Using an ITS as an Arithmetic Assistant for Teachers – 3-year Review

Chun-Hung Lu[1,2]
[1] Dept. of Information Management, National Taiwan University, Taiwan
d95006@im.ntu.edu.tw

Chorng-Shyong Ong[1]
[1] Dept. of Information Management, National Taiwan University, Taiwan
ongcs@im.ntu.edu.tw

Wen-Lian Hsu[2]
[2] Institute of Information Science, Academia Sinica, Taiwan
hsu@iis.sinica.edu.tw

Abstract

This paper summarizes the results of three years of research into using an ontology-based system called InfoMap for e-learning. Using InfoMap, we develop an empirical platform to collect information about mathematical errors made by elementary school students in Taiwan. In our intelligent tutoring system (ITS), InfoMap represents the curriculum, as well as the knowledge of experts and teachers. We use the relation between the curriculum and expert knowledge to analyze students' arithmetic errors and determine where the system's knowledge is deficient. Following the work of Brown and Burton, we develop a method called Process Map to describe procedural knowledge in our ITS. After a three-year empirical study, we found that eleven error types in Brown and Burton's "Buggy Model" do not apply to Taiwanese students. Significantly, we also found that some errors appear to apply only to Taiwanese students.

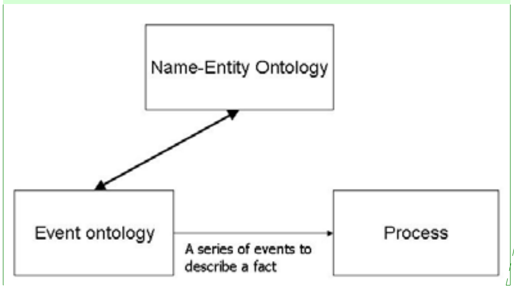
Keywords: Deficient knowledge, Error Model, Intelligent Tutoring System (ITS), Ontology, Procedural Knowledge, Student Model, InfoMap ,Process Map

Introduction

Extensive studies have been made of mental representations of knowledge. A person's cognition generally begins with noticing and remembering, and mental representation is called upon to provide knowledge. According to Anderson (1993), spatial images are comprised of two basic components: declarative knowledge and procedural knowledge. . Declarative knowledge refers to an individual's factual or conceptual knowledge. In designing a generic architecture to represent procedural knowledge, the actions defined by domain experts and control of the action flow are two important tasks. From our existing ontology tool, InfoMap, we adopt the concept of a service-oriented architecture to register all services, and use terminology that is

OWL compliant to describe the composition of services. Self (1999) showed that focusing on the process whereby knowledge is constructed is more important than focusing on the target knowledge itself. By using the descriptions of classes, properties, and instances, as well as the descriptions of their relationships in an ontology, our system can provide more robust reasoning functions.

In this research, we propose an ontological representation scheme called Process Map (PM) that represents procedural knowledge. The combination of a behavior model (procedural knowledge) and an ontology (declarative knowledge) has the advantage of allowing access to existing domain specific glossaries, taxonomies and ontologies from within the processes registered in our Process Map Management system. If we regard the repeated processes as reusable components, then we can identify (1) the activity structure from given behavioral models of components; (2) the correlations among these components; and (3) the log of information about a student's operational behavior. Most researchers use declarative knowledge as the sole basis for ontology simulation. However, in Sowa's opinion, the paradigm of declarative knowledge construction has largely failed to produce human-like cognitive processing in computers (Sowa, 1999). To address this situation, we have developed an ontology called InfoMap (Hsu, 2001) that incorporates both declarative and procedural knowledge (see Figure 1).



註解 [P1]: **Top box**: Named Entity Ontology. (Highlighted in the 1st revision and discussed last Tuesday.)

Figure 1. The conceptual architecture of our ontology

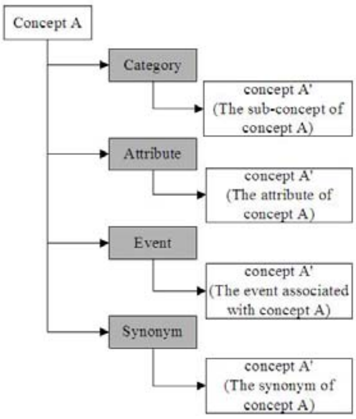


Figure 2. Ontology format of InfoMap

Representation

Our ontology is based on InfoMap, which was originally created as a named entity ontology. We extend InfoMap to include an event ontology and a process map, and use it to represent an expert module, a student model, and a curriculum.

InfoMap

Knowledge representation has long been an obstacle to simulating human understanding. Although several strategies have been proposed for natural language understanding, many have been confined to discussions in textbooks, rather than actually implemented in large-scale natural language systems. One difficulty is that different situations require different representation schemes.

Our knowledge representation scheme, InfoMap, is designed to facilitate both human browsing and computer processing of a system's domain ontology, which is constructed from structured concepts in each specific domain. Examples of concept structures range from simple concepts, such as a word, a phrase, or an event, to more complex concepts, such as a sentence, a paragraph, a script (a collection of related events), a story, and the passive tense of English. Each concept is associated with a structure (a sub-map) describing the relationships of the concept to its related concepts. The system can store a large number of events, syntactic or semantic structures, and scripts. Given a natural language sentence, the system tries to match it to a sub-map or decompose it into several events within InfoMap.

