A Metadata Editor of Exercise Problems for Intelligent e-Learning

Tsukasa HIRASHIMA
Hiroshima University, 1-4-1 Kagamiyama
Higashi-Hiroshima, 739-8527 JAPAN
tsukasa@isl.hiroshima-u.ac.jp

Abstract

In this paper, a metadata-authoring method of exercise problems is proposed and a metadata editor with the method is described. In the metadata-authoring method, a new problem is characterized using differences from the basic problem. In the metadata editor, a user can make a new problem by changing the basic problem. The changes represent the differences between the new problem and the basic problem. The new problem is then characterized by the differences from the basic problem. We call the metadata-authoring method "Differential Indexing".

1 Introduction

Our research target of intelligent e-learning is the exercises for letting students master the use of solution methods in mathematics, arithmetic, physics, and so on. Although students are usually taught a solution method using a basic problem, the students who can solve the basic problem cannot always solve exercise problems that can be also solved by the same solution method. The origin of the difficulty of the exercise problems is the differences between the basic problem and the exercise ones. Then the students learn how to deal with the differences through the exercises. Therefore, in such exercises, differences from the basic problem are the most important characteristics of the problems. Based on this consideration, in order to realize adaptive control of the difficulty of problems in e-Learning, we propose a metadata model called “differential indexing” [Hirashima et al, 2002]. Then, we have developed a prototype of a web-based metadata editor with the differential indexing. In the editor, authors make the metadata by re-placing the elements composing the basic problem with the elements composing the exercise problems. Therefore, the authors don’t have to know the format of the meta-data. Moreover, the editor can diagnose the metadata semantically. Furthermore, it can acquire knowledge that is used to the diagnosis in the metadata authoring. In addition, “web-based” means that it is possible for a lot of people to use the editor and to share the problem data. These features of the editor will be useful in order to use it in real world.

2 Differential Indexing

The differential indexing is proposed based on MIPS that is a model of problem process of the problem structure [Hirashima et al, 1992]. When the problem structure drawn from a problem is transformed to the problem structure corresponding to the one of the base problem, the solution method applied to the base problem can be also applied to the problem. In the transformation process, the differences are detected. In other words, the differences express the problem solving process of the exercise problems. Therefore, the metadata made by the differential indexing include enough information to solve the exercise problems. By comparing the two metadata, similarities and differences between the two exercise problems are derived. Then, the similarities and differences represent semantic relation between the problems.

In the differential indexing, the following two differences from a base problem can be categorized: (1) an instance difference, and (2) a structure difference. The structure difference is divided as (2a) a structure difference that can be complemented with fact knowledge, and (2b) a structure difference that can be complemented with operational knowledge. In the following section, these differences are described concretely.

3 Metadata Editor

Figure 1 is the interface of the metadata editor. (Currently, only a Japanese version is implemented. The words in the figures were translated into English). The interface is composed of (a) Problem sentences field, (b) Problem-indexing field, and (c) Problem confirmation field. When a user selects a solution method, the base problem of the solution method is given in Problem-indexing field (in this section, a

user means a problem author). In Problem-indexing field, each line corresponds to a basic relation composed of an object, an attribute, and a numerical value. The left-hand side column is the statement of the way to give the value. The "given value" is the value that is given in the problem directly. "Not given (fact)" is the value that is not given in the problem, but can be complemented by fact knowledge. "Not given (operational)" is the value that is not given in the problem, but can be derived by operational knowledge. These basic relations correspond to the problem structure of the base problem. By changing the concepts or the statements, the problem structure of the new problem is generated. The metadata of the new problem are described as the changes.

In Figure 1, the base problem of the crane-turtle method is given in Problem-indexing field. In Problem sentence field, a user then writes sentences of a new problem that can be solved by the same solution method. The sentences are not interpreted by the system but saved as the body of the problem data. The sentences are used as they are when the problem is given to a learner.

After that, the user replaces concepts in Problem-indexing field with the concepts used in the new problem. Throughout this process, the problem structure that has different instances but has the same structure with the base problem is generated. Before making a problem including structure differences, a user has to make a problem structure that has only the instance differences. Then the statements of the way to give values are changed to make the problem structure that includes structure differences. The editor then requests the user to input the way to complement the value. In the case of "not given (fact)", the editor requests the user to input fact knowledge that has the same form with a basic relation. In the case of "not given (operational)", the editor requests the user to input operational knowledge and additional basic relations. The operational knowledge is the form of the operational relation between the basic relations. The additional basic relations should be included in the problem sentences.

