T. Hirashima et al. (Eds.) (2011). Proceedings of the 19th International Conference on Computers in Education. Chiang Mai, Thailand: Asia-Pacific Society for Computers in Education

Generator of Related Problem Sets for Intelligent Practice Supporting System

Daisuke IIZUKA[a*], Tatsuhiro KONISHI[b] & Yukihiro ITOH[c]
[a]*Graduate school of Shizuoka University , Japan*
[b]*Faculty of Informatics, Shizuoka University, Japan*
[c]*Shizuoka University, Japan*
*gs10004@s.inf.shizuoka.ac.jp

Abstract: We have developed an intelligent educational system for high school chemistry, named Intelligent Practice Supporting System (IPSS). Exercise problems that existing IPSS can handle are designed independently of each other problems. However, teachers often design a set of problems that has a certain educational effect; for example, learners can focus difference of two materials by solving two problems on each of the materials. We call such set of problems "related problem set (RPS)". To design RPS is not so easy task for teachers. In this paper, we propose a method to generate RPSs by transforming representations of problem solving process of an original problem. We add this function to IPSS, and extend IPSS to be able to propose an additional exercise using generated RPS.

Keywords: Problem Generation, Intelligent Educational Systems, Related Problems Sets

Introduction

Nguyen [1] classified problem generation into the following two categories: (1) To derive solution structure/solution sentences from problem sentence/surface structure; (2) To generate problem sentence/surface structure that satisfies given solution structure/domain knowledge. In (1), they wrote it is a challenging issue to model domain knowledge and method for finding the solution from given formulated structure. We proposed a method to formalize solution structure and domain knowledge in high school chemistry, and constructed a problem solver that can derive solution structures from problems [2]. In this paper, we propose problem generation by transforming the solution structure. We focus on educational effects of problem combinations. For example, a set of problems solved by a common knowledge must have an educational effect to make learners' understanding on the common knowledge more stable. A pair of problem each of which can / cannot be solved by a knowledge lets learners focus on the boundary of applying condition of the knowledge. We call such a set of problems that has a certain educational effect "Related Problem Set (RPS)". The educational effect of RPS depends on combination patterns of problems.

In previous researches, there are systems which are making suitable problem for each learner with learners model [3] [4]. These systems don't focus on the combination of problems. Hirashima [5] proposed a method to generate simplified problem for learners who fail to solve a difficult problem. Simplification is a kind of methods to generate RPSs.

In order to design RPS, teachers have to select appropriate problems which have a certain relation to each other problem, carefully. It is not so easy task for teachers.

In order to develop a method to generate RPS, firstly we performed a case study on a problem collection for high school chemistry. We extracted RPS from it and classify the types of RPS by its educational effects. Next, we investigated relations among problems in each RPS. Based on the relations, we develop algorithm how the system should transform problem solving process of an original problem in order to generate its related problems. Then we construct problem generator that can create related problems from an original problem according to educational effect designated by a teacher.

We have developed an intelligent educational system for high school chemistry, named Intelligent Practice Supporting System (IPSS) [6].We add the problem generator to IPSS to be able to propose additional exercise with generated RPS. This result will be much effective for saving teacher's effort in preparing good exercise problems.

1. Our previous researches

1.1 Types of Problems our system handles

Our system can handle the following three types of exercise problems:

(i) Simulate a chemical phenomenon; a part of result of the simulation is the answer.
(ii) Calculate a property value of a material using numerical relation knowledge.
(iii)Problems composed of (i) and (ii).

1.2 Knowledge representation for high school chemistry

Our problem solver uses the following three types of chemical knowledge.

(1)Knowledge of Phenomenon: It represents a relation among "reactants", "changing" and "products" on a chemical reaction. It can be classified into two types: "concrete phenomenon knowledge" which is defined using only concrete materials (ex. If HCl and NaOH react, NaCl and H2O are generated), and "general phenomenon knowledge" which is defined using abstract material (ex. If acid and base react, salt and H2O are generated). There are links between knowledge of general phenomenon and of concrete phenomenon that are instances of the general phenomenon knowledge.

