Automatic Generation of Courseware for Economical Mathematics

Yukari Shirota
Faculty of Economics, Gakushuin University
1-5-1 Mejiro, Toshima-ku, Tokyo 171-8588, Japan
yukari.shirota@gakushuin.ac.jp

Abstract

Today, an increasing number of universities use distance learning systems that leverage the World Wide Web. However, teachers developing the corresponding learning materials face a cost problem in that the work requires much time, practice, and devotion on their part. To solve this issue, we have developed a system -- e-Math Interaction Agent -- that automatically generates learning materials using Semantic Web technologies, such as XML and XSLT. Knowledge databases containing math formulas and basic economic knowledge form the core mechanism of the system. Given the necessary mathematical problem definition data, the system can automate the target courseware by using these knowledge bases.

1 Introduction

Today, an increasing number of universities use distance learning systems that leverage the World Wide Web. However, teachers developing the corresponding learning materials face a cost problem in that the work requires much time, practice, and devotion on their part. To solve this issue, we have developed a system -- e-Math Interaction Agent -- that automatically generates learning materials using Semantic Web technologies, such as XML and XSLT [Shirota, 2004A and B]. Knowledge databases containing mathematical formulas and basic economic knowledge form the core mechanism of the system. Given the necessary mathematical problem definition data of which size is small, the system can automate the target courseware by using these knowledge bases.

The system differs from existing courseware automation systems in that it features

(1) interactive dialogues with a virtual character that are pre-programmed into the XSL stylesheets,
(2) a solution plan and calculations that are automated from a knowledge base of mathematical formulas and economical rules, and
(3) mathematical software that generates the mathematical expressions in MathML format and image files.

My final goal is to formalize a teaching model for a wide range of mathematical problems that includes how to solve the problems and guide students. When teachers use our system, they will be released from tedious XML programming activities and thus able to devote their energies to more creative work.

This paper describes how the e-Math Interaction Agent dynamically automates Web-based materials to be presented interactively on Web browsers. In the next section, we will explain the design principles and a system model of our proposed courseware automation. In Section 3, the developed system architecture will be described. The prototype system that automates learning materials to teach optimization problems in mathematics will be shown. Discussions and conclusions are given in the last section.

2 Automatic Generation Process Model

First, we shall explain our proposed model of automatic generation processes for math problems (See Figure 1). The input data is the definition data of a math problem. Suppose that the mathematical problem is named 'Problem A.' We wish to automatically generate a solution plan specific to Problem A. Namely, the output of the automatic generation process is the learning material specific to Problem A.

The core parts of the process are the "general solution plan model" and "the general model of teacher interaction" that are represented by the two hexagons. The "general solution plan model" describes how to solve a problem of the same or a similar type. The "general model of teacher interaction" defines what a virtual teacher dialogues with a student and how the virtual teacher guides a student. It is also defined for the same or similar types of problems. The followings are typical type names:

(1) Optimization problem of single variable functions.
(2) National income determination modeling problem.
(3) Optimization problem of multivariable functions.
(4) Constrained optimization problem with Lagrange multipliers.

