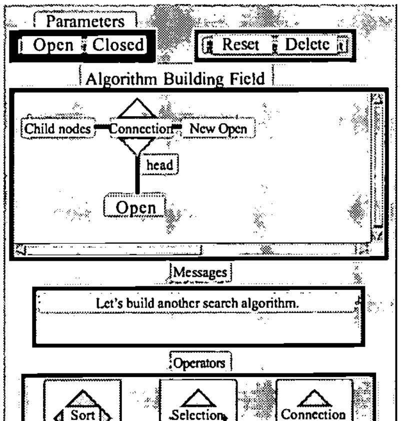
Figure 2. An example of the model ofSearch Algorithm

Figure 2 shows an example of a search algorithm built by the operations. The lozenge is the operation, and the rectangle is the list. The parameter that indicates "referred cost" or "head or tail" to specify the operator is presented at the bottom right of the lozenge. Therefore, Figure 2 means that "the child nodes that are not included in Closed are put into the hcad of Opcn." This is a kind of depth-first search that prunes using Closed.

Every part described above is necessary to build the search algorithms taught in the introductory lecture to artificial intelligence. In order to make learners understand search algorithms more deeply, our ILE provides an environment where learners can build search algorithms freely, and can receive feedback for the algorithms. In the following section, the ILE designed based on the model of search algorithms is described.

2.2 Building Search Algorithms

The interface for building search algorithms is shown in Figure 3 (currently, the interface is written in Japanese. Explanations in Figure 3 are translated to English for this paper. Japanese version is shown in [5]). Learners

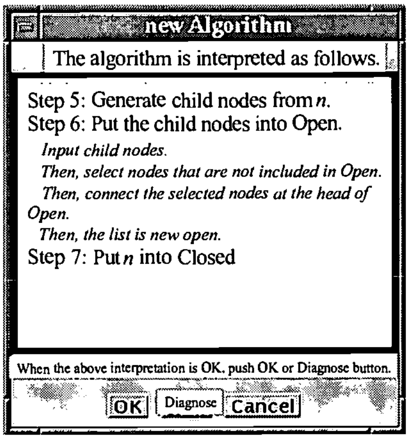


Let's build another search algorithm.

Figure 3. The Algorithm Building Field.

build search algorithms in the "building field" by assembling parts provided in the interface. At the bottom of the Interface, three operators are provided in the lozenges. The parameters specifying the operators are selected from the menu under the lozenges. The reference lists of the operators are selected from the box at the upper left. All manipulation in the interface can be done with a mouse. The algorithm in the building field is a depth-first search without having pruned.

Learners can confirm the algorithm built by themselves in two ways: a written explanation and a trace of the search tree. The explanation is generated by interpreting the operations in order of sequence in the building field. Figure 4 is the explanation of the algorithm shown in Figure 3. A search tree is generated by showing the trace results in a search space. The search spaces are provided as mazes in the environment. Figure 5 is an example of search tree that is the results of the search for the maze shown in the right in the figure.



The algorithm is interpreted as follows.

Step 5: Generate child nodes from *n*.
Step 6: Put the child nodes into Open.
Input child nodes.
Then, select nodes that are not included in Open.
Then, connect the selected nodes at the head of Open.
Then, the list is new open.
Step 7: Put *n* into Closed

When the above interpretation is OK, push OK or Diagnose button.

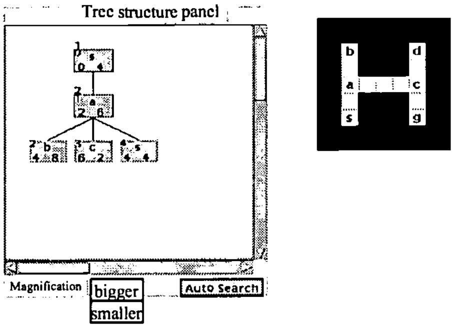
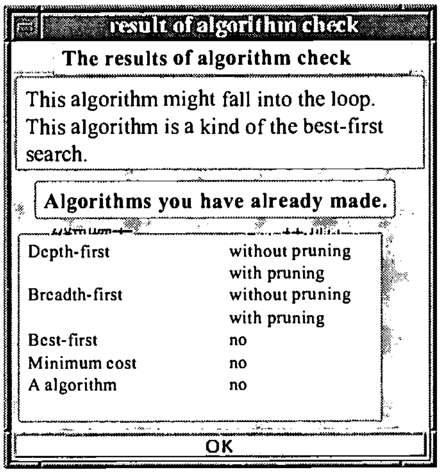


Figure 4. An Example of Explanation an Algorithm. Figure 5. An example of search tree.



The results of algorithm check

This algorithm might fall into the loop.
This algorithm is a kind of the best-first search.

Algorithms you have already made.

Depth-first	without pruning
	with pruning
Breadth-first	without pruning
	with pruning
Best-first	no
Minimum cost	no
A algorithm	no

Figure 6. An Example of the Results of Problem Diagnosis.

Learners can also ask the system to diagnose the algorithms built in the building field. The reasoning module has both the adequate combinations of operations and heuristics rules to criticize the algorithms that are not adequate. By using the adequate combinations, the adequate algorithms can be detected. By using the heuristics rules, the errors in the inadequate algorithms are detected. If no errors are detected by the heuristics rules, the reasoning modules can not judge the type of the errors. The heuristics rules are prepared from the following three points of view: the kind of algorithm, redundancies in the algorithm and the covering of the search space. An example of messages generated from the results of the diagnosis is shown in Figure 6. In the following section, the diagnosis of the search algorithms is described.

2.3 Diagnosis of Search Algorithms

In the reasoning module, the algorithms are diagnosed using heuristics rules. The heuristics rules of each viewpoint are shown in this section.

2.3.1 Type of Algorithm

The algorithms built by the learners are categorized by the following heuristics rules.

* When, after child nodes are put into the head of Open, either any nodes are not put into the head of Open or Open is not sorted, the algorithm is categorized as depth-first search.
* When, after child nodes are put into the tail of Open, either any nodes are not put into the tail of Open or Open is not sorted, the algorithm is categorized as breadth-first search.
* When, after child nodes are put into Open, Open is finally sorted in the order of consumed cost, the algorithm is categorized as minimum consumed cost search.
* When, after child nodes are put into Open, Open is finally sorted in the order of predicted cost, the algorithm is categorized as predicted cost search.
* When, after child nodes are put into Open, Open is finally sorted in the order of the total of consumed cost and predicted cost, the algorithm is categorized as A algorithm.

When the algorithm has no characteristics checked by the above rules, the kind of algorithm cannot be specified.

2.3.2 Redundancy of Algorithm

When the algorithms include the following operators, the diagnosis module judges that the operators are redundant in the algorithms.

* The same operators are used continuously.
* When several operators of sort are used, only the operator of sort used at the end has meaning.
* After using the connecting operator with a list as the parameter, the execution of the selection operation with the same list as the parameter results in deleting the added nodes.

2.3.3 Covering of the Search Space

Several search algorithms that can be built by learners can not find goals that exist in a search space. The reasoning module diagnoses whether or not the algorithm can cover the search space, by using the following heuristics rules.

* When several child nodes which might imply goals are not put into Open, the algorithm might fail to reach the goal included in the search space.
* When the algorithm that isn' t categorized as breadth-first, minimum consuming cost or A algorithm doesn' t include the selection operator with Closed as the parameter, the algorithm falls into the loop.

2.4 Feedback based on the Diagnosis

Based on the results of the above diagnosis with heuristics rules, the messages to criticize the algorithm are provided in the interface. Figure 7 shows an example of the messages. When the type of an algorithm is judged, the type is indicated. When the algorithm includes the redundant operators, the operators and the explanation of the redundancies depending on each heuristics rule are provided. When the algorithm might not cover the search space because several child nodes fail to be input into Open, the explanation prepared for the heuristics rule is shown. When the algorithm might fall into the loop, the possibility of falling into the loop is indicated.

When the algorithm includes a pruning operation, the fact is also indicated. In the interface shown in Figure 7, to motivate learners to build the next algorithms, the algorithms the learner has made correctly and hasn't made yet are shown.

3 Preliminary Evaluation

For a preliminary evaluation of the learning environment, we gathered thirty college students and asked them to use the learning environment. Those who were in the second grade or in the third grade have already taken the lecture of artificial intelligence. Their participation was voluntary. Before the experiment, we explained how to operate the environment for ten minutes. Then, we asked them to build search algorithms in the learning environment for an hour.

In the experiment, we recorded the following data: (1) the number of algorithms built by the learners, (2) the number of adequate algorithms, (3) the number of inadequate algorithms that could be diagnosed with heuristics rules, (4) the number of inadequate algorithms that couldn' t be diagnosed, and (5) the number of types of the adequate algorithms the learner made. The results are shown in Table 1. After the experiment, we asked four questions: (a) Are you interested in the system? (b) Is the system easy for you to use? (c) Would you like to use the system more? (d) Do you understand the search algorithms better than before? The results are shown in Table 2.

Table 1. The results of the students algorithm building.

Student number	(1)The total number of algorithms	(2)Adequate algorithms	(3)Inadequate algorithms (be diagnosed)	(4)Inadequate algorithms (not be diagnosed)	(5)the type of the algorithms
No.1	26	11	15	0	5
No.2	24	10	6	8	5
No.3	16	10	6	0	5
No.4	38	17	13	8	5
No.5	44	13	20	11	5
No.6	48	10	26	12	5
No.7	21	8	8	5	3
No.8	26	15	9	2	5
No.9	23	14	4	5	5
No.10	16	5	6	5	1
No.11	43	24	11	8	5
No.12	20	8	12	0	5
No.13	16	9	7	0	5
	361	154	143	64	

Table 2 . The Results of Questions.

	Yes	Maybe yes	No	No answer
Qusetion-a	10	3	0	0
Question-b	1	7	5	0
Question-c	7	5	0	1
Question-d	3	3	4	3

In Table 1, the total number of algorithms the learners made was 361, that is, 27.8 per student. The total number of adequate algorithms was 154, that is, 43 % of the algorithms. The total number of inadequate algorithms was 207 (57%). The number of diagnosed errors by heuristics rules was 143. This means that the system could detect the errors in 69 % of the inadequate algorithms. Among thirteen students, eleven students made every type of algorithm.

In Table 2, the results of Question-a and –c suggest that most of the students had interest in the learning environment. The result of Question-b indicates that the interface is not easy for the students to use. For Question-d, four students answered "no", and three students didn' t judge, that is, more than half the students didn' t think they gained a deeper understanding by using the learning environment.

Students made many algorithms in the experiment and they answered that the learning environment was interesting. In addition most of them could make every type of algorithm. These results suggest that the learning environment is promising. The answers for Question-b mean we should improve the interface. In Question-d, Six students thought they got deeper understanding by using the environment, but seven students didn' t think so. When we gathered students, we told them that we would ask them to use a learning environment for search algorithms. Therefore, most of the students participating in the experiment might have confidence about their understanding of search algorithms. This is one reason for the result for Question-4.

As for the results, the experiment suggests that the learning environment is promising to be used in the real world, but the effect couldn' t be confirmed clearly.

4 Conclusions

This paper described a learning environment for learning by design in the case of search algorithms. In the learning environment, learners can build search algorithms by combining parts by direct manipulation. Then, the environment diagnoses the algorithms in order to give feedback about the algorithms. First, the environment judges whether or not the algorithms are adequate. When the algorithms aren' t adequate, they are diagnosed using heuristics rules. The heuristics rules detect errors in the algorithms. By using the results of this diagnoses, the environment can give messages to help the learners revise their algorithms or to motivate them to build the next type of algorithms.

We havc alrcady implemented the learning environment. As a preliminary evaluation of the environment, we asked 13 students to use the environment, and gathered several types of data. As a result, the experiment suggests that the learning environment is promising to be used in the real world and that is promising, but the effect couldn' t be confirmed clearly. In the next step, we will use the learning environment in class and evaluate it in a real learning context.

References

(1) S.Papert: Mindstorms, Basic Books, Inc(1980).

(2) G.Fisher, K.Nakakoji: Empowering Designers with Integrated Design Environments, Proc. of the First International Conference on Artificial Intelligence in Design, pp105-117(1991).

(3) A.Repenning: Programming Substrates to Create Interactive Learning Environments, Interactive Learning Environments, Vol.4, No.1, pp.45-74(1994).

(4) E.Soloway, M.Guzdial, K.Hay: Learner-Centered Design: The Challenge for HCI In The 21st Century Interactions,Vol.1, No.2, pp.36-48(1994).

(5) H.Ikematsu, Myat, T.Hirashima, A.Takeuchi: Interactive Learning Environment for Search Algorithm, Proc. of Research Meeting of Kyushu Division of Information Processing Society of Japan, 3A-5, pp.101-108(2000)(in Japanese).

Applicability of an Educational System Assisting Teachers of Novice Programming to Actual Education

Hiroyuki Suzuki*, Tatsuhiro Konishi and Yukihiro Itoh****
* *Graduate School of Science and Engineering, Shizuoka University*
** *Faculty of Information, Shizuoka University*
Johoku 3-5-1, Hamamatsu, Shizuoka, Japan 432-8011
Tel: +81-53-478-1474 Fax: +81-53-478-1499
E-mail: { h-suzuki, konishi, itoh} @cs.inf.shizuoka.ac.jp

In this paper, we propose a technique for reducing processing time during program evaluation, and examine processing time of evaluating programs of which sizes are relatively large in novice programming courses. We proposed a method of constructing an automated evaluation system assisting teachers teaching novice programming. Our system evaluates learners' programs by comparing them with a standard algorithm representing teacher's intentions. By using our system, teachers can easily pick up learners' defective programs. We constructed a prototype system, and examined whether the system can evaluate programs actually submitted by learners. We confirmed that it could evaluate the programs validly. However we found that we should improve the processing time after evaluating various programs. In order to reduce processing time, we extend the matching algorithm using two ways. As a result, processing time is improved without spoiling the accuracy of matching. After that, we design a model course of novice programming based on actual courses in our university. And we examine the relation among program size, arbitrariness of teacher's intention and processing time. Then we confirm that the processing speed of our system is fast enough to be used in actual education environment.

Keywords: **educational system assisting teachers, automated evaluation system, program diagnosis, experimental evaluation**.

1. Introduction

By using program diagnosis technique, many programming education systems have been developed[1][2]. Most of them are designed to help learners, not the teachers. We think it is necessary to help teachers in order to give learners better advice. It needs much effort for teachers to evaluate many programs. So we constructed a prototype system assisting teachers teaching novice programming[4]. We have designed a model course on the basis of actual novice programming education course. Then, we have examined whether the system can evaluate programs written by learners. We confirmed that it could evaluate the programs validly[5]. However we found that we should improve the processing time after evaluating various programs. In this paper, we propose a technique for reducing the processing time (section 3), and examine processing time of evaluating programs of which sizes are relatively large in novice programming courses, then confirm that the processing speed of our system is fast enough to be used in actual education environment (section 4).

2. Our previous work

Generally, teachers teaching novice programming arrange goals for their exercises, and set exercises related to the goals. We call these goals "*teacher's intentions*". They evaluate whether each program submitted by a learner achieves the goals or not, and advise the learners according to the result of the evaluation. However, it needs much effort for them to evaluate many programs with various bugs. So we support them by developing an automated evaluation system classifying programs which perfectly satisfy *their intentions*, which partially satisfy *their intentions* or do extra work, and which don't satisfy *their intentions*. Therefore they only have to check the unsatisfied programs carefully.

We found that most of *teacher's intentions* can be represented with standard algorithms to solve exercises. So we use the standard algorithms for inputting *teacher's intentions*. We designed an algorithm representation based on PAD expression. We call the representation "Extended PAD". By using the Extended PAD, teachers are allowed to use the following structures in order to represent arbitrariness included in *their intentions*.

Non-ordering structure: It represents arbitrariness on the order among tasks,
Alternative structure: It represents arbitrariness on method to achieve a goal.

The Extended PAD consists of two types of elements: elements which correspond to a Pascal operation or control structure, and which correspond to a sequence of Pascal operations. We call the previous elements "primitive operations", and the other "macro operations".

We developed a method matching learners' programs with a standard algorithm. According to this method, the system tries to check correspondence of every combination of statements in a learner's program with the statements in a standard algorithm. If it judges that a statement in a learner's program matches with a statement in a standard algorithm, it makes a hypothesis on a correspondence of these statements and correspondences of variables referred by the statements. Then it continues matching the other statements on the basis of the hypothesis. As the matching process succeeds, the hypothesis grows up. Generally, possible correspondence of variables is not unique. So the other hypotheses containing the other correspondences of variables are set up at the same time. According to the result of matching, it outputs its judgement, "perfect match", "partial match" or "no match" based on the most plausible hypothesis. "Partial match" means that a learner's program doesn't match with a standard algorithm perfectly, but both the ratio of matched statements to the whole in the learner's program and the ratio of matched statements to the whole in the standard algorithm are higher than each threshold we defined.

3. Improvement of processing time

In order to reduce processing time, we should re-consider a method of matching programs. The system outputs the result of judgement on the basis of the most plausible hypothesis. The others are rejected. So if the system has an ability to find useless hypotheses on the way during matching process and avoid checking the correspondence of statements on the basis of the useless hypotheses, processing time must decrease.

So we extend the matching method as follows:

1) When the system intends to make new hypothesis containing correspondence of a new combination of statements in a learner's program and a standard algorithm, it calculates the ratio of matched statements to the whole under the assumption that all of the following statements will be matched perfectly. If the ratio cannot reach the threshold, the system doesn't make the hypothesis and omits the process of matching the following statements based on the hypothesis.

2) After the system has matched whole statements on the basis of a certain hypothesis, it tries to match statements on the basis of another hypothesis. In such a case, when the system finds that the ratio of matched statements to the whole cannot reach the ratio of previous trial, under the similar assumption to 1), it stops matching the following statements.

As a result, processing time is improved without spoiling the accuracy of matching. We confirm that evaluation of programs is not changed by the extension. Table 1 shows improvement of processing time necessary to judge the programs. Exercise (1), (2) and (3) are illustrated in Table 2. The computer system used for both experiments before and after the extension is an Engineering Workstation JU2/2300 (CPU: Ultra SPARC-II (300MHz) * 2, SPECint95: 12.3, SPECfp95: 20.2, Operating System: SunOS5.6, Made by: Japan Computer Corp.).

Table 1: Improvement of processing time.

	Number of programs	Average time of processing [sec/program]	
		Before extension	After extension
Exercise(1)	42	184.98	1.57
Exercise(2)	56	1.43	0.13
Exercise(3)	49	109.06	1.10

4. Applicability of our system to actual education environment

In order to discuss applicability of our system to actual scene in education, we design a model novice programming course based on actual courses in our university[5]. The exercises in the course are seen in Table 2. We use programs submitted by learners in the actual courses of our university.

We write each standard algorithm of exercise within the following restriction: the number of steps of

Extended PAD must be less than twice the number of steps of a standard program for corresponding exercise. The reason is that teachers don't prefer writing more detailed standard algorithms because of their costs.

The computer system used for this experiment is also the Engineering Workstation JU2/2300.

Table 2: The exercises in the model course.

Exercise(1)	There are several birds and tortoises. Find the number of them when the following conditions are given: (1) Sum of the heads of birds and tortoises. (2) Sum of the legs of birds and tortoises.
Exercise(2)	A character datum that is an uppercase character will be input by user. Convert it to lowercase.
Exercise(3)	A hexadecimal number will be input by user as two digits of character datum "0-9" or "A-F". Convert it to a decimal number.
Exercise(4)	Solve an equation "ax^ 3+bx^ 2+cx+d=0" by using the Newton method.
Exercise(5)	Sort integer data in an array by using the straight selection sort algorithm.
Exercise(6)	Solve simultaneous equations by using the Gaussian elimination.

4.1 Result of experiment

In order to evaluate processing time, we must consider the following factors: program size of standard algorithms and one of learners' programs, arbitrariness of *teacher's intentions*, judgement of learners' programs. Therefore we examine the relation among these factors and processing time. Table 3 shows:

- Program size of a standard algorithm and arbitrariness of *teacher's intention.*
- The number of learners' programs and average of their program size.
- Judgement of learners' programs, and the average processing time.

We measure the program size by the number of statements, blocks and variables, and the arbitrariness by the number of alternative structures and non-ordering structures including a standard algorithm. The judgement means the number of learners' programs that are judged "perfect match", "partial match" and "no match" by our system.

As program size increases, the number of combination of statements in a learner's program with the statements in a standard algorithm also increases. At the worst case, the number of the combination increases proportionally to factorial of number of the statements. However Table 3 shows gentler increasing. From the result, we think our extended matching method works well. Exercise (4) needs rather long processing time. We think that the number of statement per block is larger than the other exercises, and most of statements in the block have dependencies, i.e. a variable assigned some value by an assignment statement is often referred in the following statements in the same block. In such a case, the system must compound formulas and the process of compound needs rather long time. However, we check how often such an exercise appears in actual textbook[3] for novice programming and find that only one exercise.

Table 3: Relation among program size, arbitrariness of standard algorithm and processing time.

	Standard algorithm					Learners' programs				Judgement of Learners' programs			Processing time [sec/program]	
	Number of:					Average of:				Number of:				
	Statements	Blocks	Variables	Alternative	Non-ordering	Programs	Statements	Blocks	Variables	Perfect match	Partial match	No match		
(1)	5	1	4	0	0	42	9.2	9.1	1.4	0	4	38	0.96	
(2)	4	1	2	0	0	56	9.3	2.1	3.1	3	17	36	0.30	
(3)	11	5	4	0	0	49	19.7	7.7	5.9	0	0	49	1.50	
(4)	9	4	5	0	0	14	9.9	2.0	6.6	7	1	6	6.23	
(5)	10	4	5	0	0	14	9.8	4.0	5.1	1	10	3	0.30	(*)(#)
	10	4	5	1	0	14	9.8	4.0	5.1	1	10	3	0.31	(*)
	10	4	5	2	0	14	9.8	4.0	5.1	1	10	3	0.43	(*)
	10	4	5	3	0	14	9.8	4.0	5.1	1	10	3	0.44	(*)
	10	4	5	0	1	14	9.8	4.0	5.1	1	10	3	0.35	(#)
	10	4	5	0	2	14	9.8	4.0	5.1	2	9	3	0.52	(#)
(6)	28	12	11	0	0	24	27.3	11.8	11.0	1	11	12	9.87	(*)
	28	12	11	11	0	24	27.3	11.8	11.0	1	11	12	31.00	(*)

We evaluate our system on another standpoint. As arbitrariness of *teacher's intention* increases, possible combinations of statements in a learner's program with the statements in a standard algorithm also increase. We prepare some standard algorithms for the exercise (5) and (6), and process actual programs of learners. The result is seen in the lines with (*) and (#) in Table 3.

We prepare a standard algorithm including three alternative structures representing two methods to achieve a goal. At the worst case, the number of combination of statements in a learner's program with the statements in the standard algorithm increases 8 (=2^ 3) times. However Table 3 shows 1.5 times increasing. We also measure processing time by using a standard algorithm of exercise (6) including 11 alternative structures representing two methods. Although it seems that processing time increases 2048 (=2^ 11) times, Table 3 shows only 3 times increasing. Similarly, on exercise (5), we prepare a sample of Extended PAD including 2 non-ordering structures representing arbitrariness on the order between two tasks. In this case, it seems that processing time increases 4 (=(2!)^ 2) times. However Table 3 shows 1.8 times increasing. We think that the reason why the processing time isn't so increased is that our extended matching method works well. When our system evaluates using standard algorithms that include alternative structures or non-ordering structures, useless hypotheses are rejected at earlier stage of matching process.

In the next section, considering the result shown in Table 3, we discuss applicability of our matching method to actual education.

4.2 Estimation of processing time for actual exercises in novice programming

We survey model programs in a textbook[3], and write standard algorithms considering alternative coding methods or alternative order of statements. Then we investigate the program size, numbers of alternative structures and numbers of non-ordering structures. The result is seen in Table 4.

We can regard that our matching algorithm processes even programs belonging to the most complicated class in actual education within practical processing time. The reasons are as follows:

- The number of statement in programs of high level is less than the exercise (6) on average. And the largest number of statement is equivalent to the exercise (6).
- The number of alternative structure is almost 4 for programs of middle or high level.
- The number of non-ordering structure is less than 2.
- Therefore we think that our system can evaluate the most complicated program in the textbook with processing time similar to exercise (6) (31.00s/program).
- In fact, if a teacher takes care of 100 students in a class, the time necessary to evaluate their programs by our system is only 50 minutes or so. We think this is practical enough.

Table 4: Program size of exercises in a standard textbook of novice programming.

Exercise		Number of programs	Number of statements	Alternative structures		Non-ordering structures
				Iteration	Others	
Basic level	(the four rules of arithmetic, etc.)	19	8.5	1.1	0.2	1.2
Middle level	(sorting, etc.)	34	17.9	4.0	0.5	1.1
High level	(searching trees, etc.)	14	20.9	3.5	0.3	1.1

5. Conclusions

In this paper, we extend our matching algorithm to improve the processing time. Through the examinations and the discussion about processing time, we confirm that our system can evaluate actual programs in novice programming in practical time. Now, we are constructing a graphical user interface for describing standard algorithm. We will evaluate usability of our system after constructing it.

Acknowledgements

This research was supported in part by Grant-in-Aid for Scientific Research (c)(11680216) from Japan Society for the Promotion of Science.

References

[1] Anne Adam & Jean-Pierre Laurent, "LAURA, A System to Debug Student Programs", Artificial Intelligence, Vol.15, pp.75-122, (1980).

[2] Haruki Ueno, "Integrated Intelligent Programming Environment for Learning Programming", IEICE Transactions on Information and Systems, Vol.E77-D, No.1, pp.68-79, (1994).

[3] Nobuo Yoneda, Teruo Hikita & Takafumi Sakurai, "Pascal Programming"(in Japanese), SAIENSU-SHA Co.,Ltd., (1991).

[4] Tatsuhiro Konishi, Akihiro Suyama & Yukihiro Itoh, "Evaluation of Novice Programs Based on Teacher's Intention", Proceedings of ICCE'95, Singapore, Dec. 5-8, pp.557-566, (1995).

[5] Yukihiro Itoh, Tatsuhiro Konishi & Hiroyuki Suzuki, "On Experimental Evaluation of an Educational System that Supports Teachers of Novice Programming", Proceedings of ICCE'98, Beijing, China, Oct. 14-17, Vol.2, pp.234-238, (1998).

Case-Based Evaluating Assistant of Novice Programs

Hiroyoshi Watanabe, Masayuki Arai and Shigeo Takei
School of Science and Engineering, Teikyo University
Toyosatodai Utsunomiya-shi Tochigi, 320-8551 Japan
E-mail: {hiro,arai,takei}@ics.teikyo-u.ac.jp

This paper presents a method of implementing an evaluating assistant system that supports teachers' evaluation work of students' programs using case-based reasoning. The target evaluation tasks are to judge whether a student's program satisfies the requirements of the given problem and to give advice for the student's program. The case-based evaluating assistant system compares a program submitted by a student with evaluation cases in the case-base. If some case matches the program, the system applies the judgment and advice on the case to the program. We implemented a case-based evaluating assistant system for novice programs written in an assembly language based on the proposed method. The implemented system was utilized for actual classes and the results showed that the system reduced the teachers' evaluation work drastically.

Keywords: **program evaluation, programming classes, supporting teachers, case-based reasoning.**

1 Introduction

This paper describes a method of supporting teachers' evaluation work of students' programs. In programming education, programming exercise courses play an important role, because writing programs is indispensable to learning programming. In programming exercise classes, however, the teacher's loads of evaluation tend to be very heavy because the teacher has to read so many programs and reports. We aim to implement a computer system as an evaluating assistant that supports the teachers' evaluation work.

There are two approaches to the evaluation of students' programs. The first one is to diagnose programs and give advice by knowledge-based program recognition [1]-[3]. Most of the systems based on the approach require a huge amount of knowledge on bugs, and it would be difficult to constitute the systems practically. The second approach is to support teachers' evaluation work [4]. This approach may not necessarily aim at automating the evaluation work. It aims at implementing practical systems by limiting computers' evaluation work. We took the second approach.

We consider the program evaluation work as collaboration between teachers and computer systems and propose an evaluating assistant model of programs. We also propose a framework of the case-based evaluating assistant system.

2 Target Task and Evaluating Assistant

2.1 Task of Program Evaluation

The target evaluation tasks of a student's program are the following two tasks: (1) the first task is judging whether a student's program satisfies requirements of the given problem. When teachers set problems, they have educational intentions about what students should learn, namely concepts, algorithms, instructions and so on. Teachers read students' programs to see whether the educational intentions are achieved. Therefore, teachers accept a student's program when the program satisfies requirements of the given problem. The first task is defined on the assumption that students have to submit their programs over and over until their programs are accepted. (2) The second task is giving written advice. Teachers give advice to students whether they accept a program or not: teachers give advice about the reasons why the program is rejected, and advice about bettering the program even if the program is accepted.

2.2 Evaluating Assistant of Programs

Figure 1 illustrates an evaluating assistant in the electronic submission environment of programs. The evaluating assistant pre-evaluates submitted programs and a teacher can refer to the results when he or she evaluates the programs. If the teacher trusts the evaluating assistant, the results from the assistant can be sent to students directly. Such an evaluating assistant is expected to save a teacher a lot of time and energy.

The output of the evaluating assistant consists of evaluation results, their reasons and the degree of confidence. The evaluation results include the judgment of acceptability (accept or reject) and written advice. The degree of confidence is one of surely, probably or unknown. When the degree of confidence is unknown, evaluation results and their reasons are not given.

The evaluation results of the assistant are required to be always correct when the degree of confidence is surely. If so, the results of the assistant with surely confidence can be sent to students directly, in other words, teachers can trust the evaluating assistant.

The evaluating assistant should have the capability to learn. The final results of the teacher's evaluation are available for the learning. If the assistant is capable of learning, almost the same programs as ones which the assistant has evaluated incorrectly, are expected to be evaluated correctly in the future.

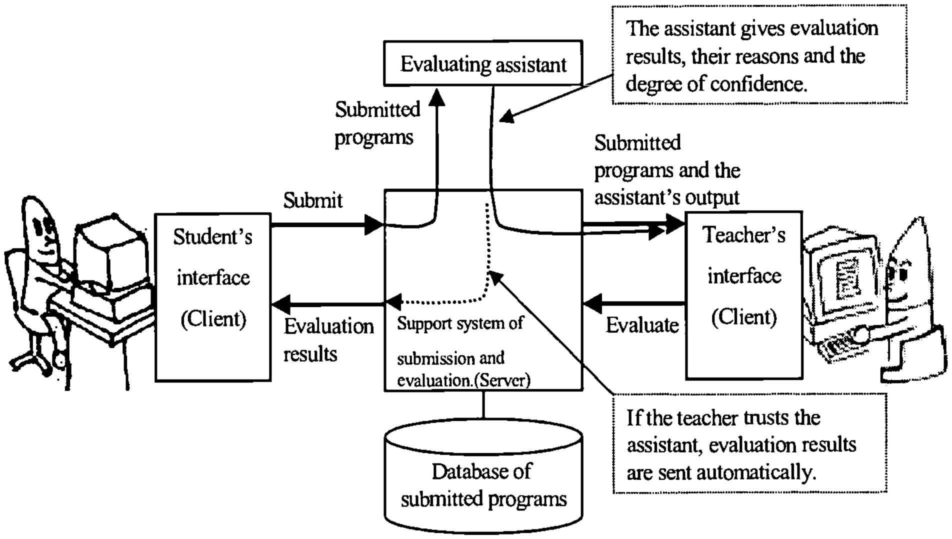

Evaluating assistant
The assistant gives evaluation results, their reasons and the degree of confidence.
Submitted programs
Submitted programs and the assistant's output
Submit
Student's interface (Client)
Evaluation results
Support system of submission and evaluation.(Server)
Evaluate
Teacher's interface (Client)
If the teacher trusts the assistant, evaluation results are sent automatically.
Database of submitted programs

Figure 1 Evaluating assistant of programs

3 Case-based Evaluating Assistant

The case-based reasoning approach is one of the best approaches to implement an evaluating assistant described in Section 2.2. Case-based reasoning systems make use of stored past cases directly in solving newly presented problems [5]. The case-based evaluation of programs is defined as "if some evaluation case of a program whose implementation is the same as the newly given program, then the evaluation results on the case are applied to the given program".

3.1 Representation of Cases

A case for the case-based evaluating assistant consists of retrieval information, problem description, solution description and maintenance information.

(1) The retrieval information includes problem identification that the program is written for and features of the program. The features of the program depend on target programming languages. For example, numbers of if-statements, while-statements and other statements are available in the case of language C.
(2) The problem description in the domain of a program evaluation task is a program list itself. A program list should be represented as a normalized form [2], or a generalized form [6], because there are many variations of program lists for the same implementation.
(3) The solution description includes the judgment of acceptability and written advice.
(4) The maintenance information includes a teacher's name and the date of adding or updating the case.

3.2 Processes of Case-based Program Evaluation

The processes of case-based program evaluation are the following:
(1) Problem analysis: The retrieval information is extracted from a student's program list.
(2) Case retrieval: Cases are retrieved using information generated by analyzing a given program. Cases that have no possibility of matching the given program should be pruned here.
(3) Evaluating and selecting cases: Evaluating cases is the process of matching a given program against cases. The purpose of the process is to investigate whether the given program has the same implementation as the cases, or not. All candidates of cases are evaluated and the best match case is selected. The method of matching programs depends on the target programming languages.
(4) Applying and adapting cases: If there is a case that matches the given program, the judgment of acceptability on the case is applied to the given program. In addition, advice sentences on the case are available for the given program, although the sentences should be adapted for the given program. If no case matches the given program, the judgment and advice is not generated.

3.3 Case-base Maintenance

The maintenance of the case-base is performed using teacher's final evaluation results. One of the most important maintenance tasks is adding new cases when the evaluation results of the assistant are different from the teacher's. New cases are also added when the confidence of the evaluating assistant is not *surely*. More advanced maintenance, e.g., generalizing, specializing and forgetting cases [7], may be needed in order to refine the case-base.

4 The Evaluating Assistant System for Assembly Language Programs

Based on the proposed idea, we implemented a case-based evaluating assistant system for novice programs written in an assembly language [6]. The target assembly language is CASL which is adopted in examinations for information-technology engineers certified by the Japanese ministry of international trade and industry.

4.1 Implementations Depending on The Target Language

In this section, implementations depending on the target language CASL are described.
(1) Evaluating the program's action: Before the case-based program evaluation, the assistant system tests the action of a submitted program using prepared sample data. Only programs executed correctly are evaluated by case-based reasoning [6].
(2) Case representation: Although cases are represented in the form described in Section 3.1, no features except for a program ID are used for the retrieval information. A program list in a case is represented in CASL itself, or its generalized form that we defined [6].
(3) Case retrieval: The implemented system retrieves all cases whose problem ID is the same as a given program. That is to say, the system does not prune candidate cases.
(4) Case evaluation (program matching): The program matching process aims at making consistent correspondences of instructions, labels and registers between a case and a student's program [6]. If the following condition1 is met, the case matches the given program.

- **Condition 1**: All instructions of the case correspond to instructions of the given program, and all instructions that correspond to nothing do not affect to the program's action.
  Especially, if the following condition is satisfied, it is called a "perfect match":
- **Condition 2**: Instructions of the case and the given program correspond one-to-one and the differences of the order of corresponding instructions are trivial.

If the best match case meets condition2, *surely* is assigned as the degree of confidence. If the best match case meets condition1 but not condition2, *probably* is assigned. In the other cases, that is, when no case meets the condition 1, *unknown* is assigned.

4.2 Experimental Results

The implemented assistant system was utilized for actual classes of the CPU and assembly language course at our university in 1999. Seventy-three sophomore students in the department of computer science took this course. Problems presented in classes of the course are the following: (P1) select bigger of the two given integers, (P2) sum the given N integers, (P3) select the maximum of the given N integers, (P4) rotate N bits to the right and (P5) check the correspondence of "(" and ")".

Table 1summarizes results of using the assistant system. The following are found from Table 1:

- The values of (f) show that the implemented case-based assistant system achieves sufficiently high accuracy of judgments. Furthermore, the accuracy of the case-based assistant system satisfies the requirements described in Section 2.2, because it is a hundred percent in cases of *surely* confidence.
- The values of (g) show that the ratios of available advice without modifying are not as high as the accuracy of judgments, although it is fairly high.
- Because teachers do not need to evaluate the acceptability when the case-based assistant system outputs evaluation results with surely confidence, it is estimated that the system reduces the teachers' evaluation work by percentages shown as (h). In other words, by using the assistant system, the evaluation work of teachers is reduced by 60 to 90 percent depending on problems.

These results demonstrate that the case-based assistant system is very effectual in reducing teachers' evaluation work. Still, there is room for improvement in the capability to generate written advice.

Table 1 Evaluation data of the assistant system based on practical use in classes

	P1	P2	P3	P4	P5
(a)Cases saved in the case-base	15	10	29	34	38
(b)Submitted programs	119	119	140	156	157
(c)Programs rejected by checking their action	44	39	44	54	79
(d)Programs evaluated by the assistant system with *surely* confidence	62	72	72	67	46
(e)Programs evaluated by the assistant system with *probably* confidence	8	3 '	5	5	0
(f)Judgment accuracy of the assistant system (%)	100 (100)	100 (100)	97.4 (100)	100 (100)	100 (100)
(g)Ratio of available advice generated by the assistant system without modifying (%)	92.9 (100)	69.3 (72.2)	88.3 (90.3)	66.7 (64.2)	91.3 (91.3)
(h) $((d)/((b)-(c))\times100$ (%)	92.7	90.0	75.0	65.7	59.0

( ) : values for programs evaluated by the assistant system with *surely* confidence only.

5 Conclusions

We have proposed a concept of a program evaluation assistant and a method of implementing the assistant by case-based reasoning. Based on the method, we implemented a system for a simple assembly language CASL and used it in actual classes; the results demonstrated that the system reduced teachers' evaluation work drastically. We plan to improve the modification functions of advice sentences (written advice) and the method of the case-base maintenance. This research was supported in part by the Japanese Ministry of Education Grant No.11680400 and No.12780293.

References

[1] Adam,A. and Laurent,J.P., "LAURA, A System to Debug Student Programs, Artificial Intelligence", vol.15, pp.75-122 (1980).

[2] Johnson,W.L., "Understanding and Debugging Novice Programs", Artificial Intelligence, vol.41, pp.51-97 (1990).

[3] Ueno,H., “Concepts and Methodologies for Knowledge-Based Program Understanding — The ALPUS's Approach”, IEICE Trans. Inf. & Syst., vol.E78-D, no.2, pp.1108-1117 (1995).
[4] Konishi,T., Suyama,A. and Itoh,Y., “Evaluation of Novice Programs Based on Teacher's Intention”, Proc. of ICCE95, pp.557-566 (1995).
[5] Kolodner,J. ,” Case-Based Reasoning”, Morgan Kaufmann Publishers,Inc. (1993).
[6] Watanabe,H., Arai,M. and Takei,S. , “Automated Evaluation of Novice Programs Written in Assembly Language, Proc. of ICCE99, vol.2, pp. 165 - 168 (1999).
[7] Watanabe,H., Okuda,K. and Yamazaki,K., “Methods for Adapting Case-bases to Environments”, IEICE Trans. Inf. & Syst., vol.E82-D, no.10, pp.1393 - 1400, (1999).

Development and Evaluation of a CALL System for Supporting the Writing of Technical Japanese Texts on the WWW

Jie Chi Yang* and Kanji Akahori**
**National Central University,*
38, Wu-Chuan Li, Chung-Li, Tao-Yuan, Taiwan 320
E-mail: yang@src.ncu.edu.tw
***Tokyo Institute of Technology*
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
E-mail: akahori@cradle.titech.ac.jp

This paper describes the development and evaluation of a Computer Assisted Language Learning (CALL) system for supporting the writing of technical Japanese texts on the WWW. To analyze discourse structure of technical Japanese texts, cohesive expressions are used as cue words. The rules for analyzing texts are based on micro-level and macro-level information, namely cohesive expressions and headlines. A CALL system for helping foreigners to learn to write technical Japanese texts is developed using Natural Language Processing (NLP) techniques. The main functions of the system are: automatically detecting headlines and cohesive expressions in technical Japanese texts, displaying this information on the WWW, and extracting examples from the corpus of technical Japanese texts. The results of a system evaluation show that the system obtained a high degree of accuracy on extraction of cohesive expressions and headlines by using the revised rules set proposed in this study. Furthermore, two evaluation experiments are conducted to examine the effectiveness of the system. The system is evaluated in terms of subjects' intuitive impression and actual usage of the system in the two experiments, respectively. The results of the study show that the instructive effectiveness of the system. The result of the interview also shows that the system is not only suitable for technical Japanese writing but also for Japanese language learning.

Keywords: **Computer Assisted Language Learning, Natural Language Processing, evaluation, technical Japanese texts**

1 Introduction

The aim of this research was to construct a Japanese learning environment for foreign students on the Internet. For students in science and technology universities, there is little time for enrolling in a regular Japanese language course, which involves spending a lot of time on experiments, studies and research, etc. The Internet environment is provided in almost all laboratories and can become an excellent virtual learning environment if there is a Japanese learning system which can be accessed on the Internet anytime and anywhere. The Internet has stimulated many new approaches to language instruction and learning, and it provides a great opportunity to learn one of the most important skills, writing. This is especially true for students in the science and engineering fields who need to write technical texts.

However, almost all CALL systems are concerned with learning how to improve one's reading and listening skills. Few systems are concerned with writing because of the difficulty of implementing an analysis of sentences typed by students who need to learn to phrase their own sentences freely without following any predefined rules. More and more researchers, therefore, use Natural Language Processing (NLP) techniques to analyze learners' typed sentence [9][16]. Recently, NLP techniques designed for use with CALL have attracted special attention (see, for example, [21][22], etc.), as this is expected to help improve writing skills.

Yang and Akahori [28][29] developed a Japanese writing CALL system using NLP techniques which can be used for learning and producing the Japanese passive voice on the WWW. Comparison of two Web-based CALL systems showed that the method of 'free input' and 'feedback corresponding to learners' typed sentence' is better than the method of 'multiple choice' and 'feedback that only displays the correct answer' [31]. Furthermore, an evaluation of the learning histories of the subjects who have actually used the system through the Internet shows that the system obtained a high degree of accuracy and instructional effectiveness [29]. These results demonstrate the effectiveness of the CALL system for writing using NLP techniques on the Internet.

Having sufficient vocabulary and grammatical knowledge is important when learning a foreign language. However, although vocabulary and grammatical rules are provided for correct sentence building in a foreign language, this knowledge alone is not enough. Being able to form correct sentences is by no means enough when it comes to expressing complex thoughts. The major problem for most foreigners learning Japanese is, apart from the writing system, the building of sentences: that is, knowing the corresponding words, the postfixes signaling the word's function (*de*, *ni*, etc.) and the position of the words (verbs final form). It is of paramount importance to learn how to structure one's thoughts: i.e., how to make an outline, how to signal the relative importance of a piece of information, and how it relates to the whole. Therefore, in order to write or to comprehend a structured sentence, it is necessary to learn how to associate sentences, in addition to having a good command of vocabulary and grammar. The connection between sentences can be described as conjunction of adjacent sentences, which is an important criterion for writing a good text as per research in *cohesion* or *discourse structure* [1][3][13][17][26]. Unfortunately, discourse structure is not amenable to single-sentence grammatical analysis, because there are no 'discourse grammars' [11].

Many methods concerning the analysis of discourse structure have been proposed in previous related works. Mann and Thompson's [18][19] rhetorical structure theory (RST) is an influential theory of text structure that is being extended to serve as a theoretical basis for computational text planning. RST postulates that a set of about 25 relations suffices to represent the relations that hold within normal English texts. Most relations have a cue word or phrase which informs the listener how to relate the adjacent clauses. RST can be applied to a computational model. There have been attempts at text generation using RST for the implementation of a prototype of the theory [10][20]. Cue words are also widely used in the identification of rhetorical relations among portions of a text [8][15][24]. Hobbs claims that coherence in conversations and in texts can be partially characterized by a set of coherence relations, which are classified into four categories. Hovy [10] collected and taxonomized the discourse segment relations; this set of relations contains three taxonomies of approximately 120 relations. Hirschberg and Litman [7] also summarize the proposed meanings of items classed as cue words in six computational and linguistic treatments.

In most of these earlier works, emphasis was put on the knowledge that is necessary for recognizing discourse structure. The problem of inference based on that knowledge was also emphasized. However, this does not mean that knowledge can be constructed easily from information available on computers. Constructing common knowledge to implement a practical system is often beyond the capabilities of current NLP techniques. Kurohashi and Nagao [14] proposed an automatic method for detecting discourse structure by checking surface information in text sentences. The information included 'clue expressions', 'occurrence of identical/synonymous words/phrases', and 'similarity between two sentences'. Their result indicates that, in the case of technical Japanese texts, considerable portions of discourse structure can be identified by incorporating the three types of surface information.

Since there are few practical CALL systems that use discourse analysis, the purpose of this study is to develop such a system for helping learners to write technical Japanese texts on the WWW. Section 2 describes the implementation of the system using NLP techniques. The authors took a similar approach to Kurohashi and Nagao [14], namely using surface information in texts. The rules for analyzing technical Japanese texts are based on micro-level (cohesive expressions) and macro-level (headlines) information. Section 3 describes the study that evaluates the effectiveness of the system in two experiments.

2 Implementation of the system

2.1 Method

The combination of cohesive expressions and headlines are employed in the implementation of the system. To examine discourse structure of technical Japanese texts, the classification of basic expressions by Yamazaki et al. [27] is adopted in this study. The reason for this is that their classification covers most of the elements of technical Japanese texts. Based on their findings, the authors have classified cohesive expressions into 15 categories as follows: *comparison, contrast, analogy, cause and reason, basis, composition and enumeration, presentation, definition, classification, hypothesis and conditions, change of state, process of change, change with prerequisites, means and methods, selection*. The total number of expressions is 82. All of the expressions are converted into regular expressions to make the rules. In all, 654 distinctions in the regular expressions were extracted from the 15 categories of cohesive expressions. These formed 654 original rules, which are used in the process of analysis.

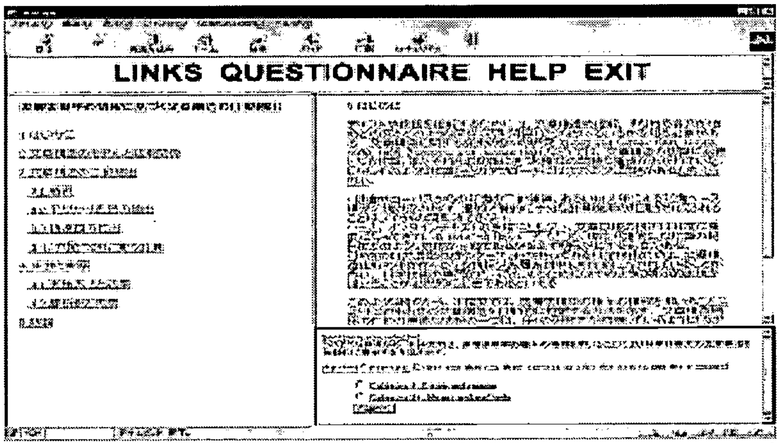
There are two patterns of rules: one is for 'simple pattern matching' and the other is for 'discourse analysis'. The former, called rule set A, is written as a regular expression form and the latter, called rule set B, is written as a regular expression combined with the result of morpheme analysis and syntax analysis. The rule in rule set B is written in a more restrictive form to improve the accuracy of discourse structure analysis. For example, if a sentence is applied to rule set A, it is then analyzed by the morpheme analysis and syntax analysis and the result will be matched to rule set B.

There are many text books on good writing, which nearly all contain a lot of material concerning the different kinds of categories or conceptual bricks at the discourse level out of which texts are built (see, for example, [4][5][6][12][25][26]. However, it is difficult to detect the text structure by just using their framework because it is too extensive and the varieties of different formats used by people for building technical texts too numerous. Instead of predefined framework, headline is used as macro-level information in this study. There are several reasons why the authors decided to use 'headline' instead. First, a well-chosen headline allows the reader to infer the text structure. Second, different formats of texts can be analyzed independently of the texts' style by using the headline. Third, it is easier to understand when the headline is displayed rather than a tree structure because the headline is a part of the original text.

2.2 The discourse structure analysis module

The discourse analysis module of the system contains 'simple pattern matching', 'morpheme analyzer', 'syntax analyzer', and 'discourse analyzer' components. First, the headlines are extracted and the Japanese texts are divided into sentences using several heuristic rules. Then all the sentences in all texts are matched with all the rules in the 'simple pattern matching' component. The 'rules for pattern matching' is used during the process of pattern matching. Because of the exclusive character of almost all of the rules, they are written in order of frequency to reduce the running time on the computer. The frequency of rules is made from the 'rules corpus'. The present system analyzes Japanese text sentences with the morpheme analyzer and syntax analyzer to check the dependency of sentences in the case grammar. Therefore, each cue word in the rules is not only matched against the word itself, but also against the 'parts-of-speech' of the cue word. Only sentences that match the rules written in restrictive form are needed for morpheme analysis and syntax analysis. This takes into consideration the problem of computer running time. The 'rules for discourse analysis' is matched again in restrictive form after the process of syntax analysis. The additional information (parts-of-speech, tense, etc.) is checked to identify the cohesive expressions, especially in the case where one sentence is matched with two or more rules.

Figure 1. One screen shot of discourse structure analysis



The learning page shows a list of technical Japanese texts. Learners can choose any one text by clicking the hyperlink on the list. When learners choose one of the texts from the list, headlines of the selected text are analyzed and displayed first to help learners grasp the whole text structure. Secondly, learners can click on the headline of any part of the text that they want to read. Then the original sentences corresponding to the headline are displayed with the extracted cohesive expressions. The cue words in the cohesive expressions are displayed in color to enable learners to focus on it more easily. Learners can click on any cue words to further find out the cohesive expressions corresponding to the sentences. They can also refer to examples that correspond to the cohesive expressions from the 'examples corpus'. Figure 1 shows one screen shot of the system (text source: [14]). As shown in this figure, the headlines of the Japanese text are analyzed and displayed on the left side of the browser. The headlines show the structure of the text. On the right side, the original sentences corresponding to the selected headline are displayed on the upper part with the cohesive expressions extracted and a link made. When the cue word '*kotoniyori*' (in the first line of the third paragraph) is clicked, the matched cohesive expressions are displayed on the bottom right side of the browser.

2.3 System evaluation of the discourse structure analysis module

A system evaluation is conducted to evaluate the performance of the discourse structure analysis module on 24 technical Japanese texts. The system evaluation is designed for text analysis in two stages (pattern matching in Stage 1 and discourse analysis in Stage 2). The analysis consists of 3 items on both stages: *headline extraction*, *cohesive expression extraction* and *frequency of the rules*. The accuracy ratio of the headline extraction in Stage 1 is 95.22% on average. After a heuristic rule is added, the result of the headline extraction using the revised rules in Stage 2 gained an exceedingly high accuracy rate of 99.17%. The accuracy of the cohesive expression extraction in Stage 1 is 70.23% on average. On the other hand, the accuracy in Stage 2 improved to 92.70% on average. This result shows that using the rules combined with morpheme analysis and syntax analysis gained a higher degree of accuracy than only using the rules of simple pattern matching. After the cohesive expression extraction, the frequency of rules is calculated. The result of 'frequency of the rules' is saved to the 'rules corpus'. The order of frequency is taken as the order of the rules to reduce the running time on the computer.

2.4 The system for supporting technical Japanese texts writing

A CALL system is developed to help learners in the writing of technical Japanese texts. The system is implemented in terms of headlines and cohesive expressions, which is based on the method of the discourse structure analysis module. For headlines supporting, a connection between headline and texts corresponding to the headline is made automatically. Learners can click on any headline to immediately link to the content of texts corresponded to it. For cohesive expressions supporting, examples with the selected cohesive expressions are automatically extracted from the corpus of technical Japanese texts. Learners can refer to these examples to help them improve their writing skills.

The flow of the system is as follows:

(1) Learners register themselves to use the system. An ID number is given after registration. The ID number is used to identify the learner because a log of all learning histories is registered during the operation of the system.
(2) The page for headlines input is appeared. Learners can free input their headlines here. When learners completed their construction of headlines, each headline is automatically linked and displayed on the left side of the browser. The left side of Figure 2 shows an example of linked headlines.
(3) When learners choose one of the headlines, a text box is appeared on the top right side of the browser. Learners can compose their texts corresponded to the clicked headline in the text box. The top right side of Figure 2 shows an example of texts input.
(4) When learners click on the 'basic expressions' button on the bottom right side of the browser, the categories of cohesive expressions is appeared on a new page. Each category is classified further into sub-categories. When learners choose one of the sub-categories from the list, examples are automatically extracted from the corpus of technical Japanese texts and the result is displayed on the bottom of the browser. Figure 3 shows that examples are displayed corresponded to the selected sub-category of cohesive expressions.

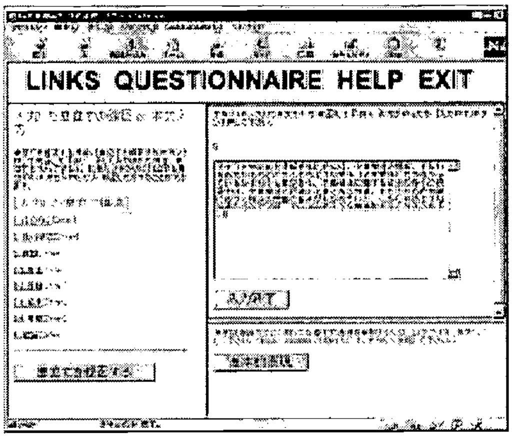


Figure 2. One screen shot of the linked headline and texts input

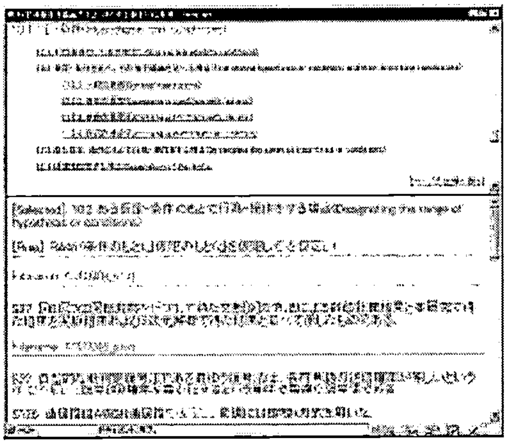


Figure 3. One screen shot of examples extracted from corpus

3 The study

Two evaluation experiments were conducted to examine the effectiveness of the system. The system is evaluated in terms of subjects' intuitive impression and actual usage of the system in the experiment 1 and the experiment 2, respectively. Thirty-three subjects participated in the experiment 1; the other seven subjects participated in the experiment 2. The subjects almost use the WWW and computer everyday.

3.1 Experiment 1

The purpose of the experiment 1 was to examine the functions of the system in terms of subjects' intuitive impression. Therefore, the experiment was designed to make a comparison between the system with the popular and well-known word processor: the MS-Word. During the experiment, the subjects were asked to look at the operation of the system and the MS-Word using video for duration of 10 minutes. The subjects were informed that they would be asked to fill in the questionnaire concerning the comparison of the two systems. The questionnaire consisted of 3 categories: *items of technical sentences writing*, *items of general sentences writing*, and *items of system operation*. The subjects were asked to rate 24 items on a 5-point scale. The subjects were also asked to make comments on the system.

Figure 4 shows the rating of the system and the MS-Word for each item with the 3 categories in experiment 1 and 2. The result of the experiment 1 shows that the system obtained a higher rating than the MS-Word on all of the items of technical sentences writing. For those items of general sentences writing and system operation, the result shows that the MS-Word obtained a higher rating than the system or there was no significant difference on the two systems. However, the system obtained a higher rating than the MS-Word on items 18 ('Sentences can be efficiently made') and 15 ('It is suitable for learning').

Comments on the system are summarized as follows: Almost all of the subjects answered that it is necessary to involve the functions to access other objects, such as figures, tables and numerical expressions, etc. Since the system is emphasized on the discourse analysis of technical Japanese texts using NLP techniques, the target of the system is limited to 'texts'. However, figures, tables and numerical expressions are important components of technical texts. Therefore, development of such visual tools for supporting these objects is expected.

3.2 Experiment 2

The result of the experiment 1 suggests that the system is preferred to the MS-Word on technical texts writing. However, actual usage of the system is not evaluated. Therefore, in order to examine the effectiveness of the system in terms of actual usage of the system by foreign students, experiment 2 was conducted. During the experiment, the subjects were asked to compose a technical Japanese text using the system. The subjects were asked to write sentences concerning their specialization instead of a given task because a variety of subjects' different fields. After the composition is completed, the subjects were asked to fill in the questionnaire concerning the comparison of the system and the MS-Word. The questionnaire is identical to experiment 1, which is divided into 3 categories. Finally, the subjects were interviewed based on their response to the questionnaires.

From Figure 4, the result of the experiment 2 shows that the system obtained a higher rating than the MS-Word on all of the items of technical sentences writing, which is consistent with the result of experiment 1. For those items of general sentences writing and system operation, the result shows that the subjects preferred the system, or the MS-Word or there was no significant difference on the two systems. Comparing this result to experiment 1, the system obtained a higher rating than the MS-Word on items 18 ('Sentences can be efficiently made') and 15 ('It is suitable for learning'), which is consistent with the result of experiment 1. On the other hand, some items obtained different result between the two experiments. These items can be divided into 3 types: First, items 7 ('I want to recommend it to my friends') and 24 ('I want to use it more') are rated from 'no significant difference' to 'a higher rating to the system'. Second, item 2 ('It is friendly') is rated from 'a higher rating to the MS-Word' to 'no significant difference'. Third, item 11 ('It is easy to see') is rated from 'no significant difference' to 'a higher rating to the MS-Word'.

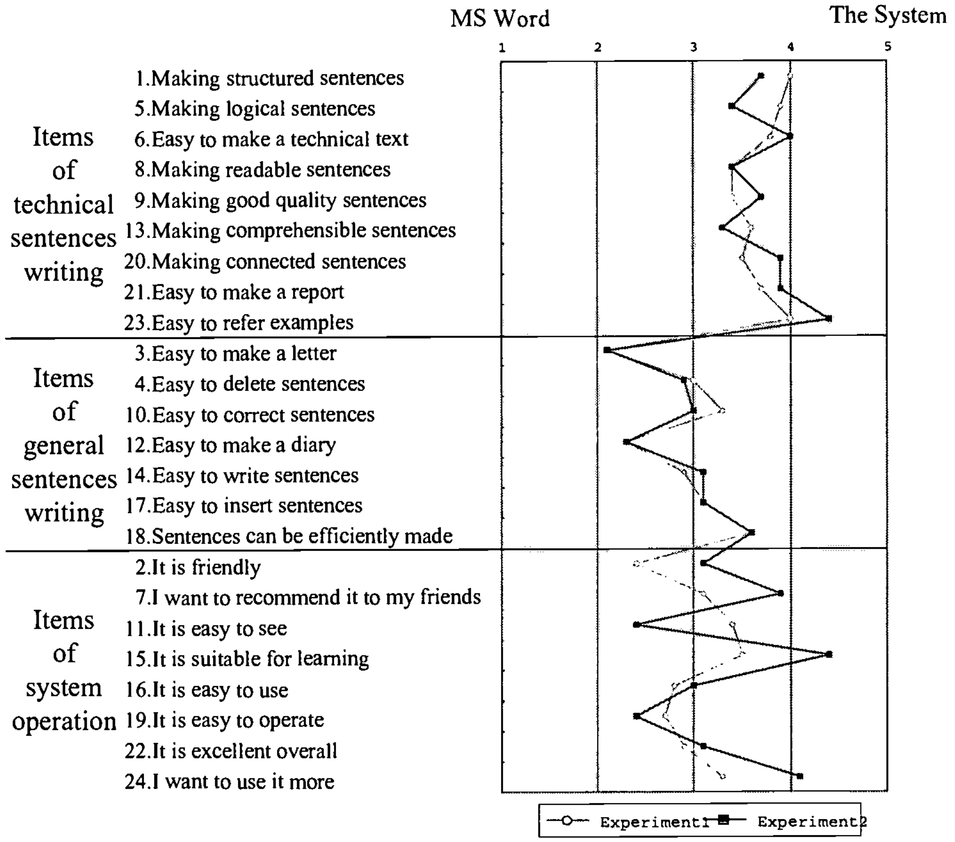


Figure 4. The rating for each item in Experiment 1 and 2

The subjects were asked to give reasons for their responses to the questionnaire items during the interview. The result of the interview concerning the functions of the system is divided into 4 types and summarized as follows: First, for automatically analyzing and displaying headlines, almost all of the subjects answered that it is very useful because they can click on any headline to immediately read the content of texts corresponded to it. The subjects also answered that headlines can be treated as an important role to help them to grasp the whole structure of the texts. Second, for automatically analyzing and displaying cohesive expressions, almost all of the subjects answered that it is very useful because they can find it is casier to convey their thoughts using explicit cohesive expressions. The subjects also answered that it is easy to find their errors because cohesive expressions in the texts are highlighted. Third, for referring to examples from corpus, almost all of the subjects answered that it is very efficient to writing because they can save a lot of time for finding examples from other references. The subjects also answered that they can imitate and learn more examples from the output of corpus. They can learn very much from the process of referring to examples in different texts, especially if there are many different usages in an expression. Fourth, for

Japanese language learning, almost all of the subjects answered that the system is suitable for learning because the system supports learners to learn technical Japanese writing in a structural way in terms of automatically analyzing and displaying headlines and cohesive expressions in technical Japanese texts. The subjects also answered that they can learn not only new cohesive expressions but also correct usages of cohesive expressions even they already know one of them.

Other comments on the system are summarized as follows: Almost all of the subjects answered that it is desired to improve the system to support the functions of electronic dictionary, thesaurus, grammar checking, etc. Therefore, construction of a good electronic dictionary for technical texts writing is considered as an important issue. Moreover, some subjects answered that it is better to extract examples form corpus according to learners' specialization than only random accessing to the corpus. From this result, constructing a corpus should not only consider the number of texts but also the balance of texts in each field.

4 Conclusion

In this paper, the authors describe the development and evaluation of a CALL system for supporting the writing of technical Japanese texts on the WWW. To analyze discourse structure of technical Japanese texts, the rules for analyzing texts are based on micro-level and macro-level information, namely cohesive expressions and headlines. A CALL system for helping foreigners to learn to write technical Japanese texts has been developed using NLP techniques. The system has the following functions: automatically detecting headlines and cohesive expressions in technical Japanese texts, displaying this information on the WWW, and extracting examples from the corpus of technical Japanese texts. The results of a system evaluation show that the system obtained a high degree of accuracy on extraction of cohesive expressions and headlines by using the revised rules set proposed in this study.

The results of the study show that the instructive effectiveness of the system. The result of the interview also shows that the system is not only suitable for technical Japanese writing but also for Japanese language learning. Based on the functions of the system, these results can be explained as follows: First, headlines can be treated as an important role to help learners to grasp the whole structure of the texts. Second, cohesive expressions often explicitly appear in the surface expressions of technical Japanese texts. Thus, it seems important and necessary to use these explicit cohesive expressions to structure one's thoughts in technical Japanese texts. Foreign learners especially may find it is easier to convey their thoughts using explicit cohesive expressions because these can be treated as an indicator of a discourse. Third, the corpus consists of the actual usage in technical Japanese texts from different fields. Instead of predefined examples, examples are automatically extracted from the corpus. Therefore, learners can learn very much from the process of referring to examples in different texts if there are many different usages in an expression. They can also save a lot of time for finding examples from other references.

In conclusion, the system is suitable for learning because the system supports learners to learn technical Japanese writing in a structural way in terms of automatically analyzing and displaying headlines and cohesive expressions in technical Japanese texts.

Acknowledgements

The authors would like to thank Dr. Kikuko Nishina (Professor in the International Student Center, Tokyo Institute of Technology) for her kind help during the analysis of technical Japanese texts. The authors would also like to thank Ms. Cheong Meng-Mei (Tokyo Institute of Technology) for her suggestions and comments on an earlier version of this paper. This study is funded by the Scientific Research Grant from the Ministry of Education of Japan (Subject No. B(2)10558019, Period: from 1 Apr 1998 to 31 Mar 2001).