(2)Knowledge of Material Concept: It has property values of a material concept, such as molecular formula. It represents also relations between a material class and its subclass by links (Hierarchy of material classes).

(3)Knowledge of Numerical Relation: It is represented by a formula and conditions under which the formula is true. The typical condition is "a certain chemical reaction happens", "a material belonging to a certain material class exists", etc.

1.3 Representation of the result of problem solving (CWM and PSPM)

As the first step of problem solving, our system reproduces the situation of a given problem from initial conditions, by chemical reaction simulator and Knowledge of Phenomenon. We call representation of the simulated situations "Chemical World Model (CWM)". Then it extracts the answer from CWM. If the problem asks a numerical value, calculation using Knowledge of Numerical Relation may be necessary. The calculation process is represented by Problem Solving Process Model (PSPM). PSPM has tree structure. Each node of PSPM has information about "name of a material", "its property" and "its property value". Root node indicates the goal of the problem. Leaf nodes indicate initial conditions of the problem. Intermediate nodes indicate sub-goals of the problem. The goal and sub-goals are calculated from lower nodes by attached formula (in Fig.1, goal 1 is calculated from term1 and term2 by formula1). Fig.1 shows example of CWM and PSPM.

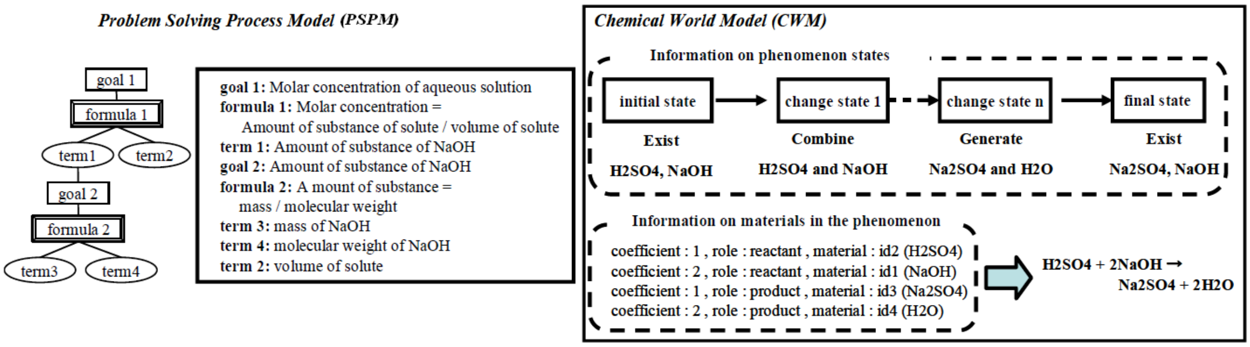

Fig.1 : Chemical World Model (CWM) and Problem Solving Process Model (PSPM)

2. Learning effects of RPS and methods of generating RPS

2.1 Case study on Learning effects of RPS and relations among problems in a RPS

We did case study on problem collections to find examples of RPSs and to extract effective combination patterns of problems. We found 16 examples of RPSs. We also found pairs of educational effects and the methods of transformation for problem generation (Table.1).

Table1 : Expected Educational Effects and Methods of Transforming

	Type of Educational Effect(RPS)	Educational Effects	Type of Transforming
1	Making a knowledge stable (target : general phenomenon knowledge)	Let learners use concrete phenomenon knowledge belonging to a general knowledge repeatedly, to make their understanding on the general one stable.	iii
2	Making a knowledge stable (knowledge of numerical relation)	Let learners use a knowledge of numerical relation repeatedly, to make their understanding stable.	ii-1,v
3	Learning how to use knowledge of numerical relation	Let learners apply a numerical relation knowledge to various situations to learn usage of the knowledge.	ii-1,ii-2,v
4	Making a knowledge stable (knowledge of material concept)	Let learners use a knowledge of material concept repeatedly, to make their understanding stable.	ii-2
5	Learning applying conditions	Let learners aware the boundary of applying condition by situations which can apply knowledge(positive example) and cannot apply it (negative example)	iv, v
6	Learning hierarchy of material classes	Let learners aware material classes by difference among problem solving processes in which a material belonging to the target class appears (positive example) and one not belonging to the class (negative example)	iv, v
7	Supporting problem solving with simplification	Help learners solve a difficult problem by changing the problem easier with simplification.	i（deletion）
8	Bringing up learners' ability with complication	Bringing up learner's ability on advanced problems by making problems more complicated	i（addition）

