Metadata for Web-Based Mathematical Learning Materials

Yukari SHIROTA †

† Faculty of Economics, Gakushuin University 1-5-1 Mejiro, Toshima-ku, Tokyo, 171-8588 Japan
E-mail: † yukari.shirota@gakushuin.ac.jp

Abstract For learning materials to be reusable, it is necessary that mathematical knowledge about them be semantically represented. Furthermore, such representation is needed to dynamically generate those materials. OMDoc, an extension of OpenMath, is widely used as a standard for representing mathematical knowledge. However, OMDoc is insufficient to describe key concepts in mathematical word problems, as its target knowledge is the description of mathematical formulas. In this paper, we address even higher level metadata for representing mathematical word problems. We also present dynamic courseware generation methods employed in our e-Math system.

Keyword courseware generation, metadata, OMDoc, mathematical word problems, solution plan

1. Introduction

For learning materials to be reusable, it is necessary that mathematical knowledge about them be semantically represented. Furthermore, such representation is needed to dynamically generate those materials. OMDoc, an extension of OpenMath, is widely used as a standard for representing mathematical knowledge. However, as its target knowledge is the description of mathematical formulas, OMDoc is insufficient to describe key concepts in mathematical word problems. Our research goal is to dynamically generate teacher dialogues and explanations that contain mathematical expressions and visual materials, such as graphs. When a teacher explains the content of a given word problem, he/she uses various terms, such as “this main balance equation,” “the right part of the equation,” and “the found solution of the equation,” all of which are key concepts in the given problem. These concepts depend on the content of the problem. The concepts should be explained repeatedly in various ways, such as in words and using graphs, until the student can understand them.

Standard OpenMath offers only descriptions of mathematical formulas, and in OMDoc, only instructional types of resources, such as “theorems,” “examples” and “proofs” are available. In this paper, we propose even higher level metadata for representing mathematical word problems. We also present the dynamic courseware generation methods employed in our e-Math Interactive Agent system.

2. Requirements

In this section, we describe the requirements for our e-Math Web-based learning material generation system. First, we define key concepts that appear in a mathematical word problem using metadata. Each concept is a term and is defined for a specific problem type. If the target is an economical math problem, the concept will have some mathematical or economical semantics. In our system, problem types are defined for sets of similar problems. Problems that belong to the same problem type can share the same solution plan. To define a new problem type, a new set of concepts is defined. For example, the following are the typical problem types:

(a) Optimization of single variable functions,

(b) Optimization of multivariable functions,

(c) Constrained optimization problems with Lagrange multipliers, and

(d) Equilibrium problem types.

In an optimization problem, the following terms and equations are defined as the above-mentioned concepts of the problem:

(1) The unary function to be set up from the given simultaneous equations with the unknown variable as the parameter.
(2) The unknown variable.
(3) The first differentiation of the unary function.
(4) The second differentiation of the unary function.
(5) The stationary points calculated by the first differentiation equation.

Each concept includes some mathematical/economical semantics about which students frequently ask the system. One of our requirements is that the generated courseware have the following help functions, so that the educational

system can respond when a student requests assistance:

(1) Provide the correct variable name of the symbol.
(2) Provide the mathematical equation/term/variable of the given concept defined for the problem type.
(3) Explain the word term in an economical sense.
(4) Explain the mathematical relationship between variables in a set of equations.
(5) Explain the economical relationship between variables in a set of equations.
(6) Provide a visual explanation of the relationships from mathematical and economical viewpoints.

As shown here, our requirements for a mathematical courseware automation system are that it dynamically generate various explanations of the relationships that appear in the problem. To implement these help functions, it is necessary to describe the semantics of equations, terms, and variables by annotating metadata.

3. Related Work

In this section, we describe existing works related to our research.

3.1. Mathematical Documentation

First, we discuss work related to mathematical documentation, in particular, OMDoc and OpenMath, in order to clarify the difference between the research goal of this work and our own.