We represent InfoMap as a tree hierarchy that contains two types of nodes, concept nodes and function nodes, as shown in Figure 2. The basic function nodes, i.e., category, attribute, synonym, and event, are used to label the relationships between two concept nodes.

Process Map

Process Map (PM), which is a means of representing procedural knowledge, can be treated as a series of processes connected by junctions and links. The direction of the flow of each instance is decided by the preconditions of each process. The steps of problem-solving can also be recorded, as long as the processes are clearly defined. This is useful if we want to detect the errors in, or track the history of, a procedure. In this section, we explain how to construct a PM and how to use it to describe procedural knowledge. In Section 4, we introduce the prototype of the process engine system.

The Structure of Process Map

The architecture of Process Map is shown in Figure 2. PM uses basic subtraction and is adapted from Brown and Burton (1978). Suppose we have a problem (p1):

```
        T3 T2 T1
-       B3 B2 B1
-----------------
```

Figure 3 shows the procedure for solving problem p1. The grey boxes are composite processes that can be further decomposed. For example, process C is a composite process that can be decomposed into processes F and G. Process G can be further decomposed into processes H, I and J. Actually, processes K, L, M, N, and O can also be represented as composite processes, but we do not show them due to space limitations. In process I, we use a different method to show the same composite idea. A white box indicates an atomic process with or without preconditions or effects. The junction "Or" indicates a one-to-many relationship and a temporal constraint between the processes connecting them (Chen-Burger, Tate and Robertson, 2002). We provide more details in the section "Representation of a Process".

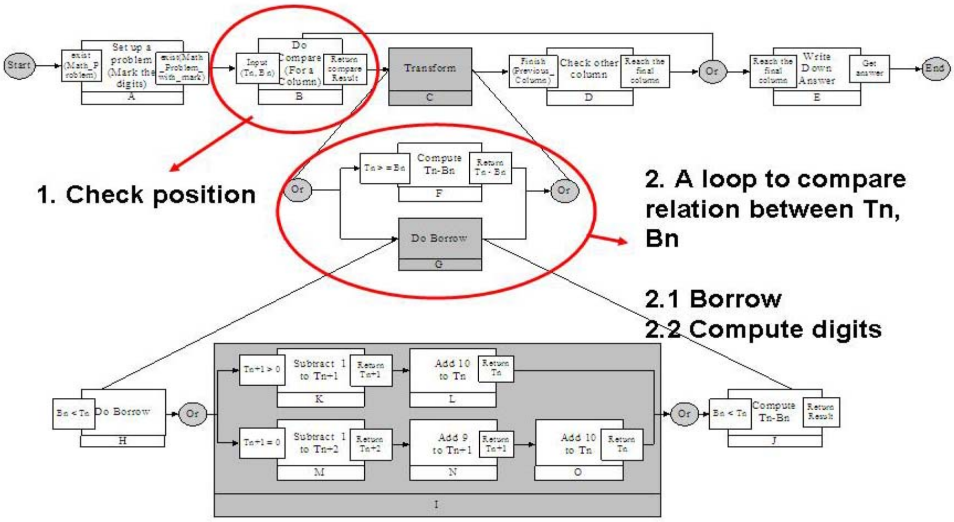


Figure 3. Visualization of Process Map

The PM in Figure 3 is an XML-format knowledge representation of the procedural knowledge in Figure 2. The original idea of PM was derived from a business process modeling language called FBPML, and a DAML-based web service ontology called DAML-S. FBPML is a visual and conceptual language that captures and describes an organization's business processes. Any business procedure can be expressed using this language (Kuo, 2002). We borrow the concepts of junctions and links from FBPML because they both play an important role in decisions about the flow of the processes. The main structure of PM is derived from the ontology for process models described in DAML-S. The primary function of the process ontology in DAML-S is process decomposition. Processes can be categorized as "Atomic", "Simple", or "Composite" (DAML Services Coalition, 2002). We adopt the first two categories. A simple procedure can be represented as a single atomic process, while a complicated procedure can be represented as a composite process (or several composite processes). The latter can be further decomposed into many composite processes or atomic processes. The advantage of this model is that it presents different views of the same procedure, so that the procedures can be represented in a more structured way. Figure 4 shows the part of the process map that represents the procedural knowledge presented in Figure 3.