Patterns of "Type of Transforming" in the right end row of Table1 are as follows.

(1)changing methods of PSPM

(i).Change on number of steps of calculation by changing the goal or initial conditions (addition or deletion of calculation process).

(ii).Change in process of calculation by changing the goal or initial conditions

(ii-1)(calculating the same (sub)goal by other knowledge)

Ex : calculate "A" using "A=B×C" ==> calculate "A" using "A=D÷E"

(ii-2)(calculating other (sub)goal by the same knowledge)

Ex : calculate "A" using "A=B×C" ==> calculate "B" using "B=A÷C"

(iii).Simple change only on numerical value included in initial conditions

(2)changing methods of CWM

(iv).Change materials without changing general phenomenon knowledge which is used for simulating phenomenon on the problem.

(v).Change materials with changing general phenomenon knowledge which is used for simulating phenomenon on the problem.

2.2 Basic Procedure of Related Problem Generation

Our problem generation procedure has three steps. (1): transform of CWM and PSPM for expected educational effect. (2): propagate the modification by (1) from CWM/PSPM to CWM/PSPM. (3): generate natural language expression of the modified problem by using templates. On step (1), our problem generator needs three inputs; "an original problem", designation of "expected educational effect" (seen in Table1) and designation on the knowledge that learners should learn with the generated problem ("Target knowledge"). It transforms PSPM, and CWM considering the expected educational effects (see the row "Type of transforming" in Table1).