OMDoc, an extension of an earlier project known as OpenMath, is widely used as a standard for mathematical knowledge representation[1,2,3]. OpenMath is a standard for representing mathematical objects with their semantics, allowing the exchange of these objects between computer programs, storage in databases, or publication on the Web[4,5]. OpenMath has a strong relationship to the MathML recommendation from the Worldwide Web Consortium[6]. MathML deals principally with the presentation of mathematical objects, while OpenMath is solely concerned with their semantic meaning or content. Although MathML facilities for dealing with content are somewhat limited, they allow semantic information encoded in OpenMath to be embedded inside a MathML structure. Thus, the two technologies may be seen as very complementary[4].

OpenMath is highly relevant for handling databases that contain mathematical expressions, such as document databases. Technical publishing requires a standard for representing mathematical objects with their semantics so that software systems and humans can exchange mathematical expressions on the Web without ambiguity. If mathematical semantics are not annotated in the documents, software systems and human users exchanging data cannot understand what the author intended the expression to represent.

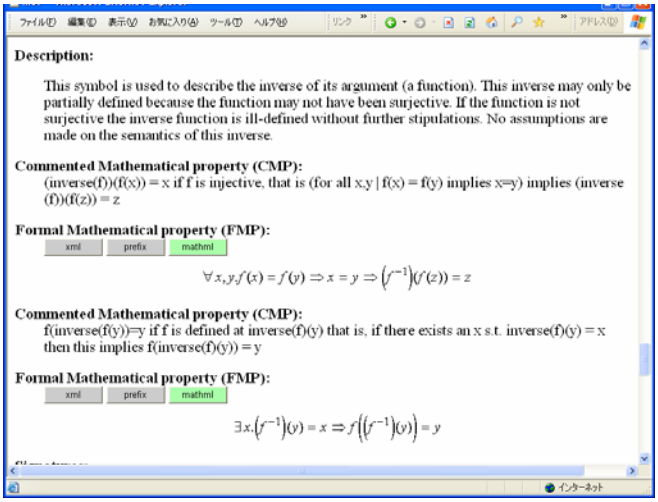

Figure1. A sample of Content Dictionaries in OpenMath.

Content Dictionaries (CDs) in OpenMath are used to assign informal and formal semantics to all symbols used in OpenMath objects. They define the symbols used to represent concepts arising in a particular area of mathematics. As an example, let us explain the definition of the symbol “inverse” that is a sample published on the OpenMath website in Figure 1 (See http://www.openmath.org/cocoon/openmath/cdfiles2/cd/fns1.html). The symbol “inverse” is used to declare inverse functions. Inverse functions are an important concept in mathematics and are defined in many mathematical textbooks[7]. As shown in Figure 1, symbol declarations contain a description of the symbol together with a set of “commented mathematical properties” (CMP) and “formal mathematical properties” (FMP). The FMP part in Figure 1 is displayed as a normal mathematical notation, instead of an OpenMath XML-representation using markup.

Differentiation and integration symbols are contributed as other examples of CD symbols, as follows (See http://www.openmath.org/cocoon/openmath/cdfiles2/cd/calculus1.html#diff):

- nthdiff: the nth-iterated ordinary differentiation of a unary function.
- partialdiff: partial differentiation of a function of more than one variable.
- int: integration of unary functions.
- defint: definite integration of unary functions.

As the above CD examples show, CDs provide definitions of mathematical formulas, not mathematical procedures (algorithms). As Kohlhase described, CDs are largely

informal because the OpenMath framework offers no support for ensuring their consistency, conciseness, or manipulation. In addition, OpenMath has no means of structuring the content of a mathematical document by dividing it into logical units, such as "definition," "theorem," and "proof"[8].

OMDoc is an extension of the OpenMath and MathML standards. It extends these formats using markup for the document and theory levels of mathematical documents so that the document author can specify them and the consumer (an OMDoc reader or a mathematical software system) can take advantage of them[1].

For example, OMDoc offers the following elements:

- Metadata: in Dublin Core and other formats, such as RDF.
- Statements: namely, definitions, theorems, axioms, examples, et cetera.
- Proofs: structured from hypotheses, conclusions, methods et cetera.