Figure 4: A Math Subtraction Process Map

Representation of a Process

In PM, a process can be categorized as atomic or composite with different parameters. An "atomic" process has three properties: ID, processName, and type; and five attributes: precondition, input, output, effect, and action.

```
- <processMap processMapName="Math Subtraction">
  - <process ID="P" processName="Math Subtraction" type="Composi
    + <controlConstruct type="Sequence">
    </process>
  - <process ID="A" processName="Setup a problem" type="Atomic">
    - <attribute>
      - <precondition>
        - <language type="IASL">
          - <logic type="AND">
              <number>N1</number>
              <number>N2</number>
            </logic>
          </language>
        </precondition>
      - <input>
          <parameters>N1</parameters>
          <parameters>N2</parameters>
        </input>
      - <output>
          <parameters>T(1)</parameters>
          <parameters>T(2)</parameters>
          <parameters>T(3)</parameters>
          <parameters>B(1)</parameters>
          <parameters>B(2)</parameters>
          <parameters>B(3)</parameters>
        </output>
      - <effect>
          <hasmark>N1</hasmark>
          <hasmark>N2</hasmark>
        </effect>
        <action>Mark the digitals</action>
      </attribute>
    </process>
```

```
- <process ID="B" processName="Do Compare(for a column)" type="Atomic">
  - <attribute>
    - <precondition>
      - <language type="IASL">
        - <logic type="AND">
            <hasmark>N1</hasmark>
            <hasmark>N2</hasmark>
          </logic>
        </language>
      </precondition>
    - <input>
        <parameters>T(N)</parameters>
        <parameters>B(N)</parameters>
      </input>
    - <output>
        <compare>Result</compare>
      </output>
      <effect />
      <action>compare the digits</action>
    </attribute>
  </process>
- <process ID="C" processName="Transform" type="Composite">
  - <controlConstruct type="Split-Joint">
    - <junctions>
        <junction type="Or" />
        <junction type="Or" />
      </junctions>
    - <processes>
        <process RID="F" />
        <process RID="G" />
      </processes>
    </controlConstruct>
  </process>
```

```
- <process ID="F" processName="Compute" type="Atomic">
  - <attribute>
    - <precondition>
      - <language type="IASL">
        - <logic type="NOT">
          - <smaller>
              <parameters>T(N)</parameters>
              <parameters>B(N)</parameters>
            </smaller>
          </logic>
        </language>
      </precondition>
    - <input>
        <parameters>N1</parameters>
        <parameters>N2</parameters>
      </input>
    - <output>
      - <result>
        - <minus>
            <parameters>T(N)</parameters>
            <parameters>B(N)</parameters>
          </minus>
        </result>
      </output>
      <effect />
      <action>Compute T(N) - B(N)</action>
    </attribute>
  </process>
- <process ID="G" processName="Transform" type="Composite"
  - <controlConstruct type="Sequence">
      <junctions />
    - <processes>
        <process RID="H" />
        <process RID="I" />
        <process RID="J" />
      </processes>
    </controlConstruct>
  </process>
  .........
</processMap>
```

Figure 4: A Math Subtraction Process Map

Properties: An ID defines a process and must be unique for each process in PM. A processName is the name of a process; it has nothing to do with the execution of the process, and is only used for human interpretation. A type indicates whether the process is atomic or composite.

Attributes: A precondition controls the execution of a process. Procedural knowledge can be represented by a series of "If-Then" rules. The preconditions of a process are very important because they determine whether the process will be executed or not. In Figure 3, the preconditions of process A stipulate that both the minuend and the subtrahend must be numbers so that the process of setting up a problem can be executed. In PM, some basic precondition terms are provided, such as "check if a variable is a number" and "compare two different variables". They can be composed in a logical form (i.e., "And", "Or" and "Not" can be used to form more complicated preconditions).

An output represents the execution results of the process. Sometimes it may also provide useful information about the preconditions of the next process. Effects indicate the additional effects (or the state changes of the object) produced by the process, but they do not belong to the output. In process A (Fig. 3), the problem marked by an ellipse is an effect of the process. For example, suppose we want to describe a reservation process. The output of the process ConfirmReservation will be a ReservationNumber. The effect will be a HaveFlightSeat status. An action tag indicates the action that will be executed in this process. We discuss this point further in the section "Logical Meaning of Junctions".

The properties of a composite process are the same as those of an atomic process, namely, ID, processName, and type. There is an additional attribute, called controlConstruct, which represents different compositions of the structures of the processes. The idea is derived from DAML-S.

We use two composition methods: Sequence and Split-Joint. A Sequence is the simpler control construct whereby the processes are executed sequentially. In a Split-Joint operation, which is more complicated than a Sequence, the processes can be executed in parallel if more than one precondition of a process is satisfied. Then, all the Split-Joint processes will be triggered and executed. A Split-Joint operation can be considered as a decision point, where the direction of the flow can be different as long as different data (information) is provided. The controlConstruct tag contains a junction tag and process tags, which represent the type of junction and the IDs of the connecting processes, respectively. For example, in Figure 3, process C is a Split-Joint composite process composed of processes F and G with Or-Or junctions. Next, we explain the logical meaning of junctions, such as And-And, And-Or, Or-And, and Or-Or, in detail. A junction is represented as:

```
<controlConstruct type="Split-Joint">
        <junctions>
                <junction type="Or"/>
                <junction type="Or"/>
        </junctions>
        <processes>
                <process RID="F"/>
                <process RID="G"/>
        </processes>
</controlConstruct>
```

The Logical Meaning of Junctions

Processes are connected by junctions, and a sequence is the simplest type of connection. A sequence does not have any logical value and only represents the order in which the connected processes will be executed. Note that a process will be interrupted if its preconditions cannot be satisfied. A split is a point at which a process can be divided into other processes. A joint is a point at which two processes can be joined. In PM, joints are always paired. The topologies can be divided into four types: Or-Or, Or-And, And-Or, and And-And (Chen-Burger, Tate and Robertson, 2002). In Figure 5(a), an And-And junction is composed of And_Split and And_Joint junctions.