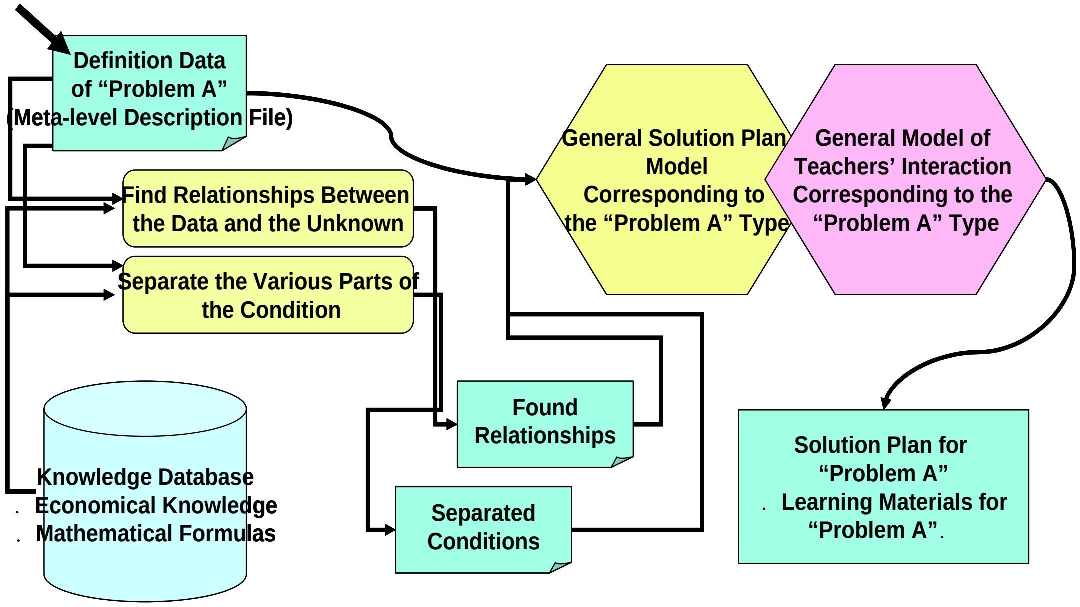

Figure 1. The automatic generation process of courseware using the general solution plan model.

For each problem type, these two models must be defined in advance by a "system supervisor" who is both a computer expert and a math teaching specialist well-versed in solving the problems and teaching them to students.

Based on the proposed model, we have developed our e-Math Interaction Agent system to automate learning objects. The above-mentioned problem types (1) and (2) have been already developed as our prototype systems. In the next section, automation of problem type (1) courseware will be shown as an example.

3 e-Math Interaction Agent System Architecture

In this section, we shall outline our e-Math Interaction agent system. The Interaction Agent is written using Perl script language and the agent invokes four sub-modules: (1) Inference Engine (Prolog interpreter), (2) Mathematical Software for symbolic mathematical processing, (3) Equation Server to generate a MathML file, and (4) Web-Page Generator, which infers the solution plan and generates the corresponding XML files. These, in turn, are displayed by the Interaction Agent using the XLST stylesheet. As the above-mentioned (2) mathematical software, we now use Maple which is widely used for math education.

We apply our proposed automation methods as a prototype system to solve mathematical optimization problems of single variable. Although many kinds of optimization problems exist, they share the same schema which includes the following steps:

(1) Determine the quantity to be maximized or minimized and write the equation for it—in words first, if necessary.
(2) Use the constraints of the problem to write the equation in terms of only one independent variable, and simplify the equation.
(3) Find the first derivative, set it equal to zero, and solve the equation.
(4) Test to determine whether critical points are maxima or minima.
(5) Check for inflection points.
(6) Answer the question posed in the problem.

Therefore the metadata properties of the problems, such as the given and unknown data, can be easily defined. Our defined attributes in a meta-level description file are as follows: (1) data, (2) unknown, (3) given, (4) relationship, and (5) find. The data schemas in a meta-level description file are explained in [Shirota 2004A]. As these data schemas are available to define all mathematical problems except proof problems, our proposed methods of automating mathematical learning materials have high descriptive power. Using the meta-level description file as the input information, the Web-Page Generator can automatically generate the corresponding XML files.

Figure 2, 3, and 4 show sample screens of the generated Web pages. The target problem there is a maximizing profit problem. As shown in Figure 2, the total revenue function (TR) and the total cost function (TC) are given. The variable "Q" represents a quantity. The student has to set up the profit function using an economical relationship "profit = total revenue – total cost". The relationship is stored in a knowledge database as an economical rule and used to set up the equation.

Then, the system finds the first derivative, sets it equal to zero, and solves the equation. The generated learning materials are shown in Figure 3. The candidate critical points are

Q=1 and Q=25. Test to determine which critical point is the maximum, the system takes the second derivative, evaluates the critical points, and checks the signs. As shown in Figure 4, the maximum point is '(25, 6750)'. In addition, the graph can be displayed which is drawn by the mathematical software Maple.

4 Discussion and Conclusions

In this paper, we have described the e-Math Interaction Agent that dynamically automates on-line Web-based materials. For such automation, Semantic Web techniques are effective. In our implementation of the e-Math Interaction Agent, we used XML and XSLT to automate learning objects.

The Interaction Agent is written using Perl script language and the agent invokes four sub-modules: (1) Inference Engine (Prolog interpreter), (2) Mathematical Software for symbolic mathematical processing, (3) Equation Server to generate a MathML file, and (4) Web-Page Generator, which infers the solution plan and generates the corresponding XML files. These, in turn, are displayed by the Interaction Agent using the XLST stylesheet.