In terms of education systems, the interesting OMDoc modules are as follows:

- Presentation: OMDoc allows the user to specify notations for content mathematical objects using XSLT.
- Applet: programs that can be executed in some way in a web browser during manipulation. The applet is called an "omlet" in OMDoc.
- Exercise/Quiz: to make OMDoc a viable format for educational and course materials.

As shown here, one extension of OMDoc is used to transform an OMDoc document into interactive courseware. These extensions have been developed by certain projects. MBASE is an open mathematical knowledge base founded on OMDoc[9,10]. ActiveMath, which is a web-based learning system, also uses OMDoc format and can dynamically generate interactive mathematical courseware[11,12]. However, OMDoc elements, such as "theorem," "example" and "proof" correspond to grammatical objects, not content objects.

Our target is metadata even higher than the targets of OMDoc. The differences between our metadata and OMDoc elements are illustrated in Figure 2. As shown in Figure 2, MathML annotation is lower than that of OMDoc.

In MathML, many tags are required to describe even a small term, such as "$(a+b)^2$," as follows:

```
<mrow>
  <msup>
    <mfenced>
      <mrow>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
      </mrow>
    </mfenced>
    <mn>2</mn>
  </msup>
</mrow>
```

In the example, element "<mrow>" is used to denote a row of horizontally aligned material[13].

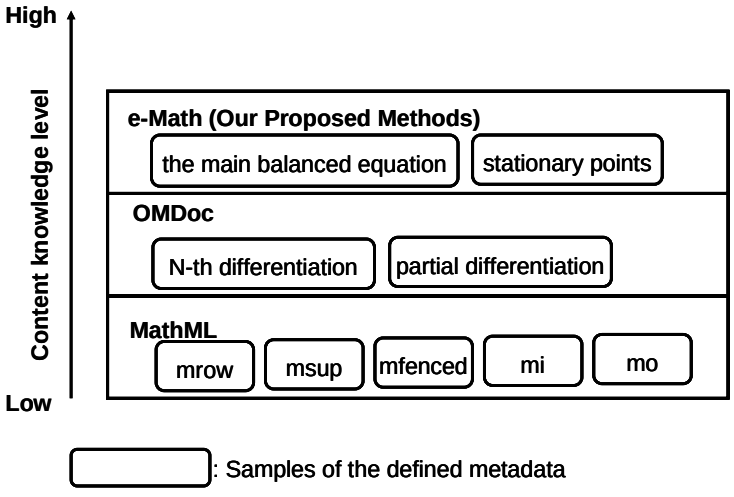

Figure 2. Comparison between our e-Math target level and others.

3.2. e-Learning Standards

Next, we discuss e-Learning standards that are not limited to the mathematical educational field. Many educational standards support content reusability goals. Our research also aims to improve reusability.

SCORM (Sharable Content Object Reference Model) is a standard for curriculum and the reuse of learning materials[14]. However, SCORM content packages do not meet their developers' initial goal regarding reusability, because the cost of defining metadata is too high[15]. In the most recent version, "SCORM 2004" (Version 1.3), it is possible to define the sequencing rules among contents to overcome the problem of the previous version. A SCORM content package can include a description of content structure and sequencing and navigation information elements[16]. However, content sequencing in SCORM is hardwired into the structured content[17]. As there is no

description of how to present materials, in most existing SCORM learning materials, physical layout representation is written in HTML in which the logical structures and physical layouts are not separated [18]. As a result, reusability decreases.

In education, instructional vocabulary is shared between teachers and learners. For example, IEEE Learning Object Metadata (LOM) defines types of learning resources [19] as follows: Exercise, Simulation, Questionnaire, Diagram, Figure, Graph, Index, Slide, Table, Narrative Text, Exam, Experiment, ProblemStatement, and SelfAssesment [20]. As Ullrich noted, the LOM types mix instructional types, such as Exercises and Simulations, and presentation formats, such as Diagrams and Graphs [21].

As shown here, metadata separation between physical representation and logical contents has not been yet defined in either SCORM or LOM.