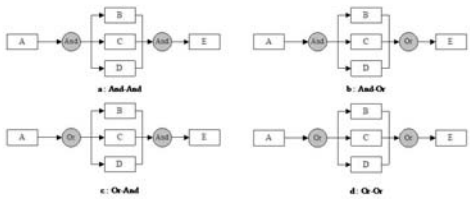


Figure 5. The different topologies of junctions

An And-And (And_Split and And_Joint) junction (Figure 5 (a)) indicates that processes B, C and D must start when process A finishes. After processes B, C, and D finish, process E can start. The And-And combination places the strictest restriction on a PM. The combination of Or-Or, shown in Figure 5(d), means that, after process A finishes, at least one of the processes B, C, or, D will be executed. Process E will not be started unless one of the triggered processes finishes. Let us suppose that processes B and D start when process A finishes. After B or D finish, then E can start. It does not need to wait for both B and D to finish. Thus, the Or-Or junction is a looser constraint than the And-And junction.

And-Or means that when process A is finished, processes B, C and D must be executed. If B, C or D finishes, process E can start. This is different from an And-And junction, which requires B, C, and D to finish before E can start. The Or-And junction indicates that at least one of the processes B, C or D will be executed after process A finishes, but process E will not start unless all of the triggered processes have finished. The triggered processes may be a combination of some processes (B + C, B + D, and so forth). Because this is an Or_Split junction, it is not necessary to trigger all the preceding processes.

Thus, it is more flexible than an And_Split junction. The different combinations of junctions, “And-And”, “And-Or”, “Or-Or”, and “Or-And”, can be used to represent and describe a complicated PM, which we consider next.

In a complicated PM, a junction can be used to represent concurrent processes, which may be Or-Or operations that can be executed separately, or And-And operations that must be executed concurrently. A junction can represent a decision point that determines different flows of a PM. For example, as shown in Figure 3, an Or-Or junction is used to compare T(N) and B(N) in problem p1. This may produce different execution results in a PM based on different input data. Each instance may have a different flow as long as different data is provided. If we want to apply this to a problem-solving procedure, we can define suitable preconditions for different processes. To monitor the flow of different instances (students), we can track the flow and extract some useful information to build a student model. We explain this point further in the section “The Teacher/Curriculum Manager Model”.

Actions

Recall that PM is used to describe a procedure. Actions are the different types of execution behavior in a PM, and can be provided by experts, teachers, or other teaching systems. Because actions are stored in a repository, relevant information about the action, such as its purpose and input parameters, should also be provided. When we describe a PM, this information is used to find suitable actions for a particular purpose. A registry and a search mechanism are provided in the action repository. The advantage of the action repository is that actions can be reused and experts’ experiences can be shared.

The repository is also flexible in that teachers can design a new PM to satisfy a new instructional goal by reorganizing the actions.

System architecture

We now introduce our system’s architecture, which is comprised of three layers, as shown in Figure 6(a). The architecture uses the same representation scheme as InfoMap to manage Process Map. Based on this representation, the modules in the architecture form a dynamic cycle (shown in Figure 6b).

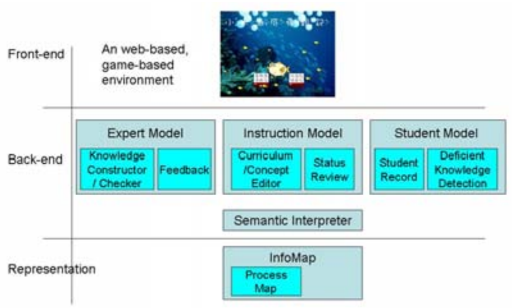


Figure 6a. The architecture of our system

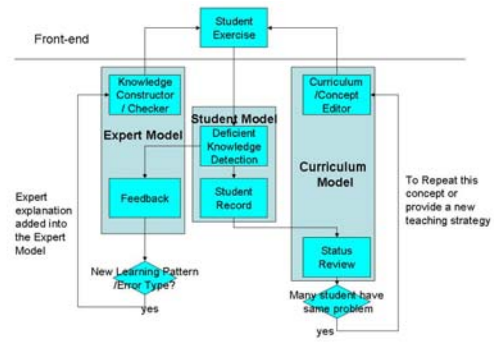


Figure6b. The process cycle of our system

In the front-end, we use a game-based environment in which each game stage is designed for one learning session that is linked to the curriculum’s sequence of sub-stages containing several smaller game units. In each stage, questions for the current learning session are loaded. If the student answers the questions with 70% accuracy, the next stage will be shown with another contextual learning session. However, if the student's correct answer rate is below 60%, the system will provide another easier, contextual learning session.

In the arithmetic problem (p1) shown in Figure 3, we can use T3*100 + T2*10 + T1 to represent augends, B3*100 + B2*10 + B1 to represent addends, and X,Y,Z to represent answers (the sum). If n represents the position, the arithmetic handles the relations between Tn and Bn. The expert knowledge of the arithmetic represented by InfoMap is shown in Figure 8.