References

[1]Chan, S.W.K. and T'sou, B.K. (1998). Analyzing Discourse Structure using Lexical Cohesion: A Connectionist Tool. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks*, V1: 657-662.

[2]Hajicova, E. (1996). The information Structure of the Sentence and the Coherence of Discourse. In Hovy, E.H. and Scott, D.R. (Eds.), *Computeational and Conversational Discourse: Burning Issues – An Interdisciplinary Account*, 97-107. Berlin: Springer – Verlag.

[3]Halliday, M.A.K. and Hasan, R. (1976). *Cohesion in English*. London and New York: Longman.

[4]Hamada, M., Hirao, T. and Yui, K. (1997). *Paper Workbook for University/Foreign Student.* Tokyo: Kuroshio. [In Japanese.]

[5]Hayes, J.R. and Flower, L.S. (1980). Identifying the Organization of Writing Processes. In Gregg, L. and Steinberg, E. (Eds.), *Cognitive Processes in Writing: An Interdisciplinary Approach.* Hillsdale, NJ: Lawrence Erlbaum Associates.

[6]Hayes, J.R. (1989). Writing as Problem Solving. In Hayes, J.R (Eds.), *The Complete Problem Solver.* Hillsdale, NJ: Lawrence Erlbaum Associates.

[7]Hirschberg, J. and Litman, D. (1993). Empirical Studies on the Disambiguation of Cue Phrases. *Computational Linguistics*, 19(3), 501-530.

[8]Hobbs, J.R. (1979). Coherence and Coreference. *Cognitive Science*, 3: 67-90.

[9]Holland, V.M. and Kaplan, J.D. (1995). Natural Language Processing Techniques in Computer Assisted Language Learning: Status and Instructional Issues. *Instructional Science*, 23: 351-380.

[10]Hovy, E.H. (1993). Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63: 341-385.

[11]Hovy, E.H. and Scott, D.R. (Eds.) (1996). *Computeational and Conversational Discourse: Burning Issues – An Interdisciplinary Account*, Berlin: Springer – Verlag.

[12]Kinoshita, K. (1981). *The Techniques of Writing for Scientist.* Tokyo: Chuoukouronsya. [In Japanese.]

[13]Kuno, S. (1978). *The Grammar of Discourse.* Tokyo: Daisyukan. [In Japanese.]

[14]Kurohashi, S. and Nagao, M. (1994). Automatic Detection of Discourse Structure by Checking Surface Information in Sentences. *Journal of Natural Language Processing.* 1(1): 3-20. [In Japanese.]

[15]Litman, D.J. (1994). Classifying Cue Phrases in Text and Speech Using Machine Learning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, V1: 806-813.

[16]Loritz, D. (1992). Generalized Transition Network Parsing for Language Study: The GPARS System for English, Russian, Japanese and Chinese. *CALICO Journal.* 10(1): 5-22.

[17]Mann, W.C. (1984). Discourse Structures for Text Generation. In *Proceedings of the 10th COLING.* 367-375.

[18]Mann, W.C. and Thompson, S.A. (1987). Rhetorical Structure Theory: Description and Construction of Text Structures. In Kempen, G. (Eds.), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, 85-95. Dordrecht: Martinus Nijhoff Publishers.

[19]Mann, W.C. and Thompson, S.A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *Text*, 8(3): 243-281.

[20]Moore, J.D. and Paris, C. (1994). Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. *Computational Linguistics*, 19(4), 651-694.

[21]Nagata, N. (1996). Computer vs. Workbook Instruction in Second Language Acquisition, *CALICO Journal.* 14(1): 53-75.

[22]Nerbonne, J., Dokter, D. and Smit, P. (1998). Morphological Processing and Computer-Assisted Language Learning, *CALL Journal*, 11(5): 543-559.

[23]Nishina, K. (1997). The Fundamental Research for the Development of Learning System of Technical Japanese. *Doctoral Dissertation.* Tokyo Institute of Technology, Tokyo. [In Japanese.]

[24]Reichman, R. (1985). *Getting Computers to Talk Like You and Me: Discourse Context, Focus, and Semantics.* The MIT press, Bradford Books.

[25]Shimizu, I. (1959). *The Manner of Japanese Writing.* Tokyo: Iwamani. [In Japanese.]

[26]Sugihara, K. (1994). *The Manner of English Writing for Scientist.* Tokyo: Chuoukouronsya. [In Japanese.]

[27]Yamazaki, N., Tomita, Y., Hirabayashi, Y. and Hatano, Y. (1992). *Handbook of Technical Japanese.* Tokyo: Soutakusha. [In Japanese.]

[28]Yang, J.C. and Akahori, K. (1997). Development of Computer Assisted Language Learning System for Japanese Writing Using Natural Language Processing Techniques: A Study on Passive Voice, In *Proceedings of the 8th World Conference on Artificial Intelligence in Education (AI-ED97).* 263-270.

[29]Yang, J.C. and Akahori, K. (1998a). Error Analysis in Japanese Writing and Its Implementation in a Computer Assisted Language Learning System on the World Wide Web. *CALICO Journal*, 15(1-3): 47-66.

[30]Yang, J.C. and Akahori, K. (1998b). A Discourse Structure Analysis of Technical Japanese Text. In *Proceedings of the Sixth International Conference on Computers in Education (ICCE98).* Beijing, China. 643-650.

[31]Yang, J.C. and Akahori, K. (1999). An Evaluation of Japanese CALL Systems on the WWW Comparing a Freely Input Approach with Multiple Selection. *CALL Journal*, 12(1): 59-79.

[32]Yang, J.C. and Akahori, K. (2000). A Discourse Structure Analysis of Technical Japanese Texts and Its Implementation on the WWW. *CALL Journal*, 13(2): 119-141.

Development and Evaluation of a Mental Model Forming Support ITS

- the Qualitative Diagnosis Simulator for the SCS Operation Activity-

Toru Miwata, Tatsunori Matsui, Toshio Okamoto and Alexandra Cristea
University of Electro-Communications, Graduate School of Information Systems
Choufu, Choufugaoka 1-5-1, Tokyo 182-8585
Tel: +81-424-43-5621
Fax: +81-424-89-6070
E-mail:matsui-t@ai.is.uec.ac.jp

In this study, we built an educational qualitative diagnosis simulator, which models SCS (Space Collaboration System: system the remote conferences and education via satellite communications) conferences. A student engages in the conference, by operating a control panel and proceeds by making the necessary selections according to the agenda of the virtual conference, and its intention and purpose, which can change at any time. The purpose of this study is supporting the student to form a correct mental model in this environment. Therefore, we incorporate an abstract model of possible computations as a logical circuit attached to the SCS system. Using this model, the system has two functions: to diagnose the student' s conceptual understanding mistakes about the SCS system and to explain to him/her the cause of these mistakes. With these functions, we expect to be able to support the student in forming a correct mental model and in understanding the SCS essentials.

Keywords: **Mental Model, Space Collaboration System, Remote Conference**

1 Introduction

Recently, with the increased awareness of the necessity of individual, subjective learning, a change occurred in the building of computer based educational systems. The existing learning supporting systems are based on automatically generating the learning method, according to the relation between the state defining parameters and the subject' s (learner' s) behavior. However, in recent years, the trend to construct systems, that positively encourage the student to work, and allow him/her to change the current state parameters by him-/herself, offer system behavior simulation, moreover, verification and correction of the student inputs, emerged. In this type of subjective/ individual learning environment, it is necessary to add a causality explanation function of the target environment. This is important due to the fact that, by letting the student/ learner adjust and change the system parameters, and then showing him/her the system behavior simulation, as derived from the current configuration and structure, fundamental system comprehension can be supported and achieved [2..11]. We have, therefore, used the above mentioned specifications and background information, to implement an educational qualitative diagnosis simulator, for supporting fundamental system comprehension and understanding. For this purpose, we have based our mental model design on the object oriented approach. The mental model is a representation of the individual comprehension about the structure and functions of the objects involved in the simulated system model. Moreover, depending on the simulation of the object functions within the learner' s mental model, it becomes possible to predict the problem solving act results. Therefore, important learning can occur and, at the same time, causality explanation within the virtual learning environment can be offered. We basd thc mcntal model used in our system on the qualitative modeling. The qualitative model is a fundamental model representation based on the causality relations that generate the target system' s behavior. The causality relations are reflected in the relations between the system' s structure, behavior and functions. Here we consider the following definitions. The **structure** reflects how the elements of the target organization are combined. The **behavior** shows how the system characteristics, expressed by the object structure, change in time. The **function** expresses how the goal, related to the object behavior, is achieved. By modeling the causality relations between the system' s structure, behavior and functions, and designing a qualitative model, the causality relation simulation becomes possible. In our system, we have constructed a qualitative diagnosis simulator for conferences via SCS. SCS, standing for Space Collaboration System, is a remote conferences and distance education system via satellite communications. The learner/ student follows the progress of the conference, by operating a control panel, and making the necessary selections, according to the agenda of the virtual conference, and its intentions and purpose, which can change in time. In this environment, we integrate a computable model abstraction of the remote conference via communication satellites, as a logic circuit. Moreover, based on this abstraction, we add a causality explanation function, and a diagnosis system of the student' s/ learner' s operation mistakes, which generate the appropriate guidance information for the student. In this way, we support the fundamental comprehension of the SCS system.

2 Qualitative reasoning

Qualitative reasoning is one of the most vigorous areas in artificial intelligence. Over the past years, a body of methods have been developed for building and simulating qualitative models of physical systems (bathtubs, tea kettles, automobiles, the physiology of the body, chemical processing plants, control systems, electrical circuits, and the like) where knowledge of that system is incomplete. Qualitative models are more able than traditional models to express states of incomplete knowledge about continuous mechanisms. Qualitative simulation guarantees to find all possible behaviors consistent with the knowledge in the model. This expressive power and coverage are important in problem-solving for diagnosis, design, monitoring, and explanation. Qualititative simulation draws on a wide range of mathematical methods to keep a complete set of predictions tractable, including the use of partial quantitative information. Compositional modeling and component-connection methods for building qualitative models are also discussed in detail [1].

3 SCS

Figure 1 displays the SCS based remote conference concept. SCS was established as a satellite communication network between universities, to enable real-time remote video conferences. Each participant' s station (called VSAT station) is enabled with a satellite communication control panel, an image and sound transceiver control panel, multiple video-cameras, monitors, and so on.

3.1 SCS constrains and limitations

The SCS conference can take place as an inter-station, bi-directional communication between two stations, or as a multiple VSAT stations communication, where only one station has the role of the moderator, and has authority upon transmission control. In the latter case, all the other station, with the exception of the moderator station, are called client stations, and can participate as such in the conference. The moderator station is decided in advance, before the actual conference, by the conference organizer, according to the requested time-schedules and conference contents. The line control is usually under the sole authority of the moderator station. However, a client station can send a request for line usage for transmission to the moderator. This operation is enabled by the proposal request button existent on each VSAT station panel. By pushing this button, a proposal request notification is sent to the control panel on the moderator station. Moreover, during the conference, it is possible for two different stations to send image and sound, namely, the carrier, at the same time, so there can be up to two distinct proposing stations. The respective client stations are depicted in the lower part of figure 1.

The communication satellite has two reception parts, and a converting switch that allows the selection of the received carrier. Depending on the existing constrains and conditions, a decision mechanism is involved, before actually sending the carrier selection from the satellite. After verifying the current constrains and conditions, the carrier is sent from the satellite. This carrier is sent without exception to all client stations. In figure 1, the sending of the carrier to all the client stations is depicted. The station carriers depicted in figure 1 as a black solid arrows show the connection between the individual stations and the transmission part of the satellite. The figure shows also that the satellite receives only two carriers at a time. Howeve; as all stations are connected with the satellite, as depicted by the solid black arrows, all stations are prepared to send a carrier.

The satellite reception part is built of a receptor, and a converting switch. In this way, by means of the restrictions set by the converting switch receptor, the satellite can receive, all in all, only two carriers. Moreover, these have to be from two distinct stations only. Also, in the case of multiple carrier reception, the moderator station operator can decide, according to his/ her free will, to commute to the receiving of one carrier only, disregarding the choices and modes of the client stations. These constrains, limitations and specifications, and the fact that the client stations can all in all send only two carriers, are depicted in the figure as dotted thick arrows. The two carriers that can be sent are named [send 1] and [send 2]. Their contents is re-sent from the satellite. The restriction that the two carriers, [send 1] and [send 2], should not come from the same station is enforced before this re-transmission. Only when al the above restrictions are fulfilled, can the received carriers be broadcasted from the satellite to all stations. At the reception of the broadcast signals, each client station can separate the two carriers, [send 1] and [send 2]. The station sending the carrier is also receiving the broadcast, without exception. Therefore, the sound and image received by the transmitting stations are:

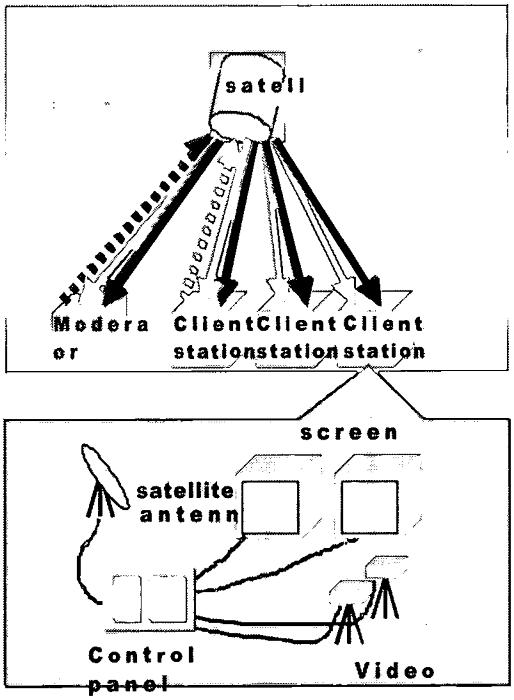


Fig. 1 Conference with SCS

(1) image+sound from the other transmitting station (if existent);
(2)the image and sound sent to the satellite by the station itself.

Moreover, as it is impossible to send the image and sound carrier to a specific station directly, by sending them to the satellite, they are broadcasted automatically to all stations. Bi-directional communication is also possible, but is actually a quasi-bi-directional communication, as the broadcast carrier of the two communicating stations is sent, at the same time, as a broadcast signal to all client stations.

3.2 SCS system frequent user errors

In table 1, the error types for different user skill levels of SCS conference practice, as gathered by surveying 4 domain specialists with over 2 years of SCS system operation experience, is shown. They were asked to give us first a list of frequently appearing user errors during the SCS usage and managing. This list is displayed in table 1 in the column headed by the label "Error/ misconception". Next, they were asked to evaluate the frequency of apparition of these errors for beginner, medium and advanced user. In table 1 their replies were represented as follows: [•] means high, [• •] means medium, and [•] means low frequency of errors. The table presents therefore the specialists' primary classification of errors according to the operation skills. To this classification, we have added a new error classification, based on the previously explained SCS system constrains and limitations. We have managed to group all errors enumerated by the specialists into four big classes of errors and misconceptions: A, B, C and D. The definitions of these classes are given below.

Table 1 Error types

Error/ misconception	beginner	medium	advanced	Error classifi-cation
Disregarding the function of the satellite· · believing direct/ dedicated transfer between fellow stations is possible.	• •	• •	• •	A,B,C, D
Believing that the sending of two carriers from the same station is possible.	• •	• •	• •	A
Believing that receiving two carriers from the same station is possible.	• •	• •	• •	A
Not understanding that, by switching the carrier to a different station, the current proposing	• •	• •	• •	B

station carrier will disappear.				
Believing that all stations can send a carrier at the same time.	. .	. .	. .	C
Not understanding the concept and necessity of the carrier request proposal.	. .	. .	. .	C
Assigning carriers to three or more stations.	. .	. .	. .	C
Not understanding why the image and sound signal sent by ones own station is received again.	. .	. .	. .	D
Believing that bi-directional communication is possible only with a specific station.	. .	. .	. .	D
Assigning the carrier to each station consecutively.	. .	. .	. .	D
Believing that only one broadcast is possible.	. .	. .	. .	D
Not making the distinction between the moderator station and the other client stations.	. .	. .	. .	static state
Believing that the client station Is in charge of the transmission control.	. .	. .	. .	static state
Not making the distinction between the two wave forms (signals), [send 1] and [send2].	. .	. .	. .	static state
Believing that the [send 1] wave form is the signal coming from the moderator station.	. .	. .	. .	static state

Class A: Misconception/ incomplete information about the sending of two different waves/ signals with the help of the judgement/ decision mechanism.
Class B: Misconception about the sending of one carrier to one station with the help of the converting switch.
Class C: Misconception/ incomplete information about the receiving of two carriers.
Class D: Misconception/ incomplete information about broadcasting to all stations.

4 The SCS qualitative model

Figure 2 shows the qualitative model of the SCS conference abstraction, in the form of a logic circuit. This qualitative model can express the structure, behavior and functions of the SCS system. In this figure, we displayed four client stations and one communication satellite. As can be seen, the satellite has two receptors, and one judgment/ decision mechanism, as a converting XOR switch between the two receptors. The two client stations sending carriers at one time can therefore have a pseudo- bi-directional communication. The structure, behavior and functions, so, the objects of the original SCS system are expressed, in this way, as a qualitative model.

The characteristics of this model make it possible to simulate the dynamic changes occurring during a distance conference, allowing to decide and evaluate the proper parameter settings for each station, moreover, to simulate the system behavior in the case of mistaken parameter settings. By using the XOR function, it is ensured that each reception part of the communication satellite can receive only one carrier from only one station. This station has sent a prior transmission proposal to the moderator station, which was accepted.

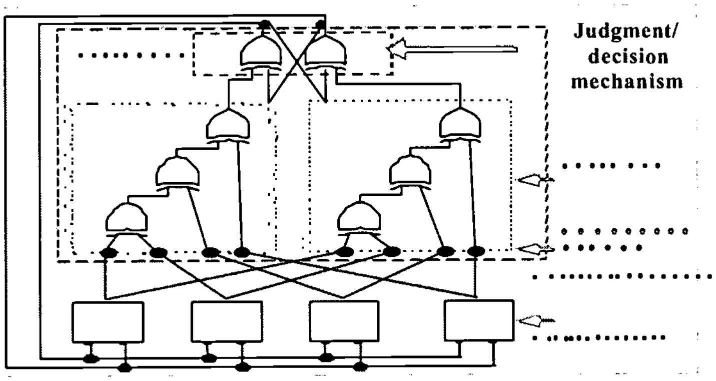
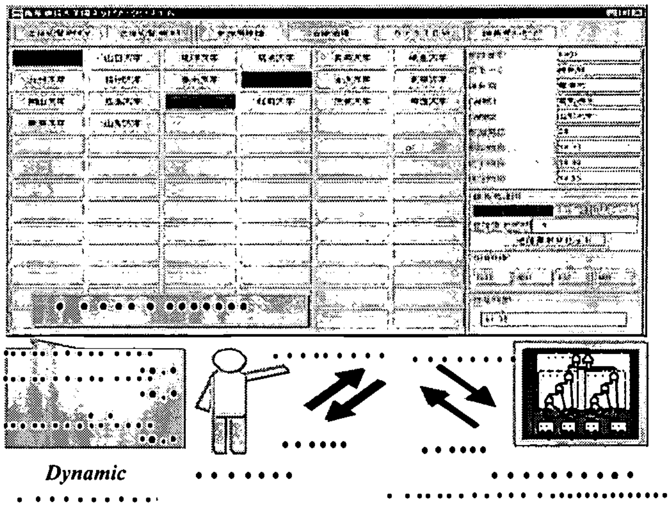


Fig 2 The qualitative model of the • • • •system

Next, it is necessary to make sure that the two accepted carriers come from two distinct stations. This restriction is enforced by the judgment/ decision mechanism. The judgment/ decision mechanism eliminates via an extra XOR function the possibility that the two carriers were sent by the same station. If the two carriers, 1 and 2, are validated by the judgment/ decision mechanism, the communication satellite broadcasts one or both to all VSAT stations. Therefore, all VSAT stations will receive the two carriers 1 and 2 and will not be able to receive any other carriers from other stations, or any wrong transmissions. Moreover, by using this model it is possible to infer the error source, as shown previously, based on the SCS system structure. The previous A, B, C, D classification can be thought of as: (A) sending of two distinct waves by using the judgment/ decision mechanism, (B) sending of maximum one carrier per station by means of the converting switch, (C) using of two carriers by means of the satellite reception mechanism, (D) existence of broadcast type of transmission only. In this way, the virtual model enables the learner to derive the cause and source of the operation error, as related to the SCS system structure. Furthermore, we have presented here a model based on only 4 client stations, that is implemented via the XOR module, but as in the case of more than 4 client stations, we can increase the number of the reception part XOR modules, adapting them to the number of stations, we can express, cope with and model therefore the converting switch for any arbitrary, greater than 2 number of client stations.

5 Learning Environment

5.1 System outline and overview



• • • • • • • Figure• •Learning environment

Figure 3 shows the overview of the system. The learner/ student is performing the conference steps by taking over the role of the moderator station operator. The goal is to cope with the dynamically changing agenda of the conference, proposed by the system. The agenda presents a description of a dynamic conference state, where bi-directional communication is required. The student can take decisions about the SCS system state and change parameter by operating the control panel. The previously described qualitative model evaluates these settings and parameters.

Next, disregarding if the parameter setup and assignment is appropriate or not, the result of the new user choices is reflected on the control panel of the interface, changing the current representation. The control panel displays also the transmission requests coming from other stations. The student has to choose the appropriate response to these requests. The student has to be able to judge the appropriateness of his/her own operations and actions, by intcrprcting thc information presented on the control panel. By repeating the above steps, the student can learn the constrains and usage of the SCS system. Moreover, to prevent deadlock situations, where the student is unable to judge his/her own errors, due to misunderstandings regarding the SCS system constrains, an explanatory function was added. This is implemented via an explanation button, which can be pressed by the student in need. The student guidance follows as has been previously shown, conform with the SCS qualitative model. In this way, the student can achieve not just a quick, superficial understanding, but also a deep, structure related knowledge about the SCS system. For example, explanation are given such as: "There are only two satellite receptors.", "There is an exclusive OR switch on each receptor, so each receptor can receive from one only station at a time.", "The judgment/ decision mechanism does not allow 2 carriers from the same station.", and so on. By leading the student to understand the connection between the parameter setup and the way the SCS system is actually built, as well as the real system components and the relations between them, via messages and state representations on the control panel, the student can be expected to perform the parameter setting by him/herself successfully in the future.

5.2 System flow

Figure 4 shows the system flow. The rapidly changing conference goal and intention of the agenda is described in chronological order. The contents of this description are on one hand, the conference state change requirements that have to be performed by the student, put into words that can be easily understood by him/her, and on the other hand, the description of the current SCS system state. In figure 4, this is expressed as [word] utterances, at the different moments in time (t0, . ,.tn):

word : state(t0) ~ word : state(tn)

For example, [word] can be a prompting message about the conference state change, with the value of "Please reply to the question from university A!", and so on. As shown in figure 4, the operation panel managing module receives from the agenda, or from the other client stations the current parameter for each given conference state, and then reflects the resulting state on the panel. For example, the button of the station, which is currently in charge of a carrier, turns red. Also, in the case of requests from other stations, the button of the station sending the carrier request signal turns also red.

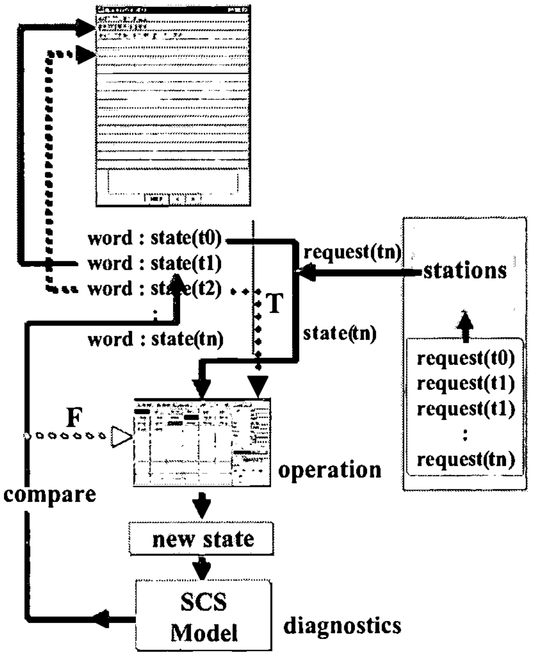


Figure 4 System flow

The student infers the present conference state from the state of the panel. Moreover, from here the student can notice if it is necessary to change the state of the conference, according to the agenda requirements. Next, to change the conference state, the student has to operate the control panel. By doing this, the parameters determining the conference are changed, and a new conference state emerges. This new state is evaluated with the SCS qualitative model. When evaluating with the SCS model, the result is compared with the next agenda. It is, in principle, possible to perform such comparisons on the SCS system without the computable module, and to judge if the operation is appropriate or not, but, in that case, the student cannot achieve a deep understanding of the SCS conference, that is, s/he cannot identify the SCS behavior as derived from structural constrains. In order 6r the learner to achieve a deep understanding, it is necessary to perform the parameter evaluation with the help of the SCS computable model. After the parameter evaluation, if the settings are judged as appropriate, the system moves to the next agenda. In figure 4, this is the case of "T" (True). In this case, the setup parameters decided by the student are handed over to the administrating module, which, in turn, reflects these changes on the operation panel. On the other hand, if, after the parameter evaluation, the settings are judged as not being appropriate, the system does not move to the next agenda. This case is shown in figure 4 as the "F" (False) case. In such a case, the wrongly set parameters are displayed on the operation panel. In this way, the deficient, real SCS state can be represented.

For example, in the case when three or more stations ask for the carrier at the same time, and the carrier is passed over to them, the moderator station's carrier disappears. The student notices that the espective state is not appropriate, and corrects the setup parameters. Moreover, in the case that s/he doesn' t notice the errors, s/he cannot continue with the next agenda. When entering a deadlock situation, the SCS qualitative model can, at the student' s request, explain to the student what kind of error s/he has done. In this way, by explaining not the protocol and process steps, but the SCS system behavior, as a result of the structural constrains, our system supports the formation of the SCS learner mental model. For instance, let us consider a case where the present transmission rights belong to universities B and C, and a proposal request is received from university A. This request is represented on the panel by the button representing university A turning red, together with a simultaneous indication message appearing in the agenda window, stating "Please answer the question from university A". If the student decides to assign a carrier to university A, without previously modifying the state of one or both stations B and C, which have the current transmission rights, the result is that the system will have 3 or more simultaneous carriers at the same time. In this case, the system represents the buttons of universities A, B, C on the panel with red color, and lets the student therefore know that the parameter setup is not appropriate.

At the same time, the agenda window will also display a message for the student. The content of this message is something like: "There are only two receptors on the satellite.", so is an explanation of the behavior, as resulting from the structural constrains.

6 Agenda

Table 2 Agenda example

agenda(t0)	The conference starts.
agenda(t1)	The moderator station is the University of Electro-Communications.
agenda(t2)	Please allocate carrier • •to Yamagata University• •
agenda(t3)	Please start sending from the lecturer camera• •
request(t4)	Carrier request • ¶sukuba University• •
agenda(t4)	Please reply to the question from Tsukuba University• •
agenda(t5)	The conference has ended• •

The SCS conference is based on a general agenda. Our system offers SCS based remote conference simulation environment and, moreover, stores typical SCS agenda models, in order to dynamically produce conferences that require conference state changes.

In this way, the student becomes the operator of the moderator station, and has to take decisions compatible to the agenda, engaging therefore in the simulated steps of the SCS conference. In table 2 we show an example of a model agenda for our system. In this table, agenda(tn) represents the agenda at moment (tn) in time, and request(tn) represents the carrier request at moment (tn) in time. In the real SCS conference, the time moment concept exists, but, in our system, we have the supplementary restriction that, only after accomplishing the current agenda, it is possible to go on with the new one. As shown above, the agenda is organized as a time series, and the student receives indications and instructions from the agenda window. The changes occurring in the conference state in the respective agenda example above correspond to a respective intention and goal. Disregarding if these intentions and goals come from the original operator's decisions, or if they were prepared by the system from the beginning, the beginner student doesn't have to loose his/her way during the SCS conference proceedings, and can give the panel operation his/her undivided attention. In other words, the indications and instructions coming from the agenda window can be thought of as an experienced operator teaching the beginner student during the SCS conference proceedings. After receiving the indications and instructions from the agenda window, the student can decide on the next conference state that seems appropriate, given the present conference state and the indications received, and operates the control panel to perform the respective change. The new state that results as a consequence of the student's operations is checked by the system, to decode if it is appropriate or not, conform with the indications and instructions of the agenda. One agenda is recorded in the system as one word and 6 state descriptors. The words are the ones that appear in the agenda window. The six possible state descriptors are shown below.

• • *station name (list of all client stations)*
• • *carrier request (list of all client stations)*
• • *carrier 1 (list of all client stations)*
• • *carrier 2 (list of all client stations)*
• • *reception 1 (list of all client stations)*
• • *reception 2 (list of all client stations)*

The state descriptor called "station name" contains a list of all client station names. Next, the carrier request, carrier 1, carrier 2, reception 1 and reception 2 state descriptors contain respective lists of [on] and [off] states corresponding to each station. In figure 3, we show the correspondence between [1] and [0] and [on] and [off]. The reason of describing all client stations carrier and reception states with [off/on] descriptors is to be able to represent also the incomplete understanding of the learner/ student, as well as his/her mistaken parameter setups and assignments.

7 Testing, experiments and evaluation

Table 3 comparison of situation before and after learning takes

Error	Pre Test	Post Test	Error classification
Disregarding the function of the satellite and believing direct/ dedicated transfer between fellow stations is possible.	**N/A**	**N/A**	**A,B,C,D**
Believing that sending of 2 carriers from one station is possible.	**3 persons**	**1 person**	**A**
Believing that receiving two carriers from the same station is possible.	**3 persons**	**1 person**	**A**
Not understanding that, by switching the carrier to a different station, the current proposing station carrier will disappear.	**5 persons**	**1 person**	**B**
Believing that all stations can send a carrier at the same time.	**3 persons**	**3 persons**	**C**
Not understanding the concept and necessity of the carrier request proposal.	**5 persons**	**1 person**	**C**
Assigning carriers to more than three stations.	**5 persons**	**1 person**	**C**
Not understanding why the image and sound signal sent by ones own station is received again.	**5 persons**	**3 persons**	**D**
Believing that bi-directional communication is possible only with a specific station.	**5 persons**	**1 person**	**D**
Assigning the carrier to each station consecutively.	**5 persons**	**1 person**	**D**
Believing that only one broadcast is possible.	**5 persons**	**5 persons**	**D**

We have performed an evaluation experiment of our system over a small sample. 5 beginner students with no SCS system experience were selected as the object of our SCS conference experiment. We have first explained them the control panel representations, meanings and operation mode, as well as the agenda window functionality, and the SCS system setup as a bi-directional communication system. They were able to consult the SCS user manual. Next, we have done a pre-test with the system without the diagnosis mechanism, and followed and checked the operations and mistakes of the beginner operator. Then, we have performed the same experiment, this time, with the help of the diagnosis mechanism. In the last step, we have compared the understanding level before and after learning. The result is displayed in table 3. A system screen display during the experiment is shown in figure 3. This figure displays a student deadlock situation, where the student has asked for an explanation about the deadlock, and the system has next checked the SCS system structure related error cause, and finally displayed it on the screen for the student to see. In the case presented in figure 3, the student hasn't realized the fact that there are only two receptors on the satellite, and has mistakenly allocated carriers to 3 stations. The explanation of his/her error is displayed on the control panel. The state of 3 stations having the carrier is represented on the panel as the respective stations' buttons turning all red (left corner of fig. 3, darkened buttons). However, if the student doesn't grasp the meaning of the representation and the cause and source of his/her errors, and asks therefore the system for help, the system will display the following message: "There are only two receptors on the satellite". With this explanation, the student understands that, as there are only 2 receptors on the satellite, s/he cannot allocate carriers to 3 stations, and will operate the panel correctly in his/her next steps.

According to our system's result shown in table 3, the students can understand the SCS system constrains and limitations, the fact that the signal has to be sent from different stations, the fact that there are only two carriers, and the concept of the XOR receptors of the satellite. However, the broadcasting mechanism was not completely understood. This is probably due to the fact that, in the current simulation system, there is no visual display of the broadcasting mechanism, of the time and direction of the transmission.

7 Conclusion

In this paper, we proposed an educational qualitative diagnosis simulator based on an object-oriented approach to mental model formation. In our model, the structure, behavior and functions of the SCS system are the objects, and from the description of the causality relations between these objects, the student can determine the cause of his/her error, based on system structure judgment.

From educational strategy point of view, QUAD implements and supports a combination of learning methods, like "Reinforcement learning", "Learning by exploring", "Learning by asking", "Learning by applying", "Self-monitoring", and so on. From educational depth point of view, the QUAD system doesn' t stop at the procedural surface level, but traces the structural implications, to gain a deep knowledge level.

For further research, we believe that, by expanding the current system, and identifying more precisely the mental model of the student, a more appropriate guidance system can be developed.

References

[1] B. Kuipers; Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, Cambridge, MA: MIT Press (1994).

[2] T. Okamoto• O. Matsuda; About the mental model of the ITS based geometrical/ topological proof process, Japanese Journal of Education and Technology,Vol.4, No.15, pp.167-182 (1992)

[3] O. Matsuda, T. Okamoto; based on hypothesis inference/ deduction : about the ITS - student model and its recognition method, IEICE conference, A, Vol. J75-A, No.2, pp.266-274 (1992)

[4] T. Okamoto; Knowledge Processing Technology and Educational Engineering, Japanese Journal of Educational Technology 20(4), pp. 207-226, (1997)

[5] T. Nishida; The basis of qualitative inference/ deduction, Journal of the AI society, Vol.4,No.5,pp.522-527,(1989)

[6] T. Okamoto; Education, Media and Artificial Intelligence, Journal of the AI society, Vol.10,No.3,pp.361 – 367 (1995)

[7] T. Okamoto; Knowledge Processing Technology and Educational Engineering, Journal of the AI society, Vol.20, No.4, pp.207 – 226 (1997)

[8] T. Nishida; The common sense thinking way model of qualitative inference/ deduction• •Journal of the AI society, Vol.2, No.1, pp.30 - 43,(1989)

[9] T. Nishida; Towards applications of the thinking ways and intelligent problem solving techniques of qualitative inference/ deduction• •Journal of the AI society, Vol.32, No.2, pp.105 • •117 (1991)

[10] T. Nishida; Aspects of qualitative inference/ deduction, Asakura publishers (1993)

[11] K. Asai, H. Ohnishi, K. Kondo, K. Tanaka, O. Matsuda; Case base and Collaboration Application based on appropriate• • •(control) and the Internet system, technical Report of IEICE, ET97-24 (1997-05)

Development of Intelligent Learning Support System with Large Knowledge Base

Katsuyuki SUGAYA and Setsuo OHSUGA
Department of Information and Computer Science, Waseda University
3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169-8555, JAPAN
E-mail: {sugaya, ohsuga}@ohsuga.info.waseda.ac.jp

The objective of this paper is to present framework for developing intelligent learning support system with large knowledge base. Recently, the need for effective learning support and training is mounting, especially in industry or engineering fields, which demand the learning of complex tasks and expertise knowledge. Intelligent learning support system is being employed for this purpose, thus creating a need for cost-effective means of developing learning support systems. In this study, intelligent learning support system is assumed as a part of the intelligent knowledge management support system. The factors necessary for the intelligent learning support system discussed here are generality and adaptability. In order to achieve the goal, a new design of the system and learner modeling technique are discussed as well as a way of generating specific intelligent learning support system.

Keywords: **Intelligent System Design, Large Knowledge Base, Learner Model, Model-based Diagnosis, Knowledge Management**

1 Introduction

The purpose of this paper is to introduce a new framework for developing intelligent learning support system using large knowledge base. This system is a part of the intelligent systems that is developing to enable the expertise knowledge management.

In daily life, human has to interact with and reason about a large number of systems. This includes physical devices as well as non-physical systems. Also in professional work a growing number of people has to be trained h operating and designing large complex systems such as airplanes, nuclear power plants, and enterprises. Consequently, the goal of education or teaching may vary from inducing insight in the physical principles underlying the behavior of the device to teaching behavior analysis in the context of system design, operation, and maintenance. In addition, recently systems in the real world are becoming larger and more complicated. Rapid progress in science and technology has created a need for people who can solve complex problems and operate and maintain sophisticated equipment. In these situations, we, human beings, have to solve various types of problems using expertise in the large and complicated systems. Therefore the need for effective learning support or training is rising, given the increasing complexity of the workspace, especially in engineering or industrial fields.

Many computer assisted instruction techniques exist that can present instruction, and interact with students in a tutor-like fashion, individually, or in small groups [3]. The introduction of artificial intelligence technology and expert systems technology to computer assisted instruction systems gave rise to intelligent tutoring systems. In the intelligent tutoring system, for example, intelligent tutors that can model the learner's understanding of a topic and adapt the instruction accordingly [2]. Although intelligent tutoring systems research has been carried out for over 15 years, few tutoring systems have made the transition to the commercial market. Authors consider that some serious problems exist in the current methodology of developing intelligent tutoring systems. As an example, each system is developed independently, and tutoring expertise is hard-coded into individual systems. h particular, the problem of learner modeling technique exists as a basic issue. The system must have learner model that represents an estimate of the learner current understanding of the domain knowledge to be used by tutor in order to give adaptive guidance and explanations to the learner. A number of learner modeling techniques have been developed [8]. However, not every model can be called complete expressing the learning condition of the learner. Hence, the motivation for this study comes from the need for effective intelligent tutoring systems, particularly development of more complete learner modeling technique.

For these problems like above we consider that the factors necessary for the intelligent learning support system discussed here are generality and adaptability. In order to achieve the goal, authors present a new framework of the intelligent learning support system those enough practical conditions. Several concepts are included in this study; expert knowledge management with large knowledge base, knowledge sharing, knowledge processing, model-based learner diagnosis, etc.

2 Expert Knowledge Management using Knowledge Base System

In this section, we introduce briefly the key concept of our knowledge base system. Our research groups have tried to solve various problems by knowledge-centered intelligent system. The main concept is Multi-strata modelling scheme [5]. This modelling scheme is applied many intelligent systems, and these systems rewarded with good results, e.g. automatic programming system [1]. And we considered that Multi-strata model is strongly support the development of intelligent tutoring systems [6][7].

2.1 Intelligent System with Large Knowledge Base

At first, we discuss to apply large knowledge base for the architecture of intelligent learning support systems, which can generate learning support systems for a wide range of domain.

In these days, with the developing of science and technology, the systems which human manages with are enlarged and more complicated. In particular, it is too difficult to transmit expert knowledge from expert engineer to novice engineers. In the engineering field, even a large system developed by many expert engineers. When the system grows lager and more complex, the knowledge that is needed to build the system is more specialized and subdivided. In these situations, some serious problems are occurred. For instance, it is difficult to communicate between expert engineer and another fields' engineers or novice one. In other words, it is too more expertise to transmission of expert knowledge from human to humans. For this reason, the expert knowledge hiding is occurred in some engineering companies.

When the knowledge is specified and subdivided, in the situation like classroom, it is not appropriate to transmit the knowledge from expert engineer to novice one e.g. next generation engineers. Therefore, we propose the transmission of expert knowledge through the large knowledge base system (Fig.1).

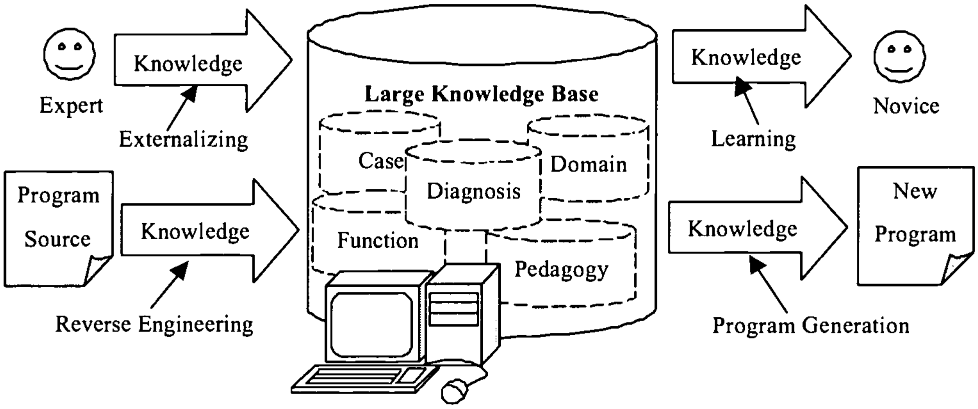


Fig.1: Knowledge Transmission using Intelligent Large Knowledge Base System

In this study, we consider that intelligent learning support system is a part of intelligent knowledge management support system. Moreover, we believe that knowledge management or learning support system is one of large and complex problem solving systems. The term problem is used here in a wide sense to mean what a person wishes to know or wants to do. There are various types of problems such as analysis, design control, decision-making, planning, and teaching. Most of them are not well dealt with by conventional software method but require the system a capability to find a solution itself in a large space. Since the space is open, self-controlled exploration in the space is necessary. The system must be provided with the various methods to solve the different type of problems, each of which is represented by a specific knowledge chunk. Furthermore, a complex problem concerns different problem domains and since a problem requires domain specific knowledge, the system must be provided with a global knowledge base including the various type of domain knowledge.

In order to use knowledge effectively, the system must be able to extract only the necessary knowledge from the knowledge base referring to the type and the domain of the problem to be solved. For this purpose knowledge must be well structured. All used knowledge is accumulated in the large knowledge base (Fig. 1).

2.2 Necessity of Knowledge Processing Language

The whole of the problem solving process is from accepting external representation of problems to generating solutions. In order to represent problems in the system a processing language is necessary. Thc language has to meet two conditions: it has to be usable for representing problems; and it has to be processable by computer processor. In ordinary computers only the procedural language is used both for processing by the processor and for representing problems. The knowledge base system, on the other hand, introduces the second language to separate the above two aspects, as well as a conversion mechanism between them. The second language is a declarative knowledge representation language. The conversion either in the declarative forms or from the declarative to the procedural form is necessary. This is the inference. It can be implemented as a procedural program on conventional computers.

The specification for the second language must be decided so that it can represent these conditions. It had to be suited for representing predicate including data structure as argument and also for describing higher-level operation such as knowledge for selecting object knowledge. KAUS (Knowledge Acquisition and Utilization System) has been developed for the purpose by our research & development team.

3 Adaptability of Learning Support System

To meet the condition of adaptability, it is necessity to represent the learner's understanding of learning domain. In this section, we discuss a learner modeling method that is applied to diagnostic techniques in artificial intelligence.

3.1 Issues of Learner Model

The performance of intelligent learning support system depends largely on how well it knows why the learner fails to solve problems. Because of the sophisticated interaction requires information about the learner, the system has to maintain some kind of model of the learner. This model may include cases about what has been done before or information about what the learner is believed to know. The process of gathering information about the learner is mostly referred to as cognitive diagnosis. Ohlsson has given a widely accepted definition of cognitive diagnosis: "cognitive diagnosis is the process of inferring a person's cognitive state from his or her performance" [4]. We consider that the point of learner model is to represent knowledge state of learner, especially his/her fails to solve problem. To satisfy this requirement, we focus diagnosis techniques.

A diagnosis is defined in terms of one or more reasoning steps that the learner cannot have solved problem. A major advantage of this approach is that it can be based solely on a model of these correct reasoning steps; no knowledge is required about the specific misconceptions that learners may have about the domain of learning. Instead we model all primitive inferences that are required to arrive at the correct solution. In addition, our approach to diagnosis of learner behavior exploits results from model-based diagnosis as it is defined in the field of artificial intelligence.

3.2 Model-based Learner Diagnosis with Case Base

Model-based diagnosis is a prominent area within artificial intelligence and emerged in the last about 15 years. The technique of model-based reasoning has been widely researched and accepted as the principal diagnosis in electronic circuit analysis, power station maintenance, medical diagnosis domains, etc. However, little emphasis has been put on its application to education or training system domain. The basic principle in model-based diagnosis is the description of system as a causal model. With the model at hand, the behavior predicated by the model is compared to the actually observed behavior. Since the predictions of the model are based on the assumption that the components work correctly, these assumptions may be partially dropped to accommodate for a detected behavior difference and thus diagnose faulty behavior.

However, there are some weaknesses in model-based diagnostic technique. The most serious weak point is the diagnosis time. It sometimes takes so much time to diagnosis. Therefore, we must be considering that it is necessary to model concerning the trade-off between the cost of a diagnosis time and its precision. Case-based reasoning, by contrast, excels in covering weak-theory domains, domains whose phenomena we do not yet understand well enough to record causality unambiguously. This feature allows case-based reasoning to be used in domains where model-based reasoning cannot be applied.

In the case-based reasoning, a reasoning engine remembers previous situations similar to the current one and uses them to help solve the new problem. However, case-based diagnostic technique has been criticized on many grounds. For example, that being specific to the system being diagnosed, they are non-constructive and that, having no analytic basis, the methods are restricted to specified faults and have a known level of competence. We think that the model-based diagnosis, being independent of the particular device descriptions, is intended to overcome these difficulties.

Therefore, we consider developing the approach of the model-based diagnosis system with case base. Model-based reasoning and case-based reasoning have the potential to complement each other quite well. However, no work has been done on specific issues of learner modeling using combine model-based reasoning with case-base. The outline of model-based learner diagnosis with case base is following. When the set of learner's behavior data input the diagnosis system, the diagnosis engine reasons the state of his/her knowledge consulting the diagnosis knowledge base include case base and object model base. The design of the model-based diagnosis system begins from describing the system as diagnosis object model. The system, which is a diagnosis object, is considered to be a set of domain models. The diagnosis object model that has knowledge of proper action, and the set of the behavior of learner as input value are given to a system. The first behavior of the system that received input is to seek whether there is a history about the same case in the case base. If the record to apply in the case is found, case base returns list of learner's knowledge, which should examine to diagnosis engine. Diagnosis engine does investigation about domain model of each record given to it, by comparing a simulation result in object model with the actual behavior of learner. Diagnosis process is finished if a trouble is recognized. When there was no record that complied with the input value in the case base, the process starts to use diagnosis domain object model. This domain object model has the hierarchical structure. A process begins from making the error model that one component in the extreme high class in the diagnosis object model is supposed to be out of order. The purpose of this process is to simulate using a made error model to examine whether the result of the simulation is the same as the behavior of learner. If there is no contradiction in the simulation result, the model-based reasoning is done again toward each domain knowledge model of the lower layer. In the same way, a diagnosis process is repeated until a trouble is recognized in knowledge component of the extreme lower layer. All process of diagnosis is knowledge processing by KAUS.

4 Discussion

The objective of this study was to develop a new intelligent learning support system, especially to focus two conditions; generality and adaptability. Authors in first propose the architecture of intelligent learning support system with large knowledge base to enough generality, which modeled by using multi-strata model. In second presented the model-based learner diagnosis to meet adaptability. All of the knowledge was represented by KAUS in intelligent learning support system that was assumed a part of the intelligent problem solving system. The issue of learner diagnosis is very important point to achieve adaptive instruction in intelligent learning support system. We proposed that fault diagnosis techniqucs bc applicd to infer the state of learner's knowledge. So we discussed the feature of diagnostic techniques, especially model-based reasoning with case base. Model-based reasoning appears to be a more promising technique than other knowledge-based methods because it can diagnose the faults that have not been pre-determined. Fails in learner's knowledge can be diagnosed automatically based on the models, which describe the correct behavior. However, because model-based approach reasons from the actual structure and function of knowledge, it is inefficient for some problems. Furthermore, obtaining domain models is sometimes either difficult or too complicated, whereas most of the fails can be diagnosed based on past experience, which is very effective if the rule base or the case base is either comparatively small or well-indexed. A better solution is a hybrid approach integrating some of the diagnostic approaches. A case base will be provided to access the solutions to some fails diagnoses occurred previously, of which the domain models are unavailable. For some diagnoses, their solutions and contexts can also be stored in the case base for reuse later. Frequently occurring fails can be diagnosed efficiently even by a few of heuristic diagnostic rules. We believe that such a hybrid diagnostic approach will perform better than any of them does. In order to achieve this goal; we have considered the division of object model and problem type. On this part, it is necessary to carry out examination that will be more profound in future work.

References

[1] Aida, T. & Ohsuga, S., "Intelligent Systems Must Be Able to Make Programs Automatically for Assuring the Practicality", Lecture Note in Artificial Intelligence 1415, pp.47-56, (1998).

[2] Costa, E, New Directions for Intelligent Tutoring Systems, Springer-Verlag Publishers, New York, (1991).

[3] Mandl, H. & Lesgold, A., (eds.), Learning Issues for Intelligent Tutoring Systems, Springer-Verlag Publishers, New York, (1988).

[4] Ohlsson, S., "Some principles of intelligent tutoring", Instructional Science, Vol. 14, pp.293-326, (1986).

[5] Ohsuga,S., "Multi-Strata Modeling to Automate Problem Solving Including Human Activity", Proc.Sixth European-Japanese Seminar on Information Modeling and Knowledge Bases, (1996).

[6] Sugaya,K. & Ohsuga,S., "Design of Intelligent Education System using Multi-Strata Modeling Concept", Proc. The Sixth International Conference on Computers in Education, Vol.1, pp.305-311, (1998).

[7] Sugaya,K. & Ohsuga,S., "Framework for Developing Intelligent Tutoring System Using Multi-Strata Model", Advanced Research in Computers and Communications in Education, Vo.1, IOS Press, pp.832-835, (1999).

[8] Wenger,E., Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge, Morgan Kaufmann Publishers, Los Altos, California, (1987).

Educational Agents and the Social Construction of Knowledge: some issues and implications

Carolyn Dowling
Australian Catholic University
115 Victoria Parade, Fitzroy, Victoria 3065, AUSTRALIA
Phone: (61 + 3) 9241 4456
Fax: (61 + 3) 9241 4546
Email: c.dowling@patrick.acu.edu.au

The use of intelligent software agents within computer mediated learning environments is currently an important focus of research and development in both AI and educational contexts. Roles envisaged and implemented include those of tutor, of 'manager', of information seeker and of fellow learner. Each of these raises its own special challenges in relation both to the capabilities of the software and to our understandings in regard to the nature of the learning process. High on the list of factors currently believed to contribute to effective learning is social interaction in the service of knowledge construction. Within many electronic learning environments we are currently witnessing the emergence of a new participant in the social interactions that mediate learning. The substitution of computer programs possessed of varying degrees of intelligence, autonomy and 'personality', for certain dimensions of human presence within the computer based classroom raises a number of questions related to the processes through which knowledge is socially constructed, and to the qualities which are necessary to ensure successful participation in those processes. Through discussion of both theoretical perspectives and practical examples, this paper explores some of these issues.

Keywords: **AI in Education, Educational Agents, Intelligent Tutoring Systems, Interactive Learning Environments, Networked Social Learning, Teaching and Learning Process**

1 Introduction

Developments in computing and information technology in recent years have rapidly propelled the notion of intelligent software agents from concept to implementation. Today, whether or not we are always aware of them, they are an integral part of a growing number of computing environments. From the invisible armies of knowbots and related entities scurrying around the Net in the service of increasingly sophisticated search engines to the cheery little characters who pop up on our screens offering assistance with anything from formatting a date to constructing a complex multimedia presentation, or the 'personalities' with whom we interact in chat rooms in happy ignorance of their purely digital nature, intelligent agents are alive and well and are multiplying rapidly.

An early but still useful conception of a software agent is, "A character, enacted by the computer, who acts on behalf of the user in a virtual environment", useful in mediating " ... a relationship between the labyrinthine precision of computers and the fuzzy complexity of man [10, p. 355]. Later definitions tend to be expressed in more functional terms, such as, "An agent can be viewed as an object which has a goal and autonomously solves problems through interaction, such as collaboration, competition, negotiation and so on" [9]. This definition has some similarities with that offered by Maes [12] who defines an agent as:

"A computational system which:

- is long lived;

- has goals, sensors and effectors;
- decides autonomously which actions to take in the current situation to maximize progress towards its (time-varying) goals" [12, slide 5].

Summarising the writings of a number of researchers, Aroyo and Kommers [1, p. 237] identify four major characteristics of agents as being autonomy, responsiveness or reactiveness, pro-activeness and social ability. Other qualities frequently proposed, but not supported by all researchers or indeed by all users, include the ability to learn from experience and consequently to respond in flexible and possibly unforeseen ways to particular situations, and the possession of a believable 'character' or personality as a basis for social interaction.

It appears that a combination of factors has contributed to the current proliferation of software agents. Apart from the technical developments which have opened up the possibility of implementing what were previously largely theoretical conceptions, there is our very real need for assistance as we operate within computing environments characterised by rapid change, large quantities of extraordinarily complex information, and a lack of common organisational structures through which information may be accessed and managed. As Laurel predicted, there are now many situations in which, in the interests of efficiency, some form of 'intelligent' mediation is required between computer systems and the needs of users.

There are, of course, different forms that this mediation could have taken. The strong propensity for most users to accept assistance in the form of a more or less personified entity as largely unproblematic undoubtedly derives at least in part from the anthropomorphic elements implicit in most computer interfaces from the earliest days of computing. It can be strongly argued that a degree of personification has always been automatically and inevitably conferred as much by a program's use of language as a component of the interface as by our everyday understandings of the 'intelligence', albeit artificial, of computers. Intelligence and language use are, after all, key defining attributes of human beings.

Not only are we accustomed to interacting with computers as though they share with us a degree of 'humanity', but in a number of areas of activity we have been persuaded to value 'social' interaction particularly highly. Education is a good example, given the extent to which our current understandings of learning depend upon an acceptance of the belief that knowledge is to a large extent socially constructed. In the current drive to move teaching and learning online, the notion of agency in computing has found a strong ally and a vehicle for expansion. Unless the social interactions that mediate learning in face to face environments can be shown to have a digital equivalent, proponents of online courses will be forever 'on the back foot', with their products being regarded by most educators as second best. While courses incorporating the communications facilities of the Internet certainly go a considerable way in promoting interactions of various types between teacher and student and also between student and student, the possibility of using software agents to create an illusion of interpersonal interaction so convincing as to achieve pedagogical outcomes equivalent to those deriving from a relationship with another human being is extremely enticing to the designers of electronic learning environments.

2 Some examples of socially interactive pedagogical agents

Johnson [7] has proposed the following definition the role of a pedagogical agent as distinct from those designed for other purposes:

> "Pedagogical agents are autonomous agents that support human learning, by interacting with students in the context of interactive learning environments. They extend and improve upon previous work on intelligent tutoring systems in a number of ways. They adapt their behaviour to the dynamic state of the learning environment, taking advantage of learning opportunities as they arise. They can support collaborative learning as well as individualized learning, because multiple students and agents can interact in a shared environment. Given a suitably rich user interface, pedagogical agents are capable of a wide spectrum of instructionally effective interactions with students, including multimodal dialog. Animated pcdagogical agents can promotc student motivation and engagement, and cngcndcr affective as well as cognitive responses" [7, p. 13].

This is a comprehensive and optimistic vision, incorporating a number of possible roles for software agents within educational environments. Types of agents currently implemented in projects around the world include record keepers, information seekers, testers, facilitators of collaboration, tutors or instructors, fellow learners, and tutees. Of special interest in regard to this paper are those that contribute to the overtly social dimensions of the learning environment. The last three listed most clearly fulfil this criterion.

2.1 Agents as instructors

There is a sense in which perceptions of the role of computers in the learning process have come full circle. Early models of the role of 'computer as tutor' in the form of drill and practice style of instructional software, generally based on Skinnerian principles and incorporating very limited interaction between user and computer, have long been rejected by most educators in favour of a range of other more acceptable guises including that of a learning tool, an information source, and a learning 'space'. With the development of agent technologies, as Johnson suggests, new possibilities now exist for incorporating computers within the learning environment in a range of socially interactive roles, including that of 'tutor', through modes of interaction more in keeping with current pedagogical theory.

It is commonly asserted that the presence of computers in classrooms has itself played a part in modifying the image of the teacher as the 'sage on the stage' in favour of a more collaborative model. Not surprisingly, thesc changing concepts are well reflected in many implementations of 'agent as teacher'. As Solomos and Avouris [18] write, for instance:

> "The user mental model of the system should be based on the metaphor of the "invited professor" rather than the "knowing everything own tutor". ... Our first findings confirm the observation that today's users, accustomed to hypertext-like interaction, are more likely to accept this collaborative teaching metaphor, according to which their tutoring system is viewed as an intelligent hypertext browser, offering links to other tutoring systems with the right content and at the right time" [18, p. 259].

The increasingly popular concept of the teacher as a facilitator of learning is also reflected in such statements as: "Each student working on the project will have an agent, operating in the background, watching progress, measuring it against the plan, and taking remedial action when necessary" [19, p. 362].

2.2 Agents as fellow learners

A style of agent of special significance in the context of socially constructivist theories of learning is the 'fellow learner', which to differing degrees might be presumed to include all participants within the learning environment. If agents are to gain widespread acceptance in the field of education, this is an important area for research and development. Since the 1980s Chan [2, 3] and colleagues have been working on a range of models of socially interactive agents for learning environments, perhaps the best known being the 'learning companion' – a software entity having limited knowledge of the domain in question, conceptualised as a fellow learner with whom the student may collaborate and even disagree. As in real life, some of these learning companions may be better informed than the student in the relevant domain of knowledge, while others may know less. Perhaps not surprisingly, in learning environments for younger students, animals are a popular choice of persona for such agents, as in this example of a networked learning environment for Taiwanese high school students, as described by Chan:

> "The Dalmation is having the same performance as the student. ... Another animal companion is Dragon, like one of those animal companions in Mulan, a Disney cartoon of this summer. This dragon will "learn" (mainly rote learning) from the student and also from other students on the Net and so may know more than the student. At certain point it'll stop learning and come back to teach the student. In a way, Dragon is protecting the student" [3].

An interesting development of this concept is presented by Sheremetov and Nunez [16, p. 310], who describe the function of a 'monitor agent' as being to modify the role, behaviour or expertise of learning companions from that of strong group leader to a weaker companion or even a passive observer, depending on its interpretation of the degree of guidance required by the learner.

2.3 Agents as pupils

We are all familiar with the common wisdom that we learn through teaching others. At the school level, many educators have long been familiar with the concept of the computer as 'tutee' through the use of the Logo programming language, in which 'teaching the turtle' was a familiar metaphor for the activity of programming. More recently, a number of researchers have explored the translation of this concept into electronic learning contexts where agents exist to be 'taught' by the student user, as in the example from Chan quoted above. A further example is described by Ju [8] who writes of a computer based peer tutoring system employing two categories of agent – an 'expert', and a 'learner':

> "... students become active learners who are guided to learn by teaching a computer. After the students watch how the computer expert solves a set of linear equations [the program] helps the human student act as a teacher in order to learn more about the subject matter. At this time, the computer plays the role of a student ..." [8, p. 559].

3 Some issues for consideration

3.1 Multiple agents

Most agent based systems utilise a number of agents, many of them capable of a complex range of interactions with the student, with one another, and increasingly with agents associated with other programs. Their individual purposes derive from theoretical analyses of the component tasks and activities that are included in the larger scale pedagogical interactions of human beings. As educators, and indeed as students, we may simultaneously enact a range of roles within the educational environment. The apparently unitary activity of 'teaching' involves such elements as demonstrating, guiding, telling, questioning, explaining, testing, motivating, criticising – even learning! Many researchers consider that the electronic medium makes it feasible to identify and separate out these diverse functions. These can then be enacted through different configurations of agents working in relationships which ranging from collaboration to competition.

An example is the Multiple Agent Tutoring System (MATS) described by Solomos and Avouris:

> " MATS is a prototype that models a "one student-many teachers" learning situation. Each MATS agent represents a tutor, capable of teaching a distinct subject. All MATS tutors are also capable of collaborating with each other for solving learning difficulties that their students may have" [18, p. 243].

Strategies for most efficaciously combining the activities of multiple agents such as these necessitate a complex agent architecture, and understandably occupy a great deal of the research agenda in this area. Of interest in relation to their participation in the social construction of knowledge is the fact that one of the most common metaphors employed by a number of researchers and courseware designers is that of a 'society' of agents, a conception reminiscent of Minsky's *The Society of Mind* [14], Gardner's multiple intelligences [6] and other related theories of cognition and behaviour. In describing the different aspects of the design of their "multi-agent, computer-based interactive environment", for example, Costa and Perkusich [4, p. 196], drawing on the work of Franklin and Graesser [5] refer to their aggregation of agents quite specifically as a 'society'.

> "The society [of artificial tutoring agents] is an open multi-agent system made up of a collection of tutoring agents that co-operate among themselves to promote the learning of a certain human learner. This society is designed to be open and dynamic in the sense that it allows maintenance operations such as the entry and the exit of agents, besides eventual modifications in the knowledge and in the inference mechanisms of an agent. Each agent defines an expert tutor in some domain, having the necessary knowledge to solve problems in this domain. These agents are cognitive and possess properties like autonomy, goal-oriented, social ability" [4, pp. 197-198].

While on the one hand, the variety of functions of agents within a multi-agent environment must also be appreciated as an attempt to realise the type of rich user interface which Johnson suggests is necessary if the pedagogical interactions within electronic learning environments are to approximate to any degree to the face to face educational experience, some educators have concerns in regard to the assumptions underlying these practices. They argue that such developments are underpinned by a reductionist rather than a holistic understanding of the processes and relationships involved in teaching and learning. In separating out the different components of pedagogical interactions, are we enabling each part to be realised more effectively, or are we failing to acknowledge that the global act of human teaching may in fact be more than the sum of its component parts? It seems reasonable to suggest that firm judgments on issues such as this must await greater experience of the roles of agents within these learning contexts.

3.2 Personification

Another focus of debate concerns the degree to which personification is helpful in fostering fruitful pedagogical interaction between the human learner and software agents. This question clearly relates more to the 'socially interactive' agents than to those fulfilling more tool-like functions, which arguably require far less in the way of 'personality'. As noted earlier in this paper, there are clear arguments for accepting that a degree of personification of computer interfaces is inevitable. As Shirk puts it:

> "Although there is some dispute among software critics concerning the advisability of having 'personalities' in computer programs, their presence seems unavoidable. Any time there is communication between a computer and a human, the information presented by the computer has a certain style, diction, and tone of voice which impact upon the human's attitude and response toward the software" [17, p. 320].

However the extent to which this should be deliberately fostered is less clear, although many feel intuitively that it should be an important element in the creation of an electronic learning environment characterised by interactions which can reasonably be described as 'social'.

An important aspect of the representation of 'character' or personality is visual appearance. Interestingly, both research and experience suggest that the relationship in the case of software agents is far from straightforward, and that a mismatch between realism in appearance and the apparent knowledge level of the agent can have a deleterious effect on credibility. The more visually realistic the representation, the higher the expectations of the user in relation to the appropriateness and 'intelligence' of utterances and actions. Agents that 'look' smart and 'act' or 'talk' dumb are poorly received by many users, who express a higher tolerance for the limitations of a 'character' more sketchily represented, for instance through cartoon-like graphics. As Masterton, writes, for instance, "A common problem with AI programs that interact with humans is that they must present themselves in a way that reflects their ability. Where there is a conflict between the ability of the system and the users' perception of that ability a breakdown occurs and users may either fail to exploit its full potential or become frustrated with its shortcomings" [13, p. 215]. He goes on to suggest the implementation of a degree of anthropomorphism intended to convey qualities such as friendliness and usefulness, without the implication of possession of full human capabilities [13, p. 211]. He describes the development and role of such an entity in the form of a VTA (Virtual Teaching Assistant) which is able to introduce topics and answer simple questions, the more complex types of exposition and interaction being left to the human teacher. In terms of a traditional scenario at university level, the VTA functions somewhat like a tutor or demonstrator as distinct from a lecturer. "In this way faculty is left free of the guiding and assisting issues of the course and is able to concentrate on more complex questions and higher level issues generated during the course" [13, p. 211].

Further instances of this principle are the examples of agents presented as animals discussed earlier in this paper. Our expectations in regard the cognitive skills of animals may well be more appropriate to the capabilities of software agents than are our experiences of human-to-human interactions.