3.3. Instructional Ontologies

Next, let us examine the research field of semantic Web technologies and ontologies for education. An ontology is a formal, explicit specification of a shared conceptualization. Ontologies are introduced to facilitate knowledge sharing and reuse between various agents. Numerous instructional ontologies exist and many semantic approaches using instructional ontologies for e-Learning systems have been researched and proposed. Mizoguchi has pointed out the advantages of a common vocabulary (ontology) and a theoretical framework for understanding and interpreting how real groups work[22]. Aroyo and Mizoguchi have proposed an ontology of instructional objects that assist authors by describing a conceptual model of the content structure. They also give instructional scenarios and strategies that can be reused[23].

The ontology of instructional items proposed by Ullrich offers a variety of learning resources types, compared with those of IEEE LOM. For example, his proposed instructional objects include Exercise, Exploration, Invitation, and RealWorldProblem under "Interactivity"[21] (See http://www.activemath.org/~cullrich/oio/oio.html). It is an ontology of instructional resource types that includes no physical format information. As Ullrich is a member of the ActiveMath group, the ontology, which is a general purpose one, may be more suitable for defining mathematical courseware. The solution plan defined in our model may correspond to his defined Procedure. Although his proposed ontology defines LawOfNature under Law, if such a subclass is defined, other new classes for non-nature laws should also be added, so that an economical rule such as "*profit = revenue – cost,*" can be defined.

The targets of current ontologies for educational systems are instructional objects and learning resource types, not yet reached to define the problem content.

3.4. Mathematical Word Problem Metadata

To our knowledge, there is only one piece of research that intends to define mathematical problems using metadata. Hirashima has proposed a metadata-authoring method using the word problem structure. In this method, when a new problem is given, the differences are detected from the base structure. Then, the solution method applied to the base problem can also be applied to the new problem to generate the solution method for the new problem[24,25]. In this sense, Hirashima's approach may be considered related to our research approach. Hirashima uses the crane-turtle method as an example of the base problem.

The crane-turtle method is an early Japanese calculus method focusing on the difference between the number of legs possessed by each creature: a crane has two legs and a turtle has four. The key concepts are (a) the number of cranes (or turtles), (b) the number of legs a crane (or turtle) has, (c) the total number of given cranes and turtles, and (d) the total number of legs of the given cranes and turtles. There are two solution methods: (1) the early Japanese method, and (2) the simultaneous equation method. The former method supposes that all animals are turtles when the unknown variable is the number of cranes. It then calculates the difference of the total number of crane legs and the given number of legs. Lastly, it divides the difference by two. It is easier to solve the problem using this early Japanese method than by using simultaneous methods.

Our proposed courseware definition method can also be applied to non-simultaneous equation-based solution processes, such as early Asian calculus methods. Various interesting solution plans other than sets of simultaneous equations can be used to solve a mathematical word problem. We would like to implement such a traditional Japanese solution plan in our e-Math system.

4. e-Math Approach

In this section, we describe our proposed method for generating Web-based mathematical learning materials. Our learning materials generation system, "e-Math," was developed in 2001 and has undergone continuous

improvement[26,27].

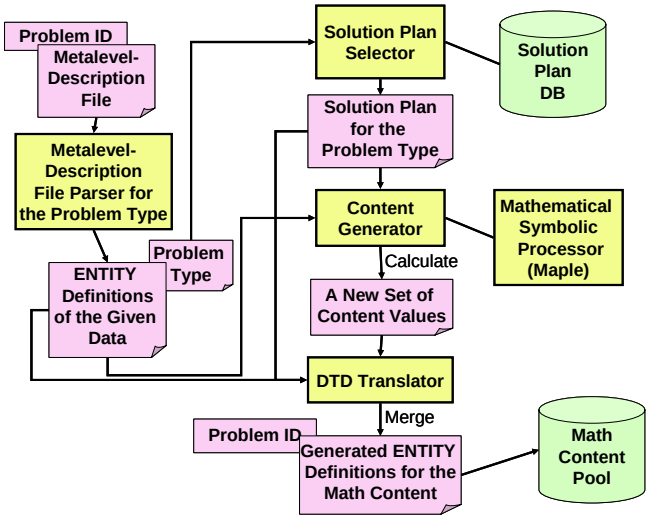

Figure 3. Creation of mathematical contents.