We apply our proposed automation methods as a prototype

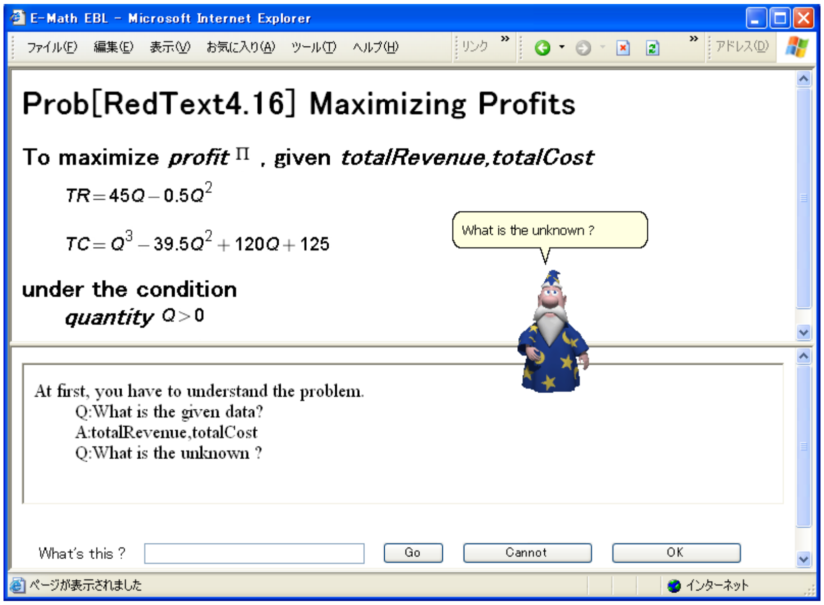

Figure 2. The generated learning materials that explain the problem of maximizing profits.

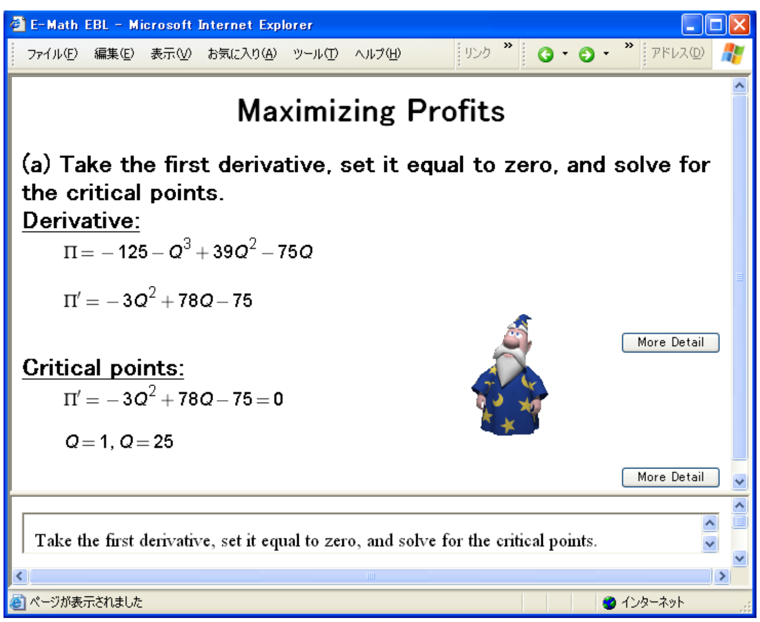

Figure 3. The generated learning materials in which the first derivative is calculated, set equal to zero, and solved.