3.3 Autonomy

Closely related to the 'intelligence' of software agents is the issue of autonomy, in particular the degree to which an agent should be furnished with pre-existing goals which might lead it to take particular action without instruction from the user, and even contrary to what the user might perceive as his or her interests and wishes. Exploring the implications of such entities existing and interacting within virtual reality environments, Loeffler [11], for instance, notes that the unpredictability resulting from significant autonomy might well result in agents who are less 'helpful' to us than we might hope or indeed expect. It is easy to slip from such considerations into the need for a contemporary version of Asimov's laws of robotics as conceived in fictional terms more than 30 years ago!

In educational contexts, the implications of autonomy, particularly in terms of control of and responsibility towards the learner, are potentially extremely complex and difficult to address without more exposure to these types of software, and indeed it is quite likely that such experience may cause community understandings in regard to appropriate relationships between the 'human' and the 'not human' in electronic contexts to develop and change over time. In the short term, current trends in educational thinking which favour giving more control and autonomy to the learner would appcar to bc morc in linc with thc thinking of researchers such as Schneiderman who favour 'direct manipulation' over the development of interactive agents with a significant degree of independence of action. Where agents are involved, they may be programmed so as to exercise control over the learner on behalf of the creator of the learning environment, or they may be configured so as to be more sensitive to a user model, and more responsive to instruction from the user/student. In the latter instance, the agent would have a greater degree of responsibility to the needs and wishes of the learner, but this may not be in keeping with the pedagogical goals of the teacher.

Trust is another aspect of the teacher/learner relationship that is complicated by the degree of autonomy with which a pedagogical agent is endowed. To the extent that the programmer chooses to delegate certain functions and responsibilities to the agent, it is their problem, but it may also be an issue for students, particularly those with more insight into the nature of the agents with which they are interacting.

A further concern in regard to the autonomy of pedagogical agents relates to the issue of intervention in the learning process. Despite the finding of Aroyo and Commers [1] that pro-activity is a quality frequently sought after in agents, there is an important issue of balance to be addressed in relation to the educational process. It is well accepted that a high degree of unsought assistance whether from a human teacher or an excessively diligent and proactive agent can be quite detrimental, in particular to the metacognitive aspects of learning. Of course this is also an issue for teachers and learners in face to face educational contexts!

3.4 Level of participation in the social construction of knowledge

The belief that it is possible for agents to participate effcctivcly in the social aspects of knowledge construction is central to the work of many theorists and researchers. Sheremetov and Nunez [16], for example, whose works derives overtly from the theoretical frameworks of Piaget and Vygotsky, argue that:
"The design of learning environments, virtual or not, aims to promote productive interactions. In this type of learning a student changes from being a passive information receiver to an active collaborator, interacting with the tutors and colleagues in the learning process. Learning does not only result from acquiring knowledge, solving problems or using tools, but also from interacting about these on-going activities with persons and agents"[16, p.305 – 306].

In relation to their specific project they write: "Our emphasis lies in the role of interactions in an artificial learning community as a group of real and artificial learners, tutors, and facilitators, working, supporting and learning from each other [16, p. 306]. But however personified and autonomous the software agent, can it really be said to participate fully in the social construction of knowledge? It has been argued quite extensively that even the most heavily personified of computer programs suffer from an intrinsic lack of ability to participate in the metacognitive aspects of learning. Pufall [15], for instance, expresses a strong belief that a computer program is unable at any level commensurate with human capacities to modify its own knowledge structures or cognitive processes, and so cannot be regarded as a co-constructor of knowledge in a meaningful sense. While this might well have been the case in relation to earlier computer based learning environments, can we continue to make the same claims with confidence today or in the future? The capacity of software to 'learn' and adapt to experience through the incorporation of new information, the appropriate modification of its representation of the context in which it functions (its 'world') and of its inference mechanisms, is undoubtedly increasing. One way of considering this question might be to look at it in terms of the type of distinction sometimes made between 'hard' and 'soft' notions of artificial intelligence. If our test of full participation depends on an understanding that the agent has 'learnt' in precisely the same way that the human has learnt, then we will have difficulty accepting the electronic entity as genuine co-constructor of knowledge. If, however, we make our claim on the grounds that it appears to the human learner that the agent has participated in the learning that has taken place, then perhaps we can at least tentatively admit such a piece of software to membership of the social milieu which has mediated the educational experience.

Conclusions

It is clear that developments in agent technology have created a range of new possibilities in terms of aligning computers more strongly with prevailing educational theories and philosophies. In considering the many issues which might be raised in relation to the nature and roles of pedagogical agents, there are three overarching questions. Firstly, do agents have the potential to enhance learning, or do they threaten to undermine those aspects of the educational enterprise that we most value? Secondly, to what extent might they assist in the replication of the social dimensions of face to face learning within online environments? Thirdly, do they go further than this, and create new possibilities in regard to the social mediation of learning? To the extent that visions such as those of Johnson [7] are able to be realised, we may be faced one day with the need to re-evaluate our attitudes regarding the relative merits of a human teacher and an electronic entity designed specifically for educational purposes. But while the rhetoric of developers often suggests an ideal surpassing the sometimes imperfect realities of human-to-human pedagogical interactions, the 'jury' of online learners and of educators is still out.

References

[1] Aroyo, L & Kommers, P., "Preface – Intelligent Agents for Educational Computer-Aided Systems", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[2] Chan, T.-W., "Learning companion systems, social learning systems, and the global social learning club", Journal of AI in Education 7 (2) (1996).

[3] Chan, T.-W. (1998), "The past, present, and future of educational agents", http://www.apc.src.ncu.edu.tw/apc/ppt_chan.html (accessed on February 3, 2000).

[4] Costa, E.deB. & Perkusich, A., "Designing a Multi-Agent Interactive Learning Environment", in Proceedings of International Conference on Computers in Education (ICCE '97), Kuching, Sarawak, Malaysia, December 2 – 6, pp. 196 – 203, (1997).

[5] Franklin, S. & Graesser, A., "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Berlin: Springer-Verlag (1996).

[6] Gardner, H., "Frames of mind: the theory of multiple intelligences", New York: Basic Books (1983).

[7] Johnson, W.L., "Pedagogical Agents", Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp.13 – 22, (1998).

[8] Ju, Y. (1998), 'Development and formative evaluation of a computer-based peer tutoring system', Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp. 559 – 566, (1998).

[9] Kinoshita, T. & Sugawara, K., "Agent oriented computing", Tokyo: Soft Research Center (1995).

[10] Laurel, B., "Interface agents: metaphors with character", in Laurel, B. (ed.) The Art of Human-Computer Interface Design, Reading, Massachusetts: Addison-Wesley (1990).

[11] Loeffler, C.E., "Artificial Life of Agents", Proceedings of the IFIP World Conference on IT Tools, Canberra, Australia, September 2 – 6, pp. 48 – 55, (1996).

[12] Maes, P., "Software Agents Tutorial", http://www.media.mit.edu/people/pattie/CHI97 (1997), (accessed on March 18, 2000).

[13] Masterton, S., "Computer support for learners using intelligent educational agents: the way forward", Proceedings of, the Sixth International Conference on Computers in Education (ICCE '98), Beijing, China, October 14 – 17, pp. 211 - 219, (1998).

[14] Minsky, M., "The Society of Mind", London: Picador/Heinemann, (1985).

[15] Pufall, P., "Function in Piaget's system: some notes for constructors of microworlds", in Forman, G. & Pufall, P. (eds) (1988), "Constructivism in the Computer Age", Hillsdale, New Jersey: Lawrence Erlbaum Associates, (1988).

[16] Sheremetov, L. & Nunez, G., "Multi-Agent Framework for Virtual Learning Spaces", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[17] Shirk, H.N., "Technical writers as computer scientists: the challenges of online documentation", in Barrett, E. (ed.), "Text, Context and Hypertext: writing with and for the computer", Cambridge, Massachusetts: MIT Press (1988).

[18] Solomos, K & Avouris, N. (1999), "Learning from Multiple Collaborating Intelligent Tutors: An Agent-Based Approach", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

[19] Whatley, J., Staniford, G, Beer, M. &Scown, P. (1999), "Intelligent Agents to Support Students Working in Groups Online", Journal of Interactive Learning Research, Vol. 10, No. 3/4, pp. 235 – 243, (1999).

Facilitating Examples Understanding through Explicit Questioning

Isaac P.W. Fung and R.H. Kemp
Institute of Information Sciences and Technology
Massey University
Palmerston North, New Zealand
Fax: +64-6-3502259
P.W.Fung@massey.ac.nz; R.Kemp@massey.ac.nz

This paper describes a novel approach for promoting undcrstanding of examples through explicit questioning. Whether being asked by the teachers or self-motivated, studying worked examples is an indispensable step for learners to acquire domain knowledge. The issue is: how could students use examples in the most effective way? Research findings indicate that the utility value of examples among different groups of learners varies dramatically. Effective learners keep self-explaining the solution statements when studying the examples while less effective learners often take each step of the statements for granted. In order to facilitate better understanding of examples, we propose to question the students explicitly on the examples content in order to stimulate their self-explanations. This paper presents the underlying computer model for generating different categories of questions from specific examples. The questions are subsequently used by a case-questioner to test the students on what they have read.

Keywords: **Artificial Intelligence, Conceptual Graphs, Intelligent Tutoring Systems, Case-based Reasoning**

1 Introduction

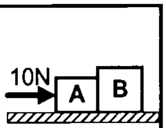
This paper presents a novel approach to facilitate the understanding of learning materials through explicit questioning. The notion we put forward in the paper forms one distinct feature of our current project: providing problem-solving advice in terms of relevant worked examples. When mapping out the project specification, there is an issue we are particular concerned with: to what extent the students benefit from the examples remains unknown. In her seminal work [2] Chi discovered the phenomenon of self-explanation among effective learners when they are presented with worked examples. Among this group of learners, they have a strong tendency to explain each example statement to themselves before moving on whereas the less effective learners tend to take the example statements for granted. In a follow-up investigation [3] Chi exploited her previous discovery in the context of learning. Not surprising, when students are deliberately prompted for self-explanation, they have shown a dramatic improvement in acquiring the knowledge. We believe the implication of Chi's study is very significant. Not only do the results shed light on understanding different learning behaviours, but the study also challenges instructors that merely informative examples do not guarantee good learning results. How the students use examples is a crucial factor in determining if they are really helping the students understand the subject domain.

As we are concerned with how the students use the examples presented, we decided not to take the present-and-go approach. Once a case (i.e. a relevant worked example) is retrieved for presentation, a case questioner will be automatically invoked to challenge the student's understanding on the knowledge embedded in the case. The questions generated are not explicitly stated in the problem statement. The rationale of this proposal involves encouraging the students to think more deeply while studying the worked examples. If the students have, in fact, understood the examples or related concepts within the domain, they should be able to answer the questions posed by the system. If not, the questions can trigger their attention towards certain aspects of the problem and stimulate their knowledge acquisition process.

2 Promoting Comprehension through Questioning

When studying worked examples, it is quite common for the students to take many solution statements for granted without trying to dig out the embedded tacit knowledge. Even if the students have the intention, they may lack the knowledge structure to find out the tacit knowledge. To put it simply, the student may know that it is helpful to self-explain the statements, but the problem is explaining what? There is research (e.g. [1], [4] and [7]) which indicates that questioning plays a significant role in understanding narrative text and therefore we argue that the same principle should also be applied in comprehending example solutions. If this argument is valid, one potentially pedagogically fruitful approach to tutoring in terms of providing examples is to question the learners on the content of the examples in a systematic way. Once the example is presented, the students will be asked questions driven by physical principles in order to detect what they know about the example and to help them discover meaningful relationships. To illustrate the argument, we consider the mechanics example shown in Figure 1.

Two blocks A & B are resting on a frictionless horizontal plane as shown. If an external force of 10N is acting on A, what is the acceleration of the blocks and the force of contact between them? (The masses of A and B are 3kg and 7kg respectively).



Solution	Net Force $_{A\&B}$	=	Mass$_{A\&B}$ × Acceleration $_{A\&B}$ *(Applying Newton's 2nd Law on*
:	External Force $_{A\&B}$		*A&B)*
	10	=	Mass$_{A\&B}$ × Acceleration $_{A\&B}$
	Acceleration $_{A\&B}$	=	(3 + 7) Acceleration $_{A\&B}$
		=	1 m/s^2
	Net Force $_A$	=	Mass$_A$ × Acceleration $_A$ *(Applying Newton's 2nd Law on A)*
	External Force $_A$ + Contact Force	=	Mass$_A$ × Acceleration $_A$
	$_A$	=	3 × 1
	10 + Contact Force $_A$	=	- 7N
	Contact Force $_A$		
	Net Force $_B$	=	Mass$_B$ × Acceleration $_B$ *(Applying Newton's 2nd Law on B)*
	Contact Force $_B$	=	Mass$_B$ × Acceleration $_B$
	Contact Force $_B$	=	7 x 1
	Contact Force $_B$	=	7 N

Figure 1: A typical Newtonian mechanics example and its solution

When presenting this example, the author must have already made many assumptions regarding the knowledge state of the reader. For instance, it will be assumed the reader knows that the weights of blocks are being cancelled by the reactions from the ground and thus the weights are not included in the calculation; the reader is also assumed to know that the acceleration of the whole system is the same as the acceleration of individual components; and that the external action on A is the same as the external action on the system as a whole in this case. However, these points may not have been mastered by some students. From the perspective of problem-solving, the solution presented is not the only way of tackling the problem. For instance, the contact force on B can be evaluated immediately by relating it to the contact force on A with which is formed an action-reaction pair. Alternatively, the problem can be tackled by solving three simultaneous linear equations with variables a, f_A and f_B which stand for the unknown physical quantities which are sought. This knowledge is not explicitly shown in the solution statements and the students whose self-explanation is less active may miss these knowledge units. Therefore, a fruitful tutorial dialogue can be created by conducting a series of question-answering episodes on the example presented.

3 A Taxonomy for Different Types of Questions

Before asking a question, the questioner must perform two steps: the first is to decide the *content of the enquiry*; and the second is to compose the *style of the queries*. To pose appropriate questions to the comprehender, the question designer must have a semantic category of questions. We have adapted the taxonomy for questions in narrative understanding originally developed in [8] into the context of physics problem-solving, and this is summarized below in Table 1. Note that except for question No.4, all the questions are relevant to the example shown in Figure 1.

CATEGORY	SPECIFIC EXAMPLES IN THE DOMAIN OF PHYSICS
1. Verification	*Is the system in equilibrium?*
2. Disjunctive	*Is force a vector or a scalar?*
3. Concept Completion	*What is FORCE?*
4. Feature Specification	*What does a convex lens look like?*
5. Quantification	*How many external forces are acting on block A?*
6. Causal Antecedent	*What caused the blocks to accelerate?*
7. Causal Consequence	*What are the consequences of the external force acting on the blocks?*
8. Goal Orientation	*In the 4th line of the solution, why are the masses of A and B summed?*
9. Enablement	*The blocks have weights; what is needed to prevent them from moving downward?*
10. Instrumental/ Procedural	*How was the acceleration of the blocks evaluated?*
11. Expectational	*What will be the magnitude of the contact force acting on A, if the mass of B is increased but the external action remains unchanged?*
12. Judgmental	*Do you think the solution presented is the only possible method?*

Table 1: Twelve Semantic Categories in Question Taxonomy (Adapted from [8])

4 Questions Generation

4.1 Based on the Definition of Concept Types

The questions in the categories 1, 2 and 3 are related to the definition of some domain-specific terminology and hence are grouped together. These categories of questions require the comprehender to grasp the definition of the focal content of the questions. For the question "Is the system in equilibrium?", the focus is on testing the readers on the precondition of a system being described as "in equilibrium". The question "Is force a vector or a scalar?" assesses the student's knowledge of the difference between vector and scalar quantities. There are two ways of generating these categories of questions: by *traversing the type hierarchy* and by *projecting* the definitional graph of the focus type into the conceptual graph [9] representing the example [5]. Based on these methods, the following scenarios can be developed. Question: "Why is the system not in equilibrium?" If the student successfully answers the net force acting on the system is not zero, another question can be generated such as "Then how can it be put into equilibrium again?"

4.2 Based on the Chaining of the Graph Nodes

In Newtonian mechanics, there are causes that are well-defined, such as the cause of acceleration being a non-zero net force; the cause of a change in velocity being non-zero acceleration; the cause of a change in position being a non-zero velocity, etc. The whole process of deriving values for unknown variables from available data can be modelled as a node chaining process, a kind of causal chaining. Figure 2 shows two subgraphs that represent the corresponding example statements:

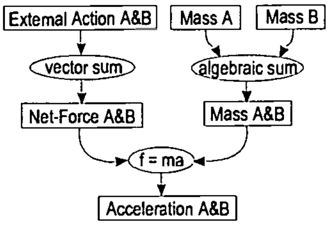
Solution Steps	Corresponding Conceptual Graphs
$\text{Net Force}_{A\&B} = \text{Mass}_{A\&B} \times \text{Acceleration}_{A\&B}$ $\text{External Action}_{A\&B} = \text{Mass}_{A\&B} \times \text{Acceleration}_{A\&B}$ $10 = (3 + 7) \times \text{Acceleration}_{A\&B}$ $\text{Acceleration}_{A\&B} = 1\ \text{m/s}^2$	 External Action A&B Mass A Mass B vector sum algebraic sum Net-Force A&B Mass A&B f = ma Acceleration A&B

Figure 2: Part of the solution steps and its corresponding conceptual graphs

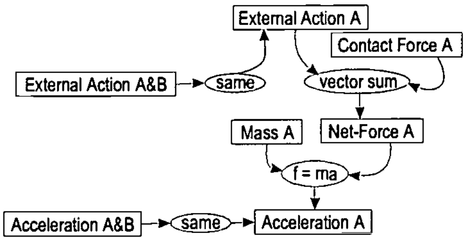
Solution Steps	Corresponding Conceptual Graphs
$\text{Net Force}_A = \text{Mass}_A \times \text{Acceleration}_A$ $\text{External Action}_A + \text{Contact Force}_A = \text{Mass}_A \times \text{Acceleration}_A$ $10 + \text{Contact Force}_A = 3 \times 1$ $\text{Contact Force}_A = - 7N$	

Figure 2 (cont'd): Part of the solution steps and its corresponding conceptual graphs

The graphs shown on the right hand side of Figure 2 provide ample material to generate questions to test students' understanding of the solution steps such as "How was the acceleration of the system evaluated?"; "How many external forces act on the block A?"; "What is the relation between the acceleration of A and the acceleration of the whole system?"; "How was the contact force on B evaluated?", etc.

4.3 Based on Propagating Qualitative Values across the Graph

Regarding the expectational question depicted in the 11th category, one should see that it belongs more to the area of qualitative reasoning (QR) [11] and this kind of question is very common in testing the knowledge of students. A QR technique had been developed in [6] and the following type of questions are successfully generated. "If the external action decreases, what would be the contact force?" "If the bottom of block A is made rough to create friction between A and the ground, what would be the acceleration of the system and the contact forces?"

5 Conclusions

This paper proposes a questioning approach to handling examples, which is intended to stimulate the student's cognitive process of self-explanation. Representing worked examples by CG allows the system to generate different categories of questions during the questioning process. We have shown that definitional, procedural and qualitative questions can all be posed to students for tutorial purposes. Due to space limitation, we have not covered all categories of questions; for instance, feature specification and enablement. At the moment, this part of the work derives only from a computational perspective and lacks empirical support. The next phase of our project is to test posing the questions to students to see if this approach would stimulate self-explanations and subsequently enable them to acquire a better understanding of the subject domain.

References

[1] Anderson, R.C. & Biddle, W.B., "On asking people questions about what they are reading." In G. Bower (Ed.) The Psychology of Learning and Motivation (Vol. 9), NY: Academic Press, (1975).

[2] Chi, M.T.H. et al., "Self-Explanation: How students study and use examples in learning to solve problems, Cognitive Science, Vol.13, pp. 145-82, (1989).

[3] Chi, M.T. H. et al., "Eliciting self-explanation improves understanding" Cognitive Science, Vol. 18, pp.145-82, (1994).

[4] Dillon, J.T., "The multidisciplinary study of questioning" Journal of Educational Psychology, Vol. 74, pp. 147-65, (1982).

[5] Fung, P.W. "Designing an case-based intelligent learning environment in physics with conceptual graphs" Ph.D. Thesis, Dept. of Computation, UMIST, England, (1996).

[6] Fung, P.W. "Generating qualitative predictive problems in a case-based physics tutor" In Proceedings of AI-ED97, World conference on artificial intelligence in education, Kobe, Japan, (1997).

[7] Graesser, A.C. & Black, J.B., "The psychology of questions" NJ: Lawrence Erlbaum Associates, (1985).

[8] Graesser, A.C. et al. "A taxonomy for question generation" Questioning Exchange, Vol. 2, pp. 3-15, (1988).

[9] Sowa, J. "Conceptual structures: Information processing in mind and machine" Reading: MA: Addison-Wesley, (1984).

[10] Trabasso, T., et al. "A model for generating questions that assess and promote comprehension" Questioning Exchange, Vol. 2, pp.25-38, (1988).

[11] Weld, D.S. & deKleer, J. "Readings in qualitative reasoning about physical systems" CA: Morgan Kaufmann Publishers, (1990).

Generating interactive explanations by using both images and texts for Micro World

Kiyoshi Yoshikawa, Isamu Takahashi, Tatsuhiro Konishi and Yukihiro Itoh
Department of Computer Sciences, Shizuoka University
Johoku 3 - 5 - 1, Hamamatsu, Shizuoka, 432 - 8011, Japan
Tel: +81-53-478-1474   Fax: +81-53-478-1499
E-mail: {cs6098, i-takaha, konishi, itoh}@cs.inf.shizuoka.ac.jp

In this paper, we propose a method of constructing an intelligent Micro-World (MW) for high-school chemistry that accepts learners' questions and advises them whenever the learners are working in the MW. We also discuss the method generating explanations using both texts and images. At first, we argue on the interaction between a learner and such a system, and classify learner's typical demands and possible educational supports by the system. Next we show the ability necessary to deal with the demands, such as recognizing learners' plan, generating a plan to achieve a goal of an experiment, reproducing the state at any step of the change in MW, controlling the initiative of the interaction, and so on. Then we propose methods in order to realize the abilities. Moreover, we illustrate how to implement the abilities and introduce our prototype system.

Keywords: **Micro World, Interactive explanation, Mixed initiative**

1 Introduction

Micro-World (MW) has a problem that it is hard to support learners who are in impasse. We are developing an intelligent MW that supports the learners[1,2,4,5]. The domain subject of the MW is high-school chemistry. The MW has the following functions:

- Simulating changes in the world model of high-school chemistry.
- Recognizing the learner' s plan by a sequence of his manipulations.
- Judging weather the learner is in impasse, by comparing the learner's plan with the standard plan that the MW generates. If the learner is in impasse, it assumes that the cause of the impasse might be lack of the knowledge necessary to perform the next manipulation which the learner should do.
- Giving the learner some advices by using texts. For example, the MW shows the knowledge which the learner doesn' t understand, the manipulation that the learner should do next, and so on.
- Accepting learner's questions at any time when the learner is working in the MW• and answering by using texts.

Our MW uses only texts in giving advices. In general, it is effective to explain something by using both texts and images. CINESPEAK [3] is one of the systems which can show explanations using both images and texts. It can generate a 3D animation and texts of explanation. It also can select appropriate camera shot corresponding to the contents of explanation. However, It can not generate explanations interactively.

We think it is necessary to avoid showing the texts and images prepared beforehand like video movies with some captions. The reason is that the explanation should be shown interactively. In other words, an educational system must not explain anything one-sidedly, because the condition of a learner is changing moment by moment while the system explains to the learner.

When an idea flashes upon a learner's mind during the explanation, the system must allow him to say his idea and respond to his remark. For example, when the system explains how to solve some problems in MW to the learner who is in impasse, if the learner requires doing continuation of the problem solving process by himself, the system should prepare MW and let him continue solving the problem on MW. Similarly, if the learner requires changing some conditions of MW and explaining the method of solving the problem, the system should stop explaining, re-plan a new method to solve the problem with new conditions, and explain it.

In this paper, we extend the user interface of our MW in order to make it more effective. The first extension is that the MW uses not only texts but also images when it shows the learners advices or explanations. The second one is that the MW generates explanations interactively. Our extended system can explain manipulations that a learner performed in a MW and the manipulations necessary to achieve a given goal by using both texts and animations simultaneously. Moreover it can explain interactively according to the learner' s demand.

In the next section, we discuss the ability necessary for the system that generate explanations interactively. In section 3 we show our basic approach to realize the abilities. In section 4 we illustrate how to implement the abilities, and we introduce our prototype system and show examples of its behavior.

2 Interactive method to explain

In order to generate explanations interactively, the system should have the following two functions.
- When a learner does not express his intention, the system must be able to lead his learning.
- The system must be able to deal with a learner' s demand whenever the system aids learning (even when it is explaining something to him).

The former is out of range of this paper, because it is the topic concerning to the teaching strategy in the field of Intelligent Tutoring System (ITS). Therefore, We concentrate the latter.

Learners' demands and the method to deal with them depend on what kind of educational supports can be provided by the system. Therefore, we must clarify:
1. the educational supports and learners' demands.
2. what kinds of ability are necessary to deal with the demands.

2.1 Possible educational supports and learners' demands

We can classify states of the system into the following two types:
- The system gives a goal and the learner manipulates the MW on his own initiative.
- The system takes the initiative then it shows advice or explanations to the learner.

We discuss learner' s demands and methods to deal with them on each state.

2.1.1 Supports and learners' demands when learner has initiative

We think the major demand on this state is to require an advice to resolve a learner's impasse. Therefore, we deal with only such type of demands as the first step of our research. In order to discuss how to deal with the demands, we classify causes of learners' impasse into the following two types.
(A):• A learner cannot understand the current state of MW.
(B):• A learner cannot decide what to do in the next step.

The system can satisfy the demand of the learner who is in impasse because of (A) by showing the following explanations:
- Explanation of a sequence of manipulations that the learner performed in the MW and the effect of each manipulation.
- Explanation of the state after each manipulation has performed.

The demand of the learner who is in impasse because of (B) can be satisfied by various ways. For example, the system identifies misunderstood or lacked knowledge and shows him the knowledge, the system explains on the similar case and lets him remind his experience, and so on. In this paper, we adopt the simplest way that the system shows the actions to be performed in the following steps. If we take the other way, we need to extend some functions to decide contents of explanations. However, the mechanism to control interactive generation of explanations is commonly reused.

As a result, the type of demands of the first state is only a demand to require some advice, and the type of explanations that the system generates is only an explanation of manipulations and the state after each one. In order to explain a manipulation and the state after it has been performed, the system generates animation showing how to perform the manipulation in the MW and texts explaining the effect of the manipulation.

2.1.2 Supports and learners' demands when the system has initiative

First, we discuss typical demands of learners who are in impasse because of (A) mentioned in the previous section. When the system explains to the learner a sequence of manipulations that the learner performed and the state after each manipulation by using animations and texts, the learner may demand that the system shows him a previous state again or a following state intermittently. In case that the learner finds his own mistakes while the system is explaining something to him, he may demand that the system stops explaining, prepares the initial environment, and lets him re-try solving his problem on the MW again. If the learner fails to resolve his impasse in spite of some explanations generated by the system, he may demand that the system show• •him the whole correct process to achieve his goal on the MW.

Then, we imagine typical demands of learners who are in impasse because of (B). In this case, the system explains him the action to be performed in the following steps. The learner may demand that:
- the system shows him the previous/following states.
- the system stops explaining in order to let him do continuation of manipulations.
- he rewrites some conditions of his problem and the system explains how to solve the problem with new conditions.

We don't argue on all of above-mentioned demands, but only ones with which our system can deal, considering possible actions by our system. Such actions are as follows:
(1) Explaining the sequence of actions which learners have performed.
(2) Explaining the sequence of correct actions by which the given goal can be achieved.
(3) Setting an environment for experiment to let learners try achieving the goal free.

Table 1. Examples of typical demands by learners

Actions of the system		type of the demand and the scene where the learner input the demand
Action before the demand	Action after the demand	
(1)	(1)	The learner wants to see another action than one shown in the current explanation on the sequence of his previous actions.
(1)	(2)	The learner can understand what he has done, and wants to see what he should do next.
(1)	(3)	The learner finds the mistakes he has made, and wants to re-try the experiment.
(2)	(1)	The learner understands correct actions, and wants to compare it with what he did.
(2)	(2)	The learner wants to see another action than one shown in the current explanation on the sequence of correct actions.
(2)	(3)	The learner understands the correct actions, and wants to re-try the experiment.
(3)	(1)	The learner loses the current state in the process of achieving the goal of the experiment. He wants to confirm the actions which he has performed.
(3)	(2)	The learner loses the way to achieve the goal of the experiment. He wants to see the correct actions.
(3)	(3)	He finds he has failed to achieve the goal and wants to re-try the experiment.

Then we can classify the demands according to which actions are performed before/after accepting the demand. Combinations of the actions are 3*3= 9 types such as "when system doing (1), a demand is input, then it does (1)", "when it doing (1), a demand is input, then it begins to do (2)", and so on. Examples of the typical demand of each type are shown in Table 1.

2.2 Abilities necessary to deal with learners' demands

In this section, we discuss abilities necessary to deal with the learners' demands mentioned in 2.1. Basically, MW should have an ability to simulate changes in the MW according to learners' actions.

In addition, in order to deal with the demands mentioned in 2.1.1, the system should have the following abilities.

(a) Ability to recognize learners' plan from a sequence of his actions.

In order to explain what learners have done by not only listing up the actions, but also showing the meanings of the sequence of the actions, the system needs the ability.

(b) Ability to generate a plan to achieve a goal of an experiment.

In order to explain correct actions which learners should perform, the system has to be able to generate plan.

(c) Ability to simulate changes in the MW according to the plan generated or recognized by itself, and ability to generate verbal explanations showing what actions has been done or what actions should be going to be done.

The system had better be able to generate explanations using both texts and images. In order to generate visual explanations, the system should be able to operate MW in a similar way as learners do. In order to generate verbal explanations, the system should be able to generate texts from the result of planning or plan recognition.

In order to deal with the demands mentioned in 2.1.2, the abilities mentioned above are also necessary. In addition, the following abilities are needed.

(d) Ability to store the history of actions by learners or the system.

The ability is needed to do action (1) or (2) as a reaction of a demand in Table 1.

(e) Ability to reproduce the state at any step of the change in MW and allow learners to manipulate the MW.

The ability is needed to do action (3) as a reaction of a demand in Table 1.

In addition, the following ability is necessary to realize mixed initiative. It is generally important to make interactive educational environment effective.

(f) Ability to control the two phases: a phase where a learner takes initiative by actions to achieve the goal, and a phase where the system takes initiative by generating explanations.

3 Methods necessary to deal with learner's demand

The basic framework of the system as a MW can be seen in [5]. An extension in this paper is that the system becomes to have two individual environments: one is the environment for experiment used by learners, and the other is the environment for explanation. Our system operates the latter environment in its explanation. We add the latter environment in order to avoid that both a learner and the system try to operate a common one at the same time. The environment for experiment has an interface and functions to accept learner's actions, and reacts as soon as it accepts an action from a learner. On the other hand, the environment for explanation cannot accept manipulations from learners (though switches similar to the environment of experiment are displayed in its window, they are dummy).

We discuss how to equip such a framework of the system with the abilities mentioned in 2.2.

(a) Ability to recognize learners' plan from a sequence of his actions.

On this ability, please see our previous paper• 1• :

(b) Ability to generate a plan to achieve a goal of an experiment.

On this ability, please see our previous paper• 1• ; too.

(c) Ability to simulate changes in the MW according to the plan generated or recognized by itself, and ability to generate verbal explanations showing what actions has been done or what actions should be going to be done.

Simulation in MW is performed by using symbolic knowledge representation. States at each step of

MW are also represented in a symbolic way. Manipulations by learners are also translated to symbolic representations. The control method of the simulator is event-driven: as soon as a manipulation is input to the simulator, the inference engine generates symbolic representation showing the next state of MW. The system draws the state of MW on the basis of the symbolic representation. Therefore, the system can simulate changes in the MW according to the generated or recognized plan, because the system can generate the input of the simulator represented symbolically from the plan.

In addition, because states of MW, manipulations to MW, and changes in MW are commonly represented in a symbolic way, the system can generate explanations in natural language on every fact in MW.

(d) Ability to store the history of actions by leamers or the system.

It is easy to store such history because all of states of MW, manipulations to MW, and changes in MW are represented in a symbolic way. The system records only the initial state and a sequence of having performed actions as the history. The system can reproduce all states and changes by simulating the change in MW again on the basis of the history.

(e) Ability to reproduce the state at any step of the change in MW and allow learners to manipulate the MW.

The system can reproduce any states in an explanation on learner's previous actions, by performing the manipulations stored as the history sequentially. On the other hand, it can also reproduce any states in the process when correct actions are performed, by performing the manipulations in the plan generated by itself. Thus, the system can set any states of an environment which learners can manipulate, by copying such reproduced states in the environment for explanation to the one for experiment.

(f) Ability to control the two phases: a phase where a learner takes initiative by actions to achieve the goal, and a phase where the system takes initiative by generating explanations.

We adopt the following strategies for controlling the phases:

- Basically, a learner takes initiative, and he acts freely in MW.
- Turn over the phase to the other phase where the system takes initiative, as soon as the learner inputs a question or demands that the system explains something.
- If the system finds that the learner is in impasse, ask him whether he hopes to turn over the phase where the system takes initiative. And if he does, turn over it.
- Accept interruption by learners whenever the system generates explanations.
- Decide the next action of the system according to the interruption. For example, if the learner demands that the system sets the phase where the learner takes initiative, set a suitable state of the environment and let him experiment freely. If he inputs a demand for the system to explain other topic than the current topic, continue explanation on the requested topic.

4 Implementation

We designed a prototype system. Figure 1 shows outline of our system.

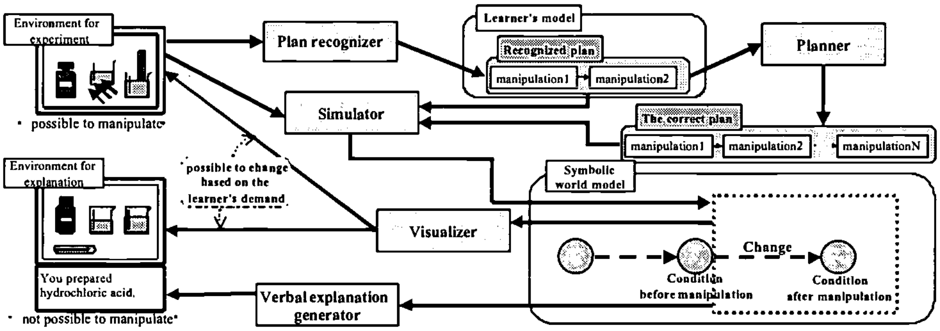

Environment for experiment
possible to manipulate
Plan recognizer
Learner's model
Recognized plan
manipulation1
manipulation2
Planner
Simulator
The correct plan
manipulation1
manipulation2
manipulationN
Environment for explanation
possible to change based on the learner's demand
Symbolic world model
Visualizer
Change
Condition before manipulation
Condition after manipulation
You prepared hydrochloric acid.
not possible to manipulate
Verbal explanation generator

Figure 1: The prototype system

The system has *environment for experiment* and *environment for explanation*. The system sets a goal and a learner tries to achieve the goal by manipulating objects in *environment for experiment*. When the learner does an action in *environment for experiment*, the simulator reproduces a change in *symbolic world model*. Then the *visualizer* draws the state after the change in *environment for experiment*. At the same time, *the plan recognizer* monitors the learner' s manipulations and recognizes his plan. When the learner becomes to be in impasse or requires some advice, *planner* generates a correct plan. Then the system visualizes environment for explanation, and starts explaining by using either recognized plan or the correct plan. In order to generate explanation, *simulator* reproduces states of the world model and *visualizer* visualizes the states in *environment for explanation*. Simultaneously *Verbal explanation generator* generates verbal explanation on the manipulation, the change, and the state.

The domain world model of this system is written by symbolic representation. In general, it is difficult to handle continuity of time and space by such representation. Therefore, our system handles time as a sequence of discrete segments of time. And it doesn' t handle strict position of objects in the world, but only relative relations which can be represented by symbols, such as "chemical materials are in the same beaker". A change is also represented by symbols which shows the initial state, the actions causing the change, the changing state, and the state after the change. Most of the subjects in high-school chemistry can be handled in the above mentioned way.

This system is implemented by using Tcl/Tk and LISP (Kyoto Common LISP). This system can deal with the 5 subjects: method of preparing a solution of a certain molarity, acidic material, basic material, neutralization, and using indicator.

We show an example of the behavior of our system when a learner does an experiment of neutralization. Figure 2 shows a user interface for *environment for experiment*. In the environment, the learner prepares hydrochloric acid, prepares nitric acid, and sodium hydroxide, pours nitric acid into hydrochloric acid, prepares phenolphthalein, and mixes it into the mixed acid. Then he finds that he has not achieved his goal. In the case that he can' t find the reason and inputs a demand for the system to explain his own actions, the system prepares an *environment for explanation* to start explaining the actions the learner has done. Figure 3 shows a user interface of *environment for explanation*. The interface has three windows: a window for displaying visual explanations and verbal explanations, a window showing a history of actions that have been taken place, and a window for inputting demands. In Figure 3, both visual and verbal explanations for the fourth action (marked in the list shown in the window for history). If the learner finds that he has made a mistake, and if he cannot find the correct way, he wants to demand that the system explains how to neutralize. He clicks on the button "*correct manipulation*" in the window for inputting demands. Then the system starts explaining the correct way (Figure 4). If he inputs a demand for the system to let him re-try the experiment, the system prepares *environment for experiment* and reproduces the state from which he wants to start experiment (Figure 5).

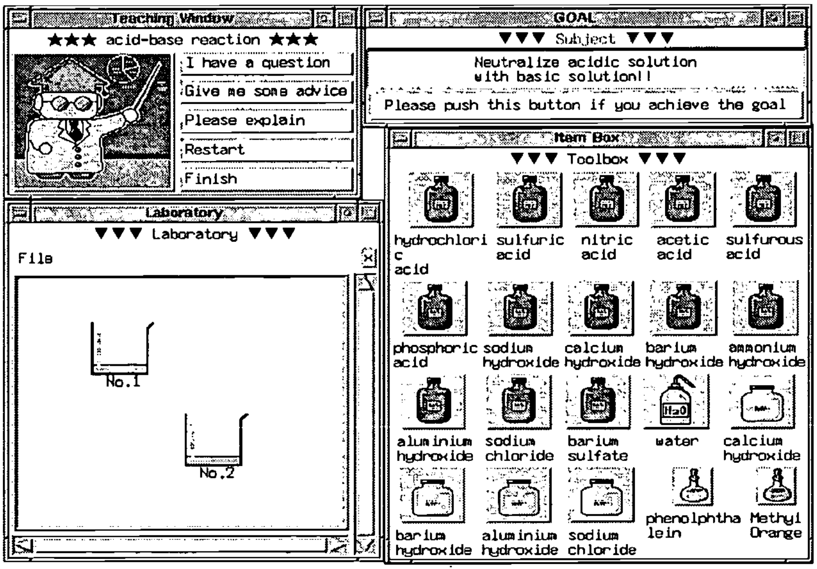
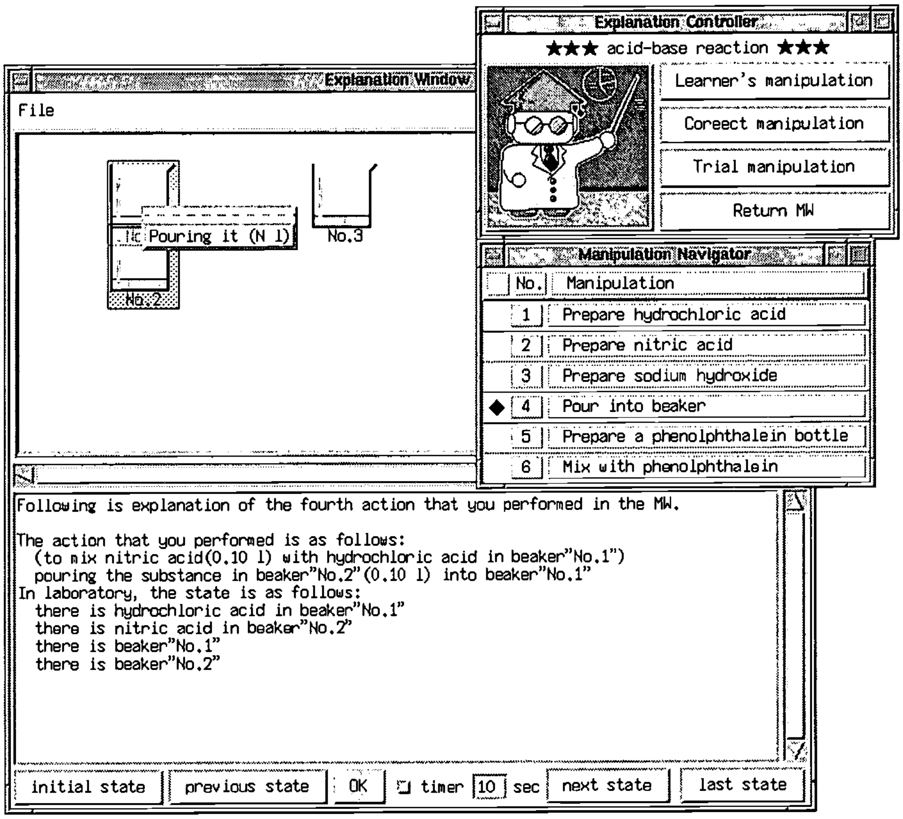

Neutralize acidic solution with basic solution!!

Figure 2: An example of environment for experiment

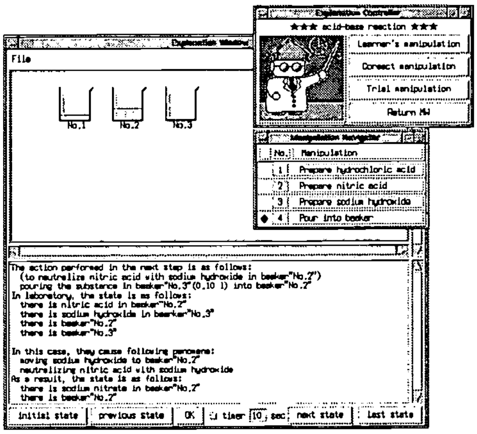


	No.	Manipulation
	1	Prepare hydrochloric acid
	2	Prepare nitric acid
	3	Prepare sodium hydroxide
◆	4	Pour into beaker
	5	Prepare a phenolphthalein bottle
	6	Mix with phenolphthalein

```
Following is explanation of the fourth action that you performed in the MW.

The action that you performed is as follows:
  (to mix nitric acid(0.10 l) with hydrochloric acid in beaker"No.1")
  pouring the substance in beaker"No.2"(0.10 l) into beaker"No.1"
In laboratory, the state is as follows:
  there is hydrochloric acid in beaker"No.1"
  there is nitric acid in beaker"No.2"
  there is beaker"No.1"
  there is beaker"No.2"
```

Figure 3: An example of environment for explanation (Explanations of a learner' s actions)

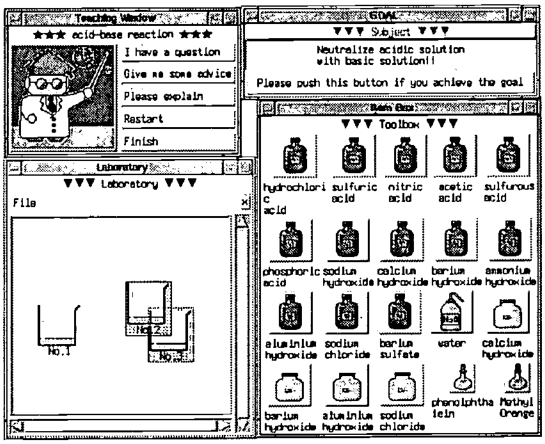


	No.	Manipulation
	1	Prepare hydrochloric acid
	2	Prepare nitric acid
	3	Prepare sodium hydroxide
◆	4	Pour into beaker

```
The action performed in the next step is as follows:
  (to neutralize nitric acid with sodium hydroxide in beaker"No.2")
  pouring the substance in beaker"No.3"(0.10 l) into beaker"No.2"
In laboratory, the state is as follows:
  there is nitric acid in beaker"No.2"
  there is sodium hydroxide in beerker"No.3"
  there is beaker"No.2"
  there is beaker"No.3"

In this case, they cause following penomena:
  moving sodium hydroxide to beaker"No.2"
  neutralizing nitric acid with sodium hydroxide
As a result, the state is as follows:
  there is sodium nitrate in beaker"No.2"
  there is beaker"No.2"
```

Figure 4: Explanations of corrects actions



Neutralize acidic solution
with basic solution!!

Please push this button if you achieve the goal

Figure 5: A reproduced environment for experiment

5 Conclusions

In this paper, we discuss a method constructing an intelligent and interactive MW generating explanations both images and texts. Our prototype system has relatively small domain knowledge base, so we have to make it larger in the future in order to increase subjects that our system can support.

When we will try to extend our system to handle other domains, the simulator underlying the system needs to deal with continuity of time and space. For example, if we deal with the field of electric circuit, the simulator needs to handle topology. If we deal with the field of dynamics of physics, the simulator needs to handle coordinate system.

Our another future work is to evaluate the effectiveness of our system experimentally.

Acknowledgements

This research was supported in part by Grant-in-Aid for Encouragement of Young Scientists (12780125) from Japan Society for the Promotion of Science.

References

[1] Itoh,Y., Takahashi,I. and Konishi,T., "On Recognition of Student's Actions in ILE", Proceedings of the International Conference on Computers in Education(ICCE'95), Singapore, December5-8, pp.630-639,(1995)

[2] Itoh,Y., Takahashi,I. and Konishi,T., "On evaluating students' knowledge by observing their actions in ILE", Proceeding of Artificial Intelligence in Education(AI-ED'97), Kobe,Japan, August18-22, pp.595-597, (1997)

[3] Stuart G. Towns, Charles B. Callaway and James C. Lester, " Generating Coordinated Natural Language and 3D Animations for Complex Explanations", Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), Madison, Wisconsin, America, July 26-30, pp.112-119, (1998)

[4] Takahashi,I., Itoh,Y. and Konishi,T., "On a method to decide the order of knowledge shown as advice in ILE", Proceedings of the Sixth International Conference on Computers in Education (ICCE'98), Beijing, China, October 14-17, vol1, pp.373-380, (1998)

[5] Takahashi,I., Konishi,T. and Itoh,Y., "On a Framework of an Educational System which Acts both as ITS and Micro-World", Proceedings of the Seventh International Conference on Computers in Education (ICCE'99), Chiba, Japan, November 4-7, pp.954-957, (1999)

Intelligent Interactive Learning Environment: Design Issues

Siu-Cheung KONG* and Lam-For KWOK**

**Department of Information & Applied Technology, Hong Kong Institute of Education, 10, Lo Ping Road, Tai Po, N.T.,Hong Kong. Email: sckong@ied.edu.hk*

***Department of Computer Science,City University of Hong Kong,Tat Chee Avenue, Kowloon, Hong Kong. Email: cslfkwok@cityu.edu.hk*

Interactive Learning Environment (ILE) provides interaction opportunities between learners and the virtual devices for productive learning. Intelligent ILE (IILE) provides quality feedback or authentic guidance to learners who need help in the ILE. This research aims to explore design implications of IILE by studying model of learner in the mathematics fraction domain. 169 primary four learners were invited to answer 10 open-ended questions on fraction addition and subtraction. A learner model on category of error and error pattern was formulated from the 423 erroneous responses. Results of the study indicated that researchers should study error patterns by understanding work of learners, distinguish careless mistakes from error patterns, and consider scaffolding support.

Keywords: **Intelligent Interactive Learning Environment, Learner Model**

1 INTRODUCTION

There are two categories of Learning Environment (LE): content-free and subject-specific [1]. A content-free LE allows participants and facilitators to formulate their own topics for discussion. Knowledge formulated from such interactions belongs to the learning community [2]. A subject-specific LE involves subject knowledge. Some subject-specific environments stress knowledge transfer like Intelligent Tutoring System (ITS) [3]. Other subject-specific environments such as Interactive Learning Environments (ILE), assisting learners to learn through exploration, put efforts on designing manipulative virtual learning devices [4]. No matter an LE is designed for knowledge transfer or knowledge formulation, subject matter of the learning domain should be carefully studied and incorporated in it [5].

1.1 Design Considerations of an ILE

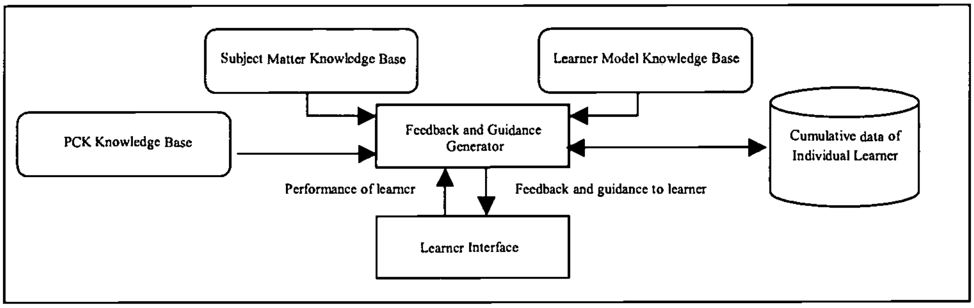
The study of subject matters plays a crucial role in designing ILE involving knowledge exploration because learners are not obtaining knowledge directly from the ILE. Learners have to learn by analogy, that is, learners have to transfer knowledge from manipulating the manipulative virtual devices of the ILE to grasp the abstract concepts of the subject domain [4]. Expert teachers are skilful in predicting how learners will think and err [6]. This diagnostic ability is tied to an expert's special understanding of the subject and is undoubtedly derived from multiple opportunities to teach the same content [7]. This knowledge includes knowing which aspects of a topic are particularly difficult, what the common misconceptions are, and what representations are important for authentic learning. Shulman [8] termed this kind of knowledge as Pedagogical Content Knowledge (PCK). It is crucial to utilize teachers' expert knowledge, especially knowledge on representation for authentic learning, to design manipulative virtual devices of an ILE.

1.2 Design Considerations of an Intelligent ILE

An ILE may provide interaction opportunities between learners and the virtual devices for productive learning. Some learners may learn the subject matter well without the assistance of the virtual learning devices. Some learners may learn well with chances to interact with the interactive learning devices of the environment. However, some learners may need guidance to learn well in the ILE [9]. An Intelligent ILE (IILE) is an ILE that provide feedback or guidance to those learners who need such help in learning the subject domain. Those learners who do not need help will not notice the existence of the auxiliary service. Learner model of learning in a subject domain may provide information about the behaviour of learners in learning the domain. Studying the learning model of learners may assist IILE designers to formulate design principles and obtain technical details such as formulating mal rules for understanding learning states of learners. A learner model thus may help to tailor-make an IILE for assisting various types of learners in learning the discipline. It is therefore important to study the learning model of learners in a specific subject domain for designing a useful and practical IILE to assist learners of various kinds in the learning process.

Three knowledge bases are therefore important for designing an IILE for learning subject-specific knowledge. They are the subject matter, the learner model of learning in the domain and the PCK of teachers in teaching the discipline. Subject matter knowledge base contains subject matter knowledge. It can provide subject matter advice and knowledge state of learners in the learning process. Learner model contains behaviour representations of learners. Learner model knowledge base may provide information about the learning state of learner. PCK knowledge base contains diverse guidance knowledge for different learning states of learners. It may provide learning advises based on PCK of experienced teachers of the subject domain who know how learners think and err in the discipline. Software agents will monitor the performance of learner in the learner interface. Software agents will determine proactive or reactive responses after a negotiation and communication process in the feedback and guidance generator. The negotiation will be a judgement of the knowledge state of the learner in the domain using both the learner model knowledge base and subject matter knowledge base of the IILE. Final decision will be an outcome after a consultation with the PCK knowledge base of the IILE and the cumulative data of an individual learner. The cumulative data records the historical learning states of each individual learner captured by the IILE. Figure 1 shows a conceptual design of an IILE for generating feedback and guidance.

Figure 1: A conceptual design of an IILE for generating feedback and guidance


Subject Matter Knowledge Base
Learner Model Knowledge Base
PCK Knowledge Base
Feedback and Guidance Generator
Cumulative data of Individual Learner
Performance of learner
Feedback and guidance to learner
Learner Interface

1.3 Chosen Subject Domain

A review of literatures indicated that many learners have great difficulties in learning the concepts and procedural knowledge of mathematics fraction [10, 11, 12]. Streefland [11] further pointed out that the main cause of such difficulties is the inadequate and inappropriate teaching in the traditional approaches. As the teaching and learning of mathematics fraction is an internationally renowned difficult topic, it is considered as an appropriate exemplar to be investigated for automation.

2 AIM AND OBJECTIVES

The aim of this research is to study the knowledge of learners in a subject-specific domain and to investigate its implication for designing a subject-specific IILE. There are two specific objectives: (1) to understand the problems of learners in learning the topic; (2) to discuss design issues of an IILE. Such findings may inform the development of IILE for providing quality feedback and guidance to learners.

3 RESEARCH METHODOLOGY

A questionnaire for studying model of primary learners on learning fraction addition and subtraction was designed.

169 primary four learners from four different schools were invited to complete the questionnaire through their mathematics teachers. All learners had completed their learning of fraction addition and subtraction before the test. Learners were requested to do the questionnaire on individual basis in a mathematics lesson for about 35 minutes. No discussions were allowed. The answer sheets were not used for any form of assessment but returned to the researcher after the administration. All 169 answer sheets returned were used for data analysis.

4 RESULTS AND DISSCSSIONS

This section will report on the quantitative and qualitative analysis results of all errors responded by participants of the survey and will discuss their implications on designing an IILE. The learner model formulated contains two areas: (1) knowledge of learners on category of error; and (2) knowledge of learners on error patterns of the domain.

4.1 Knowledge of Learner on Category of Error

Nine categories of error were identified and summarized from the 423 incorrect responses. Though incorrect response of each question may contain more than one error, this study selected the primary source of error for classification. Results were summarized in table 1. Categories were organized in descending order of percentage that account for the errors. The summarized result may serve as an important reference in designing a learner model of LE for fraction learning. Among the nine categories, categories 1, 2 and 9 directly related to the subject matter and accounted for nearly forty percent of the erroneous work. Categories 3 and 8 were common types of error in any mathematics exercise. It is interesting to investigate whether learners in this age group would commit these types of error like doing subtraction for addition at a certain level of unconsciousness. The study reflected that these factors might account for another twenty percents of errors.

Table 1: Category of error summarized from the learner model of the study

	Category of Error	Percentage Accounted
1	Improper handling of mixed number in fraction operation	20.4%
2.	Insufficient procedural knowledge for evaluating fraction	14.7%
3.	Calculation or careless mistake	13.5%
4	Unable to set up correct expression for solving word problem	11.6%
5.	Incorrect strategy for evaluating expression	11.4%
6.	Unable to identify error pattern for erroneous work	10.9%
7.	Not responding to question or the piece of work unfinished	8.5%
8.	Conducting subtraction for addition and similarly addition for subtraction	5.5%
9.	Incorrect simplification of answer to the simplest fraction form	3.6%

Though categories 4 and 5 can be purposely avoided, they do play a role in mathematics learning. Setting up expression for solving problems in a scenario may help to test whether a learner has grasped the taught concept. Strategies of evaluating numerical expressions may help to detect whether a learner has knowledge on magnitude of operands and order of evaluation on operators in an expression. The deficiency of this knowledge accounted for twenty percents of errors detected in this study. Categories 6 and 7 accounted for the last twenty percent of learners' work that might not be understandable or remain unfinished. Those 10 percent of learners' work could not be identified for any error pattern reflected that even human teachers might be unable to understand open-ended pieces of work like evaluating mathematics expressions.

4.2 Knowledge of Learner on Error Patterns

This section will report on knowledge of learners with problems in working with fractions on addition and subtraction. After careful analysing error patterns of learners in evaluating and solving simple fraction addition and subtraction problems, two categories were summarized: (1) concrete error pattern; and (2) vaguc idea on working with fractions. The first category includes some concrete error patterns that can be abstracted into mal rules. The second category contains error patterns that cannot be easily summarized into mal rules but reflect vague ideas and incomplete working procedures of learners. One of the most famous mal rules on fraction addition can be named as "Add numerators and add denominators". Learner with poor knowledge on fraction addition will adopt knowledge of arithmetic addition by adding the numerators of fractions in the fraction expression to give the numerator of the resultant fraction and similarly adding the denominators of fractions to give the denominator of the resultant fraction. There were four learners committing this type of error in this study. This rule might explain 3% of the errors. The second category of error pattern to be analysed involves high-level abstraction. The group of learners in this category showed no concrete error patterns. However, the pattern illustrated that these learners have some vague ideas of doing fraction addition and subtraction. Examples were illustrated in table 2.

Table 2 Vague ideas for evaluating fraction addition and subtraction expressions

	Error 1	Error 2
Learner 1 (3 score)	$\frac{3}{8}+\frac{1}{6}=\frac{9}{18}+\frac{9}{18}=\frac{18}{18}=1$	
Learner 2 (6 score)	$\frac{1}{2}+\frac{1}{3}=\frac{3}{10}-\frac{2}{10}=\frac{5}{10}=\frac{1}{2}$	
Learner 3 (0 score)	$\frac{1}{2}+\frac{1}{3}=\frac{1\times3}{2\times3}+\frac{1\times3}{3\times1}=\frac{3}{6}+\frac{2}{6}=\frac{6}{5}$	$\frac{3}{8}+\frac{1}{6}=\frac{3\times6}{8\times6}+\frac{1\times6}{3\times6}=\frac{54}{18}+\frac{3}{18}=\frac{12}{18}=\frac{6}{5}$

These erroneous presentations reflected that learners did have vague ideas about the working procedures on fraction addition. They need assistance to organize the disconnected nodes into a semantic net. Result of the studies indicated that some error patterns could be represented by mal rules. However, there were even more that cannot. An alternate method of studying error patterns of learners is to understand their work.

Identify Careless Mistake

The learner model of this study reflected that twenty percent of errors were derived from calculation or careless mistakes. Careless mistakes in this study mean transcription errors or simple computational mistakes form one step to another. The feedback and guidance will be different if an error is identified as a careless one. An IILE should handle not only problems generated from subject matters but also general problems of learner like careless mistake. An authentic guidance should provide not only advices or actions that can assist learners to formulate conceptual understanding of the subject domain but also offer help to learners derived from general problems such as careless mistakes. An IILE should attempt to distinguish careless mistake from other error patterns like human teachers.

Scaffolding Support

The forty percent of errors derived from inadequate knowledge of learners reflected that only immediate feedback may not help learner much and thus authentic guidance should be considered for facilitating conceptual understanding. A productive learning support should be an arrangement of a sequence of situations for facilitating knowledge construction [12]. The role of a mathematics-learning environment will be to help learners to learn, especially those fundamental concepts in mathematics, but not to replace mathematics learning in the conventional manner. Therefore it is fundamental for such kind of learning environment to provide scaffolding support to learner when assistance is needed. Support should gradually withdraw so that learner can stand on its own after leaving the system. Therefore a fraction IILE should be designed like a blank sheet for learner to work with fraction. Feedback and guidance are only provided when it is needed. On the other hand, learner working in the IILE who does not need support will not notice the IILE in behind.

5 CONCLUSION

Studying the learning model of learners may assist IILE designers to formulate design principles and obtain details for understanding learning states of learners. The learner model of this study modelled behaviour of learners in two aspects: error category and error patterns. Nine categories of error were identified. Forty percent of errors were derived from inadequate knowledge of learners on subject matters. Twenty percent could be explained by careless mistakes. Twenty percent involved general mathematics knowledge. The final twenty percent of erroneous work were difficult to be classified or work was not completed. Learner model of the study reflected that some error patterns could be represented by mal rules. However, there were even more that cannot. An alternate method of studying error patterns of learners is to understand their work. Result of the study indicated that IILE needed to apply a strategy to identify careless mistake so that appropriate guidance to learners can be provided. The forty percent of errors derived from inadequate knowledge of learners reflected that only immediate feedback may not help much and thus authentic guidance should be considered for facilitating conceptual understanding. A productive scaffolding support should be an arrangement of situations for facilitating knowledge construction. The future work of the study is to design ways and means to understand work of students, to devise strategy to distinguish careless mistake from other error patters, and to plan scenarios for assisting learners to learn by exploration in an IILE.

References

[1] Vosniadou, S., Corte, E.D., Glaser, R., & Mandl, H. (Eds.), "International perspectives on the design of technology-supported learning environments", Hillsdale, NJ: Lawrence Erlbaum, (1996).

[2] Duffy, T., & Jonassen, D. (Eds.), "Constructivism and the technology of instruction: A conversation", Hillsdale, NJ: Lawrence Erlbaum, (1992).

[3] Chan, T.W., "An introduction to intelligent computer assisted learning systems research and development", National Science Council Monthly, Vol. 23, No. 5, pp. 456-468, (1995).

[4] Akpinar, A., & Hartley, J.R., "Designing interactive learning environments", Journal of Computer Assisted Learning, Vol. 12, pp. 33-46, (1996).

[5] Wittmann, E.Ch., "Mathematics education as a 'design science' ", Mathematics education as a research domain: A search for identity, Dordrecht: Kluwer Academic Publishers, pp. 87-103, (1998).

[6] Berliner, D.C., "In pursuit of the expert pedagogue. Educational Researcher", Vol. 15, No. 7, pp. 5-13, (1986).

[7] Livingston, C., & Borko, H., "Expert-novice differences in teaching: A cognitive analysis and implications for teacher education", Journal of Teacher Education, Vol. 40, No. 4, pp. 36-42, (1989).

[8] Shulman, L.S., "Those who understand: Knowledge growth in teaching", Educational Researcher, Vol. 15, No. 2, pp. 4-14, (1986).

[9] Cheng, C.C., "Proactive guidance in computer-assisted language learning", Technical Report, LLL-T-24-95. Language Learning Laboratory, College of Liberal Arts and Sciences, University of Illinois, (1995).

[10] Behr, M.J., Harel, G., Post, T. & Lesh, R., "Rational number, ratio and proportion", Handbook of research on mathematics teaching and learning, New York: Macmillan, pp. 296-333, (1993).

[11] Streefland L., "Fractions in realistic mathematics education: A paradigm of developmental research", Dordrecht: Kluwer Academic Publishers, (1991).

[12] Ohlsson, S., "Knowledge requirements for teaching: The case of fractions", Teaching knowledge and intelligent tutoring, Norwood, NJ: Ablex Publishing Corporation, pp. 25-59, (1991).

Internet Video on Demand System of Classroom Teaching Cases

- Building "RAPSODY": An Intelligent Media-Oriented Remote Educational System for Self-Learning Support -

Hisayoshi INOUE and Toshio OKAMOTO
Graduate School of Information Systems
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-City, Tokyo, 182-8585 JAPAN
TEL: +81-424-43-5623
E-mail: {ino,okamoto}@ai.is.uec.ac.jp

Our study aims to accumulate information for teachers, about established teaching methodologies and techniques. For the purpose of our study, we construct a searching system for lesson example videos, and we develop an environment for facilitating the usage of our example videos, and for encouraging teachers' collaboration. Concretely, we focus on the domain of the new subject introduced in the Japanese curriculum, called "Information Education". In this paper, we describe the details of our video searching system, the design of the database tables, and we show an example of system operation.

Keywords: **Information Education, Teacher Education, Distance Learning, VOD, Rapsody**

1 Introduction

In now-a-days advanced information society, the demands about teachers' competence are high and diverse. Concretely, teachers are required to posses on one hand curriculum development ability, learning environment design ability, group learning, individual learning, simultaneous learning, and on the other hand the previous abilities should be supported by class practice ability, observation and evaluation ability, and last but not least, the ability to connect the textbook's world and the real world.

It is difficult to raise and form this complex set of abilities, with the help of only the presently available education and training methodology for teachers. Therefore, the necessity emerged [1] to examine the feasibility of a new systematic approach, for supporting the teachers' literacy progress, by building on their natural talents/ and abilities, and expanding these to reach the required width and breadth.

The information technology science is offering the tools for the development of an environment supporting the teachers' endowment progress. The knowledge concerning the teachers' education contents and methods is stored as multimedia information, in the form of pictures, videos and sound tracks. Moreover, by using the network environment, it is possible to make use of all resources over the net, without any constraints or restrictions of time and/or geographical location.