4.1. Generating Mathematical Contents

In this section, we shall explain the creation processes of mathematical contents in our proposed method (See Figure 3). As shown in Figure 3, the input file to our system is called a metalevel-description file. The metalevel-description file is the definition data of the math problem and includes the problem ID information, such as 'problem 1-a' as the attribute 'title.' Suppose that the input data is a metalevel-description file, as shown in Figure 4. The given problem is a word problem and the attribute 'problem-words' indicates the name of a file whose contents are words of the problem.

```
<!-- Metalevel-Description File                    -->
<!-- for   a "Equilibrium" Type Problem            -->
<!--                                               -->
<!-- equilibrium(var, eq)                          -->
<!--   var: equilibrium variable (ex. Y)           -->
<!--   eq:   main balanced equation                -->
<!--         (ex. "Ys(Y)=Yd(Y)")                   -->
title: problem 1-a
problem-words: &problem1-a.txt
data: national income, Y, , Y>0
data: total supply, Ys, , Ys>0
data: total demand, Yd, , Yd>0
data: consumption, C, C=0.8*Y+100
data: investment, I, I=50
relationship: Ys=Y, Yd=C+I
find: equilibrium(Y, Ys=Yd)
```

Figure 4. A sample metalevel-description file which corresponds to the 'equilibrium' problem type.

```
<!-- ENTITY Definitions of the Given Data     -->
<!-- by the Metalevel-Description File       -->
<!ENTITY problem-type     "equilibrium">
<!ENTITY problem-words  &problem1-a.txt>
<!ENTITY given-eqs.eq     "C=0.8*Y+100, I=50">
<!ENTITY relationships.eq          "Ys=Y, Yd=C+I">
<!ENTITY equilibrium.id   "national income">
<!ENTITY equilibrium.var           "Y">
<!ENTITY balance-left.id  "total supply">
<!ENTITY balance-left.var          "Ys">
<!ENTITY balance-right.id          "total demand">
<!ENTITY balance-right.var         "Yd">
<!ENTITY consumption.var           "C">
<!ENTITY investment.var   "I">
    ……
```

Figure 5. Generated ENTITY definitions of the given data.

```
<!-- Solution Plan for "Equilibrium"                    -->
<!-- Type Problems                                      -->
<!--                                                    -->
<!-- slove(eqs, var[, param])                           -->
<!--   eqs:          a set of equations to be solved    -->
<!--   var:          an unknown variable                -->
<!--   param:        a parameter of the unknown         -->
<!ENTITY all-eqs.eq 'relationships.eq, given-eqs.eq'>
<!ENTITY left-sided-eqs.eq
    split-eqs(all-eqs.eq, balance-left.var )>
<!ENTITY right-sided-eqs.eq
    split-eqs(all-eqs.eq, balance-right.var )>
<!ENTITY left-sided-eq.eq
    solve(left-sided-eqs.eq,balance-left.var,equilibrium.var)>
<!ENTITY right-sided-eq.eq
   solve(right-sided-eqs.eq,balance-right.var,equilibrium.var)>
<!ENTITY balance-eq.eq 'balance-left.var=balance-right.var'>
<!ENTITY equilibrium.value
          solve('all-eqs.eq, balance-eq.eq', equilibrium.var)>
```

Figure 6. A solution plan for the 'equilibrium' problem type.

The metalevel-description file parser extracts the problem type from the attribute 'find' and then, based on the 'equilibrium' problem type, parses and constructs a new set of ENTITY definitions, as shown in Figure 6. In the 'equilibrium' problem type, the equilibrium level of

the unknown variable is finally calculated under the condition of the main balanced equation. Thus, the key concepts are: (1) the unknown variable, (2) the left part of the main balanced equation, and (3) the right part of the main balanced equation. These three concepts are represented by the concept names (a) equilibrium, (b) balance-left, and (3) balance-right.

The symbols of these key concepts are defined in the attribute 'find' as "equilibrium(Y, Ys=Yd)" (See the last line of Figure 4). The parser extracts these symbols and the symbols of the other concepts, such as 'given-eqs' and 'consumption.' One concept has the four-attribute (identifier, symbol, equation, value) tuple, whose attributes are respectively represented by "id, var, eq, and value," as shown in Figure 5.