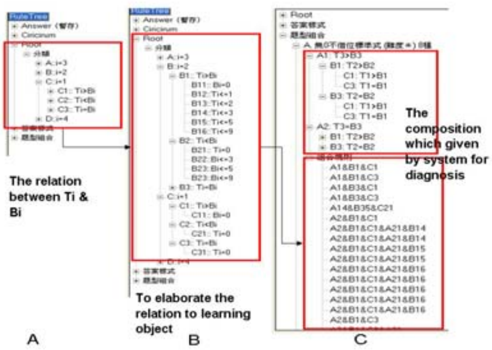


Figure 8. The composition given by the diagnostic system

The curriculum manager, shown in Figure 9, arranges the learning modules (the lesson plans), which are designed by teachers. Each module contains one or more learning objects, each of which has its own teaching strategy. The strategy and the curriculum can both be represented by InfoMap. The curriculum map gives teachers a more comprehensive understanding of what they should be prepared to teach. It eliminates sequencing errors, and helps teachers develop lessons that are truly interdisciplinary (Martin, 1994). The map is similar to an outline or a flowchart and can be described by PM. Every element in the curriculum map can be regarded as a composite process that can be divided into more detailed processes. Finally, we represent the subtraction procedure (Brown and Burton, 1978) by the Process Map shown Figure 3. The map can be used to represent the teaching strategies in the curriculum manager, and also to arrange post-conditions with error types. After teachers have collected students' problem-solving procedures and error types, they can update new learning maps for the students. The curriculum manager creates a continuous cycle: "curriculum design, teaching strategy design, recording (student's learning behavior), error analysis, and feedback on teaching strategies", which helps other teachers create good learning maps for the students.

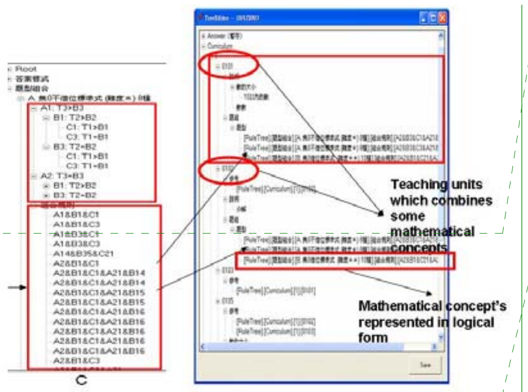


Figure 9 Mathematical concepts represented in logical form

註解 **[P2]:** **B:** The meaning of "To elaborate the relation to learning object" is not clear.
C: Change to "The composition given by the diagnostic system".
(Both points discussed last Tuesday.)

註解 **[P3]:**
Change to: 1) Teaching units, which combine some mathematical concepts
2) Mathematical concepts represented in logical form

3-Year Empirical Study

Due to the complexity of ITS, we evaluate our system with a mixed model, as shown in Figure 10.

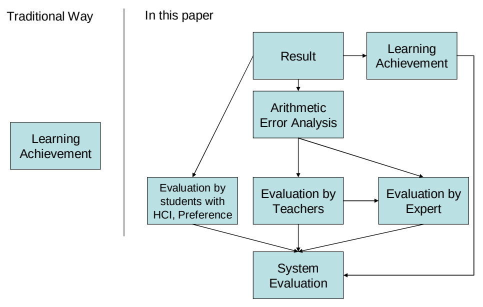


Figure 10 System evaluation model

System acceptance

According to our questionnaire, which is presented in the Appendix, the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy of 0.57 is borderline acceptable; therefore, it can be used for factor analysis. To assess the matrix for psychometric adequacy, we use Bartlett's sphericity test, the Kaiser-Meyer-Olkin measure of sampling adequacy, and inspection of the off-diagonal elements of the anti-image covariants matrix. These measures suggest that the correlation matrix is suitable for factor analysis. As well as the KMO measure, Bartlett's sphericity test demonstrates that the items are interdependent (X2 = 1074.02447492174, df = 219, P < 2.77-123).

A cross-section regression model yields the following results:

By using "I'm satisfied with this tutoring system" as an independent variable, the responses "I like to be tested by the game-based UI" and "I like the system generated questions" are significant. The reason is that we provide a game-based environment.

When using "Compared with this type of assessment, I would like to use a pencil and paper test" as an independent variable, the responses "I like using handwriting to answer a question" and "I would like to be instructed by a scenario-based voice" are significant. The experts explained the reasons for such choices are deeply rooted on our cognition. When "Using the tutoring system in the assessment will enhance my learning achievement" is treated as an independent variable, the responses "I think the tutoring system's user interface is clear" and "I think the system generated items are more difficult than traditional ways" are significant. According to several students, the system provides quick responses, which strengthens their learning motivation.

When "Using the tutoring system for assessment is better than a pencil and paper test" is used as an independent variable, the responses "I think the tutoring system's user interface is clear" and "I like the system generated items" are significant.

Proof of correctness

From our empirical study, we obtained two sets of evaluation results, as shown in Table1. The table compares the teacher's evaluations and the system's suggestions. The results are divided into three groups as follows.

1. All the same: the teacher's evaluation of a student's error is the same as the system's suggestion.
2. Partially similar: the teacher's evaluation of a student's error is partially similar to the system's suggestion.
3. Totally different: the teacher's evaluation and the system's suggestion are totally different.

The results show that the system correctly diagnoses errors in 77.32% of cases.