With this goal in mind, we are researching the development of an integrative distance education training system for supporting teachers' self-training, called RAPSODY (Remote and Adaptive System of Oriented Dynamic Teaching/Learning) . Up to now, the availability of video records and guidance plans about lessons was limited to education training centers or universities, etc. The present research intends to make the information on educational activities and practices public, and aims at joint usage and re-usage of teachers' self-learning and self-training methodologies and tools. Concretely, we develop a retrieval system based on dialogue patterns, by using a database of lesson videos.In order to jointly use the information in the distributed environment, or to be able to re-use it, we implemented a Video On Demand (VOD) system. The (teacher) user can control/manage the specification and stop/suspension of the regeneration point for the distributed VOD.

The main purpose of our research is therefore to propose a distance-learning environment on the Internet, for improving the teachers' practical abilities. In this paper, we describe the video on demand system developed until now, the indexing method of the classroom teaching movie example database, the system's functions and the system's evaluation.

2 The outline of the system

2.1 The structure of the system

Fig. 1 shows the structure of the search system. The system is built of the following three parts:

- Web browser;
- Lesson video example database;
- Video distribution server.

The web browser has the role of the user interface. The search/retrieval mechanism searches the lesson example video database via three types of relational database files.

The video distribution server stores the lesson scenes' videos. The video distribution server performs the VOD function at the users' requests. The search functions performed for the user are of the following two types.

- Keyword Search
- Feature Oriented Search

The Keyword Search (fig. 1,   ) takes place as explained below.

a-1) The user designates the search conditions.
b-1) The search mechanism compares the search conditions input by the user, with the available class example video database.
c-1) The search mechanism extracts the record(s) matching the searching conditions.
d-1) The result is displayed as the search result.

The Feature Oriented Search process (fig. 1,) is done as shown in the following.

a-2) The system dialogue mechanism enquires about the video characteristics/features desired by the user.
b-2) The user can answer to the system's enquiry vaguely [2].
c-2) The decision making table (showed later on in table 6), obtained from the user, is the basis for the evaluation of the specific search conditions. The gathered search conditions are passed over to the search mechanism of the database.
d-2) The database search mechanism compares the search conditions resulting from the evaluation with the lesson example videos contained in the database.
e-2) This result is displayed as the search result.

The search result is formatted as an URL list that is shown to the user. These URLs perform the function of linking the search result and the actual videos on the VOD server. The (teacher) user chooses the URL that s/he wants to refer (Fig. 1,).When the URL is chosen, the VOD client software, embedded via the Web browser plug-in, starts, and the video playback begins.

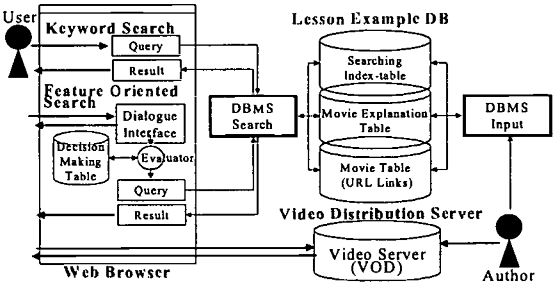


Figure 2 Sstem Organization

3. Database structure

3.1 Lesson (unit) database

The following three relational database files define the video lesson database.

- Searching Index File
- Movie Explanation File
- Movie File

In the following, we will explain in detail each database file type.

3.2 Searching Index File

The Searching Index File results from the comparison of the video database with the search items. The search items are organized in items for the Keyword Search and items for the Feature Oriented Search. In table 1, we show the Searching Index File for the Keyword Search, and in table 2, the Searching Index File for the Feature Oriented Search.

For the Keyword Search, the search conditions are given directly by the user. The Searching Index File slots are "lesson name (unit)", "learning contents (subunit)", "used information equipment/machinery/device(s)", "used tool(s) application(s)" and "class viewpoint".

In the Feature Oriented Search, the system generates the search conditions, based on the information obtained from the user. The Searching Index File (Feature Oriented Search) employs comparison of the extracted search conditions and the database, depending on the dialogue with the user. The slots of the Searching Index File for the Feature Oriented Search mechanism are "teacher activity " and "student activity ".

Table 1 Searching Index File (Keyword Search)

index frame database .basic search key.

record fields	value type
ID (primary key)	*numerical value type*
lesson name (unit)	*text type (menu selection)*
learning contents (subunit)	*text type (menu selection)*
used information equipment/ machinery/ device(s)	*text type (menu selection)*
used tool(s) application(s)	*text type (menu selection)*
class viewpoint	*text type (menu selection)*

Table 2 Searching Index File (Feature Oriented Search)

index frame database. feature search key.

record fields	value type
ID (primary key)	*numerical value type*
teacher activity	*text type (menu selection)*
student activity	*text type (menu selection)*

3.3 Movie Explanation File

Table 3 shows the contents of the Movie Explanation File, regarding the movie features. When the user is about to commence the lesson, the points, which need his/her attention, are explained via the contents of the Movie Explanation File. These explanations are used when displaying the search results.

Table 3 movie explanation file

movie feature description .video feature information)

record fields	value type
camera angle	*text type (menu selection)*
equipment existence	*text type (menu selection)*
equipment usage	*text type (menu selection)*
number of teachers	*text type (menu selection)*
teachers' movements	*text type (menu selection)*
existence learning supporter(s)	*text type (menu selection)*

number of students	*text type (menu selection)*
students' movements	*text type (menu selection)*
existence of a central student	*text type (menu selection)*

The explanation information in the Movie Explanation File (table 4) resumes the lesson scenes compiled by the video registrants, and the information on how the checkpoints, necessary for the lesson, were estimated. The slot of table 4 called "teacher's aim", corresponds, for instance, to the classification 8 presented later on in table 7.The "checkpoints 1 to 3" express the free description of the image, from the points of view shown below.

Checkpoint 1 the movie preconditions to be considered;

Checkpoint 2 what should be extracted/ understood from the current movie;

Checkpoint 3 the necessary forecast of the movie's following development.

Table explanation file

4 movie

movie explanation database

record fields	value type
ID (primary key)	*numerical value type*
teacher's aim	*text type (menu selection)*
checkpoint 1	*text type (item description , within 100 characters)*
checkpoint 2	*text type (item description , within 100 characters)*
checkpoint 3	*text type (item description , within 100 characters)*

3.4 Movie File

Table 5 shows the Movie File. The Movie File contains pointers to the real videos. The VOD server houses the real videos. Table 5 contains the Movie File slots called "thumbnail picture (still picture)", "previous movie", "movie URL (movie file name)", and "next movie".

Table 5 movie file

movie database

record fields	value type
ID (primary key)	*numerical value type*
thumbnail picture (still picture)	*text type (still movie file name)*
previous movie	*text type (URL type input)*
movie URL (movie file name)	*text type (URL type input)*
next movie	*text type (URL type input□*

For the discrete movie time-series {P(t)}, the following relationship exists:
[$P(t-1),P(t),P(t+1)$] = [previous movie, movie URL, next movie];$P(t=0)$ ={still picture};where t is the time.

4 The system's behavior

Figure 2 shows the search conditions input interface (for Keyword Search). Figure 3 shows the search result display interface. After the (teacher) user specifies the conditions for the desired video search via the search conditions input interface, the search starts. The result of this is displayed in the search result display interface [3] [4]. This interface shows the value of the slots called " still picture", "lesson contents (subunit)", "teacher's aim", "checkpoints", "teachers' activity" and "students' activity". The "still picture" can be seen in figure 3 (). Next to being a significant snapshot of the lesson video, the still picture has also the role of a pointer to the real video (a link to the VOD video file), so describes the URL (figure 3,). By clicking on the still picture, the video starts (figure 3,). The "teacher's aim" (fig. 3,) and "checkpoints" (fig. 3,) are, as mentioned before, the most important information for image explanation. The figure also presents the (teacher) user with help/support information about other items and record fields.

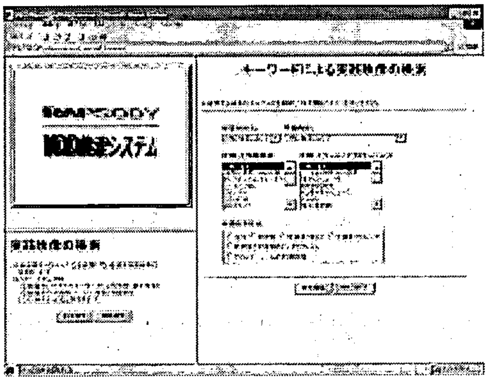


Figure 2 search conditions input picture

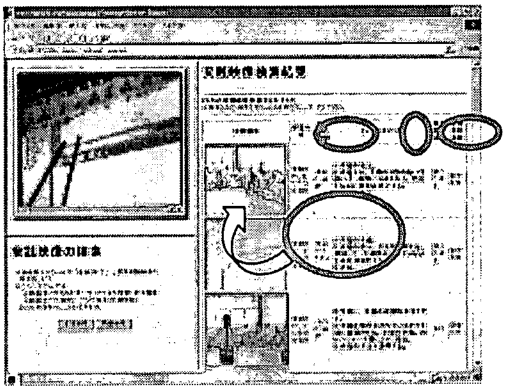


Figure 3 search result display picture

5 Conclusions

We construct a searching system for lesson example videos, and we develop an environment for facilitating the usage of our example videos, and for encouraging teachers' collaboration. Concretely, we focus on the domain of the new subject introduced in the Japanese curriculum, called "Information Education".We have presented in this paper the summary of the video search VOD system we have developed, moreover, we have shown the database organization and the system's behavior.As for the future tasks and research, we are planning to investigate about building a flexible key for the video search mechanism. We are studying at present the dialogue mechanism, with the immediate goal of using the search result's negative feedback information to the user's request, to serve as a new search key.

References

[1]Okamoto, T. (research representative/delegate) : "1998. Advanced Information Society; synthetic research on the contents, system and form of teachers' education – midway report", 1997, 9 year research funds foundation of the Ministry of Education, A)(1)09308004 stage report.

[2]Stain, A, Gulla, J.A, Muller, A. and Thiel, U.: "Conversational Interaction for Semantic Access to Multimedia Information", Intelligent Multimedia Information Retrieval, pp. 399-421, MIT Press, 1997.

[3]Noguchi, H.: "Data representation and management in image databases", IPSJ, Vol.33, No.5, pp. 457-465, 1992

Is a Learning Theory Harmonious with Others?

To form Effective Collaborative Learning Groups with Ontological Engineering

**Akiko Inaba, Thepchai Supnithi, Mitsuru Ikeda,
Riichiro Mizoguchi, and Jun'ichi Toyoda**
I.S.I.R., Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
inaba@ai.sanken.osaka-u.ac.jp

Our research objectives include constructing a collaborative learning support system that detects appropriate situation for a learner to join in a collaborative learning session, and forms a collaborative learning group appropriate for the situation dynamically. In this paper, we describe the outline of a system of concepts concerning learning goals expected to attain by learners through collaborative learning process with justification by the learning theories. We propose possibility that theory-based learning groups can be combined into one in order to help a learner attain his/her learning goals and showed an example of effective learning group formation which is formed by combining multiple theory-based learning groups. With the ontology, it will be possible to compare and synthesize the learning theories to design the collaborative learning settings.

Keywords: **Ontology, Collaborative Learning, Distributed Learning Environments**

1 Introduction

Our research objectives include constructing a collaborative learning support system that detects appropriate situation for a learner to join in a collaborative learning session, and forms a collaborative learning group appropriate for the situation dynamically. To fulfill these objectives, we have to consider the following:

1. How to detect the appropriate situation to start a collaborative learning session and to set up the learning goal,
2. How to form an effective group which ensures educational benefits to the members of the group, and
3. How to facilitate desired interaction among learners in the learning group.

We have discussed item 1 in our previous papers[10, 11], and this paper focuses on item 2. When we have clarified item 2 and extracted the desired interaction in the group, we would consider item 3.

There are many theories to support the advantage of collaborative learning. For instance, Observational learning[2], Constructivism[19], Self-regulated learning[9], Situated learning[15, 16], Cognitive apprenticeship[5], Distributed cognition[21], Cognitive flexibility theory[22, 23], Sociocultural Theory[25, 26], Zone of proximal development[25, 26], and so on. If we select a theory from these and form a learning group based on the theory, we can expect effective collaborative learning with the strong support of the theory. However, it is difficult to understand all theories because these theories are derived from a wide research area including pedagogy, sociology and psychology. Moreover, we can expect different educational benefits based on these learning theories, and observe various kinds of interaction between learners through collaborative learning process. Due to the diversity, it is difficult to list the learning theories effective to gain a specific educational benefit for a learner, and to compare the theories to form a suitable collaborative learning group for the learner.

Therefore, we have been constructing a system of concepts to represent collaborative learning sessions supported by these learning theories[12, 14, 24]. We call the system of concepts "Collaborative Learning Ontology". Although advantages of collaborative learning over individual learning are well known, the collaborative learning is not always effective for a learner. Educational benefit that a learner gets through the collaborative learning process depends mainly on interaction among learners. The interaction is partly influenced by relations among members of learning group, which suggests that how to form an effective group for the collaborative learning is critical to ensure educational benefit to the members. In this paper, we focus on "Learning Goal Ontology" which is a part of the Collaborative Learning Ontology.

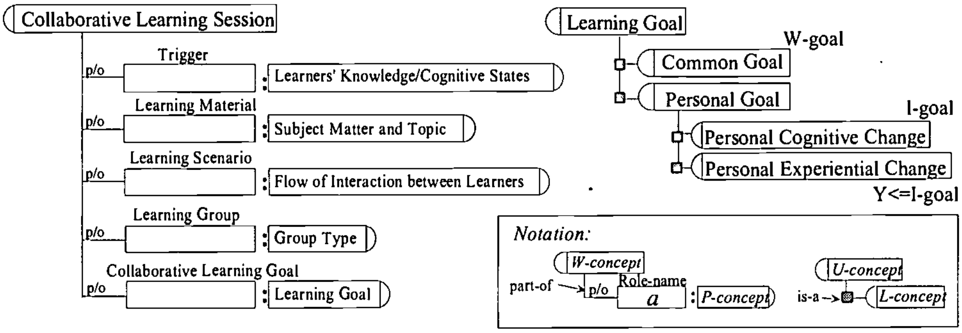


Fig. 1. Collaborative Learning Ontology

The concept “Learning Goal” is one of the most important concepts for forming a learning group because each learner joins in a collaborative learning session to attain some learning goals.

To help a learner obtain a specific educational benefit we can find several learning theories useful for the purpose and form different learning groups according to the theories. If the groups are merged into one, we may form a better learning group which is guaranteed its effectiveness by multiple learning theories. So, we also discuss the combination of learning groups supported by different learning theories.

This paper is organized as follows: we first show briefly the structure of our “Collaborative Learning Ontology” and “Learning Goal Ontology”. Then we summarize advantages and remaining tasks: how can we narrow down candidates of learning groups into one? Finally we propose a new learning group formation formed by combining multiple learning theories.

2 Learning Goal Ontology for Collaborative Learning

Through a survey of studies on collaborative learning, we picked up concepts to represent a collaborative learning session. As a result, we set up five primitive concepts to characterize the session: Trigger, Learning Material, Learning Scenario, Learning Group, and Learning Goal. Fig. 1 shows the conceptual structure of Collaborative Learning Ontology.[1] Here, we concentrate on the concept“Learning Goal” which is one of the most important concepts for forming a learning group, because each learner joins in a collaborative learning session to attain some learning goals. The “Learning Goal” can be specified as two kinds of goals: “common goal” as a whole group and “personal goal” for each learner. The concept “personal goal” can be specified as two kinds: the goal represented as a change of a learner's knowledge/ cognitive states, and the goal attained by interaction with other learners.

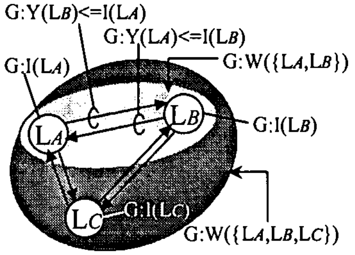


Fig. 2. Learning Goal Ontology

We classify the goal of the first person (*I*), that of the first person to interact with the second person (*You*), and that of the whole group as I-goal, Y⇐I-goal, and W-goal, respectively. I-goal, which is described as G:I, represents what a learner is expected to acquire. Y⇐I-goal, which is described as G:Y⇐I, represents what a learner is expected to acquire through the interaction. W-goal expresses the situation being set up to attain Y⇐I-goals and we describe the goal as G:W. W-goal is a common goal characterizing the whole group.

Fig. 2 represents learning goals in a group where three learners: L_A, $L_B$and $L_C$are participating. Learner $L_A$has an I-goal which is attained through this collaborative learning session and this goal is described in Fig. 2 as G:I(L_A). Both $L_B$and $L_C$have I-goals, and they are represented as G:I(L_B) and G:I(L_C) respectively. G:Y(L_B)⇐I(L_A) is a Y⇐I-goal between $L_A$and $L_B$observed from L_A's viewpoint. In other words, it means the reason why $L_A$interacts with L_B. Concerning this interaction between $L_A$and L_B, there is also a Y⇐I-goal observed from L_B's viewpoint. That is, it is the reason why $L_B$interacts with L_A. This Y⇐I-goal is represented as G:Y(L_A)⇐I(L_B). Both G:I(L_A) and G:Y(L_B)⇐I(L_A) are personal goals of

[1] Notation: the schemata define the W-concept and the U-concept. The W-concept has entity a, which is an instance of the concept P-concept, as a part. The entity a plays a specific role (Role-name) in the W-concept. The concept P-concept has a semicircle on the right sides. It means the concept is defined in other schema. The L-concept is a specification of the U-concept, and the U-concept is a generalization of the L-concept.

Table 1. W-goals

		W-goal	Definition	Src.
Singleton W-goal	M-P x S-S	Setting up the situation for Peer Tutoring $\boxed{\text{PT}}$	Setting up the situation where a learner teaches something to another learner.	[6, 7]
		Setting up the situation for Anchored Instruction $\boxed{\text{AI}}$	Setting up the situation where a learner diagnoses another learner's problem and then solve it (Problem-based Learning)	[4]
	S-P x M-S	Setting up the situation for learning by Cognitive Apprenticeship $\boxed{\text{CA}}$	Setting up the situation to learn knowledge or skill as an apprentice	[5]
		Setting up the situation for sharing (Meta-) Cognitive function between learners $\boxed{\text{SC}}$	Setting up the situation to share cognitive or meta-cognitive function between learners based on Sociocultural Theory	[25, 26]
		Setting up the situation for sharing Multiple Perspectives $\boxed{\text{CF}}$	Setting up the situation to evoke a learner's reflective thinking based on Cognitive Flexibility theory.	[22, 23]
	M-P x N-S	Setting up the situation based on Distributed Cognition $\boxed{\text{DC}}$	Setting up the situation where full participants, whom knowledge bases are different each other, discuss problems	[21]
		Setting up the situation based on Cognitive Constructivism $\boxed{\text{CC}}$	Setting up the situation where full participants discuss problems	[19]
CW-goal		Setting up the community for Legitimate Peripheral Participation $\boxed{\text{LPP}}$	Setting up the the community of practice for peripheral participant	[15, 16]
		Setting up the situation for Observational Learning $\boxed{\text{OL}}$	Setting up the situation to share other learners' learning processes	[2]

Note: $\boxed{**}$ means an abbreviation for the W-goal.
e.g., The W-goal "Setting up the situation for Peer Tutoring" is abbreviated as "PT".

L'_A. G:W($\{L_A, L_B\}$) is a W-goal of the learning group ($\{L_A, L_B\}$). G:W($\{L_A,L_B,L_C\}$) is a W-goal of the learning group ($\{L_A, L_B, L_C\}$).

We have identified goals for collaborative learning for each of the three categories, and constructed I-goal Ontology, Y⇐I-goal Ontology, and W-goal Ontology with justification based on learning theories.[2] We can expect learners to acquire not only new knowledge concerning problems they solve, but also cognitive skills, meta-cognitive skills, and skills for self-expression through the collaborative learning session (I-goals). Each I-goal has several phases of development. It is difficult to understand from a theory what educational benefit is expected to a learner, because of lack of unified systematic terminology to represent a variety of phases. So, we adopt the terminologies used in two established findings: Rumelhart & Noman's work[15] on knowledge acquisition and Anderson's one[1] for skill development. The process to acquire a specific knowledge includes three qualitatively different kinds of learning[15]: Accretion, Tuning, and Restructuring. Concerning development of skills, there are also three phases of learning: Cognitive stage, Associative stage, and Autonomous stage[1, 8].

The learner is expected to achieve these I-goals through interaction with other learners. For example, to achieve the I-goal "Acquisition of Content-Specific Knowledge (Accretion)", some learners could take the Y⇐I-goal "Learning by being Taught[5]", while some learners could take another Y⇐I-goal "Learning by Observation[2]".

Table 1 shows the W-goals. The W-goals are classified into four kinds (*i.e.*,Three kinds of singleton W-goals and one Composite W-goal) according to their structures. To form a learning group means to pick up learners who join in the group as members and to assign a specific role in the group to each member. The formation should have rationale supported by learning theories. The structure of learning goals expresses the rationality. A W-goal, which is a learning goal as a whole group, provides the rationale for the interaction among the members. It means that a W-goal specifies a rational arrangement of Y⇐I-goals. Fig. 3 shows a typical representation for the structure of a W-goal. It would be more easily to understand a learning theory by preparing the structure to represent the theory and filling in each component of the structure with suitable concepts according to the theory.

A learning theory generally argues the process that learners, who play a specific role, can obtain educational benefits through interaction with other learners who play other roles. The theories have common characteristics to argue effectiveness of a learning process focusing on a specific role of learners. So, we represent the focus in the theories as Primary Focus and Secondary Focus.

Primary Focus (P): a learner's role that is mainly focused in the learning theory. The learner who plays this role (P-member) is expected to gain the main educational benefit.

[2] The details of the ontologies are described in our previous paper[14]. Here, we show the outline of the ontologies.

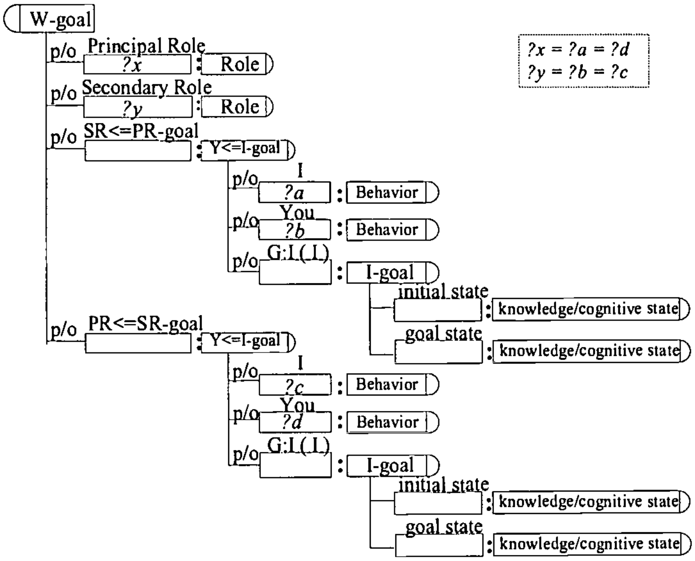


Fig. 3. Conceptual Structure of a W-goal

Secondary Focus (S): a learner's role that is weakly focused in the learning theory. The learner who plays this role (S-member) is needed as a companion to enable a P-member to attain his/her learning goals.

We classify the W-goals into the following four kinds depending on the number of the components P and S.

Singleton W-goal: Each Singleton W-goal can exist independently.

Multiple-P x Single-S: The W-goal of M-P x S-S type can have multiple P-members and single S-member.

Single-P x Multiple-S: The W-goal of S-P x M-S type can have single P-member and multiple S-members.

Multiple-P x No-S: The W-goal of M-P x N-S type has only one role for its members. In this group, each learner plays a role of companion for the other learner, while he/she gains main educational benefit.

Composite W-goal: The CW-goal includes another group as its component S.

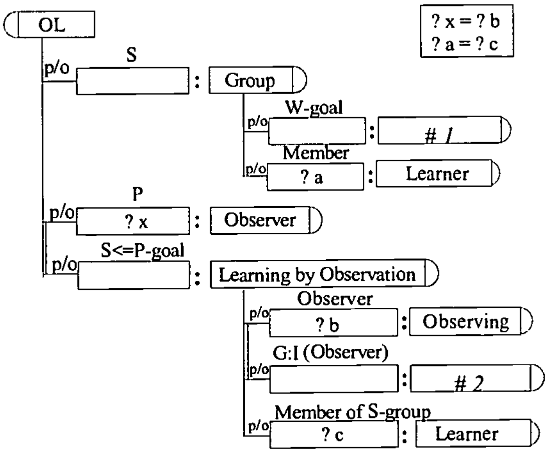


Fig. 4. An Example of CW-goal: OL

For example, in the situation of Peer Tutoring, there are two roles: *Peer Tutor* and *Peer Tutee*. Main educational benefit is tuning of content-specific knowledge by externalizing a learner's knowledge[6, 7]. So, P is identified as *Peer Tutor* and S is identified as *Peer Tutee*. From the viewpoint of assigned task, the role of main problem-solver is *Peer Tutee* who wants to get a new knowledge to perform assigned tasks, while the role of helper is *Peer Tutor*. The number of members who play *Peer Tutee* (S) should be single, the number of members who play *Peer Tutor* (P) can be multiple, and the W-goal **PT** is identified as a M-P x S-S type.

A group attaining a W-goal(W_i) can have another group, which has another W-goal(W_j), as the component S of the W-goal(W_i). We call the W-goal(W_i) "**CW-goal**" which means a composite W-goal. Fig. 4 shows the conceptual structure of the CW-goal Observational Learning[2]. The learning group has *Observers* as its component P.

The *Observers* require a group (*i.e.,*its component S) as an object to observe meaningful interaction. In the figure, the W-goal, which is set in #1, depends on what I-goal is set in #2. For example, if accretion of content-specific knowledge is set in #2 as *Observer*'s I-goal, the W-goal **PT** is recommended as S's W-goal (#1).

A W-goal has two kinds of goals of interaction as follows:

S⇐P-goal: a Y⇐I-goal which means how and for what purpose the P-member interacts with the S-member.

P⇐S-goal: a Y⇐I-goal which means how and for what purpose the S-member interacts with the P-member. In the collaborative learning session, all members of learning group are expected to get some educational benefits. So, the S-member also has an I-goal, and the P⇐S-goal should be effective to attain the I-goal.

The entities of these goals refer to the concepts defined in the Y⇐I-goal Ontology. The conditions, which are proper to each W-goal, can be added to the concepts, if necessary. Each of the Y⇐I-goals referred to by S⇐P-goal and P⇐S-goal consists of three components as follows:

I: a role to attain the Y⇐I-goal. A member who plays I role (I-member) is expected to attain his/her I-goal by attaining the Y⇐I-goal.

You: a role as a partner for the I-member.

G:I: an I-goal which means what the I-member attains.

Each W-goal can be expressed by a set of Y⇐I-goals and I-goals. We can identify a group formation to start an effective collaborative learning session with these goals.

3 Advantages and Remaining Tasks of Learning Goal Ontology for Forming an Effective Learning Group

In a traditional classroom, sometimes a teacher divides students into several subgroups, and then the students start collaborative learning in the subgroup all at once. Such collaborative learning does not ensure educational benefits for every student, because it depends on a student's knowledge/ cognitive state whether collaborative learning is effective or not, and progress in learning differs from student to student.

So, we have been proposing a network-based new learning environment to support individual learning and collaborative learning dynamically. In the environment, each learner is solving problems individually with an ITS. When the ITS detects a desired situation for a learner (triggered-learner) to shift from individual learning mode to collaborative learning mode, the ITS forms an effective learning group for the learner, and then the members of the group start a collaborative learning session. In the group, not triggered-learner but every member should be ensured to attain individual learning goals through specific interaction with the other members. To encourage the interaction, every member is assigned a specific role in the group. When the members attain their learning goals, they close the session and return individual learning mode. We call the idea of dynamic group formation *"Opportunistic Group Formation (OGF)"*.

With our Learning Goal Ontology we can represent the several group formations whose effectiveness is ensured by learning theories. It means that the ontology brings the following benefit: When a personal goal for a learner (*i.e.,*I-goal or Y⇐I-goal) is decided, we can identify learning theories which propose learning groups to facilitate that the learner attain the personal goal. And then, we can form a specific group and identify roles assigned to the members of the group according to the theory.

If there are many theories to enable a learner to attain a specific personal goal, we can form many learning groups supported by the theories as candidates. Then, we have to narrow down the candidates to one. How can we select one?

Each learner plays a specific role in collaborative learning session. Every role has necessary conditions which should be satisfied by a learner who plays the role. The conditions will work as constraints to narrow down the candidates. If there are still some candidates after checking the conditions for role assignment, there are no rules for conflict resolution between all possible learning theories.

One might want to select one of the most profitable theory-based learning groups for a learner to attain a personal goal. Every theory expresses a different learning situation. The differences between theories do not mean the differences of the degree of effectiveness, but diversity of means to attain a goal. So, it is hard to compare a theory with the others on the effectiveness for helping a learner attain a personal goal.

There is another solution of the problem for narrowing down the candidates to one. Are learning theories exclusive each other? If the candidates can be integrated into one, a stronger learning group will appear: a learner is expected to attain a personal learning goal through some kinds of interaction, and each interaction is justified by a learning theory.

4 Is a Learning Theory Exclusive or Harmonious with Other Theories?

In actual learning environment, teachers often adopt the style of collaborative learning. If the group includes a member $L_A$whose knowledge base and/or experiences are relatively poor, it would be difficult for $L_A$to discuss with other members and to solve a problem collaboratively. $L_A$is expected to grow into a senior through practice in the group. This type of learning group is similar to the group based on the theory "LPP" which describes a process in which a newcomer grows into a senior[15,16]. Fig. 5 shows typical learning group formation the W-goal "LPP" where three learners: L_A, $L_B$and $L_C$are participating. As a whole group, all members solve a problem collaboratively, and $L_A$is regarded as a **Peripheral Participant** and $\{L_B$and $L_C\}$ are regarded as **Full Participants**.

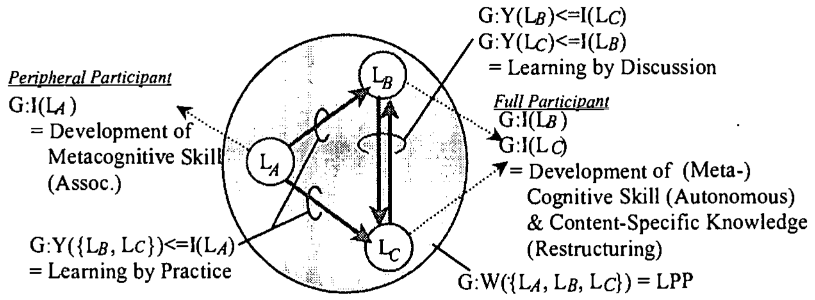


Fig. 5. An Example of Group Formation: LPP

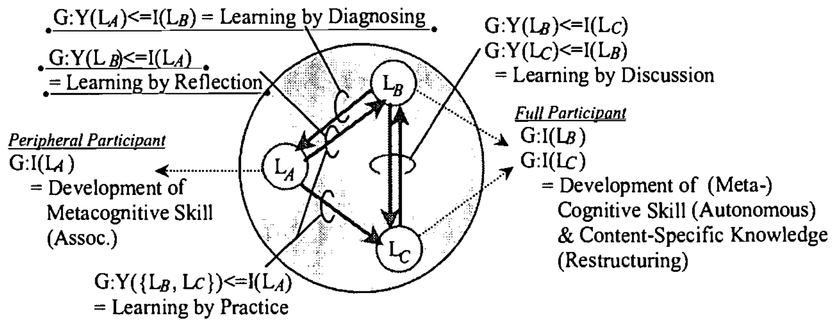


Fig. 6. An Example of Actual Learning Group

In this case, many skillful teachers will arrange for an excellent learner (*e.g.*,L_B) to help $L_A$in the group. For example, when a new student comes to our laboratory, a senior student may work as a tutor for the new student. Fig. 6 shows this type of learning group formation. We can find additional Y⇐I-goals between $L_A$and $L_B$in Fig. 6 as compared with Fig. 5. The teacher will expect different types of interaction between $L_A$and L_B, which bring additional educational benefits to them. This type of group formation can not be interpreted by a single learning theory.

In a learning group supported by "LPP", can all Peripheral Participants grow up into full participants? According to the theory "LPP", a learner (*i.e.*,Peripheral Participant) can acquire knowledge on the community and develop his/her (meta-) cognitive skills only by the learner's own practice. It is not assumed the other learners (*i.e.*,Full Participants) help the Peripheral Participant grows up. It seems that there is a gap between the Peripheral Participant and the Full Participant. Especially concerning the development of (meta-) cognitive skills, a Peripheral Participant can observe not the process in which

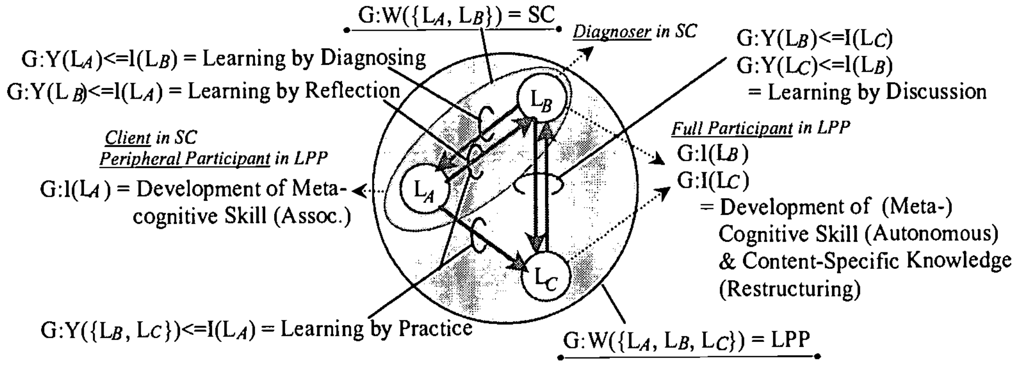

G:W({L_A, L_B}) = SC
Diagnoser in SC
G:Y(L_A)<=I(L_B) = Learning by Diagnosing
G:Y(L_B)<=I(L_A) = Learning by Reflection
G:Y(L_B)<=I(L_C)
G:Y(L_C)<=I(L_B)
= Learning by Discussion
Client in SC
Peripheral Participant in LPP
G:I(L_A) = Development of Meta-cognitive Skill (Assoc.)
Full Participant in LPP
G:I(L_B)
G:I(L_C)
= Development of (Meta-) Cognitive Skill (Autonomous) & Content-Specific Knowledge (Restructuring)
L_A
L_B
L_C
G:Y({L_B, L_C})<=I(L_A) = Learning by Practice
G:W({L_A, L_B, L_C}) = LPP

Fig. 7. An Example of Combined Learning Group: W-goal LPP & SC

a Full Participant uses the skill, but input and output for using the skill. It is difficult to learn the process by observing only input and output.

Concerning the W-goals, both W-goals "AI"[4] and "SC"[26] assume to have a "poor learner" who engages to solve a problem and a "helper" for the learner. The W-goal "AI" has a **Problem-Holder**, who has a difficulty in solving a problem, and an **Anchored Instructor**, who diagnoses the Problem-Holder's problem and gives advice to him/her. Similarly, the W-goal "SC" has a **Client**, who externalizes his/her own thinking process, and a **Diagnoser**, who diagnoses the Client's thinking process and evaluates the process. In both W-goals, a "poor learner" is expected to attain his/her I-goal, by a "helper"'s advice. Each of these W-goals can be combined with one of the other W-goals. That is, if it is difficult for a learner to attain an I-goal, we can combine the W-goal "AI" or "SC", and one of the other W-goals to help the learner attain the I-goal.

In the case of Fig. 6, we can interpret the group as a combination of two groups. One group ($Group_i$) consists of two Full Participants ($L_B$and L_C) and one Peripheral Participant (L_A). The W-goal of $Group_i$ is "LPP". Another group ($Group_j$) consists of a Client (L_A) and a Diagnoser (L_B), and the W-goal of the group is "SC". Fig. 7 shows the combination of two groups. In this learning group, $L_A$is expected to participate in the session more easily thanks to the help of L_B. For L_B, it is an opportunity for diagnosing L_A's authentic problems and helping $L_A$to participate in the collaborative learning session. Through the experience, we can expect $L_B$to develop his/her cognitive skill in two ways. For L_C, he/she will be able to get the same educational benefit with participating in the group shown in Fig. 5, because his/her activity is equal between the both groups.

For the combination of theory-based learning groups, the role of ontology is to clarify principles of combination. In combined groups, it should be guaranteed that all members can attain their own learning goals. At this stage, we store possible patterns of combining some theory-based learning groups as a pattern library. The ontology should not only represent the patterns, but also the principles which express the design rationale why the groups can be combined into one. When we can clarify the principles, an intelligent educational support system will be able to infer an effective learning group formation based on the principles opportunistically: The group formation is not picking up an appropriate one from the static pattern library. In this paper, we have described the possibility of combination the W-goal "AI" or "SC", and other W-goals. We have to consider the other types of combination.

5 Conclusions

We have discussed Learning Goal Ontology which will be able to make it easier to form an effective collaborative learning setting and to analyze the educational functions for a learning group. By considering the personal and common goals, we have identified three kinds of learning goals; I-goal, Y⇐I-goal and W-goal. In this paper, we described the outline of Learning Goal Ontology, and summarized advantages and remaining tasks for the ontology. We proposed possibility that theory-based learning groups can be combined into one in order to help a learner attain his/her learning goals and showed an example of effective learning group formation which is formed by combining multiple theory-based learning groups. With the ontology, it will be possible to compare and synthesize the learning theories to design the collaborative learning settings.

At this stage, we mainly focus on the learning goals. Future work includes to construct ontologies on remaining concepts in Collaborative Learning Ontology. Advantage of collaborative learning includes emotional factors: *e.g.,*motivation, familiarity. It is also our future work how to treat these factors.

References

1. Anderson,J.R. "Acquisition of Cognitive Skill", Psychological Review, vol.89(4), pp.369-406, (1982).
2. Bandura, A. "Social Learning Theory", New York: General Learning Press, (1971).
3. Bransford, J. D., Vye, N., Kinzer, C., & Risko, R. "Teaching thinking and content knowledge: Toward an integrated approach", In. B. Jones & L. Idol (Eds.) Dimensions of thinking and cognitive instruction. Hillsdale NJ: Erlbaum. pp. 381-413, (1990).
4. Cognition and Technology Group at Vanderbilt. "Anchored instruction in science education", In: R. Duschl & R. Hamilton (Eds.), Philosophy of science, cognitive psychology, and educational theory and practice. Albany, NY: SUNY Press. pp. 244-273, (1992).
5. Collins,A. "Cognitive apprenticeship and instructional technology", In: Idol, L., & Jones,B.F. (Eds.) Educational values and cognitive instruction: Inplications for refoem., Hillsdale, N.J.: L. Erlbaum Associates, (1991).
6. Cooke, N.L., Heron, T.E., & Heward, W.L. "Peer tutoring: Implementing classroom wide programs", Columbus, OH: Special Press, (1983).
7. Endlsey, W.R. "Peer tutorial instruction", Englewood Cliffs, NJ: Educational Technology, (1980).
8. Fitts,P.M. "Perceptual-Motor Skill Learning", In: Melton,A.W. (Ed.), Categories of Human Learning, Now York: Academis Press. pp.243-285, (1964).
9. Flavell, J. H. "Metacognitive aspects of problem-solving", In L.B. Resnick (Ed.), The nature of intelligence. Hillsdale, NJ: Erlbaum. pp.231-235, (1976).
10. Ikeda, M., Hoppe, U., & Mizoguchi, R. "Ontological issue of CSCL Systems Design", Proc. of AI-ED 95, pp.234-249, (1995).
11. Ikeda, M., Go, S., & Mizoguchi, R. "Opportunistic Group Formation", Proc. of AI-ED 97, pp.166-174, (1997).
12. Inaba, A., Ikeda, M., Mizoguchi, R., & Toyoda, J. "The Learning Goal Ontology for Collaborative Learning", http://www.ai.sanken.osaka-u.ac.jp/~inaba/LGOntology/, (1999).
13. Inaba, A. & Okamoto, T. "The Intelligent Discussion Coordinating System for Effective Collaborative Learning – What is the role of artificial intelligence in supporting collaboration? –", AIED 97: Workshop Notes IV, pp.26-33, (1997).
14. Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., & Toyoda, J. "How Can We Form Effective Collaborative Learning Groups? – Theoretical justification of "Opportunistic Group Formation" with ontological engineering –", pp.xx-xx, (2000).
15. Lave, J. "Cognition in practice: Mind, mathematics and culture in everyday life", Cambridge University Press, (1988).
16. Lave,J. & Wenger, E. "Situated Learning: Legitimate peripheral participation", Cambridge University Press, (1991).
17. Mizoguchi, R., & Bourdeau, J. "Using Ontological Engineering to Overcome Common AI-ED Problems", IJAIED, 11, (2000).
18. Mizoguchi,R., Ikeda,M., & Sinitsa,K. "Roles of Shared Ontology in AI-ED Research", Proc. of AI-ED 97, pp.537-544, (1997).
19. Piaget, J., & Inhelder, B. "The Psychology of the Child", New York: Basic Books, (1971).
20. Rumelhart, D.E., & Norman, D.A. "Accretion, Tuning, and Restructuring: Modes of Learning", In: Cotton, J.W., & Klatzky, R.L. (Eds.) Semantic factors in cognition. Hillsdale, N.J. : L. Erlbaum Associates., pp.37-53, (1978).
21. Salomon, G. "Distributed cognitions", Cambridge University Press, (1993).
22. Spiro, R. J., Coulson, R., L., Feltovich, P. J., & Anderson, D. K. "Cognitive flexibility: Advanced knowledge acquisition ill-structured domains", Proc. of the Tenth Annual Conference of Cognitive Science Society, Erlbaum, Hillsdale, NJ, pp.375-383, (1988).
23. Spiro, R. J., Feltovich, P., J., Jacobson, M., L., & Coulson, R. L. "Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquistion in ill-structured domains", http://www.ilt.columbia.edu/ilt/papers/Spiro.html (1995, accessed on March 2000).
24. Supnithi, T., Inaba, A., Ikeda, M., Toyoda, J., & Mizoguchi, R. "Learning Goal Ontology Supported by Learning Theories for Opportunistic Group Formation", Proc. of AIED99, (1999).
25. Vygotsky, L.S. "The problem of the cultural development of the child", Journal of Genetic Psychology, vol.36, pp.414-434, (1929).
26. Vygotsky,L.S. "Mind in Society: The development of the higher psychological processes", Cambridge, MA: Harvard University Press, (1930, Re-published 1978).

Learning Protocols for Knowledge Discovery: A Collaborative Data Mining Approach to Creative Science Education

Feng-Hsu Wang
Department of Information Management
5, The-Ming Road, Gwei Shan District, Taoyuan County 333, Taiwan
Tel: +886-3-3507001 ext. 3446
Fax: +886-3-3294449
E-mail:fhwang@mcu.edu.tw

One of the creative capabilities of scientists is the ability to turn data (observations) into knowledge, that is, the capability for knowledge discovery. In this paper, we propose a collaborative data mining approach to designing learning tools in educational environments for creative science education. Specifically, students can experience knowledge discovery by engaging in collaborative data mining activities that enable students to cooperate both with the computer and the other students. Data mining process is typically made up of a set of activities such as selection and sampling, preprocessing and cleaning, transformation and reduction, forming knowledge rules, evaluation and revising knowledge rules. The learning process is modeled as a set of learning protocols that properly distribute the data-mining work among students and computers. Based on these protocols, we design and implement a set of learning tools in a web-based learning environment for global climate exploration.

Keywords: **Learning protocol, knowledge discovery, data mining, learning environment, collaborative learning, science education.**

1 Introduction

Among the creative capabilities of scientists, the most important one is the ability to turn data (observations) into knowledge, that is, the capability for knowledge discovery. In this paper, we propose a collaborative data-mining approach to creative science education in learning environments. In this data-mining supported environment, students could observe real world data in different perspectives, derive their own classification rules and test the rules collaboratively, such that they can experience knowledge discovery by engaging in collaborative data-mining activities.

In this paper, we adopt learning protocols [9] to describe the learning processes. Learning protocols are a set of constraints, rules, or processes for structuring learning processes, and are externalized as executable methods, with roles, events, and actions made explicit. Learning protocols can be used to coordinate goal-directed, effective interaction in a group of learners. In this paper, we will devise a set of learning protocols that properly distribute the data-mining work among students and computers.

Based on the collaborative data-mining protocols, we design and implement a set of learning tools in the CILSE-GCE learning environment [7, 8]. CILSE-GCE is a web-based collaborative learning environment for global climate exploration. The task domain, global climate exploration, is inherently a scientific classification problem. Students are expected to induce classification rules by making observations under a couple of climatic features. These tools are designed with the intention not only to teach students the target knowledge, but also the scientific ways of study skills. We believe the students will achieve higher learning goals through the collaborative process of creating knowledge by themselves.

2 The CILSE-GCE Learning Environment

The target domain draws sources from the instructional material in the geographic climate course of senior high schools in Taiwan. One of the domain knowledge is the classification of each climate pattern, which is recognized as a specific set of the climatic attributes. In this paper, we focus on the construction of the climatic classification knowledge. Three components of the CILSE-GCE learning environment were built. They are the Virtual Classroom, Visualized Data Viewer, and Intelligent Tutor, respectively, which are outlined below.

The Virtual Classroom serves as the origin where teachers and students coordinate and collaborate. Through the Virtual Classroom, students could access the multimedia coursebase, the climatic GIS database (via the Visualized Data Viewer) and the historical literature database. These rich data sources allow students to observe, search and collect related information in different aspects regarding to the problems at hand. The CILSE-GCE environment also provides an intelligent tutor to help students induce the classification rules. During the rule induction process, a student has to identify what the settings of the relevant attributes are by exploring resources of all kinds. When he/she determines a specific set of attribute values, the intelligent tutor would evaluate the student's answer, and give suggestions to guide the student's further exploration.

A set of rich data sources are needed to allow students to observe, search and collect related information in different aspects regarding to the problems at hand. In the Visualized Data Viewer, rich climate information could be displayed in different layers of maps covering the globe. Students could select, resize and combine different information layers for display to investigate the climate attributes in different perspectives. Hotlinks to climatic data and statistical graphs associated with the typical cities are also provided to allow students to do some measurements and inferences. Up to now, we have collected more than 1700 city records of various kind of climatic information, such as latitude, temperature, precipitation, height above sea level, etc. This database is the main data source that students can collect related data and perform data-mining process to discover the classification knowledge. Figure 1 shows a snapshot of the Visualized Data Viewer.

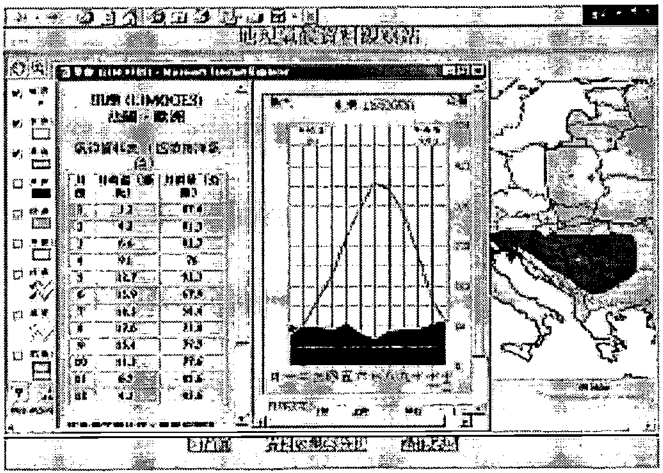


Figure1 A snapshot of the Visualized Data Viewer system.

3 Collaborative Data Mining as Knowledge Discovery

For creative science education, students are asked to acquire the learning skills of knowledge discovery, such as making observations and data collections, performing data analysis, generating hypotheses, testing hypotheses, and making conclusions. Standing from the viewpoint of knowledge discovery [2], we model the learning process as a data-mining process. Figure2 shows the set of data-mining activities, such as selection and sampling, preprocessing and cleaning, transformation and reduction, forming knowledge rules, evaluation and revising knowledge rules. Some steps of the data-mining process can be handled well with computer supports, especially those involving tedious computations and comparisons. Other steps are more suitable to be learning tasks for human students. In this section, we propose the framework of collaborative data mining within which each student member first applies the data-mining process to generate his/her private knowledge base, and then all students collaboratively integrate their private knowledge bases to a more general knowledge base, a result of social consensus process.

The first step in the data-mining process is to select a target data of interest from database, and to possibly sample the target data. The learning skills required of the students are the capability of observation and data collection. Based on the aspects they observe data, students can select all relevant attributes they think might be important to the classification problems at hand. Besides, there are so many samples in the database that students have to learn the sampling skill by selecting as typical samples as they can.

Secondly, the preprocessing and data cleaning step handles noises and unknown values, as well as accounting missing data fields. This step can be dealt with quite well with computer software. Thirdly, the data reduction and transformation step involves checking relevant features depending on the goal of the learning task and certain transformations on the data such as converting one type of data to another (e.g., discretizing continuous values), and/or defining new attributes. It is this step that testifies the hypothesis of attributes that students generated at the previous data observation step.

In the knowledge formulation step, students may apply one or more knowledge discovery techniques and tools on the transformed data set to extract valuable patterns. In this step, students can learn domain-dependent skills as well as the ability to work with computers, as is practiced by most scientists nowadays. Finally, the knowledge evaluation step involves interpreting the result with respect to the goal/task at hand. And as is often the case, students may get back to previous steps based on the evaluation results. Well-designed OLAP (OnLine Analysis Processing) tools are required for students to practice such kind of data analysis tasks. Note that the data-mining process is not a linear one. It might involve a variety of feedback loops, because any one step can result in changes in preceding or succeeding steps.

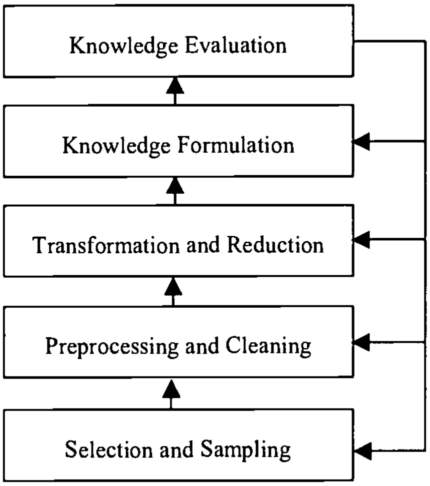


Figure2 The data-mining process.

4 Learning Protocols for Collaborative Data Mining

Learning protocols can be used to coordinate goal-directed, effective interaction in a group of learners. A learning protocol consists of a set of components. First, a protocol has a name signifying the situation type to which the protocol can be applied. Secondly, a protocol consists a sct of states and transitions. In each state the users can perform actions such as communicate or manipulate artifacts. A transition to another state is triggered by an action or a specific condition. Actually, a learning protocol can be represented as an event-driven state-transition graph. Thirdly, a protocol includes different roles pertaining to the persons involved in the enactment of the protocol. Finally, a protocol may contain various types of artifacts, such as text documents, graphical objects, test forms, etc. In the following, we design a set of learning protocols for the collaborative data mining process.

4.1 The protocol to construct member knowledge

The protocol shown in Figure 3 outlines the actions of personal data-mining process and coordinates the interactions between a student and the computer. There are totally ten states in the protocol. Each state and transition is described as follows. In the Observing Data state, the student observes the data in all aspects he/she consider important to classify the climatic patterns. The main data source is the Visualized Data Viewer. The student then defines a set of attributes (in the Defining Attribute state) that will be used to classify the climatic patterns. In the Sampling state, the student starts to collect data (cities) and fill in all the details of the climatic attributes that he/she had defined. Since some of the attributes are numeric values, the student has to transform them into symbolic ones (like temperature is high or low) in the Discretizing Attributes state for more data understandability.

In the Mining Rule state, students have to extract and write down the classification rules hidden in the collected data. For this purpose, we design a set of data analysis tool that depicts the distribution graph or dependency graph of the climatic data based on the attributes specified by the students, such as the ones shown in Figure 4.

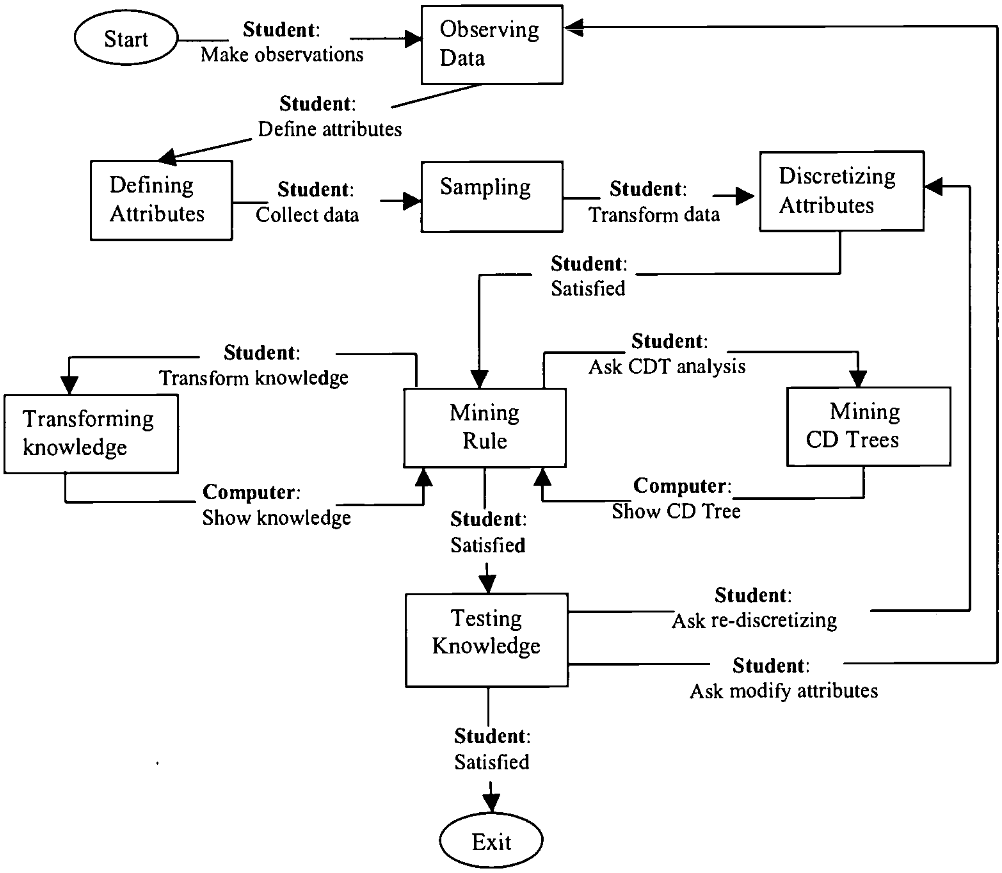
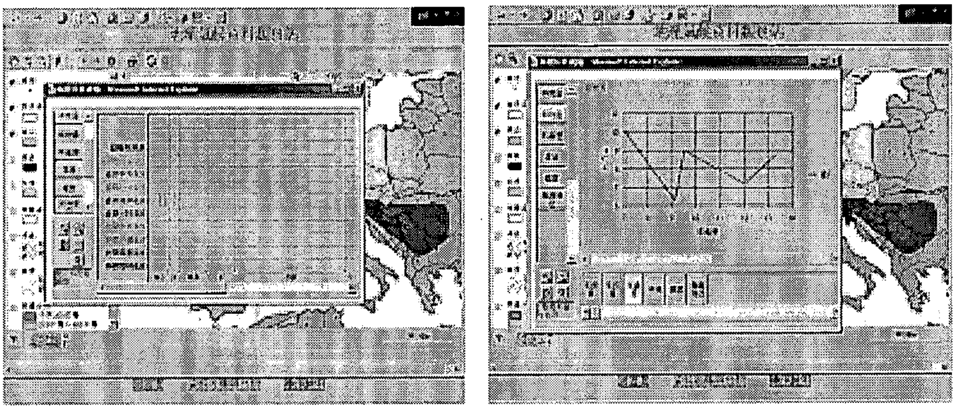


Figure3 The personal data-mining learning protocol.



(a) (b)

Figure4 (a) A data-mining support tool for comparing frequency distributions. (b) A data-mining support tool for depicting data dependency graphs.

Nevertheless, it would be still difficult for some students to discover the hidden knowledge (rules) without further computer supports. Hence, we design and implement another tool to facilitate the data-mining process in the Mining Decision Tree state. This tool uses a variation version of ID3 algorithm [4] to devise a Composite Decision Tree (CD Tree) out of the collected data. A As shown in Figure 5, students can use the CD tree to select and compose classification rules that are of more accuracy, stability and understandability. While rules provide a good local view of each knowledge unit, CD Trees provide another view that facilitates the comparison of different rule structures. . In the Transforming Knowledge state, the student can exchange the knowledge format from CD Trees to Rules, and vice versa. At last, the student can test his/her classification knowledge against the city cases in the Testing Knowledge state, and decide whether to further revise the knowledge.

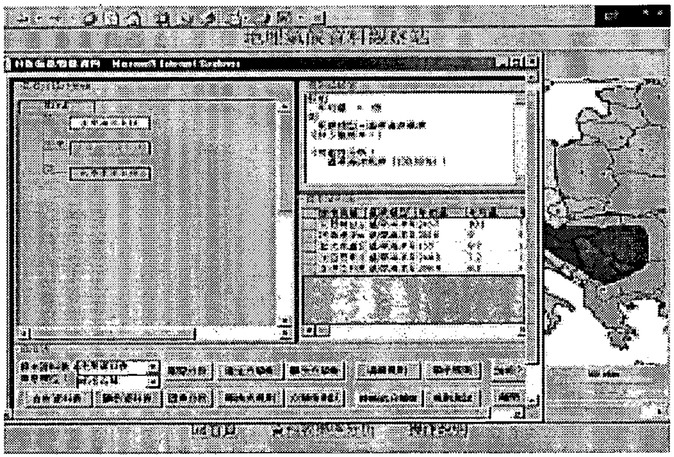


Figure5 A data-mining support tool to help students extract hidden knowledge (classification rules) from data.

4.2 The protocol to integrate group knowledge

After each student member establishes his/her own knowledge, the student group starts to perform the knowledge integration task collaboratively. The students achieve the knowledge integration goal by solving the classification problem collaboratively, trying to reach a consensus, which is the group knowledge. The corresponding learning protocol is shown in Figure 6. In the Presenting Cases state, a Coordinator (a software agent) selects a city case from the database for the student group to identify its climatic pattern. In the Classifying Case state, each student member applies his/her knowledge to solve the problem, and shows the applied rule and related information (such as the symbolic terms for each numeric attribute) in a shared working space. With the information shown in the shared working space, each student member starts revising his/her own knowledge by references to the correct answers and the colleagues' knowledge. Detail of the Revising Knowledge state is described in next protocol. Each time the member knowledge is revised, a new applied rule is sent once again to the shared working space. This process will loop until a temporary consensus is reached. At last, the Coordinator store the final rule set into the integrated knowledge base (i.e., the group knowledge). We adopt the Blackboard Architecture [3] to implement this learning protocol.

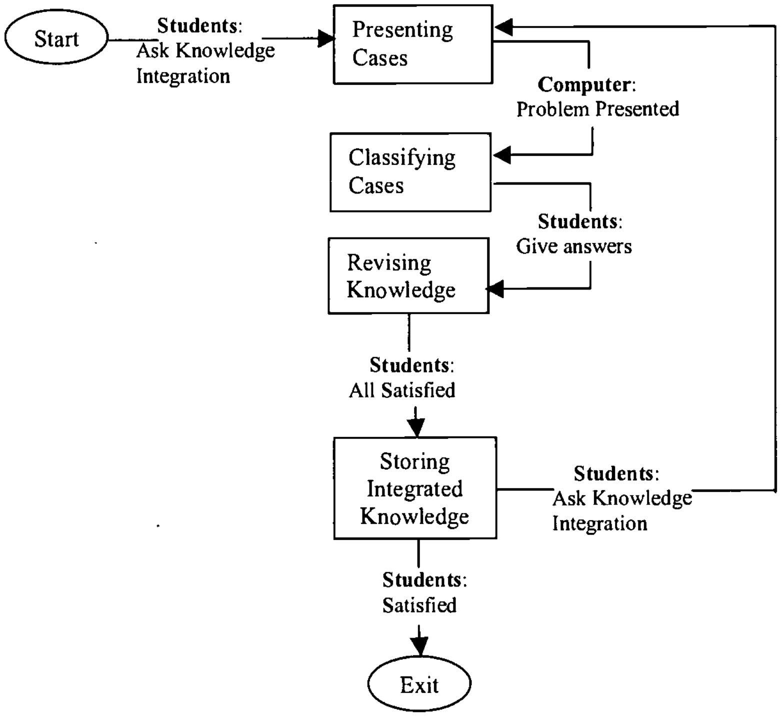

Start
Students: Ask Knowledge Integration
Presenting Cases
Computer: Problem Presented
Classifying Cases
Students: Give answers
Revising Knowledge
Students: All Satisfied
Storing Integrated Knowledge
Students: Ask Knowledge Integration
Students: Satisfied
Exit

Figure6 The collaborative knowledge integration learning protocol.

4.3 The protocol to revise member knowledge

When students ask to revise his/her private knowledge, the knowledge revising learning protocol, as shown in Figure 7, is entered. In this protocol, two kinds of knowledge operations, the knowledge generalization and knowledge specialization operations, are supported. Each student member can revise his/her private knowledge by applying the two knowledge operations and/or exchange knowledge through the Group Chatting state that involves chatting-support tools. Each kind of knowledge operation can be applied to the various artifacts such as rule structures, numeric attribute intervals, and attributes. Specifically, in Knowledge Generalization state, students can delete conditions from rules, reduce numeric attribute intervals or delete some attributes from the attribute set, while in Knowledge Specialization state, the students can add conditions into rules, extend some numeric attribute intervals or add new attributes into the attribute set. To facilitate both kinds of knowledge revision, an automated rule testing and warning subsystem is implemented to list the rules that are potential for further generalization or specialization based on the test result against any data set.

5 Conclusions

In this paper, we have proposed and implemented a collaborative data-mining support tools for knowledge discovery in creative science education. These functional extensions are being integrated to our previous Web-based learning environment, CILSE-GCE. This collaborative process fosters all the constructive design principles mentioned in [1, 5], such as observation, interpretation construction, contextualization, cognitive apprenticeship, collaboration, multiple interpretations, ownership of knowledge, self-awareness of construction process. In this collaborative learning model, students would experience the process of looking for patterns collaboratively. Besides, we find that learning protocols are very effective ways to the description and implementation of learning processes. Finally, it is indicated that during free exploration of a problem space, greater learning occurred if students adopted more systematic strategies for rule induction [6]. Further evaluation tests will be conducted to provide beneficial evidences of such kinds of discovery learning.

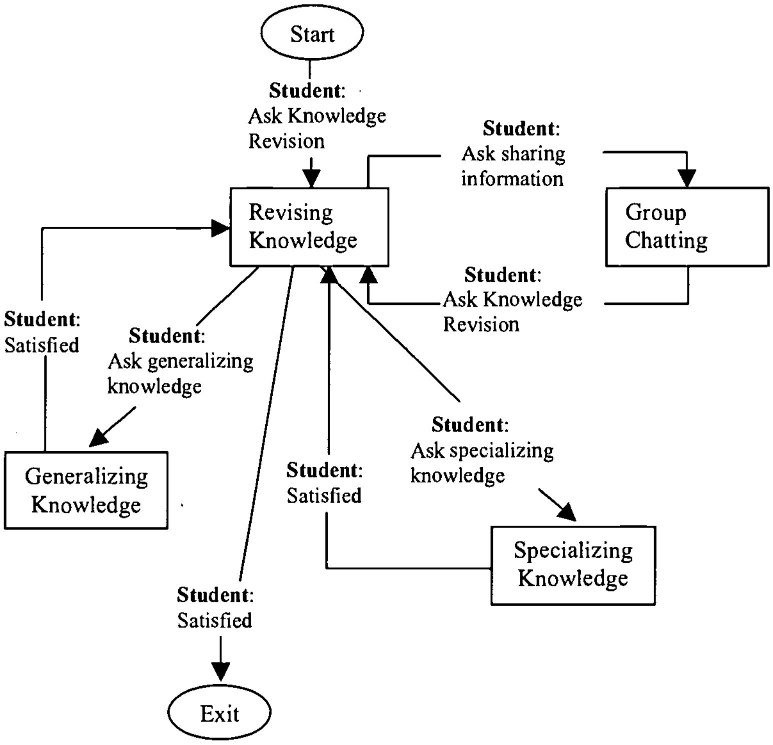

Start
Student: Ask Knowledge Revision
Revising Knowledge
Student: Ask sharing information
Group Chatting
Student: Ask Knowledge Revision
Student: Satisfied
Student: Ask generalizing knowledge
Generalizing Knowledge
Student: Satisfied
Student: Ask specializing knowledge
Specializing Knowledge
Student: Satisfied
Exit

Figure7 The knowledge revising protocol.