A solution plan is defined and stored in a solution plan DB in advance, as shown in Figure 3. The solution plan is defined using concept names, as shown in Figure 6. In the solution plan, mathematical algorithms are described in the form of a function invocation.

```
<!ENTITY problem-type       "equilibrium">
<!ENTITY problem-words      &problem1-a.txt>
<!ENTITY given-eqs.eq       "C=0.8*Y+100, I=50">
<!ENTITY relationships.eq   "Ys=Y, Yd=C+I">
<!ENTITY equilibrium.id     "national income">
<!ENTITY equilibrium.var    "Y">
<!ENTITY balance-left.id    "total supply">
<!ENTITY balance-left.var   "Ys">
<!ENTITY balance-right.id   "total demand">
<!ENTITY balance-right.var  "Yd">
<!ENTITY consumption.var    "C">
<!ENTITY investment.var     "I">
  ……
<!ENTITY all-eqs.eq
 "Ys=Y, Yd=C+I, C=0.8*Y+100, I=50">
<!ENTITY left-sided-eqs.eq  "Ys=Y">
<!ENTITY right-sided-eqs.eq
 "Yd=C+I, C=0.8*Y+100, I=50">
<!ENTITY left-sided-eq.eq   "Ys=Y">
<!ENTITY right-sided-eq.eq  "Yd=0.8*Y+150">
<!ENTITY balance-eq.eq      "Ys=Yd">
<!ENTITY equilibrium.value  "750">
```

Figure 7. Generated ENTITY definitions for the mathematical content.

In the solution creation process stage, the solution plan selector chooses the solution plan file based on the type information about the problem. Then, the content generator generates a new set of content values by calculating the given functions, such as 'solve' and 'split-eqs,' using the mathematical symbolic processor. We use Maple as the mathematical symbolic processor.

The defined function 'solve' is used to solve the given simultaneous equations in order to find the value of the unknown variable or a parameterized function of the given variable, such as "f(Y)." For example, the invocation of "solve(right-sided-eqs.eq, balance-right.var, equilibrium.var)" is first translated by the ENTITY definition of the given data to the following string:

solve("Yd=C+I,C=0.8*Y+100, I=50", "Yd", "Y").

Next, Maple solves the equations and returns the string "Yd=0.8*Y+150." Another function, 'split-eqs,' extracts the related equations from the given equations with the second parameter as the starting variable, as follows:

split-eqs("Ys=Y, Yd=C+I, C=0.8*Y+100, I=50" , "Yd") which returns the string "Yd=C+I, C=0.8*Y+100, I=50."

Finally, the DTD translator combines the ENTITY definition parts of the given data and the ENTITY definition parts of the solution plan after replacing the function invocation parts with the new set of content values. The generated ENTITY definitions for the mathematical content, which are shown in Figure 7, are then stored in the math content pool for run time use. After the solution process creation stage, the mathematical contents to be explained by a virtual teacher are fully defined.

4.2. Producing XML Documents

In this section, the process of creating learning materials is explained. This process corresponds to the presentation layer of the materials. The generated learning materials are XML documents, graph files, and equation image files, such as 'jpeg' files.

Figure 8 shows the creation processes. During run time, the problem ID and information on the page number of the learning materials are entered by the student. The problem type information can be extracted using the given problem ID. The problem ID is passed to the math content selector, as shown in Figure 8.

Then, the selector chooses the generated ENTITY definitions for the math content. Teachers' guidance plans are defined and stored in the guidance plan DB in advance. Once the problem type and page number information have been input, the guidance plan selector chooses the

guidance plan for the problem type and the page number. A guidance plan is a DTD file. The guidance plan is described in the defined concept names, such as 'balance-eq.' The concept name is translated into the entity name by the guidance generator.