For example, Problem-1 shown in Problem sentence field can be solved by the same solution method with the base problem shown in Problem indexing field. To make the problem structure of the Problem-1, the user has to first make the problem structure that includes only the instance difference. Therefore, a user replaces "the total number of legs of cranes and turtles" with "the total score of pupils" in problem-indexing field. Then, the statement of the way to give the value should be changed. In the problem, there is

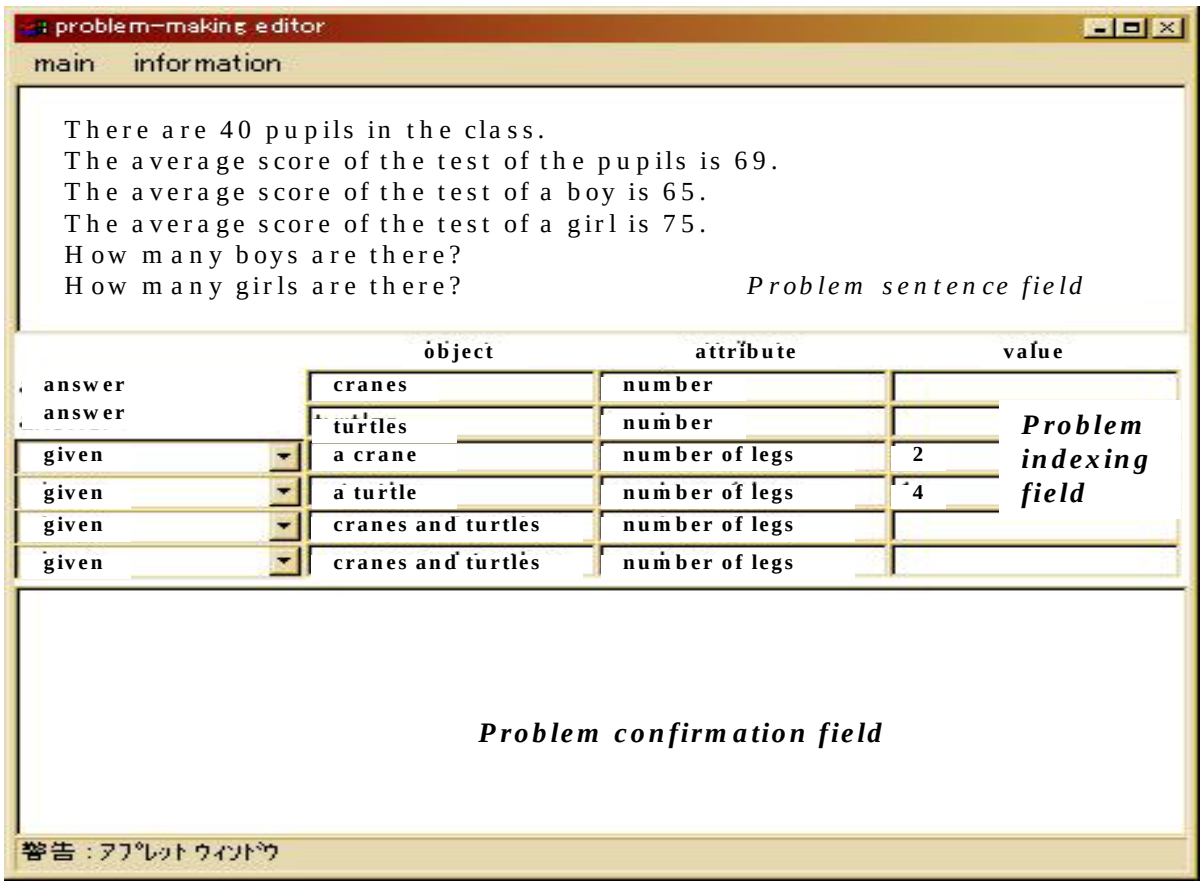

Figure 1. The Interface of Metadata Editor

no total score of a test of pupils, but it can be derived with the total number of pupils and their average score. Therefore, the user should state that the value is “not given (operational)”. Then, the user should input an additional basic relation (Object: pupils, Attribute: average score, Value: 69) and an operational knowledge (“average score of pupils” multiplied by “the number of pupils” is “the total score of pupils”).

After the user has finished changing the problem structure in the problem-indexing field, the editor provides a user with problem sentences that are generated from the problem structure to help them check the problem structure in the problem confirmation field. Because the editor can solve the problem by applying the solution method to the problem structure, it also provides the user with the explanation of the calculation used to derive the answer as the operation among the basic relations.

References

[Hirashima et al, 2002] T. Hirashima and Akira Takeuchi. Differential Indexing: A problem-authoring method for computer-based problem practice. In Proceedings of the international e-Learning Conference, pp.1612-1615.

[Hirashima et al, 1992] T. Hirashima, A. Kashihara and J. Toyoda. Providing Problem Explanation for ITS. In Proceedings of the International Conference on Intelligent Tutoring Systems, pp.76 - 83.