References

[1] Black, J. B. & McClintock, R. O., "An interpretation construction approach to constructivist design". In B. G. Wilson & D. N. Perkins (Eds.), "Constructivist Learning Environments: case studies in instructional design" (pp. 25-32), Englewood Cliffs, NJ: Educational Technology Publications, (1996).

[2] Fayyad, U. M., "Data mining and knowledge discovery: Making sense out of data", IEEE Expert, Vol.10, pp. 20-25, (1996).

[3] Nii, H. P., "Balckboard systems: Blackboard application systems from a knowledge based perspective", AI Magazine, Vol. 7, No. 3, (1986).

[4] Quinlan, J. R., "Induction of decision trees", Machine Learning, Vol. 1, No. 1, pp. 81-106, (1986).

[5] Honebein, P.C., "Seven goals for the design of constructivist learning environments", In B. G. Wilson & D. N. Perkins (Eds.), "Constructivist Learning Environments: case studies in instructional design", (pp. 11-24). Englewood Cliffs, NJ: Educational Technology Publications, (1996).

[6] Vollmeyer, R., Burns, B. D. & Holyoak, K. J., "The impact of goal specificity on strategy use and the acquisition of problem structure", Cognitive Science, Vol. 20, pp.75-100, (1996).

[7] Wang, F. H. & Chen, C. H., "CILSE-GCE: A collaborative intelligent learning support environment on world wide web", Annual Meeting of ED-MEDIA, Seattle, WA, USA., June, pp. 264-269, (1999).

[8] Wang, F.H., “On the design of an intelligent exploratory environment for geographic climates on WWW”, Computer Networks and ISDN Systems, Vol. 30, pp. 699-700, (1998).

[9] Wessner, M., Pfister, H. R. & Miao, Y., “Using Learning Protocols to Structure Computer-Supported Cooperative Learning”, Annual Meeting of ED-MEDIA, Seattle, WA, USA., June, pp. 471-476, (1999).

Monitoring and Verifying Mathematical Proofs Formulated in a Restricted Natural Language

Peter Schmidt
Department of Computer Science, University of Bonn
Römerstr. 164, 53117 Bonn, Germany
E-mail: peter@uran.informatik.uni-bonn.de

A restricted natural language is presented which is suitable for formulating mathematical proofs in the domain of calculus. A line of a proof according to the language consists of three parts: A marking, a proof statement, and a foundation of the statement. Foundations include among others the name of a theorem, the name of a concept, or a formula manipulation operation. It is demonstrated how mathematical proofs worded in that language may be automatically monitored and checked for correctness and completeness by a computer program. For that, techniques of the fields of theorem proving and of formula manipulation are applied; the lines of the original proof are transformed into a quantifier free form and checked line by line; an internal knowledge base of concepts and theorems allows for verifying proof statements which are founded by concept definitions or theorem applications. The described methods may be used in virtual or face-to-face universities for the purpose of proof exercises by students or for the purpose of automatically checking and scoring student proofs. The approach together with a medium-grained XML representation of concepts, theorems, and proofs may form the core of a learning environment which gives students the opportunity of an intensive interactive occupation with mathematical proofs.

Keywords: **Calculus Proofs, Verifying, Restricted Natural Language**

1 Introduction

Finding and constructing mathematical proofs are standard activities of persons who study mathematics or disciplines of science. For learning purposes, it would be desirable to have an interactive software system into which students could enter a mathematical proof in the usual way utilizing the natural language and the software system would monitor and verify the student's proof or provide help if needed.

From the side of the field of mechanical theorem proving, techniques and procedures are available to automatically prove theorems or check a given proof, if the theorem or the proof are worded in a formal language like first order logic or the quantifier free clause form (see e.g. [1], [5]). The main bottleneck to reach the above mentioned goal is the difficulty of processing and correctly understanding natural language input. As a solution to the problem or as a compromise we here suggest a restricted natural language to formulate proofs. The language results from an inquiry into mathematical proofs which occur in mathematical textbooks of the domain of calculus (see e.g. [8]). We chose the domain of calculus because of the importance of calculus for the edifice of mathematics and for many practical applications and because calculus belongs to the first fields which are studied at the universities.

Secondly, we discuss how proofs utilizing that restricted language may be automatically monitored and checked for correctness and completeness by a computer program. To monitor a proof, the proof is transformed into an internal form which includes the quantifier free notations of the occurring logical expressions. A proof is checked line after line like a human would do who tries to verify a given proof. The checking for correctness of the single statements relies on the techniques of the fields of theorem proving and of formula manipulation and of their combinations. Regarding the theorem proving techniques we utilize methods which are similar to the methods of Bledsoe, Boyer and Henneman to automatically prove limit theorems ([2],[3]).

Apart from providing opportunities of doing proof exercises, the described methods may be used in virtual or face-to-face universities for the purpose of automatically checking and scoring proofs of students.

Thirdly, we shortly discuss the extension of the approach to an extensive learning environment.

2 Mathematical Theorems and Proofs in the Domain of Calculus

The *subjects of calculus* include among others limits of sequences and functions, derivations of functions, determination of properties of functions, integrals, the study of special classes of functions, and many practical applications of theoretical results.

Proof methods used in calculus are multifarious and include direct proofs using the analytical definitions of concepts like limit, continuous or differentiable (epsilon-delta notation), inductive proofs, indirect proofs or proofs by counter-examples, or direct proofs utilizing chains of inferences of already proven theorems.

A large set of proofs in the domain of calculus follows a recurrent pattern. One characteristic of those proofs is the use of analytical definitions of the main concepts to establish the proof. A further characteristic of many proofs is that they employ formula manipulation methods as a central technique to establish the proof. Proofs often consist of a construction process. Those characteristics allow for monitoring proofs without a long chain of logical deductions.

3 A Restricted Natural Language to Formulate Proofs

The restricted language to word proofs is here informally described mostly by examples so that persons who are familiar with proofs of the domain of calculus can understand the scope of the various allowable statements. The language is not supposed to be exhaustive, but the current version of the language covers a large set of calculus theorems and proofs in textbooks and in collections of exercises.

The usual structure of a natural language proof in a textbook consists of a series of statements which are substantiated by one or more foundations. The statements may have a reference to other statements of the proof. The restricted language reflects that structure by dividing a proof into proof lines. Each proof line consists of up to three parts: a marking, a proof statement, and a foundation of the proof statement. By clearly separating the three parts of a proof line from each other, the variety of natural language wording reduces to a simple and easily comprehensible structure.

3.1 The wording of a proof

Basic elements of the language. There are a series of basic elements which may occur in a proof including numbers, variable names, function names, the universal quantifier (ALL), the existential quantifier (SOME), and the logical operators of negation (NOT) and of conjunction (AND). R denotes the real numbers. *Keywords* of the language generally consist of capital letters. *Intervals* play a central role in proofs and may be designated in the usual way, e.g. [a,b] for a closed interval of the real numbers, (a,b) for an open interval, or ALL x WITH |x-a|< delta for an interval with the point a in the middle of it. *Partitions of intervals* are often used in various contexts. They usually define end points and a list of intermediate points and fix the length or a maximum length of the resulting part intervals (see an example below). *Iterations* may be used in the usual way, e.g. i=0,...,n or j=1,2, ... to denote a finite or infinite sequence.

Proof statements. The current version of thc languagc comprises thc following proof statements which are described in the next paragraphs:

(1) Assignment statements. Assignment statements allow for defining new variables or functions. An assignment statement starts with the keyword LET. Examples are
LET delta = min(delta1,delta2),
where min denotes the minimum function and delta1 and delta2 are earlier defined variables, or
LET h(x) = f(x)+g(x) ALL x IN [a,b] , where the new function h(x) is defined, or
LET f : [a,b] -> R , where a function and its domains are defined.

(2) Choice statements. Choice statements describe a choice of an entity from a set of possibilities. A choice may e.g. refer to a number chosen from an interval or to a partition of an interval. A choice statement starts with the keyword CHOOSE. The format of such a statement depends on the choice situation. Simple examples are

CHOOSE eps > 0 or CHOOSE x IN [a,b].

An example which covers the choice of a partition of an interval is

CHOOSE PARTITION p OF [a,b] WITH $a=x_0 < x_1 < ... < x_n=b$ AND ($|x_i-x_{i-1}| <$ delta , i=1,...,n) ,

where [a,b] is an interval, x_i are points in the interval, and the mentioned restriction of the lengths of the intervals $[x_{i-1},x_i]$ holds.

(3) Relational statements. Relational statements, i.e. equations and inequalities, frequently occur in calculus proofs. The statements often include constraints on the appearing variables. Typical recurrent examples relate to analytical definitions of concepts and formula manipulation operations. An example which states the definition of continuity is: ALL eps > SOME delta >0 ALL x WITH |x-a| < delta: |f(x)-f(a)| < eps. Often a chain of equations and inequalities appears like ALL x IN [a,b]: |f(x)+g(x)| <= |f(x)| + |g(x)| <= M+N < INFINITY . Another simple example of a relational statement is eps/2 + eps/2 = eps , where eps is a given variable.

(4) Property statements. Property statements describe a property of an entity, e.g. the property of a function to be continuous in an interval. An example is: f IS continuous IN [a,b]. Other properties which often occur in calculus proofs are e.g. uniformly continuous, monotonously growing, or differentiable.

A series of **further statements** which often appear in a proof more or less drive or structure the proof.

(5) Proof type statements. A *proof type statement* characterizes how the proof is done, e.g. by finding a contradiction. The statement starts with the keyword PROOF TYPE and is followed by the name of a proof method from a list of proof methods, e.g.
by DIRECT, DIRECT_BY_DEFINITION, DIRECT_BY_A_CHAIN_OF_THEOREMS, INDIRECT, COUNTEREXAMPLE, SPECIALIZATION, COMPLETE_INDUCTION. The classification of the proof may be relevant regarding several aspects which are mentioned below. An example of a proof type statement is: PROOF TYPE INDIRECT.

(6) To prove statements. *To prove statements* are used to specify what must be or will be proven. There are two variants which may precede a statement: *to prove* or *sufficient to prove*. Here are examples: Let us assume that the conclusion of a theorem is: 'The function f(x) is bounded in an interval [a,b]'. Then the first line of a proof may be e.g. TO PROVE SOME m > 0 ALL x IN [a,b]: |f(x)| < m or the first line of the proof may be e.g.

SUFFICIENT TO PROVE ALL x IN [a,b]: |f(x)| < 1 . In the first case the keywords are followed by a statement which is equivalent to the conclusion of the theorem. And in the second case the keywords are followed by a statement from which the conclusion of the theorem may be inferred.

(7) Assume statements. *Assume statements* are mostly found in indirect proofs. They then state the negation of the statement of the theorem. The statement starts with the keyword ASSUME and there follows another statement. An example is ASSUME NOT [c] , where [c] denotes the marking of the conclusion of the theorem (see an example in Theorem 2 below).

(8) Contradiction statement. A *contradiction statement* states the contradiction of statements occurring in the proof. The statement starts with the keyword CONTRADICTION and its foundation contains the contradicting statements in one or the other way. An example is CONTRADICTION {[4],[6]}. The statement says that the statements marked by [4] and [6], respectively, are contradictory (see an example in Theorem 2 below).

(9) Anchor statements and **induction step statements.** *Anchor statements* and *induction step statements* serve the purpose to structure induction proofs. The statements start with the keywords ANCHOR and INDUCTION STEP, respectively. Examples are ANCHOR n = 1 and INDUCTION STEP n TO n+1.

(10) Proof finishing statement. The *proof finishing statement* consists of the keyword QED and states that the proof is assumed to be complete.

Markings. *Markings* serve the purpose to mark statements so that other parts of the proof may refer to the marked statement. The markings consist of letters and digits embraced by brackets, e.g. [A].

Foundations. A *foundation*, possibly together with other foundations, substantiates a proof statement. There are a couple of possibilities of denoting a foundation: A foundation may consist of the name of a theorem, of a formula manipulation operation, of a property of an object, or of a line number which denotes a logical line of the current proof or of the theorem. The foundation of a logical proof line is enclosed in curled brackets whereby the single foundations are enclosed in brackets and separated by commas, e.g. {[4], [5]}.

3.2 Examples of user proofs

The following examples illustrate the use of the language to formulate proofs. Note the more often occurring *double points,* e.g. one in the proof line which is marked by [2]. That double point is necessary for reasons of uniqueness to separate the prefix containing the quantified expressions from the inequality. An alternative would be to use an IF ... THEN ... statement. The foundations starting with the letters *fm* refer to formula manipulation operations, e.g. {[fm: rewriting]} in line [7]. The theorems are here not worded according to the language. A corresponding wording is necessary when the theorems and the proofs are automatically processed by a monitoring program.

Theorem 1 (Sum of continuous functions)

```
Let
  [p1] f: R->R,   g: R->R,   a in R,
  [p2] f is continuous at the point a
  [p3] f is continuous at the point a
Then
  [c] f+g is continuous at the point a
```

Proof:

```
[1] PROOF METHOD DIRECT_BY_DEFINITION
[2] TO PROVE
     ALL eps > 0 SOME delta > 0 ALL x with |x-a| < delta : |(f(x)+g(x)) - (f(a) + g(a))|< eps {[c]}
[3] CHOOSE eps > 0
[4]  SOME delta1 > 0  ALL x with |x-a| < delta1 :  |f(x) - f(a)| < eps/2  {[p2]}
[5]  SOME delta2 > 0  ALL x with |x-a| < delta2 :  |g(x) - g(a)| < eps/2  {[p3]}
[6]  LET delta = min(delta1,delta2)
[7] ALL x IN R : |(f(x)+g(x)) - (f(a) + g(a))| =  |(f(x) - f(a)) + (g(x) - g(a))|  {[fm: rewriting]}
                                             <=  |(f(x) - f(a)| + |g(x) - g(a)|    {[fm: triangle inequality]}
[8] ALL x with |x-a| < delta : |(f(x)+g(x)) - (f(a)+g(a))| <= |(f(x)-f(a)| + |g(x)-g(a)| {[7]}
                                                            <  eps/2 + eps/2  {[4], [5]}
                                                            =  eps {[fm: simplification]}

[9] QED {[2],[8]}
```

Theorem 2 (Global Monotony)

```
Let
  [p1] f: [a,b]->R  is continuous
  [p2] f  is differentiable in (a,b)
  [p3] for all x in (a,b):  f'(x) > 0
Then
  [c] f is strictly monotonously growing in [a,b].
```

Proof:

```
[A] PROOF METHOD INDIRECT
[B]  ASSUME  NOT [c]  {[A]}
[C] SOME  x1 IN [a,b], SOME x2 IN [a,b] :   x1 < x2  AND f(x1) >= f(x2)  {[B]}
[D] ( f(x2) - f(x1) ) / ( x2 - x1 )  <= 0 {[C]}
[E] SOME x0 IN (a,b) : f'(x0) =  ( f(x2) - f(x1) ) / ( x2 - x1 ) {[Mean-value theorem]}
                                 > 0   {[p3]}
[F] CONTRADICTION {[D], [E]}
[G] QED
```

4 Monitoring and Checking User Proofs

A user may enter a proof of a given theorem utilizing the above described language. The natural language proof is then transformed into a quantifier free version. That version is suitable for applying techniques of theorem proving and of formula manipulation. Each step of the user proof is checked by one of several special procedures (see below). We will first discuss the quantifier free version of the above mentioned theorems. Then we will describe the special procedures in the context of checking the proof statements of Theorem 1 and of Theorem 2.

4.1 Quantifier Free Version of a Theorem and a Proof

To check a user proof the natural language proof is transformed into a quantifier free form. Generally, the known methods of the field of mechanical theorem proving apply to get a quantifier free version (see e.g. [1], [5]), but one has to take into account some particularities which result from the fact that the proof representation exceeds first order logic:
(i) The *choice statement* corresponds to a quantification. The identifier succeeding the element CHOOSE has to be treated as a universally quantified variable, if the constraint attached to the variable represents an interval. If the constraint represents an assignment, the variable corresponds to an existentially quantified variable. An example is: A statement "CHOOSE eps > 0" has to be treated as "ALL eps > 0".
(ii) The ranges (scopes) of the quantifiers are not explicitly given in the proof. They have to be determined according to the following rule: The range ends when another quantifier with the same variable name appears or with the last appearance of the variable name.

After having dealt with those exceptions one can apply the usual transformation procedures to the proof lines which contain quantifiers. The statements of the example proofs which contain quantifiers take the following forms (an 'a' or an apostrophe is here added to the markings of the original proof lines):

The quantifier free form of Theorem 1. Figure 1 essentially shows the quantifier free form of the proof of Theorem 1 according to the transformation procedure. We assume that the reader is in general familiar with that procedure and we only mention some modifications and specific aspects which relate to the example proof.

(i) According to the transformation procedure the quantified variable names must be replaced by unique names and the existentially quantified variables are replaced by Skolem functions. In the example, the variable *eps* of line [3] is renamed into *eps0*; *delta1* and *delta2* are replaced by the Skolem functions *d1(eps0)* and *d2(eps0)* which depend on *eps0*; *delta* of line [6] is renamed into *delta0* and defined as *min(d1(eps0),d2(eps0))*; the various variables *x* are not renamed here in the example because of readability.

(ii) The equations and inequalities are assigned a corresponding interval of validity. With that we follow the proceeding of Bledsoe et al. [2].

In addition to the quantifier free version, the monitoring program utilizes a table of the occurring objects, i.e. the functions, variables, constants, and their characteristic properties. We do not here mention further details.

```
[2a] |(f(x)+g(x)) –(f(a)+g(a))|< eps                        ; x IN ( a-delta , a+delta ),
[4a] |f(x) - f(a)| < eps0/2                                 ; x IN ( a-d1(eps0) , a+d1(eps0) ) ,
[5a] |g(x) - g(a)| < eps0/2                                 ; x IN ( a-d2(eps0) , a+d2(eps0) ) ,
[7a] |(f(x)+g(x)) –(f(a)+g(a)) |= |(f(x)-f(a))+(g(x)) –g(a)) | ; x IN R
     |(f(x)-f(a))+(g(x)) –g(a)) |<= |f(x)-f(a)|+|g(x) - g(a)|  ; x IN R
[8a] |(f(x)+g(x)) –(f(a)+g(a)) |<= |f(x)-f(a)|+|g(x)-g(a)|   ; x IN ( a-delta0 , a+delta0 )
     |f(x)-f(a)|+|g(x)-g(a)| < eps0/2 +eps0/2               ; x IN ( a-delta0 , a+delta0 )
     eps0/2 +eps0/2 = eps0
```

Figure 1: Quantifier free version of the proof of Theorem 1

The quantifier free form of theorem 2. Figure 2 essentially shows the quantifier free form of the proof of Theorem 2. The quantities x0, x1, and x2 are existentially quantified.

```
[C'] f(x1) >= f(x2) AND x1 < x2
[D'] ( f(x2) - f(x1) ) / ( x2 - x1 ) <= 0
[E'] f'(x0) = ( f(x2) - f(x1) ) / ( x2 - x1 )
     (f(x2) - f(x1) ) / ( x2 - x1 ) > 0
```

Figure 2: Quantifier free version of the proof of Theorem 2

4.2 Checking a proof for correctness and completeness

The monitoring procedure of the user proof consists of checking one line of the proof after the other. The whole procedure of checking a proof falls into several special subprocedures which process the different kinds of proof statements. There are the following subprocedures which generally utilize the quantifier free versions of the original statements to process the original user statement:

PROCdef: checks the correspondence between a concept and its analytical definition
PROCfm: checks formula manipulation operations
PROClogic: checks logical manipulations
PROCassume: checks the different kinds of assume statements
PROCtoprove: checks whether the succeeding statement corresponds to the statement of the theorem
PROCtheorem: checks whether a theorem may be employed in a special situation
PROCcontradiction: checks contradicting statements
PROCqed: checks whether the theorem is in fact proven

We will describe some features of the procedures in the context of checking the example proofs and mcntion some more details which are not immediately related to the examples. It should be obvious that the subprocedures also apply to analogous proof steps of other theorems. With the description, we use the line markings of the original proofs (like [2] or [C]), and we do not additionally mention the corresponding line markings of the quantifier free versions (like [2a] or [C']), although the procedures actually utilize the transformed statements.

Checking Theorem 1.

Line [1] states the proof method as 'DIRECT_BY_DEFINITION'. That information will be used later when the 'QED' statement of line [9] occurs (see below).

Line [2] consists of a 'TO PROVE' statement and mentions the analytical definition of the continuity of the function f(x) + g(x) at the point a and as the foundation the conclusion *[c]* of the theorem. The subprocedure *PROCtoprove* uses the subprocedure *PROCdef* to verify that the user statement and the analytical definition of continuity correspond to each other. To check that statement, *PROCdef* uses an internally provided analytic definition of the concept of continuity. The user statement and the analytical definition are compared in the quantifier free form by a unification process. The user statement is regarded as correct when a unification is possible. *PROCtoprove* utilizes the foundation of the line [2] to establish the connection between the concept of continuity and the user definition. Line [2] is internally marked and used later when the 'QED' statement is processed (see below).

A 'TO PROVE' statement may also appear in a proof e.g. to state a lemma which will be used later in the proof. In that case no foundation would be needed and a connection to the conclusion of the theorem would not be established.

Statements which explicitly state the analytical definition of a concept or vice versa infer the concept from an analytical definition are frequently found in calculus proofs. They are all treated by the subprocedure *PROCdef* in a similar way.

Line [3] mentions the choice of an *eps* > 0. That statement corresponds to a universally quantified variable *ALL eps > 0*. The statement results in an entry into the table of the entities of the proof. No further operation happens.

The lines [4] and [5] reflect the analytical definitions of continuity of the functions f and g, respectively. The foundations {[p2]} and {[p3]} trigger thc comparison with the definitions of the continuity of f and of g, respectively. The subprocedure *PROCdef* establishes the correctness of the user statements as in the case of line [2]. In order to deal with the *eps/2*, in contrast to the usual *eps* without any factor, a generalized version of continuity is used: _SOME M >0 ALL eps>0 SOME delta>0 ALL x WITH |x-a|<delta: |f(x)-f(a)| < M*_ eps. A suitable factor of *eps* in the middle of the proof is often the key with continuity proofs to assure a neat < *eps* without a factor when the proof is finished. The reader will know that.

Line [6] defines the variable *delta* and its value by an expression. The statement results in an entry into the table of the entities of the proof. No further operation happens.

Line [7] gives rise to an equation and an inequality. According to the mentioned foundations, the subroutine *PROCfm* uses a simplification process to check the first equation and a *triangle inequality* subprocedure to check the second relation. Formula manipulation operations play a central role with proofs in the domain of calculus, so corresponding methods need to be available.

Line [8] divides into three relations. The first inequality is an immediate consequence of [7]. *PROCfm* checks their correspondence by standardizing the inequalities and by establishing that the interval mentioned in the line [8] is contained in the interval R of [7].

The second statement resulting from [8] establishes the statements of [4] and of [5] as foundations. *PROCfm* uses *evaluation heuristics* to handle the check of the correctness.

The third relation resulting from [8] only needs simplification which is also done by *PROCfm*.

Line [9] states that the theorem is proven. In the case of a *direct proof* one expects that the conclusion of the theorem will explicitly or implicitly occur as an inference within the proof, usually at the end of the proof. The subprocedure *PROCqed* processes the proof type of the line [1] and uses the preceding 'TO PROVE' statement which was already recognized as equivalent to the statement of the theorem to check whether the relation of the line [2] is fulfilled by the statement of line [8]. Therefore *PROCqed* uses *PROCfm* and a unification process is again employed. *PROCqed* recognizes that the proof is complete.

Checking Theorem 2.
Line [A] states the proof method as 'INDIRECT'. That information will be used later when the 'QED' statement of line [G] occurs (see below).

Line [B] mentions an 'ASSUME' statement which contains a negation of the conclusion of the theorem. The subprocedure *PROCassume* recognizes that one part of the contradiction, i.e. the part referring to the conclusion of the theorem, is established.

An 'ASSUME' statement may also be used to state something which will be proven later. That corresponds to an alternative use of the 'TO PROVE' statement.

The statement of line [C] is an immediate inference of the mentioned foundation [B]. The subprocedure *PROClogic* verifies that the statement of line [C] logically follows from the logical formula NOT [c] .

The statement of line [D] is an immediate consequence of its foundation [C]. *PROCfm* uses *evaluation heuristics* to handle the check for correctness.

Line [E] divides into two relations. The first relation consists of an application of the Mean-Value Theorem. The subprocedure *PROCtheorem* proves the correctness of the line by checking whether the premises of the mentioned theorem are fulfilled. *PROCtheorem* uses an internally provided version of the theorem. The second relation is an immediate consequence of the premise [p3] and checked by *PROCfm*.

Line [F] is founded by the statements of the lines [D] and [E]. The subprocedure *PROCcontradiction* uses *PROCfm* to check the contradiction.

Line [G] states that the proof is complete. In the case of an indirect proof one expects that a contradiction occurs and that one part of the contradiction is an inference of the negated conclusion of the theorem and the other part is a valid statement which was inferred. *PROCqed* processes the proof type of the line [A] and uses the preceding 'ASSUME' and 'CONTRADICTION' statements to verify that the proof is complete.

Error handling. In a *positive* case, a user proof can be recognized as correct and complete, that means that the occurring statements can be inferred using the corresponding foundations and that the sequence of statements actually proves the conclusion of the theorem. In a *negative* case, several types of light or severe errors may occur. From the perspective of a monitoring system which checks the various proof lines there may happen three cases in connection with each proof line:

(i) *The correct case*: The monitoring program can recognize that a statement can be inferred by using the given foundations. That positive case includes the possibility that a minor error occurred which can be clarified by a dialogue between the system and the user. The list of minor errors includes syntactical errors (e.g. regarding the language or any mathematical formula) or a lacking foundation which can be completed by the system. The completion may be possible e.g. in the case that the foundation of an obvious formula manipulation operation is missing or a reference to a preceding proof line is missing.

(ii) *The error case*: The monitoring program detects e.g. a logical error, an incorrect formula manipulation transformation, an unallowed application of a theorem, a premature 'QED' statement or no 'QED' statement. In that case the system can supply a hint to the user and the user gets the opportunity to correct the error.

The feedback in the case of multiple errors in a single statement depends on the way in which the errors are interconnected. Generally, the error possibilities are multifarious. Some multiple errors can be handled one after the other, e.g. when there are two errors in a formula. The hint that the formula is not correct may make the user rectify one error, so that only one is left.

Let us consider another example: A user enters the wrong name of the theorem which he applies and the application of the theorem is also wrong. The system would try to apply the mentioned theorem and two outcomes are possible: (a) The theorem cannot be applied or (b) the theorem can be applied. In the case of (a), a hint that the theorem is not applicable could help the user to recognize that he entered a wrong theorem name. In the case of (b), the system would state the conclusion of the theorem application. The user might then also recognize that the theorem name is wrong. In those cases the double error is reduced to one error.

(iii) *The unclear case*: The monitoring program cannot decide the correctness of a proof line. Various reasons may be responsible for that. One reason is that an important foundation is missing, e.g. a reference to the theorem which was used , so that the monitoring program cannot infer the user statement. Other reasons refer to the performance of the mentioned subprocedures: They may not be able to verify a correct statement or falsify a wrong statements in certain situations. Such a case suggests to expand the monitoring program.

5 Applications and Extensions and Pragmatics

The above described approach may be utilized for different purposes by different groups in educational institutions. Students have the opportunity to occupy themselves with mathematical proofs and do exercises which may be immediately checked for correctness and completeness.

On the other hand virtual or face-to-face universities may employ such methods in automatic on-line test systems. Proofs delivered by students could be automatically checked and scored. While students construct a proof the system might give hints in the case that foundations are missing, that there are syntactical errors, that the sequence of inferences is not complete, that a statement is just wrong, or that the student is lacking an idea how to prove the theorem. Dependent upon the amount of hints or help provided the software system might decrease the score gained.

The language as it was described above does not contain a set of symbols which are frequently used in theorems and proofs, as e.g the notation for limits, sequences, sums, integrals, or the faculty function. To integrate them one may use the notations of MathML [9]. A closer look at the proofs which are found in the text books of calculus suggests that a large set of the proofs can be worded using the above outlined language when one assumes that the usual mathematical symbols are available and some more extensions are done.

The described approach of verifying proofs demands an internal knowledge base of the concepts and theorems of calculus when proof statements are founded by concept definitions or theorem applications. Such a collection will sensibly use XML as a representation language (see e.g. [6]). See an XML representation of a theorem and of a proof on the website [7]. By utilizing that knowledge base an extensive learning environment which deals with mathematical proofs may be developed. Some aspects related to getting support with finding and constructing proofs are: One may retrieve theorems having the premises which may be used with the proof. One may retrieve a list of proof ideas of the domain and discover the one which may be useful in the current context. The roughly outlined approach to a learning environment stresses the personal proof finding and proof construction activity. A different approach to a learning environment in the field of mathematical proving relies on a general, interactive theorem prover [4].

It is obvious that one has to get used to entering a proof in the restricted natural language. An adequate interface may help to reduce the cognitive overload. Another option is to further develop the language, so that the proofs may be entered in a less restricted way and look more like textbook proofs. Such proofs might then be transformed into the restricted natural language. It is clear that the students would use such a verifying system only when the advantages outweigh the disadvantages. Some advantages are the confirmation of correctness and completeness or the detection of errors and the option of getting help.

6 Conclusions

A restricted natural language to formulate mathematical proofs in the domain of calculus was presented. It was demonstrated how mathematical proofs worded in that language can be transformed into an internal representation and checked for correctness and completeness. Some educational applications were mentioned. The extension to a learning environment was roughly outlined.

Our current prototype of verifying proofs includes an interface to enter natural language proofs, some procedures of theorem proving and an own formula manipulation system. The prototype will be further developed with respect to the methods and the knowledge bases.

References

[1] Bibel, W., "Automated Theorem Proving", Vieweg, Braunschweig, Germany, (1987).

[2] Bledsoe, W.W., Boyer, R.S., Henneman, W.H., "Computer proofs of limit theorems", Artificial Intelligence 3, pp. 27-60, (1972).

[3] Bledsoe, W.W., "Non-resolution Theorem proving", Artificial Intelligence 9, pp. 1-35, (1977).

[4] Buchberger, B., Jebelean, T., Vasaru, D., "Theorema: A System for Formal Scientific Training in Natural Language Presentation", Proceedings of the Tenth World Conference on Educational Multimedia and Hypermedia (ED-MEDIA/ ED-TELECOM 98), Freiburg, Germany, June 20-25, pp. 174-179, (1998).

[5] Chang, C.-L., Lee, R.C., "Symbolic Logic and Mechanical Theorem Proving", Academic Press, NY, (1973).

[6] Goldfarb, C.F., Prescod, P., "The XML Handbook", Prentice Hall PTR, Upper Saddle River,NJ, USA, (1998).

[7] XML representation of a theorem and of a proof, http://www.cs.uni-bonn.de/~peter/ICCE2000example.html

[8] Smith, K.T., "Primer of Modern Analysis", Springer Verlag, New York, (1983).

[9] Mathematical Markup Language (MathML) 1.01 Specification, http://www.w3.org/TR/REC-MathML.

Multimedia Intelligent Tutoring System for Context-Free Grammar

Rhodora L. Reyes, Carlo Paolo T. Galvey, Marie Christine D. Gocolay,
Eden M. Ordona and Conrado del Rosario Ruiz
Software Technology Department, College of Computer Studies
De La Sane University, Manila
2401Taft Ave., Manila
Tel: (632) 5244402, (632) 5364278
Email: ccsrlr@dlsu.edu.ph

CFG-MINTS is a multimedia intelligent tutoring system that teaches context-free grammar. The tutor model of his ITS is composed of a set of teaching strategies and an algorithm that determines which teaching action to be deployed given the goals of the system and the current state of the student model. The student model uses the Constraint-Based Modeling (CBM) approach in diagnosing the learner. CBM reduces the complexity of student modeling by focusing on the difference of the student' s solution to the ideal solution only and the analysis is reduced to pattern matching. The assumption here is that there can be no correct solution of a problem that traverses a problem state, which violates the fundamental ideas, or concepts of the domain. The system also includes features for simulating the created context-free grammar to aid in teaching.

*The paper was not available by the date of printing.

Natural Language-like Knowledge Representation for Multimedia Educational Systems

Eva Heinrich and Ray Kemp
Institute of Information Sciences and Technology
Massey University, Palmerston North, New Zealand
E-mail: E.Heinrich@massey.ac.nz R.Kemp@massey.ac.nz

The appropriate use of multimedia is becoming increasingly important in computer teaching systems. Not only are students stimulated by being presented with information in a variety of forms, but such an approach also more closely resembles the real world where they have to assimilate what they see and hear, abstracting out what is relevant. With the diversity and amount of multimedia material that may be present in these systems, a powerful form of knowledge representation is required to support navigation and knowledge retrieval. The (human or computer) tutor may wish to refer to document segments, to recap important points, provide feedback, give hints and so on. The student also may wish to refer to items previously seen or heard. The Flexible Structured Coding Language, FSCL, is a natural language-like, formalised description language which allows the formulation of rich yet structured sentences. These sentences are attached to segments of multimedia documents. FSCL provides an easily accessible approach for knowledge representation, precise and rich description of complex contents, correct and complete retrieval within the descriptions, and retrieval across data of different media types. FSCL can be extended to integrate ontologies, inference of knowledge and freeform querying performed by the learner.

Keywords: **Multimedia, knowledge representation**

1 Introduction

Computer-based educational systems have developed from standalone applications, using mainly text and graphics, which focused on teaching a restricted set of subjects or skills. Today's multimedia systems are often distributed across the web using a client-server approach and aim to integrate teaching material from multiple subjects areas. These systems collect feedback on the progress of the learner and attempt to provide material at the appropriate levels. An example of such a system is GENTLE [5].

Beside the technical challenges of managing such a system, a number of conceptual ones arise. One of these is knowledge representation and the related issue of knowledge retrieval. One problem with supplying a learner with a flexible learning environment is the need to provide a mechanism for locating appropriate information. This is a non-trivial task considering the vast amount of diverse material stored and the complexity of the concepts incorporated into the learning material. Another requirement is to give the learner a mechanism for questioning the system. This can be for retrieving specific material or for asking conceptual questions concerning the subject area.

To illustrate some of the requirements for a computer-based educational system, consider a small scenario. Imagine a web-based teaching module on the use of machinery. This module could consist of a number of multimedia documents: for example, a video showing an instructor demonstrating the use of the machinery, a set of images displaying various technical features of the machinery or a set of text documents explaining various procedures. These multimedia documents, annotated with appropriate knowledge representation mechanisms and generic domain knowledge, have to be stored. Based on this information a range of material could be retrieved: a segment of the video document showing the instructor demonstrating a specific task; additional information from images or text documents relating exactly to this task; the status of the machinery at a specific position in the video inferred from the domain rules.

After a brief overview on current approaches to knowledge representation in computer-based educational systems, we consider how the Flexible Structured Coding Language, FSCL [9,11], may be applied to this problem. We will first describe FSCL in the form it is used in its original context of studies of human behaviour and then discuss the advantages of using FSCL in computer-based educational systems. We then suggest some modifications to FSCL to provide extended support for computer-based educational systems and conclude the paper by summarising the contributions this natural language-like approach to knowledge representation can give us.

2 Current approaches to knowledge representation

To access the appropriate information in a computer-based educational system, a knowledge representation scheme is necessary. This provides a meta-level description of the contents of the educational system. In this paper, we consider the format of this meta-level description, not its technical realization in a database or file system. Before we describe some common approaches to meta-level description, we want to briefly discuss why a meta-level description is necessary and why it is not possible to extract the information directly from the learning material.

The retrieval of information from documents directly has limited scope both on a technical and on a conceptual level. Technically, searching through text based documents is easy and allows for identification of keywords, phrases or sentences. Achieving the same level of retrieval for video documents is much harder. Techniques exist to automatically parse video documents to detect scene changes [8, 23] and objects [6, 17]. However, a number of problems still have to be overcome to provide sufficient access to video content [13].

Setting the technical difficulties in accessing video or audio documents aside, there are still conceptual considerations which will demand some meta-level description of content. Retrieving appropriate information from a collection of documents will, in many cases, require access to the semantics of these documents. Searching through these documents on a keyword (or object) basis is unlikely to produce satisfactory results [2]. The transition of factual ('she was smiling', a smiling face, a sunny picture) to conceptual (happiness, pleasant atmosphere) information has to be made to access the semantics of a document. This is not possible without some meta information or description of these documents.

A number of approaches are used to facilitate the access to the semantics of documents in preparation for information retrieval. Ontologies provide a modelling scheme for a specific domain creating a shared vocabulary for the description of contents [4]. Topic maps [22] create organising principles for information by defining topics, the associations of topics and the occurrence of topics in documents. Conceptual graphs [20,21] capture knowledge about a specific domain and make this knowledge accessible to deduction using first order logic.

In the analysis of data in the social sciences, a *description* approach is common. Codes or annotations, called descriptions, are attached to specific locations of multimedia documents to assist retrieval. These can contain any kind of factual or semantic descriptions of the documents' contents. Domain specific codes or freeform textual annotations are common in analysis programs like The Observer [16], Nudist [18] or its successor, NVivo [19]. All the approaches mentioned above have been proposed to overcome the technical and conceptual difficulties of accessing the information contained in multimedia documents and to facilitate the retrieval of appropriate information. In this paper, we propose the use of FSCL as a meta-level description mechanism. In the next section we introduce the main features of FSCL. We follow this by a discussion of its advantages for knowledge representation and retrieval, and indicate how FSCL can be combined with ontologies and conceptual graphs.

3 Knowledge representation using FSCL

FSCL is a natural language-like description language. It aims to combine the expressiveness and flexibility of natural language with the rigour of formalised approaches. The main components of FSCL are its *vocabulary*, *grammar* and *categories*. The vocabulary can be freely defined by the author of the teaching material. Any word can be used and the vocabulary can be extended at any point of time. Whereas the vocabulary is likely to be defined for a specific domain, the grammar is generic. It is designed to formulate 'subject - verb - object' and 'concept - object' sentences and combinations of these elements, including conjunctions, prepositions, adjectives and adverbs. The role of the categories is to bridge the vocabulary and the generic grammar. The grammar is defined on the categories. Each word of the vocabulary has to belong to exactly one category. This construct allows for the structure of the description language to stay the same across applications in different domains. The categories of FSCL have been defined in accordance with the word classes of the English language. The categories are: Person/Thing, Activity, Concept, Conjunction, Preposition and Descriptor (which combines the word classes adjective and adverb).

FSCL has been incorporated into an information system to support the analysis of multimedia documents, called PAC [12]. Sentences formulated with FSCL can be, in a system like PAC, attached to a segment of a multimedia document. The sentences, together with document identifiers and segment specifications are stored in a database and later used for retrieval. Because the structure of the FSCL sentences is well known, it is possible to access the semantics of the information stored. The retrieval of information from FSCL descriptions is achieved using the Flexible Structured Query Language, FSQL [9].

FSQL provides three layers for querying: the first layer is based on the properties of FSCL and allows the correct and complete retrieval of information from the description sentences; the second layer provides for Boolean combinations within sets of description sentences; the third layer accesses the properties of the multimedia document segments attached to the FSCL sentences and facilitates time and position comparisons. More detailed information on FSCL and FSQL can be found in [9]. Specific information about information retrieval across multiple media formats is given in [10].

4 Advantages of using FSCL

The most convenient and expressive language available to us is natural language. Yet looking at knowledge retrieval with computer systems, natural language poses a range of well known and not yet fully solved problems. The main problem lies in the vast amount of implicit knowledge necessary to see words in the right context and to fully understand a sentence [21]. Various large scale projects are underway to attack these problems, like WordNet [15], an ontology for natural language processing, and the Cyc system [14], attempting to construct a 'complete' ontology of the world. Our approach is far less ambitious. We acknowledge that using full natural language for knowledge representation and retrieval would be highly desirable. Yet with the enormous difficulties associated with this approach we were looking for a much simpler solution. FSCL provides us with a number of advantages:

- We have a natural language-like notation. Any FSCL sentence can immediately be understood by a human reader. The importance of this is confirmed in the discussion of the five principles of knowledge representation by Davis *et al* [3].

- We have a language and can deduce the structure of our sentences. We have therefore more power than with the keyword approach commonly used in information retrieval, which suffers from low precision and low recall [21].

- We can build a powerful vocabulary by integrating the FSCL categories with ontologies.

- Of special interest to computer-based educational systems is that we can link our form of knowledge representation with multimedia documents.

FSCL has been successfully used to support the study of behaviour recorded in multimedia documents. It has given analysts the possibility to create rich descriptions of behaviour and to analyse the descriptions in a precise way [9]. We want to keep the main features of FSCL in formulating natural language-like, structured and flexible sentences attached to multimedia documents. Further, we want to adapt FSCL for a more general use in knowledge representation and retrieval. Our ideas in this direction are presented in the next sections of this paper.

5 Proposed extensions

We want to indicate several areas of possible changes and extensions to FSCL: changes to its categories and grammar forms; extensions to include ontologies; conversion of FSCL sentences to conceptual graphs to facilitate inferencing; and the introduction of freeform querying.

5.1 Changes to categories and grammar of FSCL

As described in section 3, the FSCL categories and grammar have been designed to formulate sentences of the forms 'subject - verb - object' and 'concept - object' in the context of studies of behaviour. To simplify the construction of the vocabulary, adjectives and adverbs have been combined in the FSCL category 'Descriptor' [9]. Adhering to the general FSCL principle of having a formal grammar on fixed, defined categories we are currently investigating a number of changes to FSCL to adapt it to a more general use in knowledge representation. The exact format of the changes has to be determined through applying FSCL in a range of web-based educational systems. Our current thinking centres around the following topics:

- We are investigating changes to the FSCL categories. Merging the categories Person/Thing and Concepts to a more general category, Noun, would address the potential conflict between abstract and concrete terms (see the discussion about the abstract term 'students' and the specific individuals in section 5.3). The category 'Descriptor' could be split up into separate categories of 'Adjectives' and 'Adverbs'. The grammar of FSCL had to change accordingly to accommodate the different roles of adjectives and adverbs within a sentence. The advantage over the current approach in FSCL would be that with this changc advcrbs could be positioned correctly as in natural language English sentences.

- In natural language, words occur in different grammatical forms in different roles in a sentence ('the instrúctor *starts* the motor'; 'the motor is *started*'). The current FSCL has a strict separation between its categories. While a word can be defined in its derivations in multiple categories (Activity: starts; Descriptor: started), it is not possible to create a semantic link between the different word forms. We are looking at introducing such a link together with a meta-level grammar to be able to detect semantic equivalence between sentences with word derivatives in different parts of speech.

- The grammar of FSCL could be extended to recognise a wider range of sentence structures. Clausal variations like imperatives ('Start the motor!') or questions ('Is the motor running?') can be introduced. Conditional sentences of the form 'if C then S' would support inference as outlined in more detail in the following section. A wider range of sentence structures recognised correctly by FSCL would increase the potential for knowledge retrieval and inference.

5.2 Extension to use ontologies

FSCL uses hierarchies to define the words of the vocabulary. These hierarchies are defined within the FSCL categories. They are used to group related words and to allow for a retrieval of information on different levels of granularity. These hierarchies, as they are currently used in FSCL, can be seen as simple forms of ontologies. While a number of issues have to be addressed to base FSCL on more substantial ontologies, none of these seems to pose a real problem.

- Users of FSCL define the vocabulary they need for their particular domain. The experience, so far, as reported in [9], show that users define their vocabulary as multiple hierarchies within each FSCL category. These hierarchies could be joined under the FSCL category name to build one ontology within each FSCL category.

- An ontology typically moves from the abstract to the concrete, from concepts to instances. The vocabulary in FSCL is organised in the same way. In a study on 'learning to read', e.g., individual students' names were grouped under the term 'students', individual teachers' names under the term 'teachers' [9]. A term like 'students' contains two components: it has an abstract component in describing a group of the population in general with the property of 'attending school to learn'; it has a concrete component in grouping together specific, named individuals. In the current uses of FSCL this distinction has not caused any problems.

- Not all FSCL categories contain vocabulary which necessarily should be structured as ontologies. While it can be of advantage to organisc thc vocabulary in thc FSCL catcgorics 'Conjunction' and 'Prcposition' in hierarchies these words will not build ontologies as they not define 'categories of the world'. Yet the coexistence of ontologies and hierarchies in the vocabulary of FSCL should not create a difficulty.

5.3 Conceptual graphs and inference

FSCL is an easy to understand and effective scheme for an author to create their own vocabulary and use it together with the grammar for describing the contents of a multimedia document such as a video. Currently, knowledge retrieval is performed using the complementary query language FSQL. FSQL addresses the grammatical structure of FSCL sentences, takes advantage of the hierarchy information built into the vocabulary, and offers Boolean, time and sequence query options. However, there is no deductive feature in this scheme which would allow us to be able to infer facts or relations that are not explicitly stated. For example, given the statements:

> If anyone starts the motor then the motor is running
> The instructor starts the motor

which describes the situation in a training video then we may wish to be able to answer the question:

> Is the motor running?

To be able to function at this level, we need the power of a first order logic system. Conceptual Graphs, CG, [20] give us this power.

Our proposal is that the user should describe their domain in terms of FSCL. The statements in this language can then be automatically translated into a CG format. This process is quite straight forward since FSCL is unambiguous, allowing many of the problems of natural language translation to be circumvented.

When a query is made, or some information needs to be located within the document segment then an initial attempt can be made to do this by using FSQL. If this fails then the deductive power of the CG representation is invoked. Standard theorem proving techniques within CG would enable us to check the veracity of a statement. As a bonus, we would get a step-by-step justification of the result proved, similar to the explanation given in expert systems.

5.4 Freeform Querying

Based on a limited yet flexible vocabulary and on a limited grammar, as offered by FSCL and FSQL, a query system can be developed which allows the user to pose questions to the educational system. As the structure and the vocabulary of these questions is known, the educational system can 'understand' these questions. Questions can be mapped against a repository of previously asked questions. If a semantically equivalent question is stored, the corresponding answer is retrieved and presented to the user. If a semantically close question is stored, this previously asked question can be used to facilitate the answering of the new question. As questions and knowledge representation are constructed by the same underlying mechanisms a mapping from question to knowledge representation is possible. This can be used to assist the answering of questions based on the knowledge descriptions and to find the appropriate segments of the multimedia teaching material.

The approach presented here does not attempt to answer any natural language question but a restricted set. The vocabulary is restricted to allow the construction of meaningful questions in a particular domain. The grammar is restricted to allow the construction and comparison of meaningful questions based on the vocabulary. The grammar is generic as it is based on categories which are used to organise the vocabulary across domains. The restriction of vocabulary and grammar distinguishes this approach from the AskJeeves [1] search mechanism. The existence of a grammar distinguishes this approach from keyword based search mechanisms as used in library systems or by internet search machines.

The general idea is to provide the user with specific answers to questions. These answers are retrieved from a body of stored answers only if semantic equivalence can be guaranteed. If semantic closeness is detected the relevant questions with their answers are given to a human operator who then decides on the suitability of the match.

6 Conclusions

In this paper we have considered the need for a knowledge representation mechanism for computer-based educational systems. We have first indicated a number of commonly used mechanisms and have then discussed the Flexible Structured Coding Language, FSCL. We have suggested that FSCL provides an effective mechanism for knowledge representation and subsequent knowledge retrieval, based on the nature of FSCL as a natural language-like description language which allows for flexible, rich yet structured description of learning concepts. As extensions to FSCL we have suggested the integration of more substantial ontologies, the conversion of FSCL sentences into conceptual graph structures and the introduction of freeform querying.

References

[1] AskJeeves, http://www.aj.com/index.asp (accessed on February 2, 2000).

[2] Davenport, G., "Indexes Are 'Out', Models Are 'In'", IEEE Multimedia, Fall 1996, Vol. 3, No.3, pp.10-15, (1996).

[3] Davis, R., Schrobe, H. and Szolovits, P., "What is knowledge representation?" AI Magazine, 14:1, pp. 17-33, (1993).

[4] Decker, S., Erdmann, M., Fensel, D. and Studer, R., "ONTOBROKER: Ontology Based Acess to Distributed and Semi-structured Information", Proceedings of Eighth Working Conference on Database Semantics, Semantic Issues in Multimedia Systems, R. Meersman, Z. Tari and S. Stevens (Eds.), Kluwer Academic Publishers, Boston, (1999).

[5] GENTLE, www.gentle-wbt.com (accessed on April 10 2000).

[6] Gong, Y., Zhang, H. Chuan, H. C., Sakauchi, M., "An Image Database System with Content Capturing and Fast Image Indexing Abilities", Proceedings of the International Conference on Multimedia Computing and Systems, May 1994, IEEE Computer Society Press, Los Alamitos, California, (1994).

[7] Grossmann, R., "The Existence of the World, An Introduction to Ontology", London: Routledge, (1992).

[8] Hampapur, A., Jain, R., Weymouth, T. E., "Production Model Based Digital Video Segmentation", Journal of Multimedia Tools and Applications, 1, 9-46, Kluwer Academic Publishers, Boston, (1995).

[9] Heinrich, E. A., "Multimedia Information System for the Support of Studies of Behaviour", Doctoral Dissertation, Massey University, Palmerston North, New Zealand, (1999).

[10] Heinrich, E. and Kemp, E., "Description and Retrieval across Multiple Media Formats", 4th International Workshop on Query Processing and Multimedia Issues in Distributed Systems, DEXA'00, IEEE Computer Society Press (2000).

[11] Heinrich, E., Kemp, E. and Patrick, J.D., "A Natural Language Like Description Language", 10th Australasian Conference on Information Systems (ACIS), B. Hope and P. Yoong (Eds.), School of Communications and Information Management, Victoria University of Wellington, Wellington, New Zealand, pp. 375 – 386, (1999).

[12] Heinrich, E., Patrick, J.D., and Kemp, E., "A Multimedia Information System to Support the Analysis of Human Behaviour Recorded on Video", Computers in Psychology, Conference Proceedings, CTI Centre for Psychology, University of York, York, (1998).

[13] Jiang, H., Montesi, D., Elmagarmid, A. K., "Integrated Video and Text for Content-based Access to Video Databases", Journal of Multimedia Tools and Applications, 9, pp. 227-249, Kluwer Academic Publishers, (1999).

[14] Lenat, D.B. and Guha, R.V., "Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project", Addison-Wesley Publishing Company, Reading, Massachusetts, (1990).

[15] Miller, G.A., "WordNet: a lexical database for English", Communications of the ACM, 38:11, pp. 39-41, (1995).

[16] Noldus Information Technology, "The Observer Video-Pro, Version 4.0 for Windows". Wageningen, The Netherlands, (1998).

[17] Palhang, M., Sowmya, A., "Learning Object Models from Real Images", Proceedings of the First International Conference on Visual Information Systems, Victoria University of Technology, Melbourne, Australia, (1996).

[18] QSR Software, QSR NUDIST 4 Overview, http://www.qsr.com.au/Software/N4/qsrn4.htm (accessed on June 22 1998).

[19] Richards, L., "Using NVivo in Qualitative Research", Melbourne: Qualitative Solutions and Research Pty, (1999).

[20] Sowa, J.F., "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley, Reading, Massachusetts, (1984).

[21] Sowa, J.F., Knowledge Representation; "Logical, philosophical, and Computational Foundations", Brooks/Cole, Pacific Grove, CA, (2000).

[22] Topic maps. ISO/IEC JTC1/SC34 - Document Description and Processing Languages. http://www.ornl.gov/sgml/sc34/document/0058.htm, (accessed on February 8 2000).

[23] Zhang, H., Gong, Y., Smoliar, S. W. and Tan, S.Y., "Automatic Parsing of News Video", Proceedings of the International Conference on Multimedia Computing Systems, May 1994, IEEE Computer Society Press, Los Alamitos, California, (1994).

Strange Creatures in Virtual Inhabited 3D Worlds

Jens F. Jensen
InterMedia Aalborg, Aalborg University
Niels Jerners Vej, NOVI4-215, DK-9220 Aalborg East, DENMARK
Phone: +4598156279 Fax: +4598142444
E-mail: jensf@hum.auc.dk

This paper discusses the strange creatures that currently populate 3D cyberspace and3D Internet. First, the concept of Virtual Inhabited 3D worlds are discussed and defined. Next, some of the key elements or basic entities that can be found within the horizon of Virtual, Inhabited 3D Worlds are identified and denned. Among these basic elements are objects and agents, differentiated by whether or not their primary function is to carry out an action. Agents (denned as entities, which primary function is to carry out actions) have two main forms, which have been described as relatively sharply differentiable polar opposites. This is done based on questions such as: who is controlling the agents? ' who is doing the driving?' On the one hand there are agents that react independently of the user, but which are controlled by software or AL, the so-called ' autonomous agents' or ' bots' . On the other hand, there are agents, which directly represent and are controlled by users, the so-called ' avatars. Although there is then, in principle, a differentiation, in terms of definition, between bots and avatars, the paper argues that both concepts cover a relatively wide spectrum of very different types of phenomena with differing degrees of control. There also seems to be a tendency toward the appearance of more and more hybrids- in the present context termed ' cyber-hybrids' - combining avatars and bots. Furthermore, these hybrid forms are in many ways the most interesting and most promising in the virtual worlds at the moment. Rather than considering avatars and bots as polar opposites, it may therefore be more productive to consider them as the outer points along a continuum, between which can be found all sorts of combinations or hybrids. Following this line of argument, the paper outlines a new typology of hybrid creatures, which currently populate the continuum between (objects) bots and avatars in Virtual worlds.

*The paper was not available by the date of printing.

The Application of Uncertainty Reasoning for an Intelligent Tutoring System

Hown-Wen Chen*, Teran Wu and Dong-Yih Bau
Institute of Information Management, Da-Yeh University
112, Shan-Jeau Rd., Da-Tsuen, Chang-Hwa, Taiwan
Email :hwchen@aries.dyu.edu.tw

The activity of test and evaluation is an important part of Computer-Assisted Instruction systems. In most systems, 「absolutely learned」 and 「absolutely unfamiliar」are often used to represent the status of a student in learning a novice concept. However, for each target concept, there are usually more than one related sub-concepts with different degrees of importance. Thus, it is quite difficult to instruct each individual student effectively according to his learning status in those conventional systems. A hybrid technology of fuzzy theory and uncertainty reasoning are thus used in the research. The proposed intelligent tutoring system was designed to illustrate: 1.automatically tracking and analyzing the current learning status of a pupil, especially detecting the formation of learning barriers or misconceptions; 2. autonomously leading pupils to visit assisted learning path and thus proposing tutorials to make the learning of students more effectively. 3.linguistically explaining the implicit behavior of a pupil during the whole learning process. In addition, the mathematical course of teaching Pythagoream Theorem was used as the content of our test-bed. A simulation by hand and positive feedbacks from teachers of junior high schools illustrate the reasonableness and applicability of the proposed tutoring system.

Keywords: **Pythagoream Theorem, Fuzzy Logic, Uncertainty Reasoning, Intelligent Tutoring System**

1 Introduction

Researches about Intelligent Computer Aided Instruction (ICAI) have incrementally grown since 1970, for example, standard intelligent tutoring systems [1], or participants in virtual environments [2], or a virtual instructor in a training environment [3]. However, as known, the effectiveness of education would depend on the local culture. But, there are few intelligent tutoring systems focusing on Taiwanese students have been reported. CORAL [4] was designed as an interface system, without any artificial intelligence module of teachers' expertise, to provide a long-distance collative learning environment of virtual learning. As discussed in lots of tutoring systems, the most challenging issue is how to evaluate and diagnose the learning of students. Tests are a typical and popular method of evaluation. Taking the GRE as an example, people have taken the test through computers since 1992. The IBM co. and Arthur Anderson Co. have begun to work on the development of a computerized testing system. Such systems, which change the form of tests from conventional paper-to-pencil to on-line, are proliferating rapidly. For ICAI, it becomes more popular that thc cvaluation of pupils' learning should not be simply classified as「absolutely learned」and「absolutely unfamiliar」. In addition, ways of leading each individual pupil to enjoy an efficient learning experience is also pursued. In the research, we proposed an intelligent tutoring system which can afford the most appropriate tutorials to each pupil according to his learning status and thus can prevent pupils to trap into a misconception too long.

2 The Organization of Tutorials and Maintaining Principles

Before implementing our tutoring system, some special issues and adopted techniques must be introduced.

Those topics include the organization of tutorials, a way of representing pupil's learning status, and the detection of any formed misconception.

2.1 The Construction of a Hierarchical Concept Tree

In general, tutorials would be organized as a tree hierarchy of curriculum in the order of chapter, section, sub-section, paragraph, etc. Since learning a complicate concept must depend on the success of learning all its related sub-concepts, the kind of structure cannot be claimed to be suitable for both learners and instructors. That is, too few containment or precedence information about curriculum is available. Thus, learning concepts and related tutorials are re-arranged as a hierarchical conceptual tree of containment here. According to literatures [5] and interviews with teachers of junior high schools, the concepts related to learning Pythagoream Theorem for native pupils can be analyzed and constructed as Figure 1. In the tree, the learning of any parent conceptual node must follow after at least one of its children nodes.

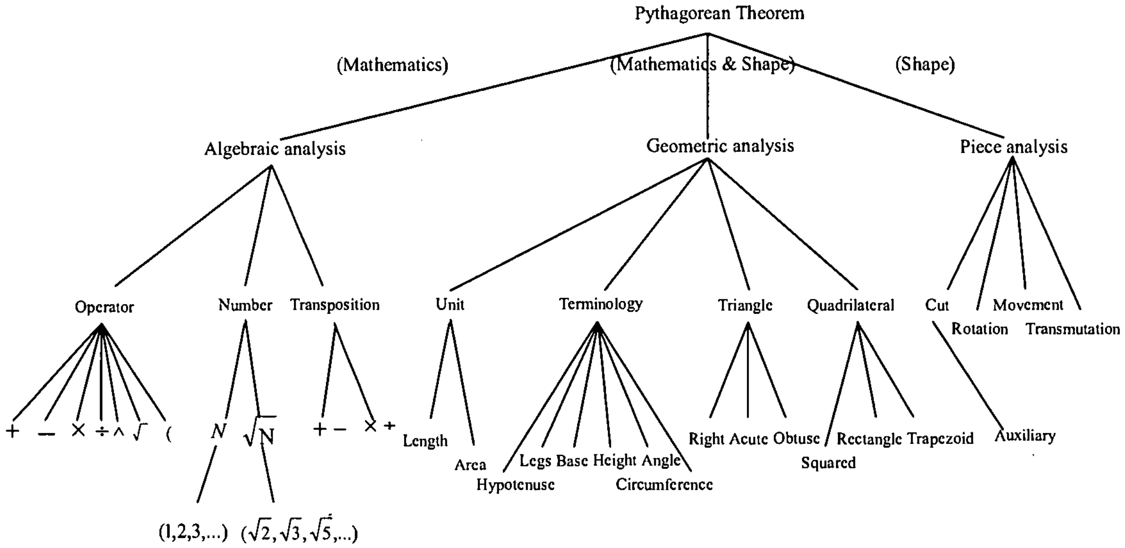


Figure 1. A hierarchical concept tree of Pythagorean Theorem

2.2 The Setting of Node Weights within the Hierarchical Concept Tree

To express the corresponding degree of importance, an integer is assigned to each testing question related to individual concept [5]. However, it is still a heavy burden even for an expert to quantitatively assess the extent. Besides, the estimated grade of importance is too subjective in general. In our system, the influence of each node on learning its parent node is defined through fuzzy theory as follows:

Step 1: Some teachers in junior high schools are asked to evaluate the relevance of nodes related to their parent node in the hierarchical concept tree.

Step 2: Fuzzy theory is included to quantify teachers' opinions in the designed questionnaire obtained in step 1. Five possible values for linguistic variables are used. Note magnitudes 0.0 and 1.0 are not adopted in the memberships because of product operations and symmetry.

Step 3: Murray's or Ishihawa's Max-Min method is used to fuzzily integrate those multiple expertise. After that, a defuzzification process to evaluate the mass centroid of fuzzy numbers is applied. The weights of nodes within the hierarchical concept tree are thus settled as shown in Figure 1.

2.3 The Maintenance of Belief Parameters

To increasing the expression power of the proposed system above "absolutely known" and "absolutely unfamiliar", a belief parameter m and another updating parameter λ described in Dempster-Shafer Theorem [6] are applied here to assess the familiarity degree of a pupil to a particular learning concept within the hierarchical concept tree. To describe the meaning of the updating parameters λ and θ (θ=1-λ), two cases must be taken in account:

Case 1: Making a correct answer

λ and θ can be used to denote the belief degree of promoting to a higher level and of staying on the same level within the conceptual hierarchy, respectively.

Case 2: Making a wrong answer

λ and θ can be used to denote the belief degree of degrading to a lower level and of staying on the same level within the conceptual hierarchy, respectively.

As to defining the updating rules of the belief parameter m, a general sub-tree structure is considered. In the tree, a node f has three children nodes labeled as a, b, c, and the interconnection links are labeled as W_{af}, W_{bf}, W_{cf}.

Case 1: *Making a correct answer in the test for the conceptual node a*

A promotion within the conceptual hierarchy must be activated. The belief parameters of the two relevant nodes a and f are thus modified as

$$m_f^t = (W_{af} \bullet \lambda) + m_f^{t-1}$$
$$m_a^t = (W_{af} \bullet \theta) + m_a^{t-1} \qquad \textbf{(eqn. 1)}$$

m_f^t, m_a^t : the magnitudes of belief after promotion

m_f^{t-1}, m_a^{t-1} : the magnitudes of belief before promotion

W_{af} : the weight of link between nodes **a** and **f**

Case 2: Making a wrong answer in the test for the conceptual node **f**

A degradation within the conceptual hierarchy must be activated. The belief parameters of the four relevant nodes, **f** and its children nodes **a, b, c,** are thus modified as

$$m_f^t = (\theta) + m_f^{t-1}$$
$$m_a^t = [\ W_{af} \bullet \lambda \bullet (1 - m_a^{t-1})\ ] + m_a^{t-1}$$
$$m_b^t = [\ W_{bf} \bullet \lambda \bullet (1 - m_b^{t-1})\ ] + m_b^{t-1} \qquad \textbf{(eqn. 2)}$$
$$m_c^t = [\ W_{cf} \bullet \lambda \bullet (1 - m_c^{t-1})\ ] + m_c^{t-1}$$

Case 3: If a correct answer is made in the topmost conceptual node, it is impossible to promote anymore. However, the belief of the topmost conceptual node is still updated with eqn. 1.

Case 4: If a wrong answer is made in the lowest conceptual node, it is impossible to degrade and the belief of the node is updated with eqn. 2.

2.4 The Strategy of Instruction

Several principles have been applied in the proposed system:

The instruction and assessment examination would only take place in the conceptual node with the largest belief. However, all assessment tests for its children nodes with weights larger than a pre-chosen threshold must be answered correctly. If the mentioned condition is not satisfied, the focus of instruction and assessment would be transferred to one of its children nodes instead.

According to Dempster-Shapfer Theorem, the procedure of normalization must be applied after each updating of belief.

There is an implicit relationship between the magnitudes of weights and belief parameter λ. To avoid the learning process to be not in progress, according to eqn. 2, the magnitude of belief updating in any child node (a) must be larger than that of parent node (f). Thus,

$$W \bullet \lambda \bullet (1) > 1 - \lambda$$
$$\Rightarrow \lambda > \frac{1}{(1+w)} \text{ for all possible w}$$

2.5 The Analysis of Learning Traces and Detection of Misconceptions

Two kinds of traversal information would be recorded during the learning process: the weighted correct rate of answering testing questions for each conceptual node, and the traversal path of all visiting nodes.