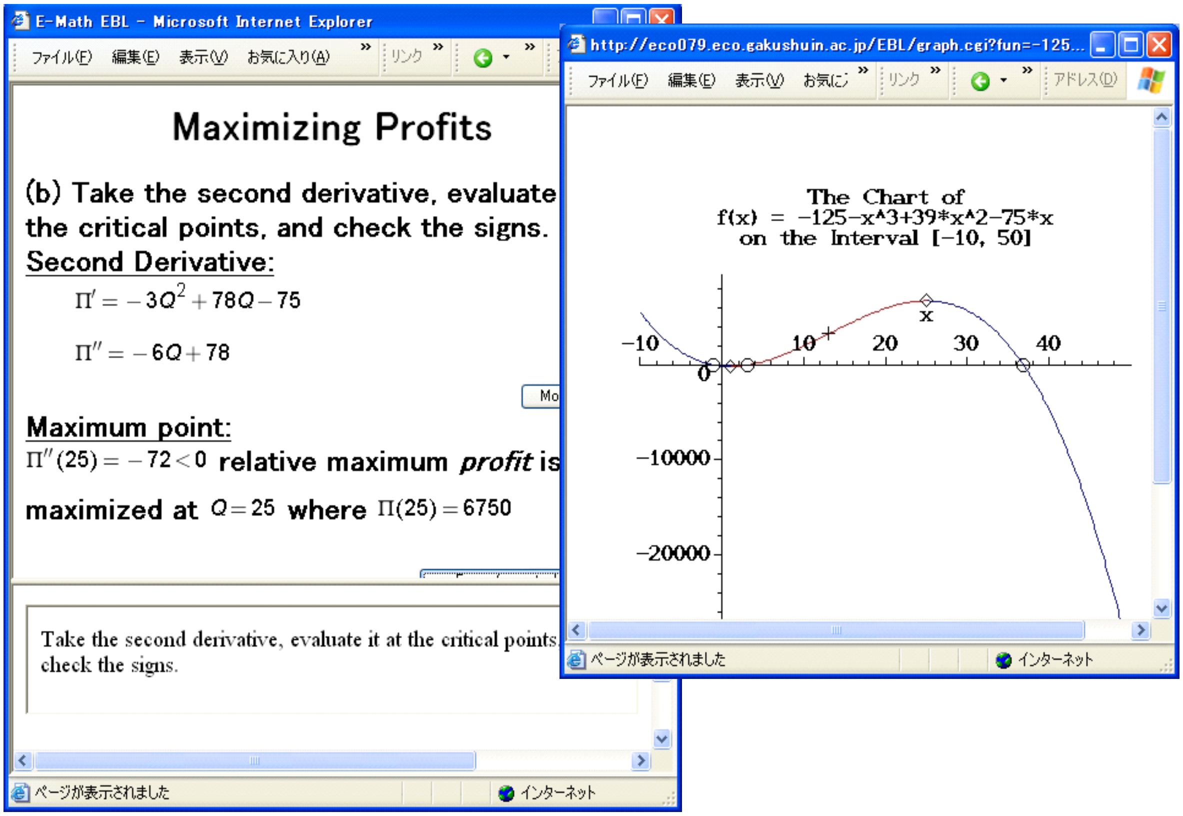

Figure 4. The generated learning materials in which the answer is shown only after the student tests to determine whether the critical points are maxima or minima.

system to solve mathematical optimization problems. Although many kinds of optimization problems exist, they share the same schema. Therefore the metadata properties of the problems, such as the given and unknown data, can be easily defined. Using the meta-level description file as the input information, the Web Page Generator can automatically generate the corresponding XML files. Our defined attributes in a meta-level description file are as follows: (1) data, (2) unknown, (3) given, (4) relationship, and (5) find. As these data schemas are available to define all mathematical problems except proof problems, our proposed methods of automating mathematical learning materials have high descriptive power.

In general, the human resource cost for Web-based learning material development is quite high. It takes teachers much time, practice, and devotion to design and develop learning materials from scratch. The cost is particularly high when teachers try to create learning materials to help students interactively and naturally. However, our proposed automation methods enable any teacher to generate his/her own Web-based sophisticated learning materials. I believe that we can leverage knowledge bases and Semantic Web technologies for most of this work, and thereby largely relinquish it to computers. To justify this belief, we have developed a prototype system that can automate Web-based learning materials. Our proposed automation methods can release mathematics teachers from tedious XML programming, so that they may devote their energies to more creative work.

Acknowledgments

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

[Shirota, 2004A] Yukari Shirota. Knowledge-Based Automation of Web-Based Learning Materials Using Semantic Web Technologies. In *Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing (C5)*, pages 26—33, Kyoto, Japan, January 2004. IEEE Computer Society.

[Shirota, 2004B] Yukari Shirota. A Metadata Framework for Generating Web-Based Learning Materials. In *Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT 2004) Workshops*, pages 249--254, Tokyo, Japan, January 2004. IEEE Computer Society.