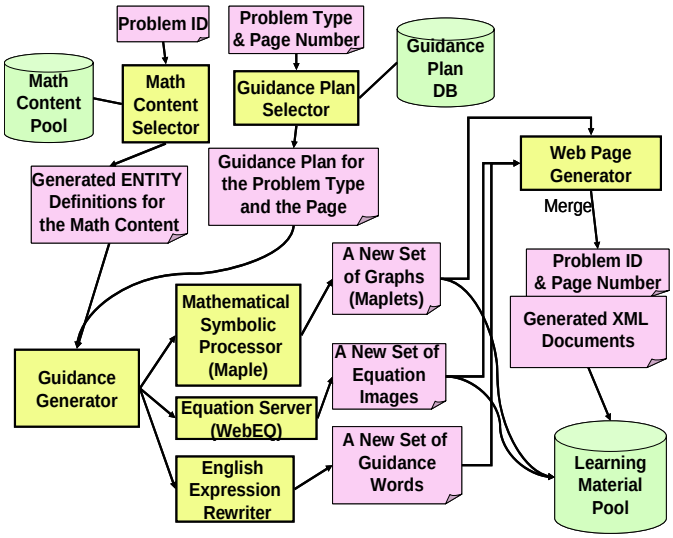

Figure 8. Creation of learning materials for Web presentation.

With the input of this guidance plan and the ENTITY definitions for the math content, an XML document of an appropriate content is generated by the guidance generator. As shown in Figure 9, the guidance generator, invokes (1) Maple to generate a new set of graphs called Maplets, (2) the equation server WebEQ to generate a new set of math equation images, and (3) the English expression rewriter.

In the guidance generation process for the 'equilibrium' problem type, the function "makeCrossGraph" is invocated to generate a Maplet file in which two function curves are intersected. Our system uses 'Maple' mathematical software. This makes it possible to automatically generate learning materials through the dynamic execution of symbolic and concrete value calculations, as well as 2D and 3D visual graphs. The codes generated to display visual graphs are called Maplet programs and are specific to Maple systems. When a student studies a math problem using the generated courseware, Maple processes for interpreting the Maplet programs are dynamically executed on the student's local computer. The generated Maplet file is invoked by Maple when the student clicks on the graph plot button. Using the equation server WebEQ, math expressions calculated in the previous phase are transformed to the image files. As math expression file formats, MathML is also available.

Finally the Web page generator combines the three sets of the materials to generate the final XML document.

5. Conclusions

In this paper, we have described the importance of metadata for key concepts that appear in mathematical word problems. Our defined concepts are in the forms of mathematical equations and terms and have some mathematical and application domain (e.g., economics) semantics. From one concept, our generation system can generate various explanation resources, using words; visual materials, such as graphs; in the form of an economical relationship, such as a mathematical expression; and using mathematical symbolic computations or concrete value computations.

Another new feature of the proposed metadata approach is the separation of a creation process to create mathematical contents and that of presentation materials. This increases the reusability of materials.

There are various solution plans for solving mathematical word problems, including interesting solution plans other than sets of simultaneous equations. We would like to implement such a traditional Japanese solution plan in our e-Math system as part of our future work.

Acknowledgements

This research is supported in part by the Japanese Ministry of Education, Science, Sports, and Culture under Grant-in-Aid for Scientific Research (C) (2)15606014.

References

[1] OMDoc: http://www.mathweb.org/omdoc/.

[2] Michael Kohlhase: "OMDoc: An Open Markup Format for Mathematical Documents (Version1.1), June 5, 2003 at http://www.mathweb.org/omdoc/.

[3] Michael Kohlhase: "OMDoc: an infrastructure for OpenMath content dictionary information," ACM SIGSAM Bulletin, Vol. 34, No. 2, June 2000, pp. 43-48.

[4] The OpenMath Society: The OpenMath website at http://www.openmath.org/cocoon/openmath/overview/index.html.

[5] The OpenMath Society: "The OpenMath Standard," 2004 at http://www.openmath.org/cocoon/openmath//standard/om20/index.html.

[6] Worldwide Web Consortium: "W3C Math Home" at http://www.w3.org/Math/.