First, the weighted correct rate can be used to indicate the current comprehension degree of a concept during the learning progresses. As known, the status near to the ending of learning should be emphasized. In other words, a pupil would be regarded as having been familiar with the concept if he can finally pass the corresponding test independent of times of previous failures. To simulate the phenomenon, three kinds of information must be kept: the number of making wrong answers W, the number of making contiguous correct answers after the last wrong answer C, and the total number of answering T. The weighted correct rate is defined as 1-W / [(T-W-C)+W+2*C], i.e., 1-W/(T+C). The interpretation of the weighted correct rate would be based on fuzzy expression in our system.

Another important issue is the way of detecting the formation of a misconception. A misconception may be caused by some blind spots of learning and thus always makes the learning process trap into a loop. A good diagnosis module of a tutoring system must have such kind of detection capability and could inform the other tutorial guidance module to show some appropriate auxiliary tutorials. If the test of each child node has passed, i.e., the learner has traversed and correctly answer all questions related to the concepts of all children nodes, the conceptual node is marked as P (Passed). If a learner cannot pass the test of a conceptual node and all its children nodes satisfy one of the following two conditions, then the learner is identified as trapping in a misconception corresponding to the conceptual node. The two conditions are <i>the child node has been marked as P; or <ii>the weighted correct rate is absolutely 1 (100%).

3 The Development and Design of Our System

Based on those described ideas, a prototype tutoring system comprising a testing and evaluation module has been developed and demonstrated. Microsoft Visual FoxPro 6.0 is used under the platform of Microsoft Windows 98. There are four modules included in our system shown in Figure 2.

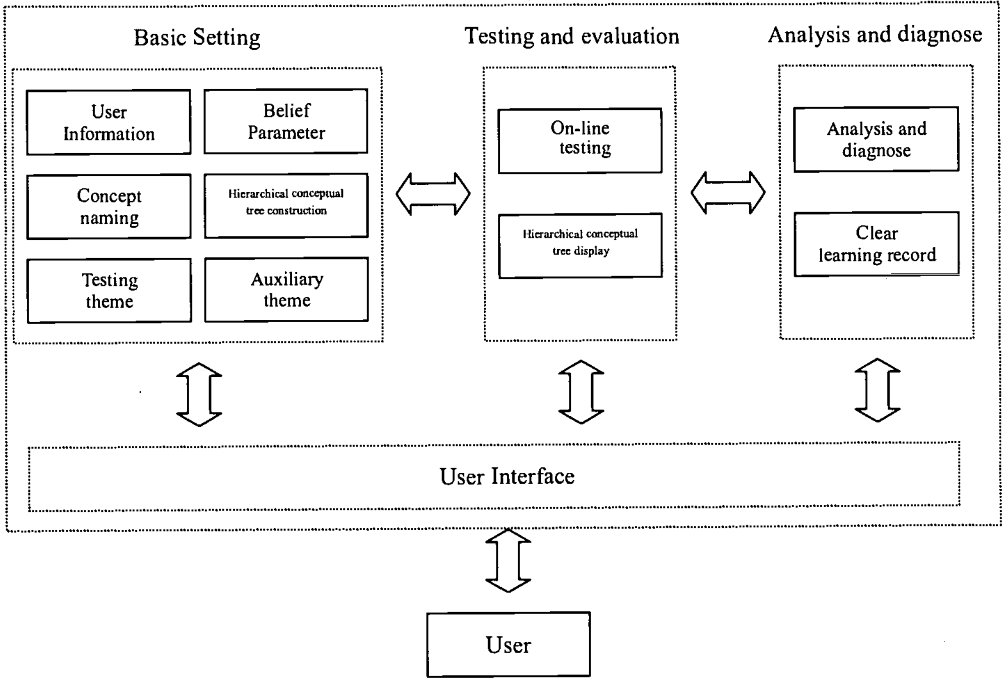


Figure 2. The architecture of the tutoring system

4 Conclusion and Future Work

In the research, techniques of fuzzy theory and uncertainty reasoning are applied to create a novel tutoring system. As demonstrated, the proposed tutoring system shows an excellent capability to present proper tutorials to guide pupils, precisely evaluates their learning status, and then shows auxiliary teaching materials to prevent pupils from trapping in any formed misconception. Finally, the traversal of learning would be analyzed and interpreted by fuzzy expressions.

Besides, some issues are worthy of deeper investigations through the study:

1. Some adaptive techniques of machine learning, e.g., genetic algorithm and artificial neural networks, should be applied to help instructors to automatically choose or tune parameters used in the tutoring system.
2. More applications about the proposed system should be examined to show its portability.

Acknowledgement

This study is supported by National Science Council of R.O.C. under contract number
NSC88-2520-S-212-003

References

[1] Anderson, J.R,"Cognitive modeling and intelligent tutoring", *Artificial Intelligence.* 42, pp.7-49, (1990).

[2] Pimentel, K. and Teixeira, K, "Virtual Reality: Through thenewlooking glass", Blue Ridge Summit, PA: Windcrest/McGraw-Hill, (1994).

[3] Rickel, J. and Johnson W. L, "Pedagogical agents for immersive training environments", Proc. of the 1St International Conf. on Autonomous agents, New York, (1997)..

[4] Chien Chou and Chuen-Tsai Sun, "The design and implementation of a cooperative distance learning environment: The CORAL experience", Education Technology and Media, 26, pp. 13-21, April, (1997).

[5] C. S. Hsu, H. F. Dou and G. J. Huang, "The application of concept inheritance in learning diagnosis system", Proc. of ICCAI'98, pp. 602-609, (1998).

[6] Shafer, G., "Mathematical Theory of Evidence ", Princeton University Press , Princeton , New Jersey ,(1976).

The Design and Implementation of Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students:AEGIS

Tsunenori Mine Akira Suganuma Takayoshi Shoudai
Graduate School of Information Science and Electrical Engineering, Kyushu University
6-1 Kasuga-kouen, Kasuga, 816-8580 JAPAN
E-mail:{ mine@is, suga@is, shoudai@i }.kyushu-u.ac.jp

Abstract

Many Internet technologies enable us to hold lectures with Web contents and even develop new lecture methods using the technologies. This paper proposes AEGIS (Automatic Exercise Generator based on the Intelligence of Students) that generates exercises of various levels according to each student's achievement level, marks his/her answers and returns them to him/her. In order to realize this feedback mechanism, we currently restrict the question-types which are generated to the following three types: multiple-choice question, fill-the-gap question, and error-correcting question. All question-types can be generated from the same tagged document. The aim of this system is to help the students understand the lecture with exploiting preexisting electronic documents.

Keywords: Artificial Intelligence in Education, Web-Based Learning, Exercise Generator

1 Introduction

As the Internet has come into wide use, WWW environments provide lots of opportunities to various fields. In the educational domain, Web data are being exploited as useful materials. We have been developing Web-based self-teaching systems and building the tools for helping students understand their subjects[1, 2, 3, 4].

We are currently focusing on the automatic student's achievement level evaluator that generates an exercise from tagged documents, presents it to students and marks their answer automatically. We call the system AEGIS (Automatic Exercise Generator based on the Intelligence of Students)[6, 7].

Creating exercises which are suitable for students is not easy. When we try to make some exercises for them in classes, we have to take at least their achievement level into considerations. The well-considered exercises are useful not only to measure the achievement level of students but also to improve their performance. It is not easy task for any teacher to make exercises of various difficulties according to their achievement level. Besides, it is very important to mark the students' answers and return the marked results to them for keeping their learning enthusiasms. This task becomes harder in proportion to the number of the students in a class[5].

This paper discusses AEGIS, which generates the three question-types from the same tagged data. Guessing the achievement level of each student from his/her trial history, AEGIS selects the most suitable question-type and exercise for him/her according to not only his/her achievement level but also the difficulty of the tagged data. After marking his/her answer, AEGIS returns it to him/her with its explanation.

The aim of this system is to exploit pre-existing electronic documents, in particular, our on-line documents shown at our Web site (http://cl.is.kyushu-u.ac.jp/Literacy) and to help students understand their lecture whose materials are set up as Web data so that they even at home can try exercises using AEGIS through the Internet.

The rest of this paper is constructed as follows: Section 2 shows related works to discuss the difference from AEGIS. Section 3 describes question-types that AEGIS deals with, considering both view points of students(answerers) and teachers(questioners) and Section 4 describes the exercise generating process by AEGIS. Section 5 shows the overview of AEGIS.

2 Related Works

A lot of automatic quiz generators have been proposed so far. Browning et. al. proposed Tutorial Mark-up Language(TML in short) to generate questions automatically[8, 9]. TML has a couple of tags to specify a question, a multiple-choice and a message. It requires a correct answer in a multiple-choice tag to mark a student's answer to the question. Carbone et. al. proposed CADAL Quiz[10], which generates a multiple-choice quiz from a question database. After marking a student's answer, CADAL Quiz returns the result to him/her and tutors. Both of them restrict the question type only to a multiple-choice quiz. On the other hand, ClassBuilder[11] generates many kinds of quizzes and grades a student's answer. However, all of them do not mention any effect of making the difficulty level of question-type change according to the students' achievement level. In order to improve their performance and keep their enthusiasm to challenge the quiz for a long time, it is indispensable to consider their performance level for generating their exercise. This point is the difference from other systems. AEGIS makes use of pre-existing electronic documents so as to embed tags into them, generates exercises automatically with tagged documents according to students' achievment levels, and reestimates both their levels and the difficulty level of the generated question through marking their answers.

3 Question-Types

There can be several types of a question in every subject. Since our aim is to get a computer generate an exercise and mark student's answer to it, we thus restrict to the following three question-types: multiple-choice question, fill-the-gap question, and error-correcting question.

Multiple-choice question. Students choose the correct answer from a given candidate list.

Example. Complete the sentence. Choose your answer from the following list.

Data structures need to be studied _____ order to understand the algorithms.

(1) an (2) in (3) on (4) at (5) by

Fill-the-Gap question. Students try to fill in the blank of a given sentence with the correct answer without any help.

Example. Fill in the blank with the right word.

Data structures need to be studied _____ order to understand the algorithms.

Error-correcting question. Students have to find the wrong expression in a given sentence and correct it.

Example. Right or wrong? Correct the sentence if it is wrong.

Data structures need to be studied an order to understand the algorithms.

All of these question-types can be constructed from a sentence by replacing one or more consecutive words with a blank or a wrong expression. We call the region replaced *hidden region*. We note that these three question-types have different difficulties even if they are constructed from the same *hidden region*. Figure 1 shows the tagged data to be used for generating the above three types of questions.

```
⟨QUESTION SUBJECT="idioms"⟩
Data structures need to be studied ⟨DEL CAND="an,on,at,by"⟩ in ⟨/DEL⟩ order to un-
derstand the algorithms.
⟨/QUESTION⟩
```

Figure 1: The tagged data to generate three question-types shown in Section 3

Students' View Point

Every multiple-choice question has surely the correct answer in its candidate list and contains the information that leads students to the correct answer. They can therefore make their choice with confidence from the list. In the case of a fill-the-gap question, they have to fill in the blank by themselves with their convinced answer without any information about the answer. Comparing both question-types, we can say that a fill-the-gap question is more difficult than a multiple-choice one. In the case of an error-correcting question, it forces them to determine whether or not there is an error in the question sentences and to correct it if it is found. An error-correcting question gives no information leading them to its correct answer, and the wrong expression in the sentences is not clear for students. We can therefore say that an error-correcting question is the most difficult one for students among those question-types.

Teachers' View Point

Once teachers set a *hidden region*, the efforts that are required to make with the three question-types are similar. The process for making exercises is as follows: in the case of a fill-the-gap question, the teachers have nothing to do. There is no information that they have to add to the exercise paper. We can say that a fill-the-gap question is the easiest one which is made among these three question-types. In the case of an error-correcting question, teachers have to think of at least one wrong expression which can be replaced with the *hidden region*. In the case of a multiple-choice question, they have to prepare several distractors to construct a candidate list. We can say that a multiple-choice question requires more information than an error-correcting one. From their points of view, a fill-the-gap question is consequently the easiest one which is made, and an error-correcting question is easier than a multiple-choice one.

4 Automatic Exercise Generating

4.1 Exercise Generating Process

The exercise generating process from teaching documents is summarized as follows:

1. Setting a *hidden region*: teachers make clear their intention why they want to ask the question to their students, that is, they consider which of the *hidden regions* is the most suitable for their intention.
2. Selecting a paragraph or sentence(s) from teaching documents: the sentences before and after *hidden regions* are often of importance to ask their students the unique answer of the question. We call the paragraph or sentence(s) a *question region*. A *question region* may have more than one *hidden region*.
3. Constructing a candidate list: a multiple-choice question requires a couple of distractors to set up a list of answer candidates. Any distractor should be natural so as to be added to the list. This list depends on the teacher's intention.

These three steps are deeply related to the teachers' intentions. It is not easy to extract such intentions automatically from the teaching documents. AEGIS system thus deals with tagged documents that already have the information such as *hidden regions* and candidate lists.

4.2 Necessary Information for Generating Exercises

In order to embed the above three kinds of information into the teaching documents, we define the following three tags: QUESTION, DEL, and LABEL.

QUESTION surrounds a *question region*, that is, the statements between ⟨QUESTION⟩ and ⟨/QUESTION⟩ are a *question region*. In the region, there can possibly be some expressions that are related to a *hidden region*. They can be good hints to lead students to the correct answer.

SUBJECT is the unique attribute of QUESTION. Its value stands for the subject or topic of *question region*.

DEL indicates a *hidden region*, which is the word(s) or sentence(s) between ⟨DEL⟩ and ⟨/DEL⟩.

A fill-the-gap question can be generated only by replacing the *hidden region* with a blank.

CAND is one of DEL's attributes. It is used to specify a candidate list.

LABEL has an attribute NAME that specifies a dependency relation with a *hidden region*. The sentence/s surrounded by LABEL tags is/are presented as a reference for the answer of a question, which will be generated with the DEL tag whose REF's value is the same as that of the NAME of the LABEL.

⟨QUESTION SUBJECT="W_S"⟩ *question region* ⟨/QUESTION⟩		
W_S	::=	word or symbol, where a backslash (\) must be added just before the symbol if it is a comma (,), double quotes ("), or a backslash (\).
⟨DEL CAND="CANDIDATE" LEVEL="PAIR" GROUP="ID" REF="ID"⟩ *hidden region* ⟨/DEL⟩		
CANDIDATE	::=	W_S | W_S,CANDIDATE
W_S	::=	word or symbol, where a backslash (\) must be added just before the symbol if it is a comma (,), double quotes ("), or a backslash (\).
PAIR	::=	LOW,HIGH
LOW	::=	an integer between 1 and 10
HIGH	::=	an integer between 1 and 10
ID	::=	keyword
⟨LABEL NAME="ID"⟩ sentences ⟨/LABEL⟩		
ID	::=	keyword

Figure 2: Tags for exercise generations

4.3 Necessary Information for Adjusting Difficulty Level of Question

The additional three attributes of DEL, which contain the information on the difficulty of solving the exercise, are LEVEL, GROUP, and REF. They specify the difficulty of each *hidden region*, and the connections to other *hidden region*.

LEVEL specifies the difficulty of the exercise to be generated from a *hidden region* itself. The value of this attribute is a pair of integers between 1 and 10. These integers specify the lowest and highest achievement level of the students who can try the exercise. AEGIS system determines whether or not the *hidden region* is worth being transformed into the exercise by comparing the student's achievement level from the both values of LEVEL.

GROUP specifies the dependency relation between *hidden regions* and holds the uniqueness of the correct answer. This GROUP is used to adjust the exercise level. If we want to generate more difficult exercises, all the *hidden regions* that have the same values in GROUP are replaced with blanks or wrong expressions at the same time. On the other hand, for generating easier ones, some of the *hidden regions* in the group are not transformed because those regions help students answer the question as hints.

REF specifies the dependency relation between a *hidden region* and other expressions than the *hidden region*. Both the region and expressions are specified with LABEL. If a *hidden region* is connected to an expression, the value of REF in the *hidden region* is the same as that of NAME in the expression with LABEL.

5 AEGIS system

5.1 Overview of AEGIS

The AEGIS system consists of three databases: *Exercise DB* (**EDB** in short), *User Profile DB* (**UPDB** in short) and *Level Management DB* (**LMDB** in short), and three main database managers: *Exercise Generator* (**EG** in short), *Answer Evaluator* (**AE** in short) and *Level Manager* (**LM** in short). The overview of AEGIS is shown in Fig. 3.

Teaching documents with the tags are compiled into the **EDB** and **LMDB**. All of the *question regions* are indexed sequentially and each *hidden region* is labeled with its own subindex of the index of each *question region*. The level of a *hidden region*, which is deeply related to the level of the question to be generated from the *hidden region*, is stored in the **LMDB** together with the index of the *hidden region*. The level of each *hidden region* in **LMDB** is reexamined regularly. **UPDB** keeps students' trial histories with their current achievement level.

EG and **AE** make communications with the users (students) through Web browsers after being invoked through CGI (Common Gateway Interface).

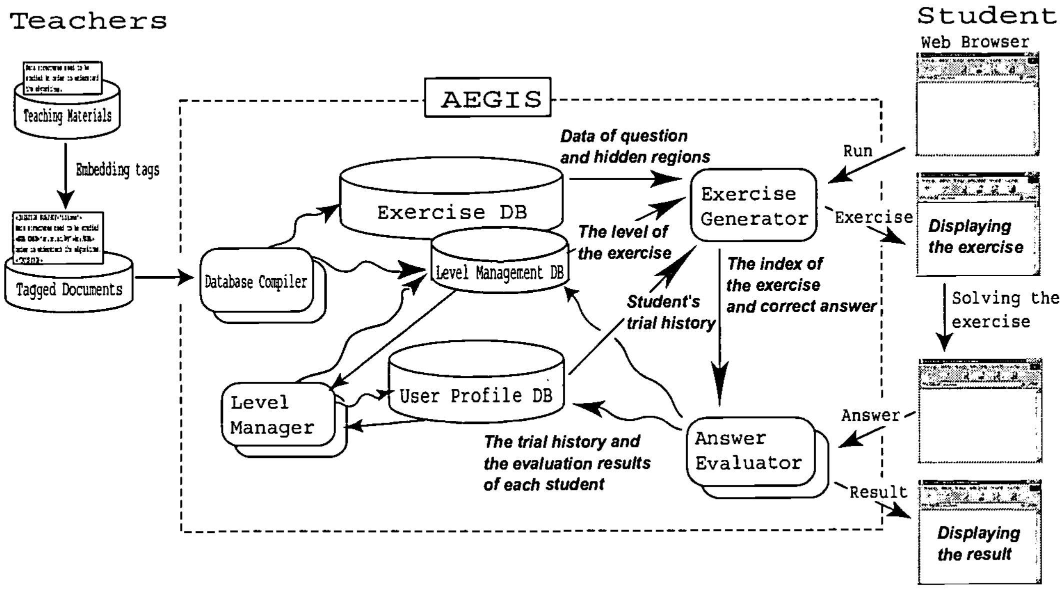


Figure 3: Overview of AEGIS

5.2 Exercise Generator(EG)

The exercise request from a student invokes **EG**. The **EG** searches the most suitable *hidden region* in **EDB** with looking over both the student's profile stored in **UPDB** and the level of the *hidden region* stored in **LMDB**, and determines the question-type of the *hidden region*. As mentioned in section 3, every question level has a relation to the question-type. **EG**'s decision process of the question-type thus employs the following strategy: If the student's achievement level is closer to the lowest number in LEVEL of the *hidden region*, **EG** selects a multiple-choice question as the question-type with high probability. On the other hand, if it is closer to the highest number in the LEVEL attribute, **EG** selects an error-correcting one.

Once **EG** determines the question-type of the *hidden region*, it is not difficult to generate the question. This is because the *hidden region* represents the correct answer of the question which is generated and teachers have already given the list of distracts explicitly with CAND attribute. Now, let's see how **EG** works when it generates the three kinds of questions:

- *Multiple-choice question*: **EG** randomly constructs one possible list for the multiple choice with both the correct answer and some distracts and outputs a question, which is generated by replacing the *hidden region* with a blank, with the list.
- *Fill-the-Gap question*: **EG** outputs a question which is generated only by replacing the *hidden region* with a blank.
- *Error-correcting question*: **EG** outputs a question which is generated by replacing the *hidden region* with one of the wrong answers specified in the CAND attribute.

Figure 4 shows an example of teaching documents with the tags. It is a piece of the teaching documents in the elementary course of Computer Literacy at our university. This course is taken by all first and second year students, about 2,300 students[5]. The teacher's intention in the example document is to teach how to use multiply and divide operations. Figure 5 shows the three question-types which are generated from the document.

5.3 Answer Evaluator(AE)

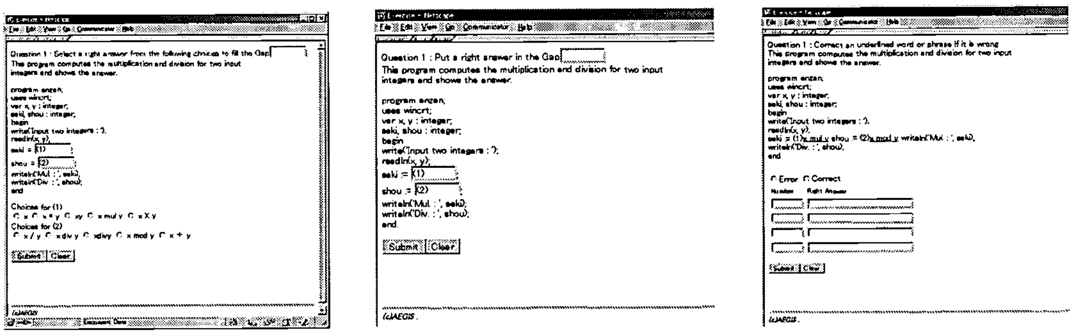
After outputting a question to the student, **EG** sends the following three kinds of information to ask **AE** to mark his/her answer: the index of a *hidden region*, the question-type, and the correct answer. After

In the previous section, we learned a program for adding two integers and showing the answer on the display. In the similar way, for all basic arithmetic operations including addition, subtraction, multiplication, and division, we can make a Pascal program in the following way.
⟨QUESTION SUBJECT="arithmetic operations"⟩
This program computes the multiplication and division for two input integers and shows the answer.

```
program enzan;
var x,y:integer;
   seki,shou:integer;
begin
   write('Input two integers : ');
   readln(x,y);
   seki:=⟨DEL CAND="x,xy,x×y,x mul y" LEVEL="1,5"⟩x*y⟨/DEL⟩;
   shou:=⟨DEL CAND="x/y,x÷y,xdivy,x mod y" LEVEL="1,5"⟩x div y⟨/DEL⟩;
   writeln('Seki:',seki);
   writeln('Shou:',shou)
end.
```

⟨/QUESTION⟩
The 7th statement multiplies x by y, and the 8th statement divides x by y. We note that the answer of "div" is an integer.

Figure 4: Example of teaching documents with the tags



Question 1 : Select a right answer from the following choices to fill the Gap

This program computes the multiplication and division for two input integers and shows the answer.

```
program enzan;
uses wincrt;
var x, y : integer;
seki, shou : integer;
begin
write('Input two integers : ');
readln(x, y);
seki := (1)
shou := (2)
writeln('Mul : ', seki);
writeln('Div : ', shou);
end
```

Choices for (1)
○ x ○ x * y ○ xy ○ x mul y ○ x X y
Choices for (2)
○ x / y ○ x div y ○ xdivy ○ x mod y ○ x ÷ y

(a) Multiple-Choice

Question 1 : Put a right answer in the Gap

This program computes the multiplication and division for two input integers and shows the answer.

```
program enzan;
uses wincrt;
var x, y : integer;
seki, shou : integer;
begin
write('Input two integers : ');
readln(x, y);
seki := (1)
shou := (2)
writeln('Mul : ', seki);
writeln('Div. : ', shou);
end.
```

(b) Fill-the-Gap

Question 1 : Correct an underlined word or phrase if it is wrong

This program computes the multiplication and division for two input integers and shows the answer.

```
program enzan;
uses wincrt;
var x, y : integer;
seki, shou : integer;
begin
write('Input two integers : ');
readln(x, y);
seki := (1)x mul y shou := (2)x mod y writeln('Mul : ', seki);
writeln('Div. : ', shou);
end
```

○ Error ○ Correct

Number Right Answer

(c) Error-Correcting

Figure 5: Three questions generated from the document in Figure 4

marking his/her answer by matching with the correct answer, **AE** shows him/her the marked result and stores it with the index of the *hidden region* and the question-type into the **UPDB**.

5.4 Level Manager(LM)

Although the initial value of the level of each *hidden region* is specified by teachers, it continues to move up and down according to the students' achievement levels, which will change as time goes by. The supplement manager **LM** processes their achievement levels statistically, computes the revised level of each *hidden region*, and stores it into the **LMDB**. **LM** increases the difficulty level of a question if a student whose level is greater than the level of question answers it wrongly, and decreases if a student whose level is less than the level of question answers it correctly. The new difficulty level of a question is consequently determined as shown in Fig.6.

After updating **LMDB**, **LM** updates the student's achievement level according to the difficulty levels of all questions he/she correctly answered.

Now, we show the formal definition of calculating both the achievement level of a student and the difficulty level of a question. Let $s_{i,t}$ and $q_{j,t}$ be the achievement level of student S_i and the difficulty level of question Q_j at time t respectively, where $1 \leq s_{i,t} \leq 10$,$1 \leq q_{j,t} \leq 10$. $s_{i,t}$ is recursively calculated with $q_{j,t}$ at stated periods and vice versa. They are defined as follows:

$$s_{i,t} = \begin{cases} 1 & \text{if } m_{s_i,t} = 0 \\ \frac{1}{m_{s_i,t}} \sum_{j=1}^{m_{s_i,t}} q_{j,t} \cdot \delta_{i,j} & \text{otherwise} \end{cases} \qquad \delta_{i,j} = \begin{cases} 1 & \text{if } S_i \text{ answered } Q_j \text{ correctly} \\ 0 & \text{otherwise} \end{cases}$$

$$q_{j,t} = \begin{cases} q_{j,t-1} + \frac{\sum_{i=1}^{m_{q_j,T}} |s_{i,\tau \in T} - q_{j,t-1}| \cdot \xi_{i,j}}{\sum_{i=1}^{m_{q_j,T}} |\xi_{i,j}|} & \text{if } \sum_{i=1}^{m_{q_j,T}} |\xi_{i,j}| \neq 0 \\ q_{j,t-1} & \text{otherwise} \end{cases}$$

$$\xi_{i,j} = \begin{cases} -1 & s_{i,\tau} \text{ is less than } q_{j,t-1} \text{ and } S_i \text{ answered } Q_j \text{ correctly} \\ 1 & s_{i,\tau} \text{ is greater than } q_{j,t-1} \text{ and } S_i \text{ answered } Q_j \text{ wrongly} \\ 0 & \text{Otherwise} \end{cases}$$

Where $m_{s_i,t}$ stands for the number of questions that S_i tried by t and τ is the latest time such that S_i tried to answer Q_j and $t-1 < \tau \leq t$. T is the set of τ. $m_{q_j,T}$ stands for the total number of students who tried Q_j in T. $q_{j,0}$, which is the initial difficulty level of the question Q_j, is given with the attribute $LEVEL$ of DEL tag by teachers.

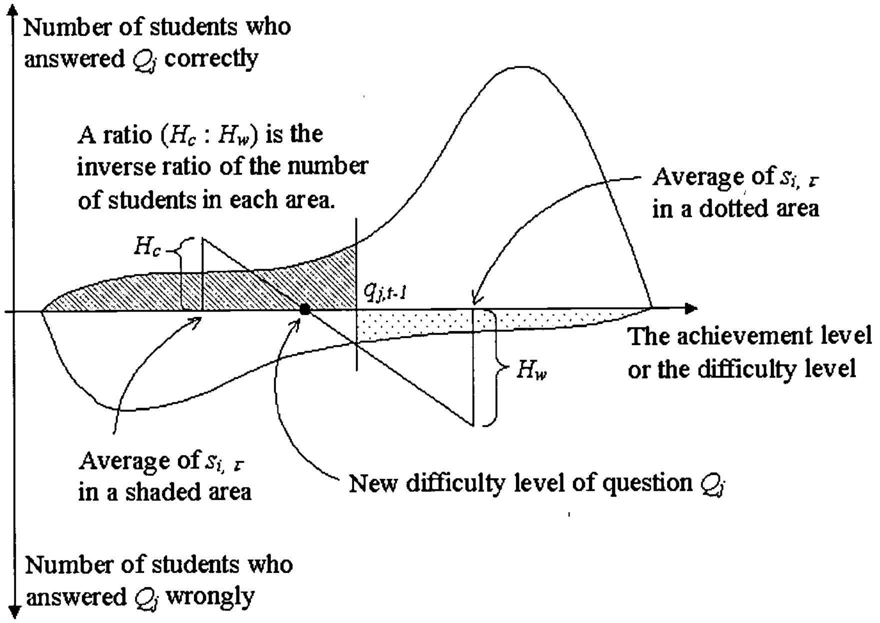

Number of students who answered Q_j correctly
A ratio ($H_c : H_w$) is the inverse ratio of the number of students in each area.
H_c
$q_{j,t-1}$
Average of $s_{i,\tau}$ in a dotted area
The achievement level or the difficulty level
H_w
Average of $s_{i,\tau}$ in a shaded area
New difficulty level of question Q_j
Number of students who answered Q_j wrongly

Figure 6: Renewing Difficulty level of Question based on Student's Achievement Level

6 Conclusions

We discussed our new Web-aided system AEGIS. The system is currently implemented in Perl scripts and CGI. We have a plan to evaluate this system by applying it to the real courses of Computer Literacy, which are taken by more than 2300 students at our university. We hope it will work fine as an educational tool for every student and help him/her to understand his/her subjects if teachers can make tags in their teaching documents. Also, we plan to implement a tagging tool and an algorithm to generate another kind of exercise that allows more than one correct answers.

Acknowledgments

We thank Dr. Teruko Mitamura at Carnegie Mellon University for her helpful comments to improve this paper.

This research was partly supported by the Grant for Special Academic Research P&P from Kyushu University and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Encouragement of Young Scientists(A), No. 11780281, 2000.

References

[1] H. Sato, T. Mine, T. Shoudai, H. Arimura, S. Hirokawa: On web visualizing how programs run for teaching 2300 students. In *Proc. Int. Conf. on Computers in Education, ICCE'97*, pages 952–954, 1997.

[2] T. Mine, D. Nagano, K. Baba, T. Shoudai, S. Hirokawa: On-web visualizing a mechanism of a single chip computer for computer literacy courses. In *Proc. Int. Conf. on Computers in Education, ICCE'98*, volume 2, pages 496–499, 1998.

[3] R. Fujimoto, A. Suganuma, T. Mine: Development of a classroom management system on the web. In *Proc. Int. Conf. on Computers in Education, ICCE'99*, volume 2, pages 756–759, 1999.

[4] R. Fujimoto, Y. Tsutsumi, A. Suganuma: Cacce: Computer aided cooperative classroom environment. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, page 1686, 1999.

[5] S. Hirokawa, T. Miyahara, T. Mine, T. Shoudai, M. Mori: Teaching 2300 students with www – practice and experience at kyushu university. In *Proc. ERI'96*, pages 59–63, 1996.

[6] T. Shoudai, A. Suganuma, T. Mine: AEGIS: Automatic Exercise Generator with Tagged Documents based on the Intelligence of the Students, Joint Conference on Knowledge-Based Software Engineering, to appear. JCKBSE2000, Brno, Czech Republic, 2000

[7] T. Mine, T. Shoudai, A. Suganuma: Automatic Exercise Generator with Tagged Documents

Considering Learner's Performance Proceedings of WebNet2000, San Antonio, Texas, to appear as a short paper, 2000

[8] P. Browning, J. Williams, D. Brickley, H. Missou: Question Delivery over the Web using TML http://www.ilrt.bris.ac.uk/netquest/liveserver/qbanks/demos/paul/lboro/ohp1.html, CAA, 1997

[9] P. Browning: TUTORIAL MARKUP LANGUAGE - A CBA SYSTEM http://www.soton.ac.uk/~ukgec/workshop/5-cba/minutes.htm#TUTORIAL, 1998.

[10] A. Carbone, P. Schendzielorz: A Web-Based Quiz Generator for Use in Tutorials and Assessment Global J. of Engng. Educ., Vol.I, No.3, 1997 http://www.eng.monash.edu.au/usicee/gjee/vol1no3/paper20.htm.

[11] ClassBuilder GradeBook And Exam Creation Software: http://www.classbuilder.com/

The Design of CAI with Thinking Activity to Progress Constructive Teaching
- An Example of Division-concept in Mathematics of Elementary School

Pou-Yah Wu*, Mei-chueh Shao, Kai-Ren Chen, Ying-Xin Zhong, Ting-Wei Lin and Cheng-Wei pan.
Department of Mathematics Education, National Tainan Teachers' College, Taiwan
**Pou-Yah Wu's e-mail:pywu@ipx.ntntc.edu.tw*

This study aims at establishing a computer assisted learning system of division-concept of networked elementary school mathematics course based on constructivism and stress on students' thinking activities. It explores how students' thinking manifest on network, how the thoughts of the learner and those of the students on-line transfer, and how the thinking of the virtual students' solving problems reflect, so as to develop a set of CAI system about constructive pedagogy. In the system, we provide the learners with diverse tools for thinking activity, letting him/her choose what he/she needs to solve problems. We use network technology to simulate the real learning situation and to make the learner and the user on the line and the virtual students to communicate and discuss immediately. By setting up the CAI system that is compatible with the mathematics education of the elementary school in Taiwan now, we expect the learner to establish the right concepts positively so as to attain constructive pedagogic concept.

Keywords: **Constructive pedagogy; Division of Mathematics; Elementary School; Networked CAI; Thinking Activity.**

1 Introduction

The course design of pedagogy in Taiwan before 1993 is based on objective theory of knowledge. However, the pedagogic design ignores the complex and interactive phenomenon practically. Therefore, mathematics of elementary school in Taiwan in 1993 adopts pedagogic theory of constructivism [6]. Constructive pedagogy improve the shortcomings of the traditional pedagogy; but it also cause the deficiency of pedagogic duration owing to the orders of discussion and reflection in case it is put into practice in the real pedagogic environment. With the popularity of network, provided constructive concepts are applied to the learning environment of network, CAI effect would be promoted further. This study aims to design networked pedagogic environment matching "basic division-concept in mathematics of elementary school" by the learner's thought, using network technology, letting the learner have an environment to learn at home. The traditional CAI system neglects the positive learning and the interaction between the learners. So, we take into how to facilitate the interactive relationship between the system and the learner. Through the transmission of the networked thought, the learners can real-time communicate, making up a whole constructive learning environment, hoping to attain the constructive pedagogic concept.

2 Principles of system establishment

2.1 Basis of learning theory

The pedagogy of constructivism lies in stressing "knowledge is constructed positively by the learner", so that pedagogic design should arrange activities of learning-orientation. In the process of learning, the teacher serves as "problem poser" whereas the students acts as "problem solver"; the teacher plays the role of assistance, and the learner should construct knowledge positively through the interactive discussion between the learners [2]. Each learner utilizes his previous concepts to expound the phenomena around, and then comes up with adjustment or assimilation toward his acquired cognitive structure to establish new concepts. Besides, the learning situation is also an important part of the content, functioning to help the learner to comprehend the differences between the perspective on conceptual traits. Thus, the learning activities ought to provide learners with quasi-actual experimental situation to manipulate, explore. By means of the cognitive conflict brought about by the students in the process of the activity, challenging his original concepts, he/she constructs the right concepts via the discussion and coordination with one another.

2.2 Basis of course content-concept of division

Division is the anti-calculation of multiplication. Both multiplication and division are thought of as the transformation of unity quantity. The so-called "transformation of unity quantity" refers to that using unity quantity as that described by calculating unit, transforming to another description by calculating unit using another unity quantity [1,3]. The situational mode of division question is categorized into two basic principles of including division and even division. Seen from the viewpoints of "transformation of unity quantity" to look at the questions of multiplication- division, the questions of multiplication is to reduce the quantity suggested in the units of higher layers (units accumulated by several units of lower layers) to the activity of transformation from the quantity suggested by units of lower layers; whereas the questions of division "including division" is on the contrary, that is, the quantity suggested by the units of lower layers changed into the transformation activity by the quantity suggested by the units of higher layers. As to even division, it is an activity of new unity quantity of high layers and unknown unity quantity.

2.3 Foundation of system establishment

This system is a learning environment constructed on the network, adopting three-tier client/server system architecture, and meaning adding a layer of service server on the original client-server two-tier client/server system architecture. In the structure of three-tier client/server master-slaver, the part of management of learning data is in the charge of database server, web server takes charge of teaching jobs, while the users of client proceed all kinds of learning activities via browser.

3 Pedagogic design of networked construction

3.1 Pedagogic design of constructive division of new course

The two questions types of division (including division and even division) should be reckoned as different ones, then helping students combine these two types of questions gradually. And by the activity of consecutive subtractions solving questions to communicate with the relationship, then introducing the format of division calculation. Thus, in the design of pedagogy, place the two combined types of characters, letting children solve problems by tangible objects or emblems and try to record the activity of solving questions. After solving the questions including division and even division successfully, try further to grasp the times of distribution including viewpoints of division when confronted with them again [4,5]. The number of unity quantity can be decided by the times of distribution to help students realize and construct the relationship containing two types of questions as to including division and even division. Finally they can introduce the processes of solving questions concerning the methods of many-steps subtraction recording including division and even division and discuss and form the formulas using " ÷ "taking notes of the common sense about the activity of solving questions including division and even division, letting children construct the whole meaningful concept of division.

3.2 CAI pedagogic design of constructive pedagogy by thinking activity

This system emphasizes the spirit of construction to help students establish the concept of division, thereby, expecting the system to become more congenial to the real pedagogic environment. We let the computer become a virtual teacher, besides posing problems, he/she can judge the students' types of solving problems and mode of operation, and providing the dialectics and clarification and discussion undertaken between the users or between the user and the virtual students. Thus, the design of the problems by this system is introduced by the ordinary ones of daily situation to make sure if students have grasped the messages of the problems and communicate and clarify the messages with each other through asking (As in Figure 1). After posing the problems and clarifying the messages, let the students solve the problems. In order to make the system grasp the process of solving problems and thinking, we design "tool table of operation of thinking activity", which contain tangible objects, representation, digits and the symbol of calculations and so on. For example, as shown in Figure 2, if learner choose "to bakery", then the tangible objects can be used to solve the problems. If the learner choose "drawing circles", then representation can be used as the tools of solving the problems (As in Figure 3); if the learner choose "to digital factory", then digits can be used as the tool of operation (As in Figure 4). By the tool of operation chosen by the user, the computer can grasp what he thinks. If the user fails to solve the problems by themselves, they can discuss with others on the line, or discuss by the activity of solving the problems of the virtual students (As in Figure 6and 7) to attain the cooperation and learning. At last, after the user solve the problems successfully, the computer will play the role of the virtual teacher, raising questions to let the user to fortify the concepts, avoiding no continual between the user's order of thought and the concept (As in Figure 5). Then posing problems again to judge the students' learning state in order to proceed another activity dynamically. In doing so gradually, the system expects the learner construct an overall meaningful concept of division.

4 Architecture and implementation of system

4.1 Design environment and tools

This system uses Windows NT server as server platform. The developing languages include HTML, JavaScript, ActiveX, ASP (Active Server page) and so on. Using ASP as the main way of control, and exercising ASP and ODBC (Open Database Connectivity) to go with it, making the user's management of teaching material simplified. In the aspect of editing course software, Authorware5 is a chief developing tool.

4.2 System flowchart

The system flowchart we designs just as Figure 8 shows, the general elucidation is as follows:

1. Pedagogic situation of networked construction: In the beginning, the system would ask the user to register data to set up the database of students' basic data. At the outset of the course, the system will judge the user's competence by the pretest; then according to the basis, the system can pose the problems. After clarifying the messages of the problems, the system lets the user proceed to solve the problems. After solving the problems successfully, it lets the user carry on a series of on-line discussion and communication with the students or virtual students. Based on the acquired knowledge, the students construct the concepts, and fortify or revise the concepts through the experience of reflection. Again, the system poses the problems to judge the students' learning situation, then it proceeds the next teaching activity.

2. Database of "student model": It consists mainly of three databases:

(1) Database of students' basic data: It is used to record the students' basic data such as name, age, the experience of using the computer and so on.

(2) Database of learning: It is used to record the course units the students have learned, the learning state and duration of each unit, and the students' learning results and so forth.

(3) Database of learning achievement: It records the students' assessment about answering and the mode of students' operation.

3. Database of "posing problems of constructive pedagogy": It stores the material content of division pedagogy. The content contains two types of division problems (including division and even division) and various types of processes pedagogic activities.

4. Database of problems: It stores the problems for pretests and posttests.

4.3 Function of on-line communication

Because the system aims at establishing a more compatible with the learning environment of real pedagogy, so that this system design a series of communicative mechanism on the line to help students proceed the learning activities to produce the learning effect. The details will be narrated as follows:

1. Group of discussion: It is an open but not synchronized function on the line. Once the user encounters the learning difficulty, he/she can put the problems on the discussion place, and when other users see them, they can put forth the ways of solving these problems.

2. Room for discussion: It is an open and synchronization for communication. This on-line unction can improve the fact that the single CAI system fails to undertake the defects of communication and discussion immediately. Take Figure9 for example, the user in the room for discussion can carry on the mutual discussion, communication to solve the problems with other users on the line for their learning difficulty.

3. On-Line call: This is a one-to-one synchronous communication way, enabling the learners to proceed one-to-one discussion and forward the brief introduction to other users on the line.

4.4 Operation flowchart for User

When the user enter the system by using browser for the first time, the system would demand the user to register, thereby getting the user's data to set up student model basic data for database, and letting the user accept the pretest to judge the user's level of operation, and recording the user's answering situation. Utilizing the user's answer for reference, modifying the connection dynamically, letting the user connect the courses properly. Afterwards, whenever the user enters the system, he has to register user name and password as the recognition. The system then will proceed next activity according to the user's previous record. When the user undertakes the learning activity, the system will take down the learning state each time, so as to analyze if the user's learning state will attain the expected aim and will be used as learning analysis.

5 Conclusions

With the approach of eased network age, the network will definitely become the trend. Thus, establishing CAI system on the network cannot be delayed. In the light of these, we hope the constructive pedagogy combine with network to make up for the deficiency of pedagogy, letting the learners have more learning space, so as to acquire the real mathematics concepts. This study proceeds to test by the pedagogic content of "division-concept" of elementary school, presently testing all the functions provided by the system, hoping to reassess pedagogic content and system in many months, looking forward to reaching the learners' interaction, fulfilling the pedagogic concept indeed, letting children construct whole mathematics concept.

References

[1] 甯自強, "單位量的變換(一):正整數乘除法運思的啓蒙", 教師之友, Vol. 34, No 1, pp. 27-34, (1993).

[2] 甯自強, "國小數學科新課程的精神及改革動向—由建構主義的觀點來看—", 科學教育學刊, Vol. 1, No 1, pp. 101-108. (1993).

[3] 甯自強, "單位量的變換(二)", 教師之友, Vol. 36, No. 5, pp. 35-44, (1995).

[4] 楊瑞智, "國民小學數學新課程中年級除法教材的設計"八十二學年度數學教育研討會,嘉義,台灣,pp.119-136(1993).

[5] 楊瑞智, "國民小學實驗課程整數乘除算則的教材處理",八十六學年度數學教育研討會,嘉義,台灣 pp.98-113(1997).

[6] 蘇育任, "建構主義式教育的迷思與省思", 國民教育集刊, No. 5, pp. 121-139, (1997).

[7] 吳博雅, 王偉仲, 謝堅, 曾彥鈞, 陳詩茹, 牛晉媛, "網路建構式電腦輔助教學系統之研究—以國小數學科面積基本概念爲例", 1999 中華民國 CAI 研討會論文發表, pp. 159-165, (1999).

[7] Dalgarno, B. Constructivist Computer Assisted Learning: Theory and Techniques. WWW. 21Oct. 1997. Available: http://services.canberra.edu.au/~barney/articles/ascilite96/paper.htm (1997)

[8] Hiltz, S. R. Teaching in a Virtual Classroom. 1995 *International Conference on Computer Assisted Instruction ICCAI'95* Hsincku, Taiwan http://it.njit.edu/njIT/Department/CCC/VC/Papers/Teaching.htm

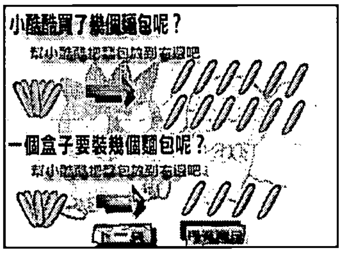


Figure 1:The Clarification of the problem

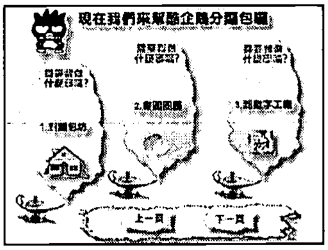


Figure 2:The choice of operation tool of problem solving

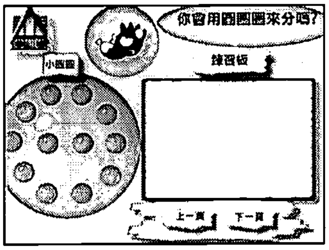


Figure 3:The presentation of thinking activity - representation

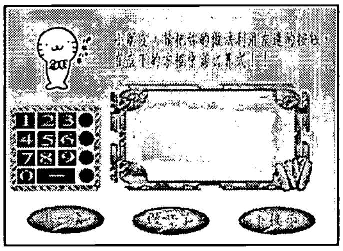


Figure 4:The presentation of thinking activity - digital and operator symbol

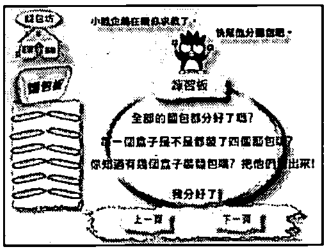


Figure 5:Reflection and discussion

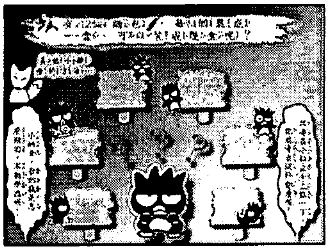


Figure 6:The strategies of virtual students

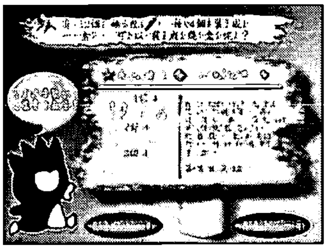


Figure 7: The communication of solving methods of virtual students

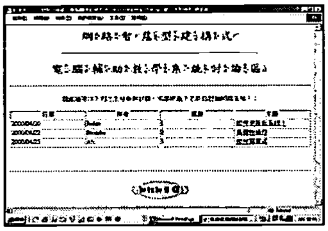
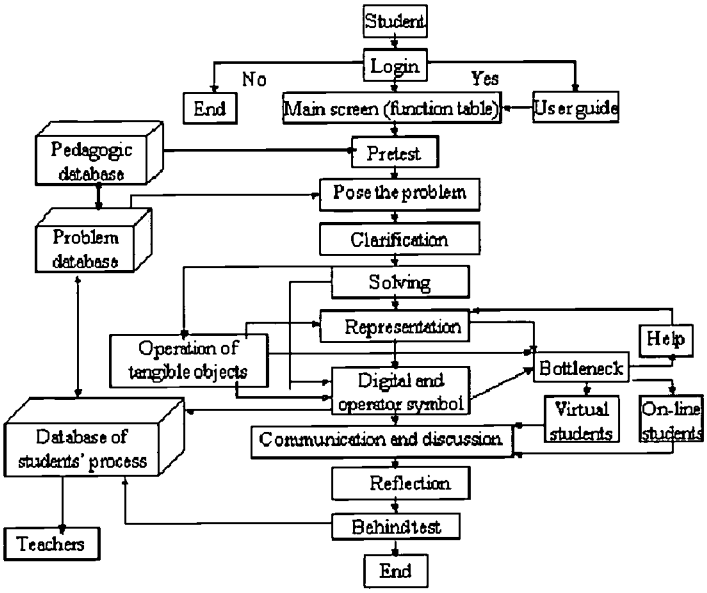


Figure 9: Group of discussion



Figurre8: The design of networked constructive pedagogy with thinking activity

The Estimation of Music Genres Using Neural Network and Its Educational Use

Yasuhiko TSUJI*, Kanji AKAHORI*, Atsuhiro NISHIKATA*
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan
Tel: +81-3-5734-2991
Fax: +81-3-5734-2991
E-mail:tsuji@ns.cradle.titech.ac.jp,
akahori@cradle.titech.ac.jp,
nisikata@cradle.titech.ac.jp

To develop a learning support system of music genre, a neural-network-based system was developed that can estimate the genre of music from partial information of a standard MIDI file of music. Standard MIDI files of 120 music titles has been identified into 4 genres, Japanese Popular Ballad, Jazz, Hard Rock and Heavy Metal after the Neural network of the system had been trained. Comparison shows that, the system developed, has a higher judgment rate than that of subjects. Next, the weight of the links were examined by an expert, 5 of the nodes in the Hidden Layer could be extracted.

Keywords: **Music Education, Neural Network, Intellectual Learning Support, MIDI**

1 Backgrounds and Objectives

Recently, popular music, for example *Beatles* etc, is included in recent music textbooks of Elementary, Junior High and High Schools in Japan. So, it is thought that music education using popular music will increase more and more in course of time. When students learn popular music, music genre of the music is an important factor[1]. In order to learn the musical feature of each genre, it is thought to be very effective. Systematic genres studying of popular music, in which students seems to be interested, is thought to be a way of the students' music experience enrichment.

An "Automatic Composition MAGIC (Music system for Arrangement and Intelligent Composition) Considering Music Style" was developed [2] by Minamikata in 1989 is one of the researches in the research field that treats plural genres of popular music. This System supports composition and adaptation using heuristic rules divided by music taste of gcnrc. It is said that rule-based system like this is effective when the system reproduces a already-known music taste or rule for the system, but there is an anxiety that generated music is conventional, and it is a problem for an unknown taste.

It can be said that the genre of popular music is the combination of different music. Now, many researches have done the grouping of music. Concerning Neural Network-based research, the research of Sakamoto (1999) grouped the music according to the sensibility information by using SD method [3]. If consider the flexibility and generality grouping by neural network differs from that of grouping by rules or multiple different analysis. So, it is said that moderate result can be expected for any unknown input by the process of grouping by Neural Network.

Based on the above research, we aim to develop the learning support system which can provide feedback on "Feature as the genre" of an unknown music with the Neural Network training of the music of various genres. Based on the above-mentioned background, we conducted this research in the following way. At the beginning, reserve experiment was done by an expert of popular music to confirm the factor for the estimation of the genre. Based on the obtained finding, we trained the Neural Network. Here the Neural Network was composed using the partial information as input signal and genre of the music as output or teaching signal. In order to use this system for education purpose in the future, the meaning interpretation for each factor of the Hidden Layer of the trained Neural Network was identified by an expert of popular music. Then, the genre estimation experiment was done using the subjects who seemed to have general experience of popular music. Lastly, the estimated average result of the subjects and the estimated result of this system was compared to show the effectiveness of this system.

2 Estimation of Music Genres by Expert

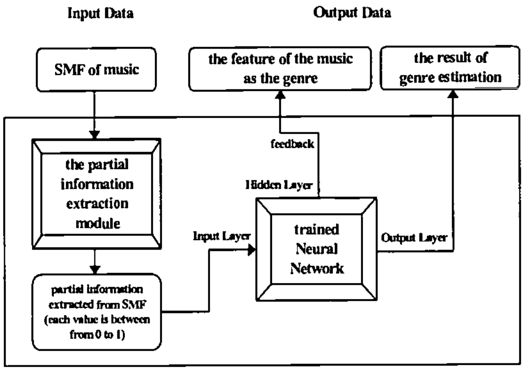
When music and genre are trained to the Neural Network, the problem is that we should take data to make an input signal from a long standard MIDI file. Therefore, we examine the mounting method of this system by knowing how the person judges the genre. For that, in the preliminary experiment we ask the expert about the factor of the genre estimation. The subjects had different musical instrument performance experience for ten years or more. The procedure was that they were made to listen ten in total of five genres. Also the factor to estimate the genre was interviewed. As a result, the following factors were found.

(1) The factor to estimate the genre is various according to the genre, and it's vague information.
(2) The factor to estimate the genre is local & partial information.

From (1), at first we got to the hypothesis that the estimation of music genre based on rules is very difficult and not proper. Under the above hypothesis, we propose to use Neural Network to deal with vague information in this research. As the input from (2), we judged that it was appropriate to extract partial information that seemed to be necessary for estimating the genre of music, and to assume it to be an input value of the Neural Network. The standard MIDI file (Hereafter, it is abbreviated as SMF) that is already a descriptive language was used as music.

3 Genre Estimation System

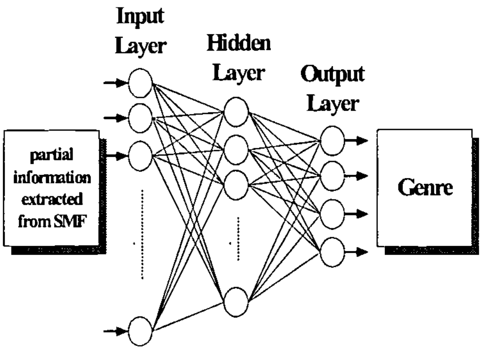
Figure.1 shows the composition of the genre estimation system. The flow of this system is as follows. When the user inputs SMF of music, the partial information extraction module extracts some partial information from the music. Then, it is put to the Input Layer of the Neural Network that has already been trained for music and the genre. The Neural Network feeds back the result of estimating the genre obtained from the Output Layer. Moreover, the feature of the music as the genre obtained from the Hidden Layer is planned to use as feedback in the future. If the module is developed, the user will be able to learn the genre.


Input Data
Output Data
SMF of music
the feature of the music as the genre
the result of genre estimation
feedback
the partial information extraction module
Hidden Layer
Input Layer
trained Neural Network
Output Layer
partial information extracted from SMF (each value is between from 0 to 1)

3.1 Extraction of Partial Information from SMF

SMF of the General Midi correspondence was used in this research. SMF includes various musical information such as Note-On (time of starting to ring each music sheet), Note-Off (time of finishing to ring each music sheet), Velocity (the strength of each attack), Note Number (pitch), and Program Number (kinds of musical instruments and tones) etc. The following three information of these score information were decided to use in the partial information extraction module.

1. Kind and tone of musical instruments extracted from Program Number (henceforth, we call it "Musical Instruments and Tones", which is expressed by an array of 128 Boolean type variable. Each valuable shows whether musical instruments (tones) of Program Number 1-128, were used in that music or not.).



Distribution of Rhythm extracted from the statistics of position of Note-On per a bar (henceforth, we call it "Distribution of Rhythm", which is expressed by an array of 16 integer type variable. Each variable shows the frequency for which Note-On event is held at the rhythm in one bar in the SMF.).

Distribution of Pitch extracted from the note number (henceforth, we call it "Distribution of Pitch", which is expressed by an array of 12 integer type variable. Each variable shows the frequency for which each pitch of 12 music scales is used in the entire music of SMF.).

3.2 Composition of Neural Network

Figure 2 shows the composition of the Neural Network. We adopted the Back-Propagation algorithm as the learning algorithm of the Neural Network. For the input signal, we used a combination of the values.

4 Outline of Genre Estimation Experiment using this system

4.1 Method

By the above-mentioned methods, the genre estimation experiment by this system was performed. 120 music titles of SMF which are composed of 30 titles each in Japanese popular ballad, Jazz, Hard Rock, and Heavy Metal, tried to be learned by the Neural Network. In this research, the combination of the following partial information was learned as an input data.

Musical instrument and tone	128bit
Distribution of rhythm	16bit
Distribution of pitch	12bit
Musical instrument and tone, Distribution of rhythm (+)	128+16=144bit
Distribution of rhythm, Distribution of pitch (+)	16+12=28bit
Musical instrument and tone, Distribution of pitch (+)	128+12=140bit
Musical instrument and tone, Distribution of rhythm, Distribution of pitch (+ +)	128+16+12=156bit

The number of units of Hidden Layer in each Neural Network is assumed to 10-30. The number of units of Output Layer is as many as the number of genres that the Neural Network learns. In this case, it requires four units in Output Layer, because there are four genres.

4.2 Result

The result of training is shown in Table 1. In the Table 1, "NN" means Neural Network, and - in NN means the Neural Network whose input information is described above. The result of training, NN was converged about 650 learning times, and

NN	Input Layer	Hidden Layer	Output Layer	Judgment Rate	Judgment Percentage	Learning Time
	128	20	4	119/120	99.2	X
	16	10	4	97/120	80.1	X
	12	10	4	103/120	85.8	X
	144	30	4	120/120	100	About 650
	28	30	4	111/120	92.5	X
	140	30	4	119/120	99.2	X
	156	30	4	120/120	100	About 1100

NN was about 1100 learning times, but other NN were not converged within ten thousand learning times. So, the trained Neural Network was able to judge the genre of learned music at 100%.

From this, it is suggested that the Neural Network like - that has single partial information in Input Layer can't finish learning, But the combination of those partial information make it enable to learn. This result supports the findings of experts at the preliminary experiment in Chapter 2 whose also says that the factor to estimate the genre is various according to the genre.

4.3 An Analysis of Hidden Layer

The Hidden Layer in the Neural Network is analyzed here. There is a heuristic method that each cell's tendency in which it is likely to make active or inactive is found by an expert, and then the meaning of factor is obtained[4],[5]. We used that method here.We focused on the weight of the link between Hidden Layer and Output that is above 10. Each unit from No.1 to 5 are activated by following genres.

Unit No.1:Hard Rock

Unit No.2:Hard Rock, Jazz

Unit No.3:Hard Rock, Jazz, Japanese Popular Ballad

Unit No.4:Heavy Metal

Unit No.5:Japanese Popular Ballad

Table 2. Analysis of Each Factor in Hidden Layer

Unit	Name of Factor
1	Hard Tone Factor
2	Synthesizer Tone Factor
3	Jazz-Acoustic Factor
4	Rhythm Tendency Factor
5	Combination Factor of Electric Instruments and Rhythm Tendency

Finally, each unit was named by a music expert. The summarized result is shown in Table 2.

5 Experiment by Subject

To investigate at how much rate can the subjects, twenty-five female university students were asked to listen to eight music titles of 4 genres of SMF with MIDI sound randomly, and to judge the genre and the factor for each music.The judgment rate of all the subjects was 66.5%.

To compare the judgment of subjects with this system, Neural Network was trained with 119 titles, and was made to estimate the genre of subtracted one as unknown music.

As a result, both Neural Network and have a judgment rate of 100% for eight unknown music titles. From this, the judgment of this system is higher than that of subjects with general experience of popular music.

6 Summary of Results

In this research, development and evaluation of genre estimation system ware performed aiming for the development of learning support system of music genre. The results are summarized as follows:

(1) The preliminary experiment for experts with an experienced popular music was performed, and a result that says that the factor to estimate the genre tends to be local & partial information was obtained.

(2) From this finding, genre estimation system using Neural Network was developed.

(3) 120 music titles have been identified into 4 genres, Japanese Popular Ballad, Jazz, Hard Rock and Heavy Metal at the rate of 100% by training the Neural Network to identify these 4 genres.

(4) The judgment rate was 66.5% as the result of the estimation experiment for subjects with general experience of popular music.

(5) This system was made to estimate 8 music titles, as an unknown music, out of 120 which were used in the genre estimation experiment by subjects. As a result, the estimation rate of 100% which is higher than that of the subjects (66.5%) was obtained.

(6) Each unit of Hidden Layer in trained Neural Network was enable to be named, and the factors of each unit were able to be extracted by the expert of popular music.

From this finding of 6, providing feedback on the features of the music from Hidden Layer becomes possible by the way of observing the result of meaning explanation of Hidden Layer in which the Neural Network has the feature of the music as a genre, observing the state of fire, and observing the input units which have tendency to make active to the fired units in the Hidden Layer.

From the result described above, the possibility of the development of a learning support system using this system for music genre is shown. And, it was thought that the trained Neural Network of this system has the application possibility not only to the learning support system but also to the supporting composition and adaptation.

Acknowledgement

The authors would like to thank Mr. UZZAMAN MD. ANIS and Assistant. Taizan SUZUKI for their cooperation.

References

[1]Hiroshi UEDA, Kazumi AKAIWA etc.: Hajimeteno ongakutsuu:50 genres 500 CDs, Ongakunotomosha,1993

[2]Junichi MINAMITAKA, Mayumi INO, Kunio SATO, Shigenori MORIKAWA: Automatic Composition MAGIC Considering Music Style, Collected lecture papers of the 39-th National Convention of Information Processing Society of Japan, No.3, pp.2242-2243,1989

[3]Takashi SAKAMOTO, Yoshinobu KAJIKAWA, Yasuo NOMURA: Classification of Group According to Tune's Impression Using Nonlinear Distinction Analysis in the Music Sensibility Space, Transactions of Information Processing Society of Japan, Vol.40, No.4, 1999

[4]Hisayoshi ITO, Hiroki ITO, Kanji AKAHORI: An Analysis for the Neural Network Supporting Users' selection of Menu Items,IEICE technical report,（Educational Technology）,Vol.96, No.500, pp.53-60, 1997

[5]Michio MATSUBARA: Evaluation of Global Processing of Students in Electric Circuit, Transactions of Society Japan Science Teaching, Vol.39, No.1, 1998

The Externalization Support System of Self-explanation for Learning Problem-Solving Process

Kazuhide Kanenishi[*], Takahiko Mendori[**], Masafumi Sawamatsu[***] and Yoneo Yano[***]
**Center for University Extension, Tokushima University,*
1-1, Minamijyousanjima, Tokushima, 770-8502 Japan,
mailto: marukin@is.tokushima-u.ac.jp
***Department of Information Systems Engineering, Kochi University of Technology,*
185, Miyanoguchi, Tosayamada-cho, Kochi, 782-8502 Japan,
mailto: mendori@info.kochi-tech.ac.jp
****Department of Information Science and Intelligent Systems, Tokushima University,*
2-1, Minamijyousanjima, Tokushima, 770-8502 Japan,
mailto: {yano, sawa}@is.tokushima-u.ac.jp

When a learner dose various tasks in the computer, the interaction of the learning support system is so a transition which happens inside the learner. At that time, educational effects such as knowledge structuring occurs due to externalization of representation. We developed the prototype system in so that externalization of self-explanation of the problem-solving process was supported. A learner externalizes the self-explanation ‘How do I solve the exercise?”. At this time, she/he explains using not only words but also diagrams, in order to exploit the effect of diagrams. Self-monitoring happens with self-explanation, and the acquisition of a problem solving strategy on is learned. In this paper, we construct a model of ‘externalization on the computer,”and we consider the occurrence of cognitive load. Learning is a kind of load, therefore any reduction of the load (as opposed to its extinction) is of assistance. We propose the presentation of an operation list as a method of load reduction. The memory dependent evaluation becomes more coherent by such a list. In other words, the cognition perspective improves. In this paper, it is shown in the prototype system how externalization is accomplished.

Keywords: **Self-explanation, Reflection, Externalization, HCI**

1 Introduction

Recently, the contents of interaction have been reconsidered in the context of the learning support system[1]. So far, a computer playing a teacher's role is aimed at the transfer of domain knowledge expressed by the system. The computer asks a learner several diagnostic questions. The computer is central to such interaction. The perspective ‘For whom is this interaction?”is absent. In this paper, we propose an learnig environment (externalization support system) that promotes the understanding of problem solving resulting from the externalization in the computer. The learner works independently on the computer with the interaction we propose, then the learner rewrites his internal state. Learning occurs at that time.

Recently, attention has focused on ‘externalization “and ‘meta-cognition.” For example, a vague idea is sometimes clarified by writing text. Externalization promotes the arrangement of knowledge and learning. On the other hand, meta-cognition is a psychic activity of higher order, involving self-monitoring (reflection), and is concerned with deep level learning such as the acquisition of strategies, the transfer of knowledge.