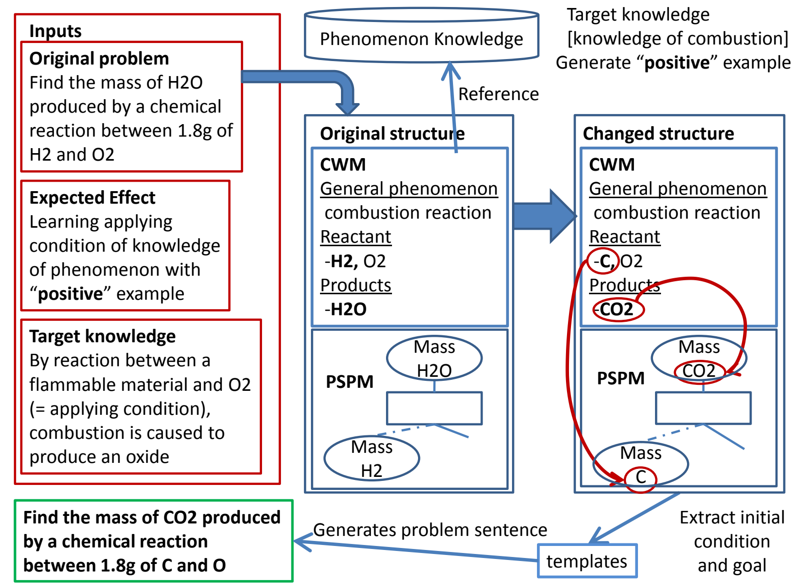

Fig.2 : Image of problem generation flow

We show an example of problem generation. For example, the original problem is "Find the mass of H2O produced by a chemical reaction between 1.8g of H2 and O2", the expected educational effect is "Learning applying condition of knowledge of phenomenon", and the target knowledge is general phenomenon knowledge that "By reaction between a flammable material and O2 (= applying condition), combustion is caused to produce an oxide". The educational effect of this example is the type 5 in Table1. The system applies the procedure (iv) in table1. The applying condition is combination of "a flammable material" and "O2". They correspond to "H2" and "O2" in the original problem. Change them another combination that also satisfies the applying condition. In order to generate a positive example, retrieve a concrete phenomenon knowledge belonging to the target general phenomenon knowledge. Our current system selects a knowledge from retrieved candidates at random. However, considering effective learning on boundary of applying conditions, it may be better to select the candidate including a material that has the maximum distance in the hierarchy of material classes from the material in the original problem. In this example, we assume the system selects chemical reaction between C and O2. It changes CWM using this knowledge of phenomenon. When system changes CWM, it makes correspondence list in which combinations of original material and changed material are described. In this example, "H2" corresponds with "C" as a flammable material, and "H2O" corresponds with CO2 as an oxide. Then the system propagates the modification from CWM to PSPM, to keep consistency. The system replaces "mass of H2O" as the goal in PSPM with "mass of CO2". It checks whether applying conditions of all knowledge of numerical relation are satisfied under the modified CWM. If any applying conditions of some knowledge have become not satisfied, it replaces such knowledge with available one. After the modification, the system extracts initial conditions and the goal from modified CWM and PSPM, to generate a problem "Find the mass of CO2 produced by a chemical reaction between 1.2g of C and O2"(Numerical values in the initial condition are modified randomly in this step).

3. Methods of applying the problem generator to IPSS

We implement a problem generator. Our current problem generator can generate RPS of (1)~(6) in Table1. Now we are designing practice proposing unit. After a learner finishes solving a problem, the unit proposes him/her additional practice with a RPS generated by our problem generator. Expected learning effects used for problem generation are set by teachers beforehand, or set by the system in order to treat learner's weak points. For example, if a learner couldn't use a knowledge at the beginning and then he/she finds how to use the knowledge with advices by IPSS, practice proposal unit proposes additional practice using a RPS for which the knowledge is useful ("Making a knowledge stable" effect on Table1).

4. Conclusion

In this paper, we focused RPS (related problem set) and developed a method to generate RPS from an original problem, designation of expected educational effect, and designation of the target knowledge. Our basic approach is transforming problem solving process of the original problem. We extended our existing practice supporting system to be able to propose additional practice. We have not evaluated our system practically yet, so our future work will be experimental evaluation on effectiveness of the generated problems.

References

[1] Nguyen-Thinh, LE., Kojiri, T. (2010). Question and Problem Generation - State of The Art. *Workshop of the 18th International Conference on Computers in Education*, 47-55.

[2] Ishima, N., Ueda, T., Konishi, T., Itoh , Y. (2006). Developing a Practical Domain Knowledge Base and Problem Solving System for Intelligent Educational System of High School Chemistry. *Proceeding of ICCE2006*, 115-118.

[3] Amuruth, K. (2005). Rule-Based Adaptive Problem Generation in Programming Tutors and its Evaluation. *Proceedings of 12th International Conference on Artificial Intelligence in Education, AI-ED'2005*, 36-44.

[4] Martin, B. and Mitrovic, A. (2002). Automatic Problem Generation in Constraint-Based Tutors. *proceedings of Intelligent Tutoring System 2002, LNCS 2363,* 388-398.

[5] Hirashima, T., Ueno, T., Yamamoto, S. (2009). Problem Generation as Structure Simplification Following Problem-Solving Process. *ICCE 2009 Workshop Proceedings of the Modeling, Management and Generation of Problems/Questions in eLearning*, 10-14.

[6] Konishi, T., Okada, Y., Iizuka, D., Itoh, Y. (2010). Development of an Intelligent Practice Supporting System for High School Chemistry. *Proceeding of ICCE2010*, 66-70.

[7] Kojima, K., Miwa, K. (2005). A System that Generates Word Problems Using Problem Generation Episodes. *Proceedings of ICCE 2005*, 193-200.

[8] Yamamoto, S., Waki, H., Hirashima, T. (2010). An Interactive Environment for Learning by Problem-Changing. *Proceedings of ICCE 2010*, 1-8.