[7] M. J. Strauss, G. L. Bradley, and K. J. Smith: Calculus,3rd Edition (Sention 1.4 “Inverse Functions;Trigonometric Functions”), Prentice Hall, 2002.
[8] Erica Melis, Jochen Büdenbender, Georgi Goguadze, Paul Libbrecht, and Carsten Ullrich: “Semantics for Web-Based Mathematical Education Systems,” Proc. of the WWW2002 International Workshop on the Semantic Web, Hawaii, May 7, 2002.
[9] The MBase Mathematical Knowledge Base: http://www.mathweb.org/mbase/.
[10] Michael Kohlhase and Andreas Franke: “MBase: Representing Knowledge and Context for the Integration of Mathematical Software Systems,” Journal of Symbolic Computation Vol.23, No.4, 2001, pp. 365 - 402.
[11] ActiveMath: http://www.activemath.org/.
[12] The ActiveMath group: Erica Melis, Jochen Büdenbender, George Goguadze, Paul Libbrecht and Carsten Ullrich: “Knowledge Representation and Management in ActiveMath,” Annals of Mathematics and Artificial Intelligence , Vol.38, No.1－3, 2003, pp.47-64.
[13] Worldwide Web Consortium: “Mathematical Markup Language (MathML) Version 2.0 (Second Edition),” W3C Recommendation, October 21, 2003 at http://www.w3.org/TR/MathML2/chapter2.html#fund.overview.
[14] ADL (Advanced Distributed Learning): “SCORM ContetnAggregation Model Version 1.3”, 2004.
[15] Tetsuya Nobuhara et al.: “Adaptive e-Learning System Corresponding to Learners’ Performance,” DBSJ Letters, Vol.3, No.2, 2004, pp. 85-88 (in Japanese).
[16] ADL: "SCORM 2004 2nd Edition Overview", July 22, 2004 at http://www.adlnet.org/.
[17] Peter Brusilovsky: “KnowledgeTree: A Distributed Architecture for Adaptive E-Learning,” Proc. of WWW 2004, May 17-22, 2004, New York, New York, USA, pp. 104-113.
[18] Nariomi Shoji et al.: “Design of a Teaching-Material Database for e-Learning,” DBSJ Letters, Vol. 3, No. 2, 2004, pp. 89-92 (in Japanese).
[19] IEEE Learning Technology Standards Committee: http://ltsc.ieee.org/wg12/.
[20] Mikael Nilsson: "IEEE Learning Object Metadata RDF binding," at http://kmr.nada.kth.se/el/ims/md-lomrdf.html.
[21] Carsten Ullrich: “Description of an Instructional Ontology and its Application in Web Services for Education,” Proc. of ISWC’04 workshop SW-EL'04 (Semantic Web for E-Learning), Hiroshima, Japan, November 7-11, 2004, pp. 17-23.
[22] Riichiro Mizoguchi and J. Bourdeau: "Using Ontological Engineering to Overcome Common AI-ED Problems," International Journal of AIED, Vol. 11, No.2, 2000, pp. 107-121.
[23] Lora Aroyo and Riichiro Mizoguchi: “Ontologies for Authoring of Intelligent Educational Systems,” Proc. of ISWC’04 workshop SW-EL'04 (Semantic Web for E-Learning), Hiroshima, Japan, November 7-11, 2004, pp. 89-94.
[24] Tsukasa Hirashima, Akihiro Kashihara, Jun'ichi Toyoda: “Providing Problem Explanation for ITS,” Proc. of the International Conference on Intelligent Tutoring Systems, 1992, pp. 76-83.
[25] Tsukasa Hirashima: “A Metadata Editor of Exercise Problems for Intelligent e-Learning,” Proc. of ISWC’04 workshop SW-EL'04 (Semantic Web for E-Learning), Hiroshima, Japan, November 7-11, 2004, pp. 99-102.
[26] Yukari Shirota: "Knowledge-Based Automation of Web-Based Learning Materials Using Semantic Web Technologies,” Proc. of The Second International Conference on Creating, Connecting and Collaborating through Computing (C5), Kyoto, Japan, January 29-30, 2004, pp.26-33.
[27] Yukari Shirota: " A Semantic Explanation and Symbolic Computation Approach for Designing Mathematical Courseware,” Proc. of The Third International Conference on Creating, Connecting and Collaborating through Computing (C5), Kyoto, Japan, January 28-29, 2005, (to appear).