We paid attention to the self-explanation of the problem-solving process, and we have researched that support[2]-[3]. At present, a learning support system that externalizes self-description of the problem-solving process has been utilized as an experiment. Exercises in statistical scales of geography were used. A learner expresses how she/he solved a problem. Geography though is considered mere memorization, the learner can acquire an understanding of the problem-solving process by self-description. An example of this type of exercise is shown in figure 1.

We propose the usage of figure 1 as a method of externalization. We are constructing the environment where a learner can do self-explanation by writing memoranda. In this system a learner draws on the character and diagram of the explanation of the problem-solving process. Furthermore, the examination process of externalization is supported from the cognition perspective. The activities scrutinized are internal (understanding the behavioral reason of the learner) and are supported by the presentation of the operation history. This supports the self-monitoring that is crucial to meta-cognition

Idea support system, idea sketch, etc. are proposed in the HCI researches. However, we think learning involves a kind of load, and our purpose is to recommend support by control of the load, rather than by elimination of the load. The consideration of support by reduction of the load is a different point.

In chapter two, we describe the educational effect of externalization. We propose a support method of externalization targeting self-explanation of the problem-solving process in chapter three. The summary of our system is shown in chapter four. We present a summary in the final chapter.

2 The outline of externalization

2.1 The educational effect of externalization

Many researchers point to the educational effect of externalization[5]-[6]. The effect of diagram use in externalization has been acknowledged as well. The learner can acquire the educational effect if self-description is externalized by use of diagrams[7]-[8].

Externalization is the expression of internal psychic activities (images). We mention clarification of knowledge, structuring, etc. as a general effect. Moreover, internalization occurs by repeated externalization, and internal processing proceeds smoothly.

Self-description involves special explanation of a point[9]. Externalization is unique as well. Self-monitoring is enhanced by a learner's repeated externalization of the self-explanation.

2.2 Externalization model on the computer

We construct a model of externalization in this section; our objective is not to clarify the mechanism of externalization. In a sense, the model is to employ educational effect. Various models of externalization are available; we have elected to choose a model of externalization in the computer.

Externalization is considered to consist of several functional modules. The module described here is the functional unit that is comparatively independent.

The model of externalization and the state of repetition of each module are shown in figure 2. We classify modules of externalization into four ways.

(1) Image generation
(2) Expression form generation
(3) Operation sequence generation
(4) Examination (evaluation)

Qucstion: What are items(1),(2),(3)?

Item	Ranking	1	2	3	4
(1)	Exporting country Amount of export	U.S.A 9,637	Australia 5,300	Poland 2,800	U.S.S.R. 2,000
	Importing country Amount of import	Japan 7,909	France 2,295	Italy 1,905	Canada 1,567
(2)	Exporting country Amount of export	Cuba 675	France 300	Australia 255	Brazil 253
	Importing country Amount of import	U.S.S.R. 672	U.S.A. 270	China 222	Japan 180
(3)	Exporting country Amount of export	Japan 1,612	West Germany 1,458	France 675	Belgium* 460
	Importing country Amount of import	U.S.A. 1,802	England 490	West Germany 483	Italy 460

(1)and(2)unit is 10,000 tons. (3)unit is 10,000,000 dollars. *:Includes Luxembourg.
(1):1982, (2):1983, (3)1980

Fig. 1 Example of geographical exercise.

Module (1) is the creative impulse that forms an internal image. Externalization consists not only of the creative impulse of expression but also of the underlying representation. Though module (1) is a heterogeneous activity, it is a part of externalization. Creativity is a very complex psychic activity, and is beyond the scope of this paper.

Module (2) is the expression of vague internal images. Expression is based on the rule of generation. As a case in point, form of presentation (rule) that has been configured freely and formalization that has already been completed, may be employed. For example, in the case of pictures, expression is free, and a person who draws a picture decides the form of presentation. On the other hand, when we illustrate a phenomenon with a formula, formalization is predetermined, and we must obey mathematical rules. As for externalization of writing memoranda, existing formalization and independent formalization are being used together. We can think of module (2) as consisting of the following three usage forms.

(a) Existing formalization
(b) Independent formalization
(c) Existing formalization + Independent formalization

The burden of usage of the existing formalization is that a learner must understands formalization. However, when a learner acquires existing forms, internalization progresses, and the representation in module (1) becomes simplified. For example, when a learner is skilled in the use of the Japanese abacus, she/he becomes capable of mental arithmetic using a mental image of the abacus. The effect of the Venn diagram in the understanding of the set theory is similar.

When externalization is done on the computer, the process of module (3) is remarkable. The rate of this part increases when the expression is done indirectly using the computer. Expression can't be generated if the computer lacks the appropriate software. Thus, a learner plans an operation sequence to configure expression, and she/he will move the mouse based on that plan.

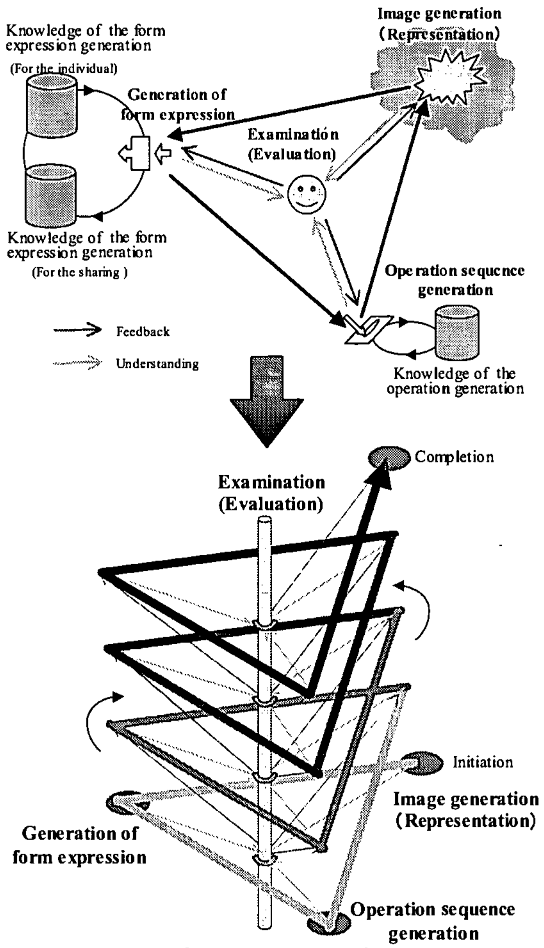

Knowledge of the form expression generation (For the individual)
Generation of form expression
Image generation (Representation)
Examination (Evaluation)
Knowledge of the form expression generation (For the sharing)
Operation sequence generation
Feedback
Understanding
Knowledge of the operation generation
Completion
Examination (Evaluation)
Initiation
Image generation (Representation)
Generation of form expression
Operation sequence generation

Fig. 2 Model of externalization on the computer.

Module (4) is different from the other modules. Examination is the evaluation of each process from module (1) to (3) with feedback. In other words, examination is a meta-level activity when compared with the other processes.

Modules (1) to (3) become a cycle. The processes from (1) to (3) are evaluated by process (4), which provides feedback. This cycle is repeated until a learner judges by examination. That the activity has been completed

3 Externalization support of self-explanation

3.1 The support of externalization by load reduction

The general support method of externalization is considered in this section. First, we remove the cognitive load intuitively for externalization support. However, the purpose of learning is to put load on the learner. Even if a computer estimates the intention of the learner (even if an explanation is formed automatically,) learning does not progress. Hence, removal of the load does not support learning.

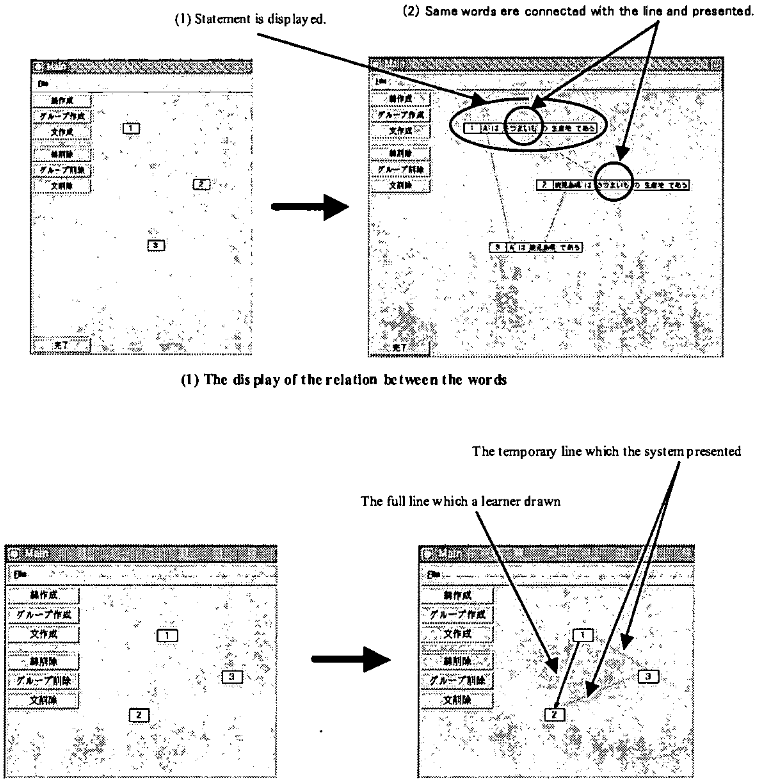
The real nature of the load lies in the multiple combinations of the loads. The load is classified by cause and category, it is necessary to separate the load that aids learning from the load that does not. We aim to reduce the load. In other words, consideration of the relevant control of the load is necessary. Arrangement and classification of each load is necessary for its support.

3.3 Drawing method of the problem-solving process

Support of the expression form generation process serves to prepare for the effective expression method. We show (by way of drawing) the problem-solving process following. Problem-solving is a process, and it has a certain structure. Diagram usage is effective to express structured information. Graph expression is a relevant method like Tweedier's indication as an informational expression that has a flowing structure[10]. For example, though various methods are proposed, a flow chart is still used for the expression of a program structure.

We use a style that combines the use of words and figures. We propose the arrow diagram that expressed structure between statements was symbolically used with the text of the items together. The externalization task with our learning environment consists of two tasks: text creation and graph drawing. Expression is constructed in the following procedures:

(1) The Entering of a simple sentence
+
(2) Placement (migration)
(3) Connection
(4) Grouping
(5) Attribute addition to the object


(1) Statement is displayed.
(2) Same words are connected with the line and presented.
The temporary line which the system presented
The full line which a learner drawn

(1) The display of the relation between the words

(2) The display of the temporary line which can be connected

Fig. 3 Model of operation support.

First, a learner constructs the explanation of the character form. The explanation of the solution reflects the problem-solving process of the exercise. We exclude compound sentences and complex sentences, and use only simple sentences. Each simple sentence during expresses each state of the problem-solving process. A learner is conscious of the order, and notices that problem-solving proceeds by the items. The order of the explanation copes with the process of the problem-solving.

Next, each simple sentence is connected with an arrow line to form a "node". A learner is specifically conscious of the structure by drawing these lines. As for the structural expression of the explanation, each statement is associated with the arrow line toward the explanation of the expressed goal from the explanation of the initial state. A learner encloses some statement, and gives color attribute as a supplementary activity. The diagram drawn with the arrow line is completed by repeating the above activity.

The part that decides expression is first and the part in which a learner himself can determine expression are both present in the above expression method. Such a method guarantees freedom of expression for the learner, while at the same time accepting ambiguity. Thus a learner can write a memorandum of the meaning of the attribute appended by the learner. For example, the system asks a learner for a reason when a learner changes the color of the line. This function is actualized as a fraction of the support of the look-over.

3.4 The support of the operation sequence generation

We describe the support of the operation sequence generation in this section. The following function is provided because the learning environment should be made convenient.

(1) The simplification of the operation
(2) The intersection of the operation and the phenomenon

The operations of the learning environment are statement creation, statement delete, line drawing, line erasing, movement, grouping, and the color alteration of the object. It prepares only for easy operations. A phenomenon can be easily imagined from these identifiers. We achieve a single function in our system. An operation and a phenomenon correspond one-to-one, and the understanding of the operation becomes easy.

The learning environment is tailored to notify the learner of deficient explanations and feasibility of the expression by the following operation support. The state of the support is shown in figure 3.

(a) The display of the related word information
(b) The display of drawing line feasibility

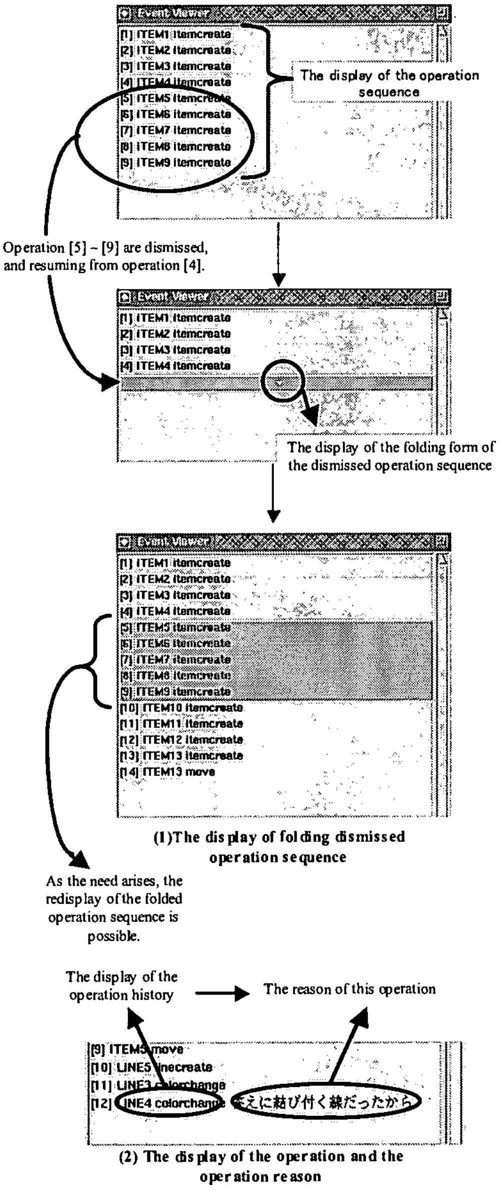
A simple sentence is displayed on the learning environment as a symbolic icon. One component is made according to statement, a learner can focus on structure between the statements. On the other hand, a certain word sometimes has significant meaning for structural grasping of the explanation. The learning environment manages word information, and employs it for support.

The indication of related word information This is the method in which the structural understanding of the statement is accelerated. When the same word in is used several statements, the learning environment shows them. When the same word is shown repeatedly, the system is made conspicuous. The system displays a statement next to the statement icon, and gives color to that word. The same word in different statements is connected by a line.

The indication of connection feasibility This is the method in which drawing between the statements is supported. The system presents the link that can be connected in the statement nodes as a temporary line. The system estimates the statement that relate to other statements in the placement step (which the statement icon on the workspace finished). The feasibility a line drawn is high in the statement which satisfies the following conditions:

(A) The statement operated just before
(B) The statement that encompasses the same word

The system presents a temporary line to the learner according to the stage at which placement was finished. All candidates are displayed when some lines are presumed. When a learner chooses a temporary line, the system re-establishes that temporary line as a permanent line. Because drawing lines is a possible option, the work of drawing is reduced for the learner.


Event Viewer
The display of the operation sequence
Operation [5] ~ [9] are dismissed, and resuming from operation [4].
Event Viewer
The display of the folding form of the dismissed operation sequence
Event Viewer
As the need arises, the redisplay of the folded operation sequence is possible.
The display of the operation history
The reason of this operation

```
[1] ITEM1 Itemcreate
[2] ITEM2 Itemcreate
[3] ITEM3 Itemcreate
[4] ITEM4 Itemcreate
[5] ITEM5 Itemcreate
[6] ITEM6 Itemcreate
[7] ITEM7 Itemcreate
[8] ITEM8 Itemcreate
[9] ITEM9 Itemcreate
```

```
[1] ITEM1 Itemcreate
[2] ITEM2 Itemcreate
[3] ITEM3 Itemcreate
[4] ITEM4 Itemcreate
```

```
[1] ITEM1 itemcreate
[2] ITEM2 Itemcreate
[3] ITEM3 Itemcreate
[4] ITEM4 Itemcreate
[5] ITEM5 itemcreate
[6] ITEM6 Itemcreate
[7] ITEM7 itemcreate
[8] ITEM8 Itemcreate
[9] ITEM9 itemcreate
[10] ITEM10 Itemcreate
[11] ITEM11 Itemcreate
[12] ITEM12 Itemcreate
[13] ITEM13 Itemcreate
[14] ITEM13 move
```

(1)The display of folding dismissed operation sequence

```
[9] ITEM5 move
[10] LINE5 linecreate
[11] LINE3 colorchange
[12] LINE4 colorchange 考えに結び付く線だったから
```

(2) The display of the operation and the operation reason

Fig. 4 Representation of the operation history list.

3.5 The support of the examination by cognitive perspective

We aim at the realization of the farsightedness of the cognitive perspective with the support of examination. Externalization is done so that a learner may learn about himself. We aren't aiming at deputy by the system. The approach of automating conception and drawing isn't embraced. Support toward examination of evaluating one's act is necessary. Therefore, we propose a method that assists the self-monitoring by the learner. The system uses the following two methods, as shown in figure 4.

(1) The display of the operation history
(2) The collection and display of the operation reason

Reflection on the personal task must depend on the current aspect of the activity subject and on memory. However, subject activity does not express variations in the middle of the task. Moreover, memory is often a temporal effect, and often can't be extracted when necessary. Therefore the system preserves task history, and history and task process are reproduced to the learner. The system makes linear operation sequences and a learner can do an operation again from the arbitrary juncture of the operation sequence that is presented. In that case, a recent operation sequence occurs from the point that an operation was done again, and a previous operation sequence is dismissed.

However, when a learner reflects on an error, dismissed operational sequences encompass significant information. Because reflection support is critical, the system preserves all operation sequences. When a recent operation sequence occurs, the previous operation sequence is hidden temporarily from the learner. Then, if a learner requires, the previous operation sequence can be displayed. Cognitive perspective in the examination process improves due to the presentation of the operation history.

The display of the operation history provides an opportunity to look back at the operation. Furthermore, we not only present operation history but also present the reason of the operation. The system requires comment input at every operation. The set list of the operation and the reason are stored in the system. It is understood that comment is useful in order to reconsider program source. Ambiguity is present in expression, and the degree of freedom of expression is guaranteed. A learner himself sometimes forgets the intention of the figure on one side. A memorandum of the operation is useful in such a case as well. The presentation (collection) of the memorandum of the operation is not the active intervention of the system.

4 Self-explanation externalization support system

4.1 The configuration of the self-explanation externalization supporting system

Our system before this paper externalized a self-explanation only in character[3]-[4]. However, figures and text are mixed in the natural externalization. Drawing activity is separated from text creation activity in self-explanation, and it is unnatural to draw after text is written. Therefore our system was designed to enter text and to draw simultaneously.

We implemented three functions in the system.

(1) Explanation management

(2) Explanation structural management (visualization management)

(3) Operational history management

Module (1) manages information on the explanation sentence. This module is shared the entry of the explanation sentence and the display of the explanation sentence. A learner enters a simple sentence. The morphological analysis of each simple sentence is done by "Cyasen" developed with NAIST. Information on the word is extracted. By this processing, information on the noun and verb in a simple sentence is extracted. The system preserves the information with a simple sentence.

The order of the statement input can't be employed as an order of the explanation sentence. When a learner completes a drawing, the system decides the order of the explanation sentence based on the following information:

(a) Related to the arrow

(b) Grouping

(c) Related to the place

(d) Input order

First, the system gives priority to arrow line information. The beginning point and end point of the arrow shows a context. Next, simple sentences are grouped to the same level. The system fundamentally introduces the order of the input as the order of the explanation. When there is no grouping or arrows in the explanation figure, the system decides the order of the explanation sentence based on the co-ordinate information of the icon on the "canvas "screen. If there is a top-to-bottom relationship among icons, this relationship becomes the context. The order of input is used except in the case above. After the system shows the order, the alteration of the order by the learner is possible.

Module (2) manages the drawing task and the information acquired from this task. A learner enters a simple sentence, and next drawing becomes possible. Module (2) manages the whole drawing task on the "canvas", and it displays support information.

Module (3) presents the task history (operational history) of the learner. .A learner sees this operation history, and adds various modifications to the externalized figure.

Various methods of presentation of the operation history are proposed. We think the list form is easy to understand. Various methods of Undo (Redo) are proposed as well[11]. We consider an operation to be a series of persistent sequences. We introduce the interface in which an operation can be done from an arbitrary part of the operation history list and the previous sequence folded under the recent sequence. The current sequence is important for the learner, so a dismissed sequence is rendered temporarily invisible. Hidden operation history can unfold the folded part if necessary (the icon indicates it has been folded.), but a learner only confirms a folded operation sequence, and a dismissed sequence can't be redone midway in the process.

4.2 Outlook of the system

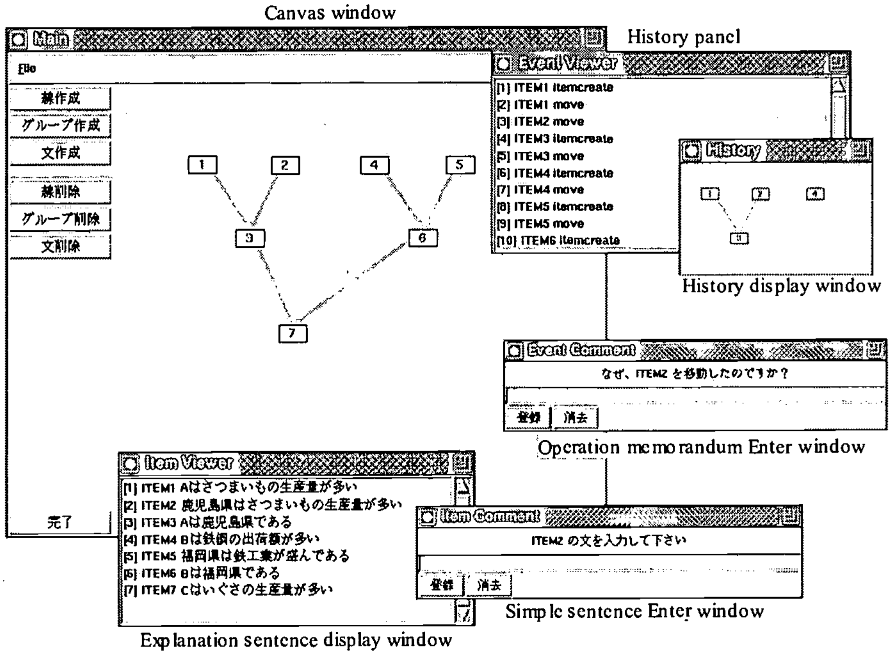
We show how self-explanation is externalized on the system in this section.Six windows are displayed in the system that was manufactured. The screen configuration of the system is shown in figure 5.

(1) Canvas window (Operation button is encompassed.)
(2) Simple sentence input window
(3) Explanation sentence display window
(4) History panel
(5) Operation memorandum input window
(6) History display window

A learner does a drawing task on the canvas window (1), and then the drawing consequence is displayed. The learner starts an explanation through this window, and the explanation is written and modified. Various operation buttons are configured. This window performs the role of the console panel of the whole system.

The simple sentence input window (2) is a one-line editor for a learner to input a simple sentence. This window is invoked by the sentence-creation button on the canvas window, and closes when the input is finished.

The explanation sentence display window (3) displays an entered simple sentence with the items. Each sentence is displayed on this window by an entered order. When a learner estimates that externalization has been terminated (the stage in which the "completion" button is pushed), the order is evaluated, and the line of the explanation sentence is replaced. However, a learner can alter the line of the explanation sentence by using the mouse to "drag & drop".



[1] ITEM1 itemcreate
[2] ITEM1 move
[3] ITEM2 move
[4] ITEM3 itemcreate
[5] ITEM3 move
[6] ITEM4 itemcreate
[7] ITEM4 move
[8] ITEM5 itemcreate
[9] ITEM5 move
[10] ITEM6 itemcreate

なぜ、ITEM2 を移動したのですか？

[1] ITEM1 Aはさつまいもの生産量が多い
[2] ITEM2 鹿児島県はさつまいもの生産量が多い
[3] ITEM3 Aは鹿児島県である
[4] ITEM4 Bは鉄鋼の出荷額が多い
[5] ITEM5 福岡県は鉄工業が盛んである
[6] ITEM6 Bは福岡県である
[7] ITEM7 Cはいぐさの生産量が多い

ITEM2 の文を入力して下さい

Fig. 5 Screen shot of the prototype system.

The history panel (4), the operation memorandum input window (5) and the history display window (6) are related mutually. The operation history of the learner and memoranda are displayed in the history panel. Though the memorandum input window is displayed at every operation, a learner doesn't necessarily need to enter memoranda. Entered comments are displayed in the history panel with the associated operation. When a learner chooses an operation from the operation history, the history display window displays the screen image when that operation is done. It is important to simultaneously present the screen image to support cognitive perspective. Moreover, if a learner chooses an operation from the operation history, she/he can redo the operation from there.

5 Conclusions

In this paper, we described a learning system in which externalization of self-explanation of the problem-solution process was supported. Recent attention is founded in the educational effect of externalization; however, the mechanism of externalization isn't clear, and the recommendation of the usage method isn't sufficient, either.

Therefore, we surveyed externalization first, and described the educational effect. Next, we considered the support method of externalization, and proposed the support method of externalization of self-explanation. The first is the method of externalization of self-explanation that employs expression by words and diagrams, such as memoranda. The other method is that of collecting the operation history and presenting it to the learner from the viewpoint in which the cognitive perspective is important for the examination process of externalization. Furthermore, we attempted a system that could leave operation reason when the operation history was collected. Self-monitoring becomes smooth by presenting operation and reason. Finally, an overview of our trial production system was shown. The state in which a learner externalized in that system was shown.

In the future, we will improve the system, targeting each operation to reduce user load, and we will evaluate the system.

Acknowledgment

This research was supported in party by the Grand-in-Aid for Science Research (A) No. 11780131 from the Ministry of Education, Science, Sports and Culture in Japan.

References

[1]Norman, D. A. :Turn Signals Are the Facial Expressions of Automobiles, Addison-Wesley, (1992).
[2]Metcalfe, J. and Shimamura, A. P. :Metacognition, MIT Press, Massachusetts, (1996).
[3]Kanenishi, K. and Yano, Y.: Guided Understanding for Problem Solving Process Using the Refining Self Explanation, Proceeding of HCI95(Vol.2), pp. 97-102 (1995).
[4]Kanenishi, K. and Yano, Y.: Development Self-Explanation Learning Environment Using the Explanation Refinement, Proceeding of IEEE SMC96(Vol.1), pp. 568-573, (1996).
[5]Stenning, K. and Oberlander, J. :A Cognitive Theory of Graphical and Linguistic Reasoning: Loginc and Implementaion, Cognitive Science, Vol. 19, pp. 97-140, (1995).
[6]Zhang, J. :The Nature of External Representations in Problem Solving, Cognitive Science, Vol.21, No.2, pp.179-217, (1997).
[7]Larkin, J. H. and Simon, H. A. :Why a Diagram is (Sometimes) Worth Ten Thousand Words, Cognitive Science, Vol.11, pp.65-100, (1987).
[8]Marcus, N., Cooper, M. and Sweller J. :Understanding Instruction, Journal of Educational Psychology, Vol.88, pp.49-63, (1996).
[9]Chi, M.T.H., Bassok, M., Lewis, M. W., Reimann, P. and Glaser, R. : Self-Explanations: How students study and use examples in learning to solve problem, Cognitive Science, Vol.13, pp.145-182, (1989).
[10]Tweedie, L.: Characterizing Interactive Externalizations, Proceeding of CHI'97 (ACM Conference on Human Factors in Computing Systems), pp. 375-382, (1997).
[11]Myers, B. A. et al.: The Amulet Environment: New Models for Effective User Interface Software Development, IEEE Transaction on Software Engineering, Vol. 23, No. 6, pp.347-365,(1997).

Traversing the Case Graphs A Computer Model for Developing Case-based Learning Systems

Isaac P.W. Fung & R. H. Kemp
Institute of Information Sciences and Technology
Massey University
Palmerston North, New Zealand
Fax: +64-6-3502259
P.W.Fung@massey.ac.nz; R.Kemp@massey.ac.nz

This paper presents an extended theory for representing cases in a case-based physics learning environment. There are two issues with which developers of case-based tutoring systems often contend: one is assessing and retrieving similar cases from the case library; the second one is delivering the case contents to the students. Whilst an earlier paper has addressed the former issue, this paper focuses on the latter by defining a computational mechanism that is used for delivering the case content. The mechanism is developed by defining a procedural semantics on the case graph which incorporates the dynamic modelling capability of petri nets. A case is initially opaque to the student. During case interaction, however, it will be made transparent gradually by engaging the students with problem-solving activities. The activities are modelled using the notions of marking places and firing transitions, where places and transitions represent case variables and operations, respectively. The idea is illustrated with an example of providing guidance to students solving problems in the domain of Newtonian mechanics.

Keywords: **Artificial Intelligence, Conceptual Graphs, Intelligent Tutoring Systems, Case-based Reasoning**

1 Introduction

This paper presents an extended theory of representing problem-solving cases proposed in [5] for the purpose of modelling instructional activities between the cases and the learners within the context of case-based tutoring systems (CBTS) [11]. In response to the classic criticisms [12] leveled at the first-generation of computer-assisted learning software that frequently have to go back to inflexible, pre-compiled problem solutions, CBTS is very attractive for several reasons. Two of them are particularly appealing to us. From an instructional perspective, students are highly influenced by past examples (i.e. real cases) to guide their problem-solving activities [1] or completing cognitive tasks [8]. Our project sponsor demands that the final system should faithfully reflect what students actually do when completing their homework. It is, therefore, our aim to ground our system design at the outset on sound psychological findings about pupils' learning behaviours. Secondly, from a technical viewpoint, case-based adaptation techniques are powerful in adapting interface components to the user's need [14].

Individual learner's needs, style and progress do differ substantially. Case-based reasoning technology [7] endows the system with the capability of inferring what is considered 'best' for the students by referring to their past learning histories. [5] proposed the use of conceptual graphs (CG) [13] for representing tutorial cases. While this method elegantly tackles the issue of assessing case similarity, how the case graphs are built remains a 'black-box'. The case users have no way to inspect the internal processes for constructing the graph. To ensure the cases are useful in tutorial contexts, the knowledge components of the cases need to be 'available' to the students. What we mean by 'available' is making the case solution transparent, i.e. the system is capable of justifying each problem-solving step being shown to the students in terms of the underlying physical principles.

The procedural semantics defined on case graphs which forms the core contents of this paper, provides a way of making the solution procedures explicit to the students. The idea is to synthesize a CG and the actor graph defined in [13] into one single global graph instead of treating them separately. The resulting structure is a tripartite graph that has three types of nodes: concept nodes, symbolic relation nodes and mathematical relation nodes. The mathematical relation nodes are for handling mathematical calculations in the domain of Newtonian mechanics, the targeted subject domain of our project These calculations are important in many science and engineering applications. In making the synthesis, two important ontological commitments were made. Firstly, human cognitive functions in studying a concrete case are viewed as a process of constructing graphs. Relevant concept nodes are created and linked to each other via some appropriate relation nodes (whether symbolic or mathematical). A case represented by the graph consists of sets of concept nodes and relation nodes, but to what extent the students understand the case contents remains unknown until some observable actions are seen. Secondly, the process of building the graph is based on the notion of concept node marking. Initially, the sets of nodes in a case are all opaque to the users because they are not yet marked. The set of nodes representing the initially given physical quantities are marked first. Each problem-solving step is viewed as generation of new graph nodes, but they are implemented as the nodes states change from unmarked to marked. To mark a set of nodes, the mathematical relation nodes (or operators) which link the marked and the unmarked nodes have to be fired. The procedures of solving the problem are defined as the firing sequence for marking the target concept nodes. The subgraph associated with a particular fired node represents the semantics of the knowledge behind its firing.

2 Formal Definition of the Case Constituents

We represent a typical case abstractly by a directed graph which is composed of

∗ Three disjoint sets of vertices C, R *and* R_m (i.e. $C \cap R = \emptyset$; $C \cap R_m = \emptyset$; $R \cap R_m = \emptyset$ and $C \cap R \cap R_m = \emptyset$) where C represents the set of concept nodes; R represents the set of symbolic relation nodes; and R_m represents the set of mathematical relation nodes.

∗ A set of directed arcs E such that $E \subseteq (C \times R) \cup (R \times C)$. Each arc $e \in E$ connects a concept $c \in C$ to a symbolic relation $r \in R$ or vice versa.

∗ A set of directed arcs E_m such that $E_m \subseteq (C \times R_m) \cup (R_m \times C)$. Each arc $e_m \in E_m$ connects a concept $c \in C$ to a mathematical relation $r_m \in R_m$ or vice versa.

Shown in Figure 1 is an example case graph where

$C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$;
$R = \{r_1, r_2, r_3, r_4\}$;
$R_m = \{r_{m1}, r_{m2}, r_{m3}, r_{m4}\}$;
$E = \{(c_1, r_1), (r_1, c_2), (c_3, r_2), (c_4, r_2), (r_2, c_2), (c_5, r_4), (r_4, c_2), (c_6, r_3), (r_3, c_1)\}$; and
$E_m = \{(c_1, r_{m1}), (r_{m1}, c_2), (c_2, r_{m2}), (c_2, r_{m3}), (c_5, r_{m3}), (r_{m2}, c_3), (r_{m3}, c_4), (c_3, r_{m4}), (c_4, r_{m4}), (r_{m4}, c_6)\}$.

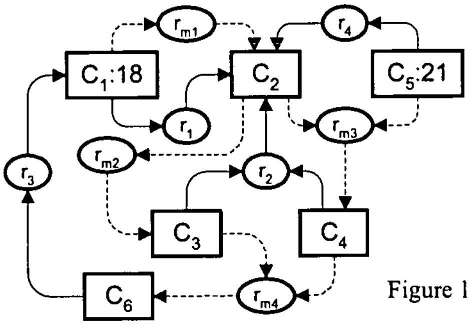


Figure 1

∗ For every $r_m \in R_m$, there exist an input set $I_r(r_m)$ and an output set $O_r(r_m)$ such that

$I_r(r_m) = \{ c \in C \mid (c, r_m) \in E_m\}$; (c, r_m) is called the input arc of r_m, and c is called the input concept of r_m; and $O_r(r_m) = \{ c \in C \mid (r_m, c) \in E_m\}$; (r_m, c) is called the output arc of r_m, and c is called the output concept of r_m.

For example, the input/output set of the node r_{m3} in Figure 1 are $I_r(r_{m3}) = \{c_2, c_5\}$ and $O_r(r_{m3}) = \{c_4\}$ respectively.

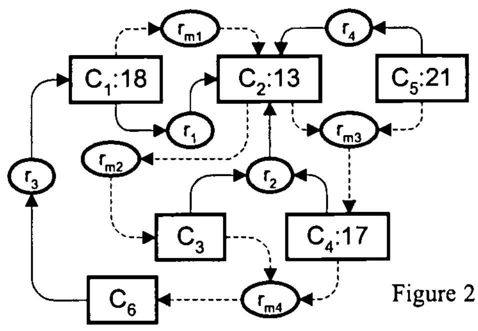


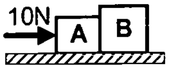
Figure 2

- ∗ For every $c \in C$, it is defined as *marked* if it is being instantiated to a specific individual. In Figure 1, c_1 and c_5 are marked whereas the others are non-marked.
- ∗ The marking μ of a graph G can be represented by a *n-vector*: $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, where each $\mu_i \in (T, F)$. For example, the graph in Figure 1 has the marking $\mu = (T, F, F, F, T, F)$.
- ∗ A mathematical relation node $r_m \in R_m$ is *enabled* whenever each concept $c \in I_r(r_m)$ is marked. In Figure 1, only r_{m1} is enabled at that marking.
- ∗ When a mathematical relation node is enabled, it can be fired at any time and every time a mathematical relation is fired, every $c \in O_r(r_m)$ will be marked[1].
- ∗ For every $c \in O_r(r'_m)$, where r'_m is the fired mathematical relation, the content of c is evaluated according to the formulas inscripted in the respective $r'_m \in I_c(c)$.
- ∗ Supposing the formulas inscripted in r_{m1} is $c_1 = c_2 + 5$ and r_{m3} is $(c_2 + c_5) / 2$, the firing of r_{m1} will mark c_2 which enables r_{m3} because c_5 has already been marked. If r_{m3} is fired later, a new marking (shown in Figure 2) will be formed and become $\mu = (T, T, F, T, T, F)$.

3 Representing Mechanics Problem-solving Cases

In our application domain, Newtonian mechanics, two categories of physical entities are identified with respect to the cases we use for tutoring: *physical objects* and *physics concepts*. Both are represented, however, as rectangular-shaped concept nodes. In each case, a number of physical objects are involved, such as a block, a car, a plane, a spring, etc., but they are normally described abstractly just as a physical object. Various meaningful relations obtain between the objects, which essentially represent the physical configuration between them. For instance, it makes sense to represent the 'rest_on' relation that holds between a block and a plane whenever the block is on the plane. Other meaningful relationships are: 'above', 'contact_with', 'moves_on', and so forth. There are attributes, intrinsic and motion-related, of the physical objects which refer to one object only. For example, 'acceleration'(a motion-related attribute) and 'mass' (an intrinsic attribute) applies to a single physical object on its own. In representing a physical situation, there are some other domain-related ideas such as external force or friction, which characterize the case being described. All these concepts are categorized as physics concepts as they are used to describe the state of the world depicted by the case. Figure 3 shows a typical case adopted from a standard physics textbook.

Two blocks A & B are resting on a frictionless horizontal plane as shown. If an external force of 10N is acting on A, what is the acceleration of the blocks and the force of contact between them? (The masses of A and B are 3kg and 7kg respectively).



[1] The notion of marking and firing is borrowed from the petri nets formalism [9]

Solution: Apply Newton's 2nd Law on A&B

$$\begin{aligned} \text{Net Force}_{A\&B} &= \text{Mass}_{A\&B} \times \text{Acceleration}_{A\&B} \\ \text{External Force}_{A\&B} &= \text{Mass}_{A\&B} \times \text{Acceleration}_{A\&B} \\ 10 &= (3+7)\ \text{Acceleration}_{A\&B} \\ \text{Acceleration}_{A\&B} &= 1\ \text{m/s}^2 \end{aligned}$$

Apply Newton's 2nd Law on A

$$\begin{aligned} \text{Net Force}_{A} &= \text{Mass}_{A} \times \text{Acceleration}_{A} \\ \text{External Force}_{A} + \text{Contact Force}_{A} &= \text{Mass}_{A} \times \text{Acceleration}_{A} \\ 10 + \text{Contact Force}_{A} &= 3 \times 1 \\ \text{Contact Force}_{A} &= -7\text{N} \end{aligned}$$

Apply Newton's 2nd Law on B

$$\begin{aligned} \text{Net Force}_{B} &= \text{Mass}_{B} \times \text{Acceleration}_{B} \\ \text{Contact Force}_{B} &= \text{Mass}_{B} \times \text{Acceleration}_{B} \\ \text{Contact Force}_{B} &= 7 \times 1 \\ \text{Contact Force}_{B} &= 7\ \text{N} \end{aligned}$$

Figure 3: A typical Newtonian mechanics case and its solution

As the complete graph representing the case occupies too much space, the whole graph is divided into several subgraphs. To illustrate the idea, three representative subgraphs are shown in Figure 4, 5 and 6. The subgraph in Figure 4 represents the physical objects involved in the case and their relationships. The (component) nodes encodes the part-whole relationship between the whole system A&B and its constituents A and B. The tuple [Blocks: A&B] → (component) → [Block: B] depicts the block labelled as 'B' as part of the whole system labelled as 'A&B'. The other relation nodes essentially represent the spatial relationships between the objects.

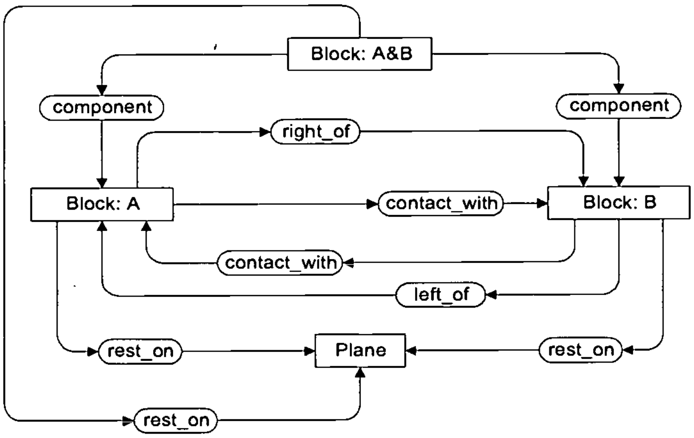


Figure 4: The subgraph showing the physical objects involved in the case and their relationship

The subgraph shown in Figure 5 concerns the attributes, both intrinsic and motion-relationed, of block A, and other relevant physical concepts centred around it. The absurd type [**T**] as the agent of the Net_Force_A and External_Force_A indicates it is something that is of no relevance to us. In Figure 6, those concept types that participate in some sort of mathematical relations are shown. Note that most of the arcs in Figure 6 are dotted indicating they are different from the usual symbolic relations.

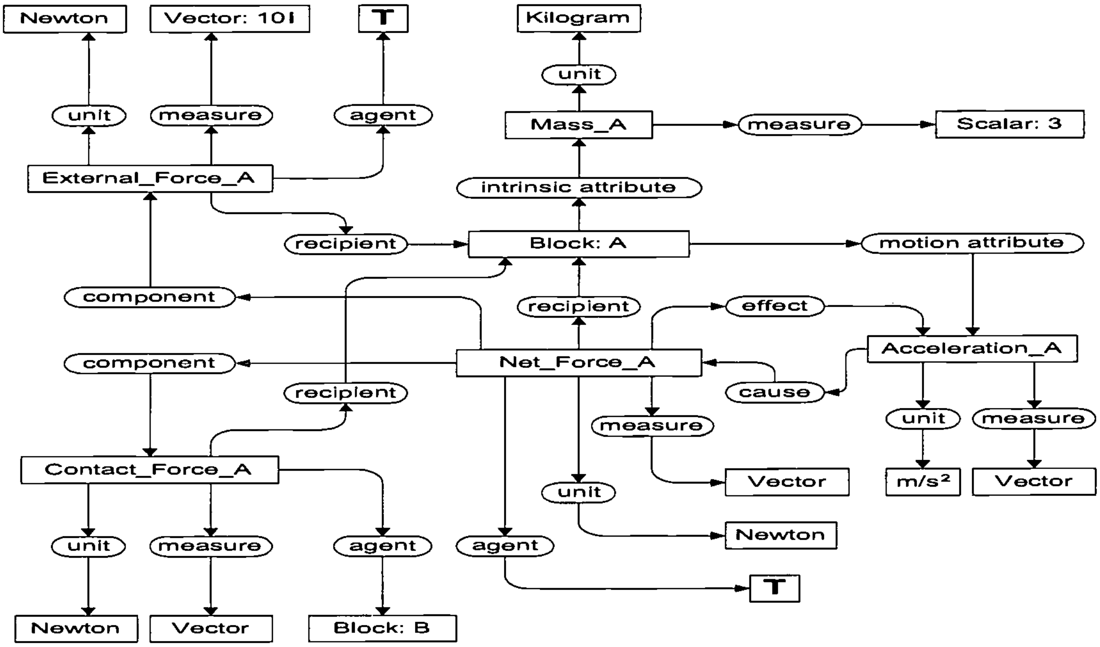


Figure 5: Subgraph showing the attributes of block A and other relevant physical concepts.

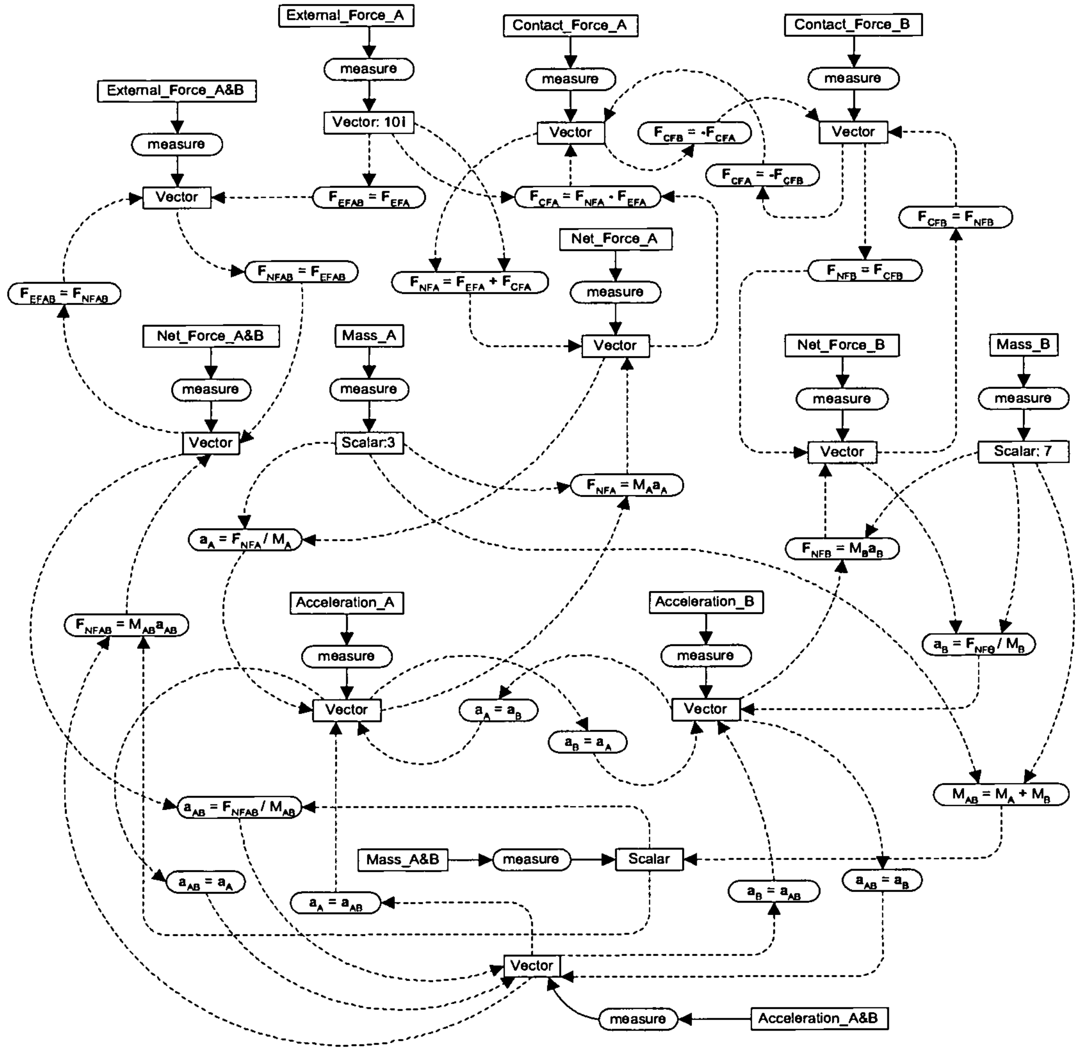


Figure 6: Subgraph showing the mathematical relationships between the relevant concept types

4 Modelling Variables Instantiation as Node Marking

Once a case has been encoded with the formalism, problem-solving activities can be modelled. When given a problem to tackle, the students will generally be asked for a new value from a set of given data. This is modelled as marking the concept nodes such as C_1 and C_5 in Figure 1. The goal is to get the concept node C_6 marked. At the initial marking, only r_{m1} is enabled and therefore any attempt to trigger other mathematical operations is not allowed and, thereby, invites tutorial intervention. The whole process of creating successive markings can be illustrated with a search tree (see Figure 7). The tree

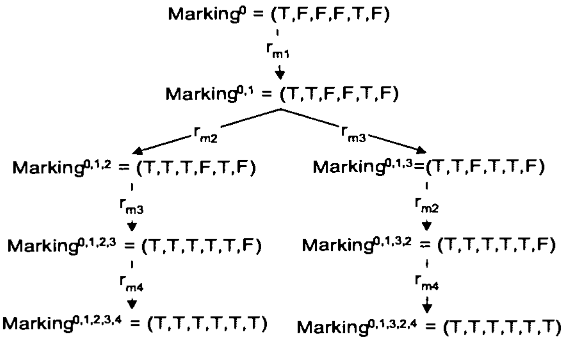

$Marking^0 = (T,F,F,F,T,F)$
r_{m1}
$Marking^{0,1} = (T,T,F,F,T,F)$
r_{m2}
r_{m3}
$Marking^{0,1,2} = (T,T,T,F,T,F)$
$Marking^{0,1,3}=(T,T,F,T,T,F)$
r_{m3}
r_{m2}
$Marking^{0,1,2,3} = (T,T,T,T,T,F)$
$Marking^{0,1,3,2} = (T,T,T,T,T,F)$
r_{m4}
r_{m4}
$Marking^{0,1,2,3,4} = (T,T,T,T,T,T)$
$Marking^{0,1,3,2,4} = (T,T,T,T,T,T)$

Figure 7

indicates the student can gain access to a large solution space for him/her to explore but in the mean time the tutor can keep track of what can/cannot be done.

5 CLASP: A Case-based Learning Assistant System in Physics

A system called CLASP, has been developed to implement the idea. At the current stage of development, two types of activities associated with examples have been identified: providing solutions for studying, and exercises with answers; hence the modes of interaction in the CLASP prototype are also designed around these two themes. When the users issue a request (in terms of the problem description of their own problems) the system will search through its whole case library and provide them cases which match their request. The style of presenting the case will follow the user's wishes, but only two modes of interaction (solution studying and guided-problem-solving) are available. This is to reflect the common way of using examples in physics textbooks. In the study mode, the system presents the whole case (i.e. both the problem and solution statements) for the user to study. This looks like an electronic reference book and the student may browse through the relevant cases. In the guided-problem-solving mode, the system only presents the problem situation to the users, but appropriate system guidance will be provided in solving the problems. The schematic architecture of CLASP is shown in Figure 8. The students interact with the system with the support of the back-end knowledge base.

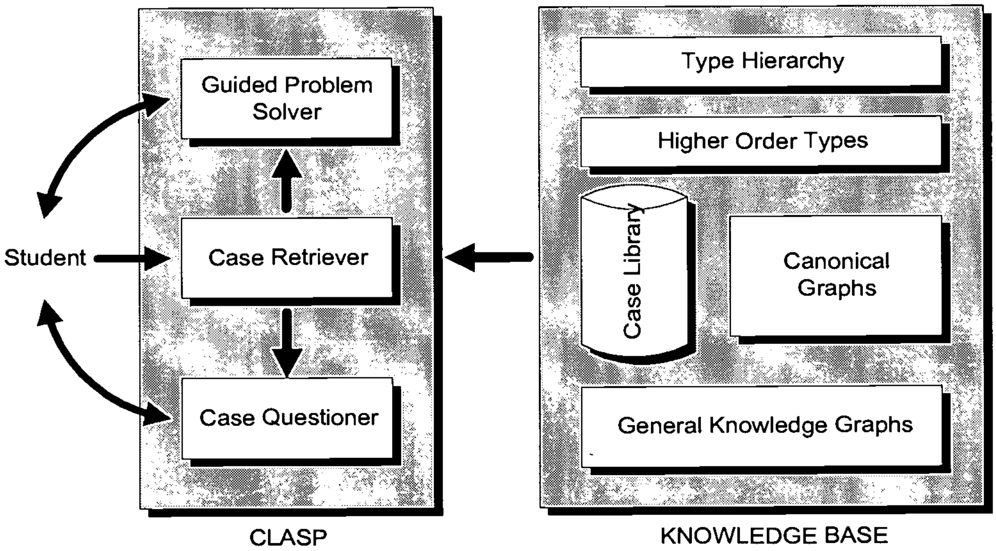


Figure 8: Schematic description of the CLASP architecture

Problem-solving in CLASP is modelled as a graph search. When a problem situation, such as the one shown in Figure 3, is encountered, the initial data are represented as concept nodes being instantiated to specific values and they are displayed to students on the working pad (Figure 9). Now the problem-solver can start tackling the problem by searching through the graph and seeing what additional information can be inferred from the initial given data. For the system to perform the tasks, the expertise has already been encoded in the case graphs, therefore the next step to be taken is searching the graph to find out which operators can be fired. The inferred steps may be unfolded or kept hidden for a while as a hint to advise the student. The intelligence of the system's problem-solving ability comes from its inference engine, being implemented by different graph search methods.

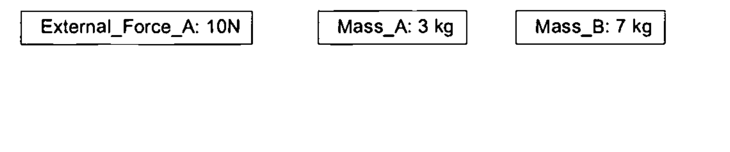
Problem Space	System's Comments:
	Initial Conditions: M_A = 3kg; M_B = 7kg; and External Force $_A$ = 10N.

Figure 9: The working pad and the corresponding system responds

The explanatory capability of the system comes from the matching of the input-operator-output nodes with the consequences of the general knowledge graphs. Whenever an operator is fired, the associated nodes will be matched against the consequences of the general knowledge graphs. If one is found, and it should be, then that particular graph will be tagged. If the student requests a justification of the step taken, the system can explain the graph in general terms. For example, the firing of an algebraic summation operator on the values of masses of two physical objects will match the consequence of the general knowledge graph in Figure 8 so the whole graph can be retrieved for explanation (Figure 10). The working pad, showing the problem space, and the explanation combinations supply the integration of what and why the step happened and the whole process becomes transparent to the student.

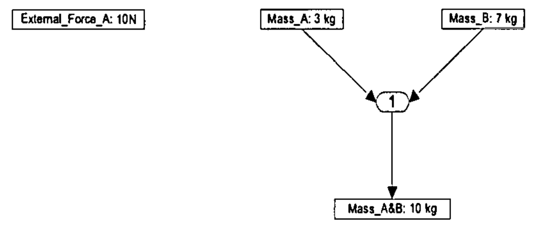
Problem Space	System's Comments:
	Step 1: For a system comprising two components, the mass of the whole system is evaluated by the algebraic sum of the masses of their individual components. $M_{A\&B} = M_A + M_B$

Figure 10: The working pad and the corresponding system responds

6 Conclusions

Case-based reasoning (CBR) is a versatile AI technology and can been found in many industrial applications [2] but its potential in training and education is still not fully explored. The work reported here may serves to strengthen the position of CBR in developing instructional systems.

The contribution of the paper to the endeavour of computer-assisted learning is twofold. Firstly, technically, a formal framework for representing cases for learning purposes has been developed. Its formal basis provides a solid foundation for developing robust computer-based instructional systems. With this methodology, the developers only have to concentrate their effort on collecting and encoding the cases. The rest (generating relevant instructional activities from the cases) will be taken care of by the system. This approach offers another advantage for rendering the cases amenable to further analysis. This may be used for providing tool to verify the case-base for internal consistency. Secondly, educationally, our approach paves the way for systematic educational software engineering because it is built on the needs of users, not the technical skills of the developers. Often, educational software developers have adopted a technically-driven design philosophy. Such systems run the risk of losing sight of what is actually happening in the real learning setting.

Our approach avoids the temptation of jumping onto the hi-tech bandwagon but, instead, concentrates firstly on what the students really need. The reason we developed a case-based learning system was not due to the existence of the technology and trying to find what role the technology can play in learning. Rather, we choose to develop a case-based approach to learning because students do learn from referring to past cases. This principle we consider crucial in determining if the final system proves itself useful to our students. Other features of the system have not been described due to space limitation. They include generating different categories of questions from a case graph [6] to promote self-explanation from the students. The model proposed in this paper can also perform qualitative reasoning [4], and causal order between system variables can be represented succinctly.

References

[1] Anderson, J.R. et al., "Learning to Program in LISP", Cognitive Science, Vol. 8, pp.87-129, (1984).

[2] Bergmann, R. et al., "Developing Industrial Case-based Reasoning Applications", Springer-Verlag, (1999).

[3] Fung, P.W., "Designing an Intelligent Case-based Learning Environment with Conceptual Graphs", PhD Thesis, Dept. of Computation, UMIST, England, (1996).

[4] Fung, P.W., "Generating Qualitative Predictive Problems in a Case-based Physics Tutor" In B. du Boulary & R. Mizoguchi (Eds.) Artificial Intelligence in Education: Frontiers in Artificial Intelligence and Applications Vol. 39, Ohmsha: IOS Press, (1997).

[5] Fung, P.W. & Adam, A., "Conceptual Graphs for Similarity Measurement in a Case-based Physics Problem Solver" In Proceedings of AI-ED 95 – World Conference on Artificial Intelligence in Education. Washington D.C, (1995).

[6] Fung, P.W. & Adam, A., "Questioning Students on the Contents of Example Solutions" In Proceedings of European Conference on Artificial Intelligence in Education, Lisbon, (1996).

[7] Kolodner, J. "Case-Based Reasoning" San Mateo, CA: Morgan Kaufmann Publishers, (1993).

[8] LeFevre, J.A. & Dixon, P., "Do Written Instruction need Examples?" Cognition and Instruction, Vol.3, pp.1-30, (1986).

[9] Peterson, J.L., "Petri Net Theory and the Modeling of Systems." Englewood Cliffs. NJ: Prentice-Hall Inc. (1981).

[10] Sanders, K.E., Kettler, B.P. & Hendler, J.A. "The Case for Graph-Structured Representations" In D.B. Leake & E. Plaza (Eds.) Case-based Reasoning Research and Development. Lecture Notes in Artificial Intelligence, Vol. 1266, pp. 245-54, Springer-Verlag, (1997).

[11] Schank, R.C., "Case-based Teaching: Four experiences in Educational Software Design." Report No.7, The Institute of Learning Sciences, Northwestern University, (1991).

[12] Self, J., "Student Models in Computer-aided Instruction." International Journal of Man-Machine Studies, Vol. 6, pp. 261-76, (1974)

[13] Sowa, J., "Conceptual Structures: Information Processing in Mind and Machine" Reading, MA: Addison-Wesley, (1984).

[14] Specht, M. & Weber, G., "Episodic Adaptation in Learning Environments" In Proceedings of the European Conference on Artificial Intelligence in Education, Lisbon, pp.171-176, (1996).

Use of abstraction levels in the design of intelligent tutoring systems

Ruddy Lelouche
Département d'informatique
UNIVERSITÉ LAVAL Québec G1K 7P4 CANADA
Tel.: (+1-418) 656 2131, ext. 2597 Fax: (+1-418) 656 2324
e-mail: lelouche@ift.ulaval.ca

In problem-solving domains (mathematics, physics, engineering, and most "exact science" disciplines), the knowledge to be acquired by the student is twofold: the knowledge describing the domain itself, but also and mainly the knowledge necessary to solve problems in that domain. As a result, an educational system in such a domain encompasses three knowledge types: the domain knowledge and the problem-solving knowledge, i.e. the knowledge to be acquired and mastered by the student, and the tutoring knowledge, used by the system to facilitate the student's learning process. In this paper, we show how these three knowledge types can be modelled, how they should interact with one another in order to fulfil the system educational purpose, and above all how abstraction levels can shed a uniformizing light on the system operation and make it more user-friendly. We thus hope to bring some contribution to the general and important problem of finding a generic architecture to intelligent educational systems.

Keywords: **Intelligent tutoring systems, Abstraction, Complexity, System Design.**

1 Introduction

Teaching is a very complex process in itself. Teaching strategies and activities vary considerably: by the role and autonomy they give to the learner, by the type of interactions they trigger with him/her, by the evaluations they enable, by the relationships they make between theory and practice, etc. From that last perspective, teachable domains can be classified according to the type of knowledge to be acquired by the student: "know", "know-how", and "know-how-to-be". Examples of such domain types are respectively: anatomy or a language grammar, the skill to solve a mathematical or medical problem, and the capability to adapt to one's environment or to deal with personal relationships. We are more particularly interested in the second type.

Moreover, almost all teachable domains vary in complexity, from simple basics to relatively complex problems to solve or situations to deal with. Thus, a student should learn and master the basics of such a domain before being taught wider notions. And when a human tutor detects errors or misunderstandings, s/he usually draws the student's attention on a small subset of the involved knowledge, so that s/he may correct his/her errors and/or misunderstandings, focusing either on a given set of the domain knowledge or on the scope of knowledge involved by a given problem.

Problem-solving (PS) domains are the ones in which we are interested here. In such a domain, the knowledge to be acquired by the student is twofold: the domain knowledge itself, but also and mainly the knowledge necessary to solve problems in that domain. As a result, an education-oriented system in such a domain, which we here call a PS-ITS, must encompass three knowledge types: the domain knowledge and the problem-solving knowledge, constituting the knowledge to be acquired and mastered by the student, and the tutoring knowledge, used by the system to facilitate the student's learning process.

This paper has two goals: to present each of the three types of knowledge involved in a PS-ITS, and for each type of knowledge, to show how abstraction and complexity levels appear and how we think it is possible to deal with them.

To do so, we present in section 2 our domain knowledge modelling and how we exemplify it in a few PS domains . Next, in section 3, we focus on the advantage of separating the problem-solving knowledge from the domain knowledge in a PS-ITS, and we present some problem-solving activities in various domains. In section 4, we briefly describe some principles of tutoring knowledge modelling in a PS-ITS. In each of these three sections, we show how to use abstraction and complexity levels, exemplifying them in a few typical domains.

Finally, section 5 presents the educational interests of using abstraction and complexity levels when modelling the three types of knowledge involved in a PS-ITS.

2 Domain knowledge

In order to describe the domain knowledge, we first present its characteristics in a general PS-ITS (section 2.1). We then show how we model it in a few PS domains (section 2.2), and how such an approach lets us introduce the notions of abstraction and complexity levels (section 2.3).

2.1 General

The first type of knowledge involved in every ITS, the *domain knowledge* (DK), contains all theoretical and factual aspects of the knowledge to be taught to the student. Although its specific structure can be varied, it typically may include concepts, entities, and relations about the domain [Brodie & al., 1984], object classes and instances [Kim & Lochovsky, 1989], possible use restrictions, facts, rules, [Kowalski, 1979; Clocksin & Mellish, 1981], semantic or associative networks [Findler, 1979; Sowa, 1984], etc.

The main system activities centred on this knowledge type are:

- providing the student with theoretical presentations and explanations about the various knowledge elements and their relationships in the teaching domain;
- providing the other modules of the ITS, i.e. problem-solving and tutoring, with the necessary background of domain knowledge that they need.

2.2 Application to a few domains

In the particular domain of **cost engineering**, Lelouche and Morin [1997; Morin, 1998] represent this type of knowledge with concepts, relations, and a special case of relations modelled as concepts, the factors.

Concepts can be basic entities like investment, interest, investment duration, present and future values, compounding, compounding period, interest rate, annuity, etc.

Concepts are linked to one another by various types of *relations*: either usual knowledge-representation relations, like *subclass of*, *element of*, *sort of*, etc., or numerical relations represented by formulæ. Such a formula is:

$$F = P \times (1 + i)^n \tag{1}$$

which, given the present value P of an investment over n periods at interest rate i, computes the corresponding future value F of that investment.

A formula such as (1) can be rewritten as:

$$F = P \times \Phi_{PF,i,n} \text{ where } \Phi_{PF,i,n} = (1 + i)^n \tag{2}$$

$$P = F \times \Phi_{FP,i,n} \text{ where } \Phi_{FP,i,n} = (1 + i)^{-n} \tag{3}$$

thus introducing the *factors* $\Phi_{PF,i,n}$ and $\Phi_{FP,i,n}$. Factors allow us to separate their definition (rightmost equalities above, a quantitative aspect) from their possible uses in the application domain (leftmost equalities, a qualitative aspect).