Table 1 Teacher’s evaluation of student errors compared to the system’s suggestions

	Result	Addition	Sub-traction	sum
All the same	1.1 The evaluation is the same	32	17	49
	1.2 Unable to judge either result	0	1	1
Partially similar	2.1 The system provides more information than the teacher.	68	16	84
	2.2 The teacher provides more information than the system	12	4	16
Total Different	3.1 The system cannot give any suggestion, but the teacher makes some suggestions.	11	4	15
	3.2 The teacher cannot give any suggestions, but the system provides some suggestions.	20	1	21
	3.3The evaluation of the teacher and the system’s suggestion are totally different.	8	0	8
Illegible sample		4	4	8

Table 2 The experts’ assessments (shown in parentheses) of the results in Table 1

	Result	Addition	Sub-traction	sum
All the same	1.1The evaluation is the same	32(32)	17(17)	49
	1.2 Cannot judge either result	0	1(1)	1
Partially similar	2.1 The system provides more information than the teacher.	68(68)	16(16)	84
	2.2 The teacher provides more information than the system.	12(12)	4(4)	16
Totally Different	3.1 The system cannot give any suggestion, but the teacher provides some suggestions.	11(11)	4(4)	15
	3.2 The teacher cannot give any suggestion, but the system provides some suggestions.	20(16)	1(0)	21
	3.3The evaluation of the teacher and the system’s suggestion are totally different.	8(0)	0(0)	8
Illegible sample		4	4	8

We invited two experts who majored in Mathematics to review the results. Table 2 shows the experts’ assessments of the results in Table 1. After the expert’s review, the system accuracy rate increased to 85.57%.

Error analysis

In 2003, we conducted an experiment in which 2,590 students from 10 schools participated in a test. Error analysis was preceded by two phases. In the first phase, the errors were divided into three groups:

1. Careless errors
2. Systematic and predictable errors
3. Random errors

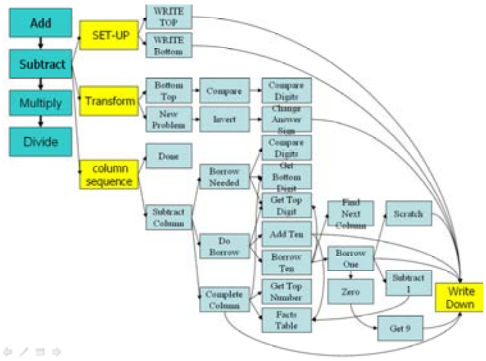


Figure 11. The procedural network of subtraction reported by Brown and Burton

The results can be summarized as follows:

(1) 31 types of addition errors.

(2) 51 subtraction errors, including 11 local errors (i.e., unique to Taiwanese students) not reported by Brown and Burton (Brown and Burton, 1978; VanLehn, 1990). Also, there were 57 subtraction errors reported by Brown and Burton (Brown and Burton, 1978) that do not apply to Taiwanese students.

First, we translated the systematic errors described by Brown and Burton and our findings into a logical format. For example, we translated the error "0－N = 0/after/borrow" into the logical representation "T2=1, T1＜B1, X=T3－B3, Y=0, and Z=T1＋10－B1". Then, we used the student's error data from the previous experiment to test the correctness of the logical representations. After finishing the test, we divided the errors into 40 categories and added some descriptions. We then asked five mathematics teachers to explain why the students made the mistakes.

In the third step, we use semantic information to relate explanations of the different error types and logical representations, to the Curriculum module. The module provides information about the current session, including the main concepts and contextual sessions. The system traces the student's previous record and compares it with the expert knowledge recorded in the system so that the components responsible for the error can be identified.

Finally, the system combines the results of the error types, curriculum, and process maps into an ontological rule description represented by InfoMap. It then provides some suggestions to help the student in areas where his/her knowledge is deficient. The user interface for the final test is shown in Figure 12.

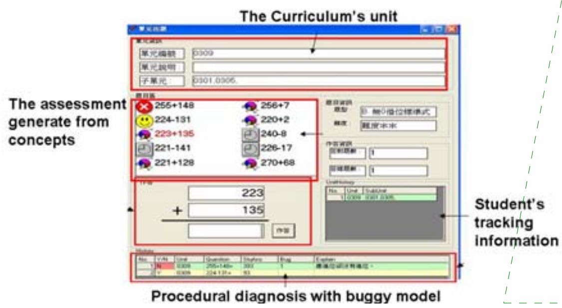


Figure 12 The assessment generated from concepts

註解 **[P4]:** Change to: The assessment generated from concepts.

In 2006, we conducted another experiment in which 56 students participated in a test. Of the 202 error samples collected, there were 36 that the system could not analyze. The errors were divided into five groups:

Table 3. The explanation of final review

格式化: 義大利文 (義大利)

Number	Errors that do not fit Taiwanese students
A3	Ti + Bj & i≠j
A16	Ti + Bi = 9 & [i] =10
A24	Ti<Bi & [i]=Ti-Bi+10 & [i+1]= Ti+1-Bi+1-1 & [i+1]= Ti+1+Bi+1-1
A25	[i]= Ti*Bi
S13	Ti-1<Bi-1 & [i-1] = Ti-1+10-Bi-1 & [i]≠Ti-1-Bi & 2<=i<=5,
S23	Ti-1<Bi-1 & [i-1]=0 & [i]= Ti-1-Bi, 【Zero/after/borrow】
S24	Ti-1<Bi-1 & Ti=0 & [i-1]= Ti-1+10-Bi-1 & [i]=9-Bi & [i+1]= Ti+1-Bi+1 & 2<=i<=5, 【Borrow/from/zero & left/ok】, 【Borrow/from/zero & left/ten/ok】, 【Borrow/from/zero】
S27	Ti-1<Bi-1 & Ti=Bi & [i-1]= Ti-1+10-Bi-1 & [i]=1 & [i+1]= Ti+1-Bi+1 &2<=i<=5,【N—N=1/After/Borrow】
S34	Ti=0 & Bi≠0 & Ti+1>Bi+1& [i+1]= Ti+1-Bi+1 & [i]=Bi& 1<=i<=5,【DIFF/0—N=N】
S35	Ti=0 & Bi≠1 & Ti+1>Bi+1 & [i]=9 & [i+1]= Ti+1-Bi+1 & 1<=i<=5 , 【Treat/top/zero/as/nine】
S38	Ti-1<Bi-1 & [i-1]=0 & [i]= Ti-Bi-(Bi-1- Ti-1) & 2<=i<=5, 【Borrow/unit/diff】
Number	**Insufficient samples**
A9	Ti-1+Bi-1≧10 & [i] = 10
A14	Ti-1+Bi-1≧10 & 1+Ti+Bi≧10 & [i]=Ti+Bi & [i+1]≠Ti+1+Bi+1 & [i+1]= Ti+1+Bi+1 &[i]≠Ti+Bi
A18	Ti-1+Bi-1≧10 & Ti+Bi +1<10 & [i] = Ti+Bi +1 & [i+1] = Ti+1+Bi+1 +1
A23	Ti,Bi≠0 & [i]=Ti | [i] =Bi
S5	Ti-1<Bi-1 & Ti-Bi =0 & Ti+1-Bi+1= Ti+1-Bi+1-2 & 2<=i<=5, 【Borrow/Across/Zero/Over/Zero】
S7	Ti-1=0 & Bi-1>0 & [i]= Ti-Bi & [i-1]=10-Bi & 2<=i<=5, 【Treat/top/zero/as /ten】
S9	Ti-1<Bi-1 & [i]=Ti-Bi & [i-1]= Ti-1+9-Bi-1 & 2<=i<=5 ,【Borrow/once/without/recurse】,【Borrow/from/zero】
S12	Ti<Bi & [i]=Ti+10-Bi & [i+1]≠ Ti+1+10-Bi+1 &1<=i<=5, 【Borrow/skip/equal】
S14	Ti<Bi & Ti+1=0 & [i]=""& [i+1]="" & 1<=i<=5, 【Borrow/won't/recurse】
S17	Ti = Bi=0 & Ti-1<Bi-1 & [i-1]= Ti-1+10-Bi-1 & [i]=0 & 2<=i<=5 , 【Borrow/from/zero/is/ten/carrying/answer/overflow】
S21	Ti-2<Bi-2 & Ti=Ti-1=0 & [i+1]= Ti+1-Bi+1-1 & 3<=i<=5, 【Decrement/multiple/zeros/by/number/to/right】
S26	Ti=1 & Ti<Bi& [i]=1 & [i+1]= Ti+1-Bi+1 &1<=i<=5, 【DIFF/1—N=1】
S40	Ti-1<Bi-1 & Ti<Bi & [i-1]= Ti-1 +10-Bi-1 & [i]=Ti+9-Bi & [i+1]= Ti+1 - Bi+1 & 2<=i<=5, 【Borrow/only/once】,【Borrowed/from/don't/borrow】,【DIFF/0-N=N/When/Borrow/From/Zero】, 【Decrement/one/to/eleven】, 【Borrow/ten/plus/next/digit/into/zero】, 【Can't/subtract】, 【Sub/copy/least/bottom/most/top】,【Decrement/by/two/over/two】,【Once/Borrow/Always/Borrow】

1. Careless & Random errors
2. Unpredictable errors, for example, 2+7=8.
3. The student answered questions, but did not know the answers.
4. Operating error, for example, 840+342=110,082.
5. Mixed error, for example, 2,700-1,024=1,778.
(It would be interesting to see how the above errors are separated/defined, i.e., the criteria used to judge them.)

We also found that there were 8 types of addition errors and 16 types of subtraction errors in Brown and Burton's analysis that are rarely made by Taiwanese students. Based on interviews with experts and a literature review, we divided the 24 types of error into two groups, as shown in Table 3.
1. 11 error types do not fit Taiwanese students because they are not classified in the same way in Taiwan's mathematics curriculum.
2. The remaining 13 error types do not have sufficient samples.

Observations of an Expert

A mathematics education expert made the following observations.
1. The system classification seems too complicated for teachers. The more overloaded it is with error information, the less the teacher will be able to adapt to it. Because teachers must spend a lot of time coaching students, maybe more than two hundred, he/she does not have time to help a student find the possible reason for an error..
2. Unlike a teacher who is familiar with a students' profile, the system cannot distinguish between carelessness and random errors.
As suggested by the experts, we implemented a hierarchical classification scheme similar to that in Brown & Burton's work.