Similarly, the factor $\Phi_{AP,i,n}$ converts a series of identical annual amounts A into a unique present value P:

$$P = A \times \Phi_{AP,i,n} \text{ where } \Phi_{AP,i,n} = \frac{(1+i)^n - 1}{i\,(1+i)^n} \tag{4}$$

Actually, $\Phi_{AP,i,n}$ is a sum of Φ_{FP} factors (see details below). The factor $\Phi_{PA,i,n}$ does the reverse process:

$$A = P \times \Phi_{PA,i,n} \text{ where } \Phi_{PA,i,n} = \frac{i\,(1+i)^n}{(1+i)^n - 1} \tag{5}$$

There exist other factors converting gradient and geometrical series of amounts into a present or future value; such factors are also computed as a sum of $\Phi_{FP,i,n}$ factors.

In **geometry**, *concepts* are basic elements like point, line, segment, and later more elaborate elements like angle, then square, rectangle, circle, ellipse, polygon, solid, polyhedra, etc. Examples of *relations* between concepts are adjacency (of segments or angles), parallelism (of lines or line segments), complementarity (of angles), etc. Upper-level, more abstract concepts are then defined using lower-level ones, as well as relations between these lower-level concepts (e.g. a triangle is a set of three segments adjacent pairwise).

In **mechanical physics**, we similarly introduce *concepts* like time, distance, velocity, acceleration, mass, force, and later angle, angular velocity, angular acceleration, moment of inertia, torque, etc. We also introduce *relations* like the one defining velocity as the variation in distance per unit of time, or the one stating that the acceleration a is proportional to the force F that is applied. Introducing a generalization from linear to rotational movement, another relation defines angular velocity as the angle variation per time unit, and another one states that the angular acceleration α of a solid body is proportional to the torque τ that is applied to it. More precisely, we have:

for a linear movement $F = M \times a$ where M = total mass of the body (6)

and for a rotational movement $\tau = I \times \alpha$ where $I = \sum (m \times r^2)$ (7)

Equation (6) expresses Newton's second law. In equation (7), I is the moment of inertia and is expressed in terms of the mass m of each of its particles and of its distance r to the rotation axis. Obviously M in equation (6) and I in equation (7) play the role of factors as in cost engineering.

Although formulæ like (2-7) related to factors essentially involve quantitative aspects, the similarities and differences between them, and the circumstances regulating the use of either one, are of a deeply qualitative ground. In **cost engineering**, if the *value* of factors is indeed calculated from two or three numerical parameters, the *context* in which they are defined depends on whether we have to timewise move a unique amount or a series of amounts, identical or not, or conversely to compute an equivalent annual amount, etc. In fact, this context corresponds to the type of conditions that govern the investment, or *investment conditions type*, without respect to the amounts and durations involved, and is thus essentially qualitative. Similarly, in **physics**, the proportionality between force and linear acceleration, or between torque and angular acceleration, expresses a qualitative relationship. Only if the need arises, the exact relationship can be expressed by using the actual mass M in formula (6) or the result of the computation of the moment of inertia I in formula (7), which in the general case involves a simple or double integral. Indeed, did not the use of qualitative reasoning originate with qualitative physics?

2.3 Towards the notions of abstraction and complexity levels

In most PS-domains, abstraction most obviously appears in the *definition of the domain concepts* themselves, like we showed in all three domains above.

If *factors* are used in the domain, it also appears that every factor introduces an additional intermediate abstraction level between the concepts implied in the equation defining it. For example, in the case of formula (1), or equivalently formulæ (2) and (3) in **cost engineering**, or in the case of formula (6) and (7) in **physics**, we have (see figure 1):

- at the bottom of the hierarchy, basic concepts "making technicalities explicit" if necessary: the interest rate and the number of periods in cost engineering, the distribution of mass within the body volume in physics ;
- above them, concepts more fundamentally related to the problem being solved, namely in cost engineering the present and future values of the investment, and in physics the force and acceleration, or the torque and angular acceleration;
- between these two levels, an intermediate level created by the introduction of the factor (Φ_{FP}, Φ_{PF}, M, or I).

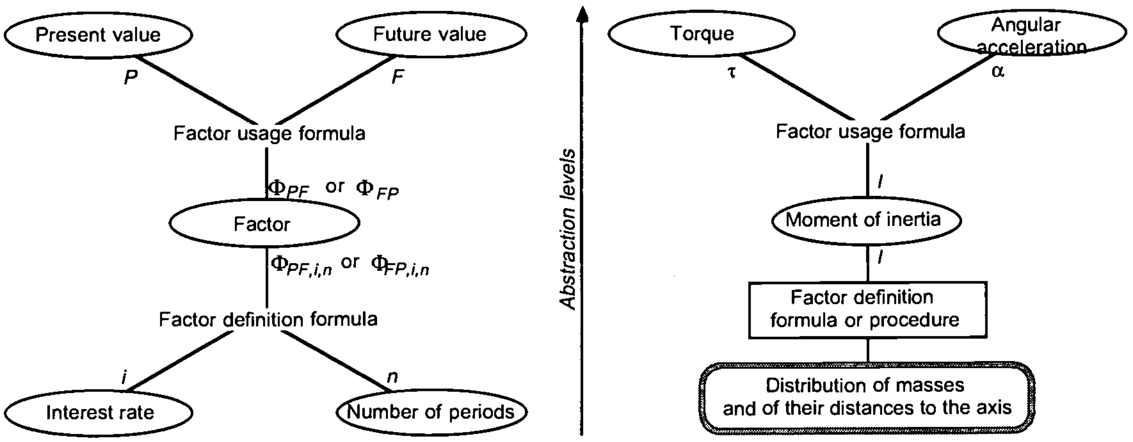

Present value
Future value
P
F
Factor usage formula
Φ_{PF} or Φ_{FP}
Factor
$\Phi_{PF,i,n}$ or $\Phi_{FP,i,n}$
Factor definition formula
i
n
Interest rate
Number of periods
Abstraction levels
Torque
Angular acceleration
τ
α
Factor usage formula
I
Moment of inertia
I
Factor definition formula or procedure
Distribution of masses and of their distances to the axis

Figure 1 — Representation of a factor as a concept.

That intermediate status of the factor, originally just an intermediate variable in computations [see formulæ (2) and (3) or (6) and (7)], makes it appear as a pedagogically oriented concept, which clearly separates

- the computational, quantitative aspect of the factor definition,
- from the practical, qualitative aspect of the factor usage in a domain problem.

This follows the theory [Lenat & al., 1979; Malec, 1989] according to which the use of multiple *abstraction levels* eases the modelling process and simplifies inferences which may be made on the domain concepts.

Most interestingly, our scaffolding approach can be made more general, at least in certain domains, where we may present and use higher-level factors built upon these first ones. Indeed, in cost engineering, "above" Φ_{FP} and

Φ_{PF}, the factors used to express the present and future values of a series of identical amounts (and vice versa) are a first way to generalize this concept hierarchy. For example, the Φ_{AP} factor is indeed a sum of Φ_{FP} factors:

$$\Phi_{AP,i,n} = \sum_{k=1}^{n} \Phi_{FP,i,k} = \sum_{k=1}^{n} (1+i)^{-k} = \frac{(1+i)^n - 1}{i\,(1+i)^n}$$

where the last expression results from computing the geometrical series shown. This example constitutes a proof of (4), but also and mainly shows that the Φ_{AP} factor is at a higher level than Φ_{FP}. Note that this refers to a *complexity level* rather than an abstraction level, since it is due to the way the Φ_{AP} factor is defined and computed. Similarly, the moment of inertia of a complex body can be (and often is) computed as the sum of elementary moments of inertia, and therefore is at a higher complexity level.

3 Problem-solving knowledge

In order to describe the problem-solving knowledge, we now present the general characteristics regarding problem-solving knowledge modelling in a PS-ITS (section 3.1). As we did in section 2, we then examplify our model in the cost-engineering and physics domains (section 3.2).

3.1 General

The second type of knowledge is specific to PS-domains [Ganeshan & al., 2000; Gertner & VanLehn, 2000], henceforth to PS-ITSs. We call it *problem-solving knowledge* (PSK). It contains all inferential processes used to solve a problem resulting from the instantiation of a practical situation based on the domain knowledge [Kowalski, 1979; Patel & Kinshuk, 1997]. In other words, in order to be able to solve a problem, the problem-solving knowledge needs a theoretical background, which is found in the domain knowledge. The processes stored in PSK may be represented in various ways, using any or all of: logic [Kowalski, 1979], procedural networks [Brown & Burton, 1978], semantic networks with procedural attachments, (augmented) transition networks, production rules [Goldstein, 1979; Anderson & Reiser, 1985], etc.

The main system activities centred on this knowledge type are:

- providing the inferential tools for problem solving, by the system or by a student;
- providing the inferential tools for coaching a student in a problem-solving session.

The main advantage of separating the problem-solving knowledge from the domain knowledge is that it emphasizes the distinction between the domain itself and the skills used to solve a practical problem in that domain, thus simplifying the learning process. That knowledge separation into DK and PSK is common to all PS-domains; this is why we believe that PS-ITSs, which are aimed at helping the student to learn how to solve problems, should display the same knowledge separation.

Besides, following [Lelouche & Morin, 1997], we can use — we believe in a novel way — that separation between DK and PSK to define four *generic operating modes* in a PS-ITS, based on the type of knowledge involved (DK or PSK), and on who "generates" that knowledge (the system or the student).

- In *domain-presentation mode*, the student asks the system some information about a domain theoretical element, and the system reacts by transferring to the student the required information or knowledge. The knowledge involved in this category is always DK, system-generated.
- In *demonstration mode*, the student asks the system to solve a practical problem or to coach him/her while s/he solves a problem. In the first case, the problem typically comes from the student him/herself, whcreas in the latter one the problem typically comes from the tutoring system. In either case, the main level of knowledge involved is PSK, system-generated.
- In *domain-assessment mode*, the system prompts the student to develop a domain element, and the student thus expresses his/her understanding of that element. If judged necessary, the system may then intervene to correct that understanding. The knowledge involved in this mode is essentially DK, student-generated.
- Finally, in *exercising mode*, the system prompts the student to solve a practical problem. The student then solves it step by step, showing what s/he understands of the involved problem-solving knowledge and of the associated domain knowledge. If necessary, the system may decide to intervene in order to help him/her reach his/her goal or to correct it. The knowledge involved in this mode is naturally PSK, student-generated.

3.2 Application to a few domains

Several problem-solving activities are domain-independent, like:

1. identify and instantiate the given problem data;
2. identify and instantiate the expected result(s);
3. apply a formula;
4. apply a theorem.

Every PS-domain also has its own domain-dependent activities. For example, in **cost engineering**, we have:

5. draw a temporal diagram to represent the relevant events;
6. compare amounts located at the same date;
7. compare amounts located at different dates;
8. add amounts situated at the same date;
9. add amounts situated at different dates;
10. choose a reference date;
11. move an amount from one date to another;
12. collapse a series of periodic amounts into one single amount;
13. explode an amount into a series of periodic amounts.

Similarly, in the subset of **mechanical physics** refered to above, some activities would be:

14. compute a torque;
15. compute an angular acceleration;
16. compute a moment of inertia.

In many cases, a PS activity can be rephrased into, restated as, a different one, of a lower *abstraction level*, because more immediate, more down-to-earth, closer to the problem to be solved. For example, in **mechanical physics**, assuming that the torque and the moment of inertia of a given solid body are known (either given or previously computed), the activity "compute the angular acceleration" (activity 15) would be expressed as, or translated into "apply formula (7)", an instance of the lower-level activity 3. A PS physics tutor is presented in [Gertner & VanLehn, 2000].

Sometimes, a PS task may also be divided into smaller ones, letting us use again the notion of *complexity levels* in these tasks. For example, in **cost engineering**, comparing two amounts situated at different dates implies:

- first, choosing a reference date at which to make the comparison;
- then, moving either (or both) amount(s) from its (their) present date(s) to the reference date;
- finally, comparing the amounts, now both located at the same reference date.

These subactivities (of types 10, 11, and 6 respectively in the sample list above) thus appear to be of a lower complexity level than the initial one (of type 7). However, it is interesting to note that, although activity 7 turns out to be more complex than activity 6 (the latter is part of the former), both are stated using the same *abstraction level.*

It may also happen that some lower-level activities can *only* appear as components of a higher-level one. For example, still in **cost engineering**, the activity "drawing a temporal diagram" (type 5 above) implies the following tasks, which can only be accomplished as part of that activity (hence their identification in this paper from 5a to 5d):

5a. draw a timeline to encompass all periods implied by the problem data;
5b. draw arrows representing the amounts involved in the problem data;
5c. if necessary, split an amount (or each amount in a series) to simplify the computations;
5d. qualitatively draw a special arrow to represent the expected result of the computations to be made.

In that case, activity 5 is both of a higher complexity level and of a higher abstraction level than any of its subactivities.

4 Tutoring knowledge

We now briefly present the *tutoring knowledge* (TK) in order to help the reader to better apprehend the relationships of that knowledge with DK and PSK. This third type of knowledge contains all tutoring processes enclosed in the ITS. It is not directly related to the teaching domain or to problem solving, but is there to help the student understand, assimilate, and master the knowledge included in DK and PSK [Gagné & al., 1992; Gagné & Trudel, 1996].

The main system activities using TK are:

- ordering and formatting the topics to be presented to the student;
- monitoring a tutoring session, i.e., triggering the various tutoring processes according to the system tutoring goal and the student's actions; such monitoring may imply giving explanations, asking questions, changing to another type of interaction, etc.;
- in a PS-domain, while the student is solving an exercise, monitoring the student's PS activities: understanding and assessing these activities, giving advice to correct or optimize them, giving hints or partly solving the exercise at hand (as required by the student or by the tutoring module), etc.;
- continuously analysing the student's progress in order to improve the efficiency of the tutoring process.

The advantage of separating the tutoring knowledge from the knowledge of the domain to be taught has been emphasized long ago [Goldstein, 1977; Sleeman & Brown, 1982; Clancey, 1986; Wenger, 1987], and lies in the reusability of TK in various domains. In the case of PS-domains, the domain to be taught clearly encompasses both DK and PSK; indeed, the term "domain knowledge" applies to DK if referring to the knowledge type, and to DK + PSK if referring to the knowledge to be acquired. Therefore, as shown in the introduction, in a PS-ITS, knowledge ends up being separated into three categories rather than two.

We believe that the tutoring processes are triggered by *tutoring goals* which depend on the current educational setting and learning context. The role of tutoring goals has been discussed in several works, some of the most recent ones dealing with task and instruction ontologies [Mizoguchi, 1999]. In the current state of our research, our assumption is the following: the underlying hierarchy or hierarchies governing the way tutoring processes interact with one another is not related to these processes *per se*, but rather to the *current goal* to be attained when they are invoked. The current goal varies during the session, depending on the student's actions or difficulties, following a dynamically built abstraction-based hierarchy. If our assumption turns out to hold, then the *dynamic structure of educational goals and subgoals* — which itself depends on the student's desires or abilities, the main underlying objective of the tutoring system, the student's state (e.g. of tiredness, etc.) and performance, etc. — will determine the succession of tutoring processes activated and tutoring interactions taking place. To our knowledge, the use of abstraction levels to induce a dynamic hierarchy of tutoring goals is new, as is the assumption that such a hierarchy will play a major role in activating the various tutoring processes and student–system interactions. Learning goals have been used by Towle [2000], but only for educational simulations, not for tutoring processes in general.

5 Educational interests of abstraction and complexity levels

In the above sections, we have sketched a complexity– and abstraction–level approach to help model the three types of knowledge involved in a PS-ITS. In this section, after clarifying these notions in section 5.1, we present the educational interests of our model. Sections 5.2 to 5.4 focus on the type of knowledge respectively presented in sections 2 to 4. Section 5.5 summarizes that discussion with some overall pedagogical interests of our approach.

5.1 An informal definition of abstraction and complexity levels

In the first three sections, we only refered to abstraction and complexity levels. Here, we try to define these notions better and in a more generally applicable way. Both notions are based on the common notion of *refinement*, but differ in how the refinement is made: in a general way, abstraction is based on, or refers to, *expressiveness* or *scope*, whereas complexity is based on, or refers to, the *number of components*.

For *concepts*, taking geometry as an example, a polygon has a higher abstraction level than a triangle or a square, because the number of sides in a polygon is indefinite, but a lower abstraction level than a set of segments, because these segments in a polygon are forced to be pairwise adjacent; a square has a higher complexity level than a triangle, because it has more sides, and also because there are constraints (re. size and angles) between these sides. In cost engineering, we saw that the factors Φ_{FP} and Φ_{AP} are expressed at the same abstraction level, although Φ_{AP} has a higher complexity level, because of the way it is defined and computed. A similar distinction between abstraction levels and complexity levels holds for the *relations* they express.

For *problem-solving activities*, we have similar distinctions, as shown in section 3.2 with several examples.

Finally, the same holds for *tutoring processes*, or student–system interactions. For instance, the ITS task of tutoring a student while s/he is solving a problem will turn out to be of a higher complexity level if the student encounters more difficulties, although the abstraction level of this process does not depend on the particular student being tutored or on the particular problem being solved. On the other hand, reacting to a student request for hint, or for explanation, is of a lower abstraction level than the previous one; however, there again, the complexity of that task will depend on the specific student request (some simply formulated questions may have quite complex answers!), and will eventually depend also on the way the student is or is not satisfied with the initial system response.

Such level-based distinctions have also been made, for example, by Mizoguchi [1999]. Note that, although the statement "A has a higher abstraction level than B" is clear and may be true, we think that the number of abstraction levels between A and B is not defined, because that number would depend on the modelling effected; for the same reason, it would be even more meaningless to try to assign a numeric value to these levels.

5.2 Domain modelling

The definition of concepts from the simplest to the most complex induces a long-time known *presentation order* for the subject matter. Similarly and in addition, the factor hierarchy described in section 2 for cost engineering lets us derive an *order for the presentation of factors* to the student, from the lowest (simplest) level up to the highest, i.e. with increasing understanding complexity. That does not imply that such an order is unique, or even the best (e.g. a student's personal interests might make another order more motivating for him/her), but it is justified by our model. This presentation order may itself induce, like for domain concepts, a possible *order for prerequisites*; e.g., if a student experiments difficulties to deal with Φ_{AP}, has s/he well mastered Φ_{FP}, a conceptually simpler factor?

Moreover, the factor-induced intermediate abstraction levels will permit the ITS to exhibit a *sharper modelling of conceptual errors*. For example, the source of an understanding error concerning one of the two relations in equation (2) or (3) or (7) (see also figure 1) is much easier to identify using the corresponding factor, either as a definition error or as a usage error, than an error concerning the global equation (1), where the definition and application relationships are not made explicit, and therefore are impossible to distinguish. Similarly, an error using a Φ_{AP} factor may be diagnosed as possibly resulting from an insufficient mastery of the simpler factor Φ_{FP} as concept (which in turn will be diagnosed as related either to its definition, or to its usage). Similarly in physics, if the student stumbles on concepts like angular acceleration or moment of inertia, has s/he mastered the simpler although similar concepts of acceleration or mass?

Abstraction and complexity levels on domain elements (concepts and relations, possibly including factors) can then be used to introduce various abstraction levels of explanations. Such explanations can then be tailored to the student's questions, and adapted to the reminders possibly needed by the student.

5.3 Problem-solving modelling

The problem-solving activities briefly presented in section 3 naturally display abstraction and complexity levels. Indeed, a standard problem can usually be divided, possibly in more than one way, into major steps, which can then be split into simpler substeps. As explained in 5.1, each subactivity in that case may be either simpler (lower complexity level) or more concrete (lower abstraction level) than the original one, or both.

In a first development stage, these abstraction and complexity hierarchies, both for domain elements and for problems to be solved, can ease the *definition of exercise types* to be implemented into the ITS, and can ease the tutor module task of *choosing the exercise type* to challenge the student with. Later, once that basic system is operational, the same hierarchies can help develop an automatic *exercise generator* dealing with the domain elements to be mastered by the student. That approach will then help the student to acquire a better critical mind about the relative importance of *problem–solving* knowledge vs. *domain* knowledge.

As for domain elements, abstraction and complexity levels can be used to introduce various types of *explanations* about the problem to be solved, varying both in abstraction (focus level, terms used, references made) and in complexity (quantity of details, possible references to the problem substeps). Moreover, our approach will lead the student to focus specifically on the *activities* for which s/he needs more tutoring, with the abstraction and complexity levels appropriate to his/her individual case.

5.4 Tutoring modelling

Functioning mode		Domain-presentation mode	Demonstration mode	Domain-assessment mode	Exercising mode
Main type of knowledge involved		Domain knowledge	Problem-solving knowledge	Domain knowledge	Problem-solving knowledge
Student's main goal		To learn (acquire or improve knowledge)		To assess his/her learning	
Direction of the knowledge transfer		System → Student		Student → System	
Typical interaction	Trigger (start)	The student asks the system...		The system prompts the student...	
		some information about a domain theoretical element	to solve a practical problem or to coach him/her in problem solving	to develop a domain element	to solve a practical problem
	Knowledge exchange	The system presents...		The student presents his/her view of...	
		the requested element	a possible solution to the requested problem	the requested element	a possible solution to the given problem
	Result (closure)	The student expresses his/her understanding...		The system assesses the student's answers, and possibly corrects them	
		of the element	of the problem solution		

Table 2 — Characteristics of the four typical operating modes of a problem-solving ITS.

As presented in section 3.1, the distinction between DK and PSK leads to the natural definition of four operating modes. Their main characteristics are recalled in Table 2.

The successive tutoring goals aimed at by the system (see section 4) are likely to result in a chain of recursive calls of the tutoring processes invoked. This recursivity will or will not be direct, depending on the tutoring interaction types being chained: the system might decide to temporarily change between interaction types, e.g. to respond to the student's actions or requests. However, the potential length of this chain is only apparent: because of the abstraction hierarchy of tutoring goals, each newly invoked process will be called with a *narrower scope* and/or a *lower complexity*, which eliminates the risk of "forgetting" the initial tutoring goal or of running into an infinite loop.

More generally, tutoring the student may take the form of explanations, guidance, hinting, or even partially solving the exercise on which the student is currently working. The level at which these will be conducted will depend on the current tutoring goal (see section 4). We think our approach is close to that of VanLehn and his colleagues [2000], although they focused their attention on fading and deepening (a particular result of the tutoring interactions) rather than on the current pedagogical goal (the cause for these interactions).

5.5 Overall interests of these abstraction and complexity levels

Abstraction levels are certainly not new. What we think is new is to use them in a systematic way to shed a uniformizing light on the ITS design and operation, and to make it more user-friendly once implemented.

First, they may help to give a better tailoring to the system tutorial interventions to fulfil the student's needs and the system tutoring goals, thus improving its conviviality and efficiency.

Then, all the capabilities presented above should result in smoother, more "natural", human-like interactions with the student. This improved ability to reproduce a human teacher's behaviour contributes again to make the system more user-friendly, and more likely to be used by the student.

Finally, although that aspect is not in the scope of this paper, our refinement of the three types of knowledge as described in sections 2 to 4 paves the way to the conception and the implementation of a structured error model, and eventually of a structured student model.

6 Conclusion

This presentation of a possible knowledge structure for PS-domains, which emphasizes the separation between domain knowledge and problem-solving knowledge, shows how a general functioning theory of such an ITS — namely the four operating modes described in sections 3.1 and 5.4 — can naturally be derived.

Moreover, the abstraction and complexity levels highlighted throughout this paper can be used as a common guideline to help finding an appropriate representation for each one of the three knowledge type, and thus can help creating more efficient ITSs. More generally, this guideline can shed a uniformizing light on the system design, although it has never been used in a systematic way in the design or implementation of an ITS.

We thus hope to bring some contribution to the general and important problem of finding a generic architecture for intelligent tutoring systems.

References

[1] Anderson John R.: *Cognitive Modelling and Intelligent Tutoring*. Washington, DC: National Science Foundation (1986).

[2] Anderson John R. & B. J. Reiser: "The LISP Tutor". *Byte* **10**, no. 4, p. 159-175 (1985).

[3] Brodie Michael L., John Mylopoulos & Joachim W. Schmidt, eds.: *On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages*. New York: Springer Verlag (1984).

[4] Brown John S. & Richard R. Burton: "Diagnostic models for procedural bugs in mathematical skills". *Cognitive Science* **2**, p. 155-192 (1978).

[5] Clancey W. J.: "Qualitative student models". *Annual Review of Computer Science* **1**, p. 381-450 (1986).

[6] Clocksin William F. & Christopher S. Mellish: *Programming in Prolog*. Berlin: Springer-Verlag (1981).

[7] Findler Nicholas V., ed.: *Associative Networks*. Orlando, FL: Academic Press (1979).

[8] Frasson Claude, Gilles Gauthier & Alan Lesgold, eds.: *Intelligent Tutoring Systems*, Proc. of the *3rd International Conference, ITS'96*, Montréal (Canada), 12-14 June 1996. LNCS 1086, Berlin: Springer (1996).

[9] Gagné Denis & André Trudel: "A highly flexible student-driven architecture for computer-based instruction". [Frasson & al., 1996], p. 66-74 (1996).

[10] Gagné Robert M., Leslie J. Briggs & Walter W. Wager: *Principles of Instructional Design*, fourth edition (1st edition published in 1974). Orlando, FL: Harcourt Brace Jovanovich (1992).

[11] Ganeshan R., W.L. Johnson, E. Shaw & B.P. Wood: "Tutoring diagnostic problem solving". In [Gauthier & al., 2000], p. 33-42 (2000).

[12] Gertner A.S. & K. VanLehn: "ANDES: a coached problem-solving environment for physics". In [Gauthier & al., 2000], p. 133-142 (2000).

[13] Goldstein Ira P.: *The Computer as Coach: an Athletic Paradigm for Intellectual Education*. AI Memo 389, AI Laboratory. Boston, MA: Massachusetts Institute of Technology (1977).

[14] Goldstein Ira P.: "The genetic epistemology of rule systems". *International Journal of Man-Machine Studies*, Vol. 1, No. 1, p. 51-77 (1979).

[15] Gauthier Gilles, Claude Frasson & Kurt VanLehn, eds.: *Intelligent Tutoring Systems*, Proc. of the *5th International Conference*, Montréal (Canada), 19-23 June 2000. LNCS 1839, Berlin: Springer (2000).

[16] Kim Won & Frederick H. Lochovsky, eds.: *Object-Oriented Concepts, Databases, and Applications*. ACM Press, Reading, MA: Addison-Wesley Publ. (1989).

[17] Kowalski R.: *Logic for Problem Solving*. Berlin: North-Holland (1979).

[18] Lelouche Ruddy & Jean-François Morin: "Introduction of pedagogical concepts in domain modelling for an ITS". *European Journal of Engineering Education*, Vol. 23, No. 2, p. 255-271 (June 1998).

[19] Lelouche R. & J.-F. Morin: "Knowledge types and tutoring interactions in an ITS in a problem-solving domain". *FLAIRS 1997, Proc. of the 10th International FLorida Artificial Intelligence Research Symposium*, Daytona Beach (Florida, U.S.A.), 10-14 May 1997, p. 62-66 (1997).

[20] Lenat D., F. Hayes-Roth & P. Klahr: *Cognitive Economy*. Working Paper HPP-79-15, Stanford Heuristic Programming Project, 46 pages. Stanford, CA: Stanford University (June 1979).

[21] Malec J.: " Knowledge elicitation during dynamic scene description ". *ACM–SIGART Newsletter: special issue on knowledge acquisition*, No. 108, p. 162-163 (April 1989).

[22] Mizoguchi Riichiro: "Ontology-awareness for intelligent instructional systems". Invited talk. *Advanced Research in Computers and Communications in Education, Volume 1* (G. Cumming, T. Okamoto & L. Gomez, eds.), Proceedings of the *7th International Conference on Computers in Education, ICCE 99*, Chiba (Japan), 4-7 November 1999. Amsterdam, Netherlands: IOS Press, p. 45-52 (1999).

[23] Morin J.-F.: *Conception of an intelligent tutoring system in cost engineering: knowledge representation, pedagogical interactions, and system operation.* Master Thesis, Département d'Informatique. Québec, Canada: Université Laval (1998).

[24] Patel Ashok & Kinshuk: "Intelligent tutoring tools in a computer-integrated learning environment for introductory numeric disciplines". *Innovations in Education and Training International*, Vol. 34, No. 3, p. 200-207 (1997).

[25] Sleeman D. H. & John Seely Brown, eds. : *Intelligent Tutoring Systems*. London: Academic Press (1982).

[26] Sowa John F.: *Conceptual Structures*. Reading, MA: Addison-Wesley (1984).

[27] Towle B.: "Using student task and learning goals to drive the construction of an authoring tool for educational simulations". In [Gauthier & al., 2000], p. 173-181 (2000).

[28] VanLehn K., R. Freedman, P. Jordan, C. Murray, R. Osan, M. Ringenberg, C. Rosé, K. Schulze, R. Shelby, D. Treacy, A. Weinstein & M. Wintersgill: "Fading and deepening: the next steps for ANDES and other model-tracing tutors". In [Gauthier & al., 2000], p. 474-483 (2000).

[29] Wenger, Étienne: *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann (1987).

Using Decision Networks for Adaptive Tutoring

Peng-Kiat Pek* and Kim-Leng Poh**
** Singapore Polytechnic*
500 Dover Road, Singapore 139651
E-mail: baiychen@magix.com.sg
*** National University of Singapore*
10 Kent Ridge Crescent, Singapore 119260
E-mail: isepohkl@nus.edu.sg

This paper reports a research project that uses dynamic decision networks in providing teacher with information on students' misconceptions and students with online tutoring. A set of Bayesian networks models the conditional dependencies between learning objectives and goals which are associated with the curriculum. Student' s responses to test items are recorded and transformed as evidence into a relevant Bayesian network to compute his likely state of knowledge mastery. The personalized Bayesian network is then converted into a dynamic decision network by adding utility and decision nodes. Tutoring policy is followed through and necessary responses from the student are solicited using additional test items. The student Bayesian network is updated when new evidence arrives, and is again converted to a decision network to determine the next tutoring policy. This process is repeated until the pre-requisites are achieved. The results generated by the system and future directions are discussed.

Keywords: **Adaptive Tutoring, Decision Network, Student Model, Tutoring Strategy**

1 Introduction

Tutoring of students is an ill-structured problem that is characterized by:
(a) Uncertainty of student's knowledge mastery.
(b) Preferences, judgements, intuition, and experience of teacher.
(c) Criteria for decisions are occasionally in conflict, and highly dependent on the teacher' s perception.
(d) Decisions must be achieved in limited time.
(e) The student's mental states evolve rapidly.

This study attempts to address these issues by using an intelligent *decision-theoretic* approach. The framework of this research has contributed to the development of an intelligent decision support system called *iTutor*, for tutoring Engineering Mechanics at Singapore Polytechnic.

Probabilistic or Bayesian networks [9] and decision analysis [5] have shown to be capable of solving many real-world problems involving reasoning and decision marketing under uncertainty. Bayes's nets allow for efficient reasoning and inference about combination of uncertain evidence. Student modeling with Bayes's nets for intelligent tutoring had achieved successes, see for example in [16], [11], and [2]. The differences in these works lie mainly in the choice of variables and granularity of the models.

In Villano' s Knowledge Space Theory, the basic unit of knowledge is an *item* (in the form of a question). The student' s knowledge state is defined as the collection of items that the student is capable of answering. The collection of all feasible states is called the *knowledge structure*, and it is connected by the *learning path*. By incorporating uncertainty at each node, the knowledge space can be transformed into a Bayes's net. The Bayes's net then constitutes a student model where probabilistic reasoning can be performed when evidence is available. Reye on the other hand, uses pre-requisite relationship of domain knowledge and *dynamic belief network* for modeling student' s mastery of a topic. Finally, Conati and Vanlehn make use of teacher' s solution(s) as the ideal model to track student' s faulty knowledge as the student solves a problem.

Our work here differs from others in that we construct relevant Bayes's nets by modeling learning objectives (L), evidence (V) from student responses, application of knowledge to different situations (C), and learning goal (G). A decision network [3] is then formed by adding decision and utility nodes to the Bayes's net. As it is computationally intractable to track student' s solution in real time, we use sequential decisions to generate tutoring strategy that anticipates students' responses.

This paper is organized as follows: Section 2 provides an overview of the conceptual framework for the decision theoretic intelligent tutoring system called *iTutor*. The transformation of student's responses to evidence is discussed in Section 3. Section 4 illustrates how the student model is constructed from a set of Bayes's nets, while Section 5 presents the tutoring strategy model using two-step look-ahead decision network. The results of a typical *iTutor* session are illustrated in Section 6. It emphasizes the automation of decision network construction and shows that when student's responses are available, the system is able to diagnose student's misconceptions and to provide adaptive tutoring using the generated strategy. Finally, we conclude by discussing future directions.

2 Framework of Adaptive Tutoring

Figure 1 shows the essential components of adaptive tutoring in *iTutor*.

The ***Evidence Model*** converts the student response (x_{ijk}) to item i into evidence of knowledge mastery for a relevant learning objective (v_{jk}).

The ***Student Model*** consists of a set of Bayes's nets with nodes that are either Evidence, Case, Learning Objective, or Goal. These nodes are initialized with prior information from the teacher' s judgement and theoretical probability models. The student model can be subsequently updated to reflect a student' s knowledge mastery when evidence is available.

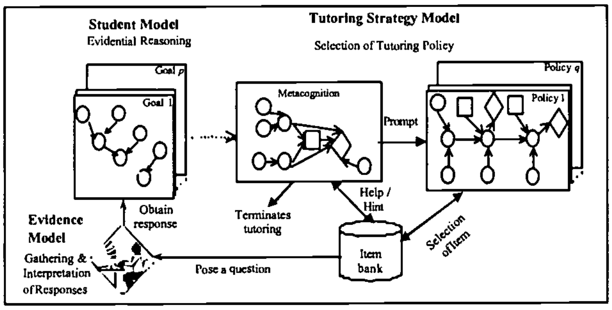


Figure 1: Inferencing Kernel of Adaptive Tutoring

The ***Tutoring Strategy Model*** uses decision-theoretic approach to select *satisfying* [14] learning objectives for tutoring student. The metacognition sub-module determines the appropriate tutor' s action: providing more help or hint, prompting another question, or stop the tutoring session. Dynamic Decision Network (DDN) provides approximate solutions for partially observable Markov decision problems, where the degree of approximation depends on the amount of look-ahead. If the decision is to obtain evidence of mastery on a learning objective, an item of difficulty b_i that matches the student' s ability θ will be selected. Student's response is collected, evaluated, and transformed into evidence at the relevant nodes in the student model. The *chance* nodes in DDN are updated and a decision policy is generated. In this way, the system is able to adapt tutoring to the needs of the student and achicvc the objectives of the curriculum.

3 Evidence Model

The student' s responses are processed in the evidence model. Let V_{jk} be the evidence node that indicates the student' s (j) mastery state of learning objective k. Let X be the set of responses and $x_{ijk} \in X_{ik} \subseteq X$ be the response to item i which tests the k^{th} learning objective, then

$$\Pr(v_{jk} \mid x_{ijk}) \propto \Pr(v_{jk}) \prod_i \Pr(x_{ijk} \mid v_{jk})$$

where $\Pr(v_{jk})$ is the prior probability which can be obtained statistically from past data. $\Pr(x_{ijk} \mid v_{jk})$ is the likelihood of correct-answer score. An example of the likelihood function is $\delta_{ik} \exp(-b_i v_{jk})$ where δ_{ik} is the importance of knowing learning objective k so as to answer item i correctly and b_i is the difficulty index for item i.

4 The Student Model

The Student Model consists of a set of Bayes's nets, and each Bayes's net models the student' s mastery of a key concept (*goal*). In Section 4.1, the structure of the student model is defined. The construction of Bayes's net and the conditional probability assignment are discussed in Section 4.2. Instantiation of an evidence node activates a message passing process in the Bayes's net. This process results in the updating of marginal probabilities at the nodes. Most commercial software for developing probabilistic network possesses efficient algorithm [1] for implementing the message passing process.

4.1 Semantics of the Student Model

The Student Model is a directed acyclic graph (DAG) that represents a joint probability distribution of a key concept and several learning objectives. A node represents the learning objective as a random variable, and an arc represents possible probabilistic relevance or dependency between the variables. When there is no arc linking two nodes, it indicates probabilistic independence between the variables. In this study, the variables are classified into four types Evidence, Case, Learning Objective, and Goal as shown in Figure 2.

Symbol	Name	Function
N_v	Evidence Node	It contains knowledge states based on student' s response.
N_C	Case Node	It contains knowledge states that reflect ability to apply knowledge in different situations (cases).
N_L	Learning Objective Node	It contains knowledge states of key learning objectives (defined in the syllabus).
N_G	Goal Node	The concept student is expected to know. Each Bayes' s net must have at least one Goal node.

Figure 2: Types of Nodes in Student Model

More formally, a student model in *iTutor* is a DAG $\Im = (N,\psi)$ where $N = N_v \cup N_L \cup N_C \cup N_G$ are the nodes such that N_v is a set of evidence nodes, N_L is a set of learning objective nodes, N_C is a set of case nodes, and N_G is a set of goal nodes.

$\psi = \psi_{pL} \cup \psi_{pC} \cup \psi_{pG}$ are the arcs such that $\psi_{pL} \subseteq N \times N_L$ are arcs into learning objective nodes, $\psi_{pC} \subseteq N_v \times N_C$ are arcs from evidence nodes to case nodes, and $\psi_{pG} \subseteq (N_L \cup N_G) \times N_G$ are arcs from learning objective or goal nodes to the goal nodes.

Notice that evidence nodes have no parent node and only evidence nodes could be the parents of case nodes. Goal nodes are always sink nodes and they have parents that are either learning objective nodes or goal nodes. This signifies that mastery of a concept (goal node) is dependent on the mastery of learning objective(s) and/or pre-requisites (other goal nodes).

4.2 Construction of a Bayes's Net

Figure 3 shows a Bayes's net on mastery of a hypothetical concept (*goal*) "XYZ". Each node has three knowledge statcs: *non-mastery, partial-mastery,* and *mastery*. The granularity of Bayes's net depends on the number of nodes and its states. However, as the granularity becomes finer, the number of entries in the conditional probability table grows exponentially.

Values at the root nodes are known as *prior* probabilities while that at other nodes are *conditional* probabilities. To use the probabilistic network the random variables must be initialized with prior probability values. These values may be based on teacher's belief or past statistics. An intuitive method is to generate a probability table based on seven-category of thc difficulty of learning objectives (see Table 1). These probability values are to be input as the prior probability of the related evidence. The teacher also has the flexibility to amend the values based on their belief and context of usage. On the other hand, the probability values can be obtained from statistics of previous tests/examinations. A simple procedure for the use of past statistics is:

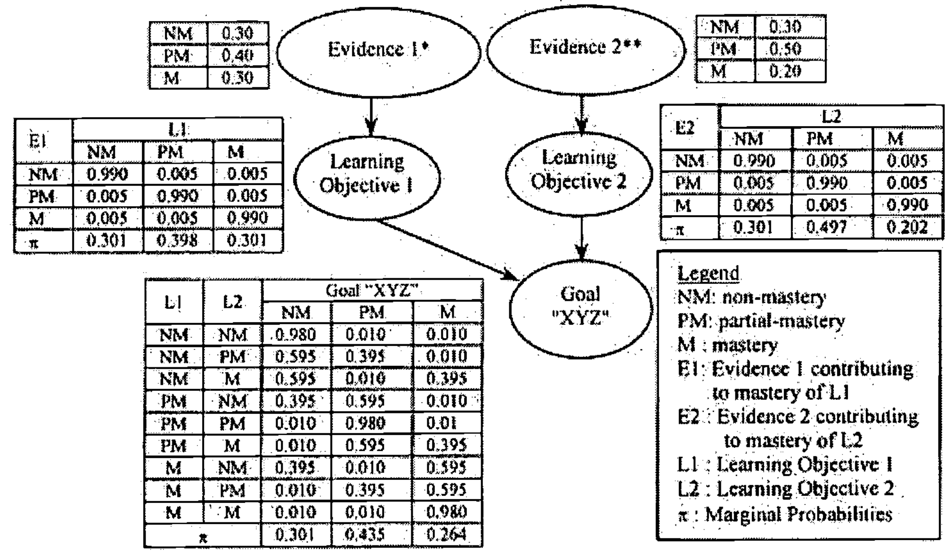
Table 1 Category of Difficulty in Mastering the Learning Objective

	Probability values		
Category	NM	PM	M
very easy	0.001	0.009	0.99
easy	0.01	0.09	0.90
fairly easy	0.05	0.15	0.80
Neutral	0.10	0.20	0.70
fairly difficult	0.20	0.30	0.50
difficult	0.30	0.40	0.30
very difficult	0.40	0.50	0.10

a) Assigned learning objectives to each question;

b) Enter student's responses (in terms of percentage) for the questions that she has answered;

c) Compute the average number of students (in percentage) for each mastery category: 040 (non-masterystate), 40-70 (partial-mastery state), and 70-100 (mastery state).



NM	0.30
PM	0.40
M	0.30

NM	0.30
PM	0.50
M	0.20

E1	L1		
	NM	PM	M
NM	0.990	0.005	0.005
PM	0.005	0.990	0.005
M	0.005	0.005	0.990
π	0.301	0.398	0.301

E2	L2		
	NM	PM	M
NM	0.990	0.005	0.005
PM	0.005	0.990	0.005
M	0.005	0.005	0.990
π	0.301	0.497	0.202

L1	L2	Goal "XYZ"		
		NM	PM	M
NM	NM	0.980	0.010	0.010
NM	PM	0.595	0.395	0.010
NM	M	0.595	0.010	0.395
PM	NM	0.395	0.595	0.010
PM	PM	0.010	0.980	0.01
PM	M	0.010	0.595	0.395
M	NM	0.395	0.010	0.595
M	PM	0.010	0.395	0.595
M	M	0.010	0.010	0.980
π		0.301	0.435	0.264

Legend
NM: non-mastery
PM: partial-mastery
M : mastery
E1: Evidence 1 contributing to mastery of L1
E2 : Evidence 2 contributing to mastery of L2
L1 : Learning Objective 1
L2 : Learning Objective 2
π : Marginal Probabilities

Note: * The prior probability for Evidence 1 is obtained from Table 1 (difficult Learning Objective 1)
** The prior probability for Evidence 2 is obtained from statistics

If a probability distribution function is able to describe the statistics, it can be used. In Figure 3, the values Pr(E2=non-mastery) = 0.30, Pr(E2=partial-mastery) = 0.50, and Pr(E2=mastery) = 0.20 are obtained from statistical data for this particular evidence. It is acceptable for another person to assign different probability values so long as it is consistent with the probability axioms [12]. Since the decision theory approach is normative rather than descriptive, it is able to explain the actions of the decision-maker.

For any node n_q, the conditional probability required to specify the Bayes's net is computed based on the relative importance (*weights*) of the parent nodes $pa(n_q)$ to itself.
If the state of n_q and pa(n_q) is the same, then $\Pr(n_q \mid \mathrm{pa}(n_q)) = \sum_{\mathrm{pa}(n_q)} (w_{pq} - (c-1)\kappa)$

$$\text{else } \Pr(n_q \mid \mathrm{pa}(n_q)) = \sum_{\mathrm{pa}(n_q)} \kappa \qquad (1)$$

where *c is the number of states and* $0 \le w_{pq} \le 1$.

$\kappa$is a constant and a measure of uncertainty such as careless errors, lucky guesses, changes in the student knowledge state due to learning and forgetting, and patterns of student responses unanticipated by the designer of the student model. The weights w_{pq} are either assessed based on the teacher's subjective judgment or past students' responses to closely related items.

Referring to Figure 3, since Learning_Objective_1 is dependent only on Evidence_1, w_1 = 1. Let Learning_Objective_1 has greater influence on mastery of goal "XYZ" than Learning_Objective_2, w_{1g} = 0.6, and w_{2g} = 0.4. Assigning κ= 0.005, the conditional probability tables can be computed using equation (1).

5 Tutoring Strategy

When a student logon to *iTutor*, the system automatically searches his ability index from the database. The ability index is either computed from the tests taken previously by the students, or from her knowledge states in the student model (see Section 5.1). Human tutors consider the student' s emotional state in deciding how to respond. Similarly in *iTutor*, the system considers factors such as response time, response pattern, student knowledge structure to determine tutoring actions: give more hint, help, ask another question, or stop the tutoring session. If the decision is to prompt another item, a learning objective and an appropriate item will be selected to coach her (see Section 5.2). Section 5.3 discussed the generation of tutoring strategy based on student' s response.

5.1 Mapping of Knowledge State to Student Ability

Let the student' s ability be $\theta_j = (\theta_{j1}, \theta_{j2}, ..\theta_{jm}, ..\theta_{jp})$. A function $f: v_{jm} \to \theta_m$ where v_{jm} is the evidence at the goal node (g) of m^{th} Bayes' s net. An example of such function is:

$$\theta_{jm} = \begin{cases} N(1.5, 0.6) & v_{jm} \geq 0.7 \\ N(0.5, 1) & 0.4 < v_{jm} < 0.7 \\ N(-1, 1.2) & v_{jm} \leq 0.4 \end{cases}$$

where N (μ, σ) denotes a *normal* distribution with mean μ and standard deviation σ

The computed ability index is then used to categorize (Advance, Intermediate, or Beginner) the student. An appropriate learning objective is selected based on the heuristic shown in Table 2. Value assignment is used to compute the path length of Bayes' s net and is used as preference for tutoring policy generation. They are as follow:

$$\begin{aligned} &\text{value}(G) = 0 \ \text{ for } G \in \{\text{Goal nodes}\} \text{and ch}(G) = \phi \\ &\text{value(ch}(N)) = 0 \text{ if ch}(N) = \phi \\ &\text{value}(N) = \text{value (ch}(N)) + 1 \ \text{ for node } N \\ &\text{where ch}(N) \text{ is the child node of } N \qquad (2) \end{aligned}$$

Table 2 Search Heuristic for Identifying Learning Objective for Coaching

Category	Identification of first Learning Objective for tutoring
Advance	25% of pathLength*
Intermediate	50% of pathLength
Beginner	75% of pathLength
New#	50% of pathLength (⌈pathLength/2⌉)

Note:

* pathLength denotes max(value(N)) ∀N ∈ Bayes's net.

The value(N) assignment is stated in equation 2

# The category *New* refers to students who login to the system the first time

5.2 Item Selection

Each item is tagged with an index (b_i) that estimates the minimum ability to answer it correctly with 0.5 probability. The items are assumed to be independent and the index obtained through statistic of past students' attempts or assigned using teacher' s belief. Subsequent update of item difficulty index may be performed through item response theory [4] such as Rasch model [10].

From the set of items related to a learning objective, an item i is selected based on: $\theta_j - b_i < \varepsilon$ where ε is a

Table 3 Utility of Various Outcomes

Condition / Expression	Preference
Decision: Stop	
$S(N)$ = "M" & value(N) = 0	1
$S(N)$ = "N" & value(N) = 0	0
k = number of $N_k \in \{(N_k, S_k)\}$ with same S_k	$1 - k/5$
number of tries, n, for the same learning objective	$1 - n/5$
Decision: Ask item on same N	
$S(N)$ = "M"	0
$S(N)$ = "N"	1
$S(N)$ = "P"	$\left(1 - \frac{\text{value(N)}}{\text{pathLength}}\right) \times \eta$ where η is a constant
Decision: Ask item on ch(N)	
$S(N)$ = "M"	1
$S(N)$ = "N"	0
$\gamma = \max(\Pr(S(\text{ch}(N)) = \text{"M"} \mid x = 1) - \Pr(S(\text{ch}(N)) = \text{"N"} \mid x = 0)$	
Decision: Ask item on pa(N)	
$S(N)$ = "M"	0
$S(N)$ = "N"	1
$\gamma = \max(\Pr(S(\text{pa}(N)) = \text{"M"} \mid x = 1) - \Pr(S(\text{pa}(N)) = \text{"}N\text{"} \mid x = 0))$	

Remarks: $S(N)$ denotes the knowledge state of node N
ch(N) denotes child node of node N

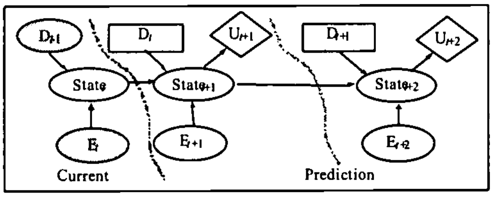


Figure 4: Dynamic Decision Network

Table 4 Utility Values for Item Difficulty Level Selection

Current Knowledge State ($State_t$)	Current Response (E_t)	Next Question Typ (D_{t+1})		
		Easy	Ave.	Diff
Non-Mastery	Correct	-0.2	1	0
	Wrong	0.4	-0.2	-0.4
Partial Mastery	Correct	-0.4	0.2	1
	Wrong	0.4	1	-0.2
Mastery	Correct	-0.4	-0.2	-0.2
	Wrong	-0.2	0.2	-0.2

pre-defined small value. This ensures selected item is challenging and likely to be solved by the student. Teacher' s solution will be displayed upon student' s request so that she can learn from her mistake. This strategy assumes student' s ability is dynamic and can be raised to higher levels through self-paced computer-aided tutoring.

5.3 Tutoring Policy Generation

To bring the probabilistic network one step closer to being a useful *intelligent tutoring system*, automated decision-making capability has been added. When asked to provide a tutoring policy for the student, the system generates a course of action based on her current mastery states. The tutoring policy aims to use a series of items with differing difficulty to determine more precisely her mastery of specific learning objectives. Items are categorized into *easy*, *average* and *difficult*. In this project, a two-step look-ahead dynamic decision network is recommended so as to compromise between the need to invoke policy generation routine for a decision and the long computing time to generate policy with many decisions.

Figure 4 shows a dynamic decision network (DDN) used in this study. In addition to the decision nodes for current and fiture time steps, the DDN also contains the previous decision, D_{t-1}, as an evidence node. When the evidence for state t arrives, the probability distributions of $State_t$ are updated [1] using the prediction-estimation process (see Figure 5). After the initial prediction of probabilities (Bel*), State_{t+1} estimates the new belief based on projected evidence [13]. This process repeats for State_{t+2}. Eventually, the expected utility is evaluated by a sequence of summations and maximizations. Tables 3 and 4 show the utility functions for node U_{t+2}. Selecting the outcomes with maximum expected utility value constitute the tutoring policy.

Define a Bayes's net with nodes N,
E_t be the evidence at time t,
$\text{Bel}(N)$ be a vector of probabilities (updated over time),
d_t be the decision at time t, and
α be the normalization constant.

Prediction 1: $\text{Bel}^*(N_{t+1}) \leftarrow \Sigma_{N_t} \Pr(N_{t+1} \mid N_t, d_t)\text{Bel}(N_t)$

Estimation 1: $\text{Bel}(N_{t+1}) \leftarrow \alpha\Pr(E_{t+1} \mid N_{t+1})\text{Bel}^*(N_{t+1})$

Prediction 2:
$\text{Bel}^*(N_{t+2}) \leftarrow \Sigma_{N_{t+1}} \Pr(N_{t+2} \mid N_{t+1}, d_{t+1})\text{Bel}(N_{t+1})$

Estimation 2: $\text{Bel}(N_{t+2}) \leftarrow \alpha\Pr(E_{t+2} \mid N_{t+2})\text{Bel}^*(N_{t+2})$

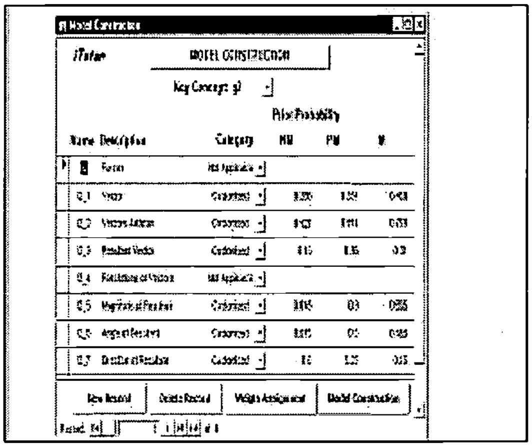
$$\text{Expected Utility}: \sum_{N_t} \text{Bel}(N_t) \max_{d_t} \Sigma_{N_{t+1}} \Pr(N_{t+1} \mid N_t, d_t) \times \max_{d_{t+1}} \Sigma_{N_{t+2}} \Pr(N_{t+2} \mid N_t, d_t, N_{t+1}, d_{t+1}) \times U(N_t, d_t, N_{t+1}, d_{t+1}, N_{t+2})$$

Figure 5: Prediction-estimation Process

6 An Illustration

6.1 Construction of a Decision Network

In this project, the construction of all probabilistic networks is performed using Netica API [7]. A module leader enters the learning objectives and the weights of the key concept *Forces* using Microsoft Access [6]. The probabilistic values shown in Figure 6 are entered based on past examination results. By clicking the button "Model Construction", a Bayes's net (see Figure 7) and a decision network (see Figure 8) on "Forces"



			Prior Probability		
Name	Description	Category	NM	PM	M
[illegible]	Forces	Not Applicable			
e2_1	Vectors	Categorized	[illegible]	[illegible]	[illegible]
e2_2	Vectors Addition	Categorized	[illegible]	[illegible]	[illegible]
e2_3	Resultant Vector	Categorized	[illegible]	[illegible]	[illegible]
e2_4	Resolution of Vectors	Not Applicable			
e2_5	Magnitude of Resultant	Categorized	[illegible]	[illegible]	[illegible]
e2_6	Angle of Resultant	Categorized	[illegible]	[illegible]	[illegible]
e2_7	Direction of Resultant	Categorized	[illegible]	[illegible]	[illegible]

Figure 6: A Snapshot on Data Entry for Model Construction

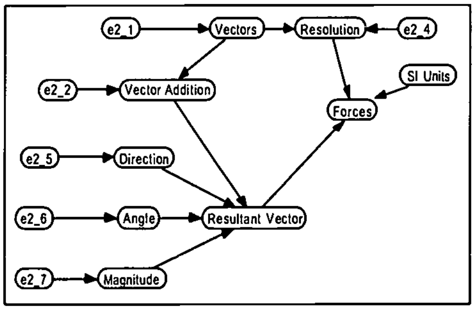


Figure 7: Bayes's Net on *Forces*

will be created. Teachers who are familiar with Netica application [8] can use the generated Bayes's net to perform *what-if*analysis. For exa mple, a teacher may want to determine the likely student's improvement if he provides remedial instructions on "Resolutions of Vectors". He can do so by instantiating the evidence node *e2_4* to "Mastery" state, and observe the probability of mastery in the goal node labeled *Forces*.

6.2 Diagnosis of a Student's Misconceptions

The items to be presented to the students are coded by the teacher using Scientific Notebook [15]. With *iTutor*, the teacher is able to monitor student's progress through the database management tool. Figure 9a shows a snapshot of a student who had answered item "Force_001" correctly and partially correct for item "Force_004". The teacher can track a student's mastery states by clicking the "Advice" button. The system transforms the responses to evidence, and instantiates the evidence nodes in the Bayes's net as shown in Figure 9b. The posterior mastery states are displayed (see Figure 9c). The output also provides the teacher information on specific learning objectives to tutor. In addition, he can also examine the detailed strategy by clicking the "Tutorial Strategy" button. This action causes the generation of a decision network (see Figure 9d). Figure 9e shows items to be posed to the student if she continues with the online tutorial. At any stage, the teacher may intervene by providing personal coaching.

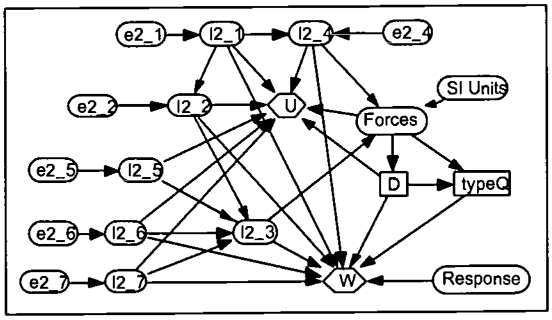


Figure 8: Decision Network on *Forces*

7 Conclusions

Presently, the students' knowledge states remain unchanged until additional evidence is available. The system also uses a constant learning rate for all students. One future direction is to include additional parameters to model student forgetting and learning rates. Another area is to provide a user interface for teachers not familiar with Netica application to perform *what-if* analysis. In this way, the teacher will be able to focus on student's issues rather than to learn another software tool. The next future direction is to include probability functions other than *Normal* distribution. This is essential when the ability distribution of student cohort is not symmetric.

A significant result of this project is the use of Bayesian networks to generate sound probabilistic inferences. Another contribution is the automation of decision networks construction. The recommended strategy is used in adaptive tutoring. With *iTutor*, teacher is able to monitor the student's progress and yet had time for lesson preparation and coaching of weaker students. In addition, the teacher has accessed to the student's knowledge states and actions taken by *iTutor* at every stage of the tutoring process. Moreover, it enables students to have tutorials customized to their needs.

References

[1] Castillo, E., Gutiérrez, J. M., & Hadi, A.S., "Expert Systems and probabilistic network Models", Springer-Verlag, (1997).

[2] Conati C. & Vanlehn K., "Teaching meta-cognitive skills: Implementation and evaluation of a tutoring system to guide self-explanation while learning from examples", Proceedings of AIED '99, 9th World Conference of Artificial Intelligence and Education, Le Man, France, (1999).

[3] Deam, T. L. & Wellman, M. P., "Planning and Control", Morgan Kaufmann, San Mateo, California, (1991).

[4] Hambleton, R.K., Swaminathan, H., & Rogers, H. J., "Fundamentals of Item Response Theory", Sage Publications, Inc, (1991).

[5] Howard, R.A. & Matheson, J.E. "Readings on the Principles and Applications of Decision Analysis", Menlo Park, California: Strategic Decisions Group, (1983).

[6] Litwin, P., Getz, K., & Gilbert, M. , "Access 97 Developer's Handbook", 3rd edition, Sybex Inc, (1997).

[7] Norsys, "Netica API Programmer's Library Reference Manual", version 1.06, Norsys Software Corp., (1999)

[8] Norsys, "Netica Application Reference Manual", version 1.12, Norsys Software Corp., (1999).

[9] Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann Publishers, Inc, (1988).

[10] Rasch, G., "Probabilistic Models for Some Intelligence and Attainment Tests", Danish Institute for Educational Research, (1960).

[11] Reye, J., "A belief Net Backbone for Student Modeling, Proceedings of Intelligent tutoring Systems", Third International Conference, ITS '96, Springer-Verlag, (1996).

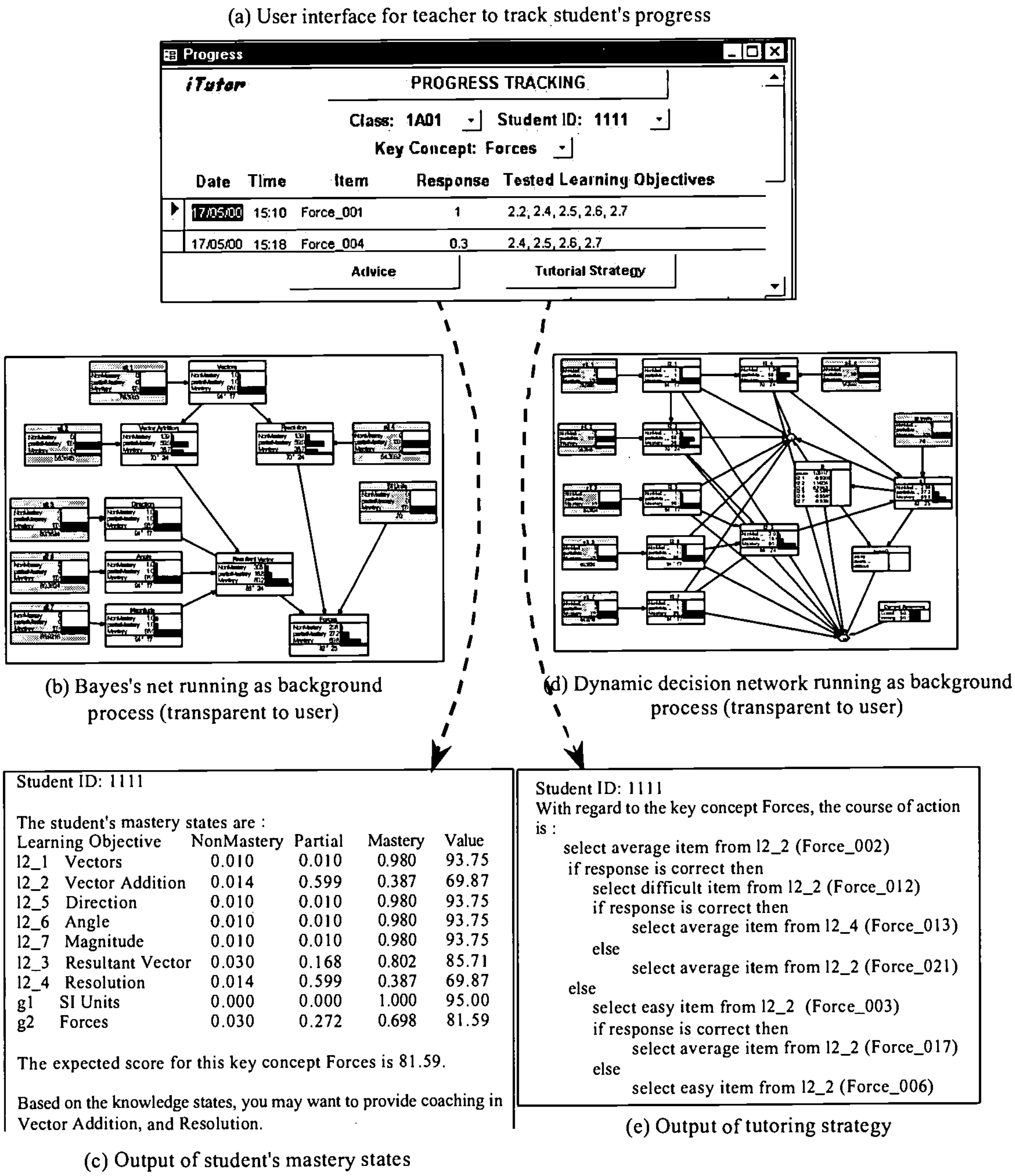
[12] Ross, S., "A First Course in Probability", Prentice-Hall International, Inc., 5th edition, (1998).

[13] Russell, S. & Norvig, P. , "Artificial Intelligence: A Modern Approach", Prentice Hall, (1995).

[14] Simon H.A., "The New Science of Management Decisions", New York: Harper & Row, (1960).

[15] TCI Software Research, "Scientific Notebook version 3.00 Build 939", Brooks / Cole Publishing Company, (1998).

[16] Villano, M., "Probabilistic Student Models: Bayesian Belief Networks and Knowledge Space Theory", Proceedings of Intelligent Tutoring Systems: Second International Conference, ITS '92, Springer-Verlag, (1992)



(a) User interface for teacher to track student's progress

PROGRESS TRACKING

Class: 1A01 Student ID: 1111

Key Concept: Forces

Date	Time	Item	Response	Tested Learning Objectives
17/05/00	15:10	Force_001	1	2.2, 2.4, 2.5, 2.6, 2.7
17/05/00	15:18	Force_004	0.3	2.4, 2.5, 2.6, 2.7

(b) Bayes's net running as background process (transparent to user)

(d) Dynamic decision network running as background process (transparent to user)

Student ID: 1111

The student's mastery states are :

Learning Objective		NonMastery	Partial	Mastery	Value
12_1	Vectors	0.010	0.010	0.980	93.75
12_2	Vector Addition	0.014	0.599	0.387	69.87
12_5	Direction	0.010	0.010	0.980	93.75
12_6	Angle	0.010	0.010	0.980	93.75
12_7	Magnitude	0.010	0.010	0.980	93.75
12_3	Resultant Vector	0.030	0.168	0.802	85.71
12_4	Resolution	0.014	0.599	0.387	69.87
g1	SI Units	0.000	0.000	1.000	95.00
g2	Forces	0.030	0.272	0.698	81.59

The expected score for this key concept Forces is 81.59.

Based on the knowledge states, you may want to provide coaching in Vector Addition, and Resolution.

(c) Output of student's mastery states

Student ID: 1111

With regard to the key concept Forces, the course of action is :

```
select average item from 12_2 (Force_002)
 if response is correct then
     select difficult item from 12_2 (Force_012)
     if response is correct then
           select average item from 12_4 (Force_013)
     else
           select average item from 12_2 (Force_021)
 else
     select easy item from 12_2  (Force_003)
     if response is correct then
           select average item from 12_2 (Force_017)
     else
           select easy item from 12_2 (Force_006)
```

(e) Output of tutoring strategy

Figure 9: Overview of an *iTutor* Session