Concluding Remarks

Using an ontology, we have developed an empirical platform to collect information about mathematical errors made by elementary school students in Taiwan. Based on a three-year empirical study, we found twenty-two types of error made by Taiwanese students' that different from those in Brown & Burton's model. We also conducted statistical analysis of the frequency of error types for use in the future. We will continue to collaborate with mathematics education experts to find the most effective ways to understand the cognitive concepts that lead to addition and subtraction errors.
In our future work, we will use this design idea to study errors made in multiplication and division, which are two of the four fundamental operations. Most textbooks provide a standard rounding rule for multiplication and division, allowing students to follow and solve problem. However, this raises the following question: What is the basis of the standard rule for multiplication and division? We will investigate whether we can use logical representation in our ontology to find the errors made by Taiwanese students in these areas.

References

[1] Anderson, J. R., Rules of the Mind. Hillsdale, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ , 1993

[2] Brown, J.S., & Burton, R.R. "Diagnostic models for procedural bugs in basic mathematical skills.", In Cognitive Science, 1978, pp. 155-192.

[3] Burns, H. L. and Capps, C. G., "Foundations of Intelligent Tutoring Systems: An Introduction. Foundations of Intelligent Tutoring Systems." , Lawrence Erlbaum Associates, Hillsdale, NJ., 1988.

[4] Carpenter, T. P., "Learning to add and subtract: An exercise in problem solving. In E. A. Silver (Ed.)", Teaching and learning mathematical problem solving: Multiple research perspectives, Hillsdale, NJ: Erlbaum., 1985, pp. 17-40

[5] C. W. Hue C. H. Kao, M. Lo, L. Y. Tu and W. L. Hsu , "NTUs: An Intelligent Tutorial System Fosters Number Concepts Through Computational Scaffolding," Proceedings of ICCE'02, Auckland, (2002).

[6] H.L. Wang, W.K. Shih, C.N. Hsu, Y.S. Chen, Y.L. Wang, and W.L. Hsu, "Personal Navigating Agent", Proceedings of AGENT'99, 1999.

[7] H.L. Wang, S.H. Wu, I.C. Wang, C.L. Sung, W.L. Hsu and W.K. Shih, "Semantic Search on Internet Tabular Information Extraction for Answering Queries", Proceedings of the Ninth International Conference on Information and Knowledge Management, Washington DC, 2000.

[8] Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E. & Terry, A., " Evaluation of expert systems: Issues and case studies." In F. Hayes-Roth, D.A. Waterman, & D.B. Lenat (Eds.), Building expert systems. Reading, Massachusetts: Addison-Wesley, 1983.

[9] L.Y., Tu, W. L. Hsu and S. H., Wu, "A Cognitive Student Model – An Ontological Approach," ICCE'02, Auckland, 2002.

[10] Martin, D. J., "Concept Mapping as an aid to lesson planning: A longitudinal study.", Journal of Elementary Science Education, 6(2), 1994., pp. 11-30

[11] Mcfarlane A., Sparrowhawk A. and Heald Y., "Report on the educational use of computer games. Teacher Evaluating Educational Multimedia report. Retrieved from http://www.teem.org.uk/publications/teem_gamesined_full.pdf

[12] Roger. T. Hartley., "Representation of Procedural Knowledge.", Department of Computer Science Technical Report, New Mexico State University, February, 1998.

[13] Self, J. The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. International Journal of Artificial Intelligence in Education(10), 1999.

[14] Sowa, J.F., "Knowledge Representation: Logical, Philosophical, and Computational Foundations." , Brooks Cole Publishing, Pacific Grove, CA, 1999

[15] W.L Hsu, "Elementary school Math. tutoring agent", Proceedings of Agent Technology Workshop, 1997.

[16] W.L Hsu, Y.S. Chen and Y.K. Wang, "Natural language agents – An agent society on the Internet" , Proceedings of PRIMA'99, 1999.

[17] W.L. Hsu, Y. S. Chen and S. H. Wu, "Event Identification Based on the Information Map - INFOMAP, " NLPKE'01, 2001, Tucson

Appendix

Dimension	Questions	
the questions design of Tutoring System in exercise	I feel the addition questions are very difficult.	
	I feel the subtraction questions are very difficult.	
	I like the way that the system generates questions.	
	I like using handwriting to answer the questions.	
	I think the system generated questions are more difficult than those of the traditional tests.	
	I like the system generated questions.	
	I like simple inline computing (addition & subtraction).	
	I like transverse computing. (addition & subtraction)	
User Interface Design	I like the story-game design of the system.	
	I like to be tested by the game–based UI.	
	I would like to be instructed by a scenario-based voice.	
	I think the tutoring system's user interface is clear	
	I think the tutoring system's user interface is easy to use.	
	Even without any instruction, I can finish this tutorial by myself.	
	This tutoring system will help me finish the assessment quickly.	
Students' Satisfaction Survey	I am satisfied with this tutoring system	
	Compared with this type of assessment, I would like to take a paper and pencil test.	
	Using the tutoring system for assessment will enhance my learning ability.	
	Using the tutoring system for assessment is better than a paper and pencil test.	
	I feel disappointed with this system	