Semantic Web Applications based on Physical State Model of Products

Noriyuki Matsuda, Michiyasu Hiramatsu, Hirokazu Taki,
Wakayama University,
Sakaedani 930, Japan
{ matsuda, s041051, taki }@sys.wakayama-u.ac.jp

Satoshi Hori,
Institute of Technologists, Japan

and Norihiro Abe
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology, Japan

Abstract

We developed two semantic web applications based on metadata that is expressed as a physical state. We assume that a physical state model for a product is defined as independent of the product. Operations which cause the physical state transition depend on the products. Two applications are a search system for an instruction for a cellular phone and automated services for two coffeemakers.

1 Introduction

We developed two semantic web applications that use metadata of a product instructional manual. We define a product as an object that causes transformation of a physical state or situation around the product. We regard an instruction book as a reference between the operation of a product and the transformation of a physical situation caused by the operation. We assume that an operation depends on the product but the transformation of a physical situation is independent of the product, and so the transformation of a physical situation can be described in general for a product genre. A description of the physical situation around a product, called a Product State Model (PSM), is used as metadata for the instruction document of a product.

We choose a cellular phone and a coffeemaker as a product for the application. The operation of adjusting the sound volume for a phone call depends on each phone. A physical state around the cellular phone, such as the "state of a phone call" and the "state of the sound volume adjustment" can be described for a general genre of cellular phones. One application provides searching function of a cellular phone's instructional manual with metadata. Another application provides automatic product execution from an instructional page with metadata. For this execution, first the application maps the elements of the PSM to the materials of the real product, and then maps the transformation of the PSM to the operation of the real product.

2 Product State Model

2.1 Model Representation

PSM M is defined below by element p, which represents a physical state around a product, and value v of this element.

M = {(p,v)}

An example of a PSM for a cellular phone is below.

M = {(p="receiving", v="calling"), (p="sound", v="on")}

We assumed that the product expert define PSM for a domain. To develop the applications, we defined cellular phone PSM for search application and coffee recipe PSM for coffeemaker application.

2.2 Model to Product

PSM represents general situations of a product. Operations are depends on a product. To find operations of a product from PSM, mapping from PSM to

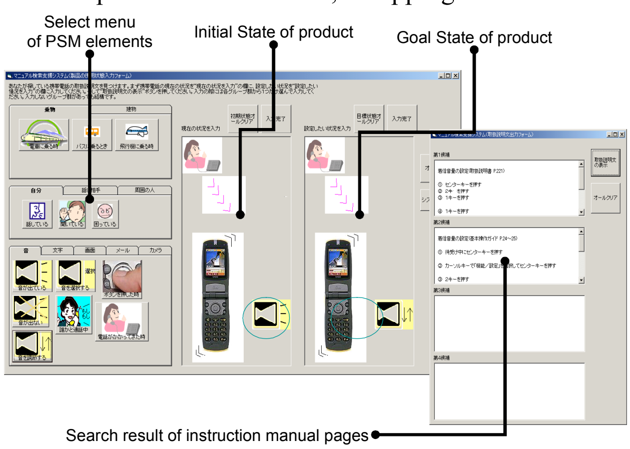

Figure 1. Application for manual search

operations is needed.
Our mapping procedure is the following: (1) attempt to map all elements in PSM to the corresponding materials, (2) check the existence of the operations to transform between the elements, (3) when there are no operations, the applications redo the mapping, (4) when operations satisfying all the transformations is accepted as a link between the PSM and the product.

3 Semantic Web Application

3.1 Manual Search

For example, we considered the scenario in which a novice user of a cellular phone, who knows only the basic operations of the phone and not most functions, searches the required page from an instruction document to use these other functions.
Because the product has more and more functions, the instruction document is thicker. Therefore, finding the required page is harder for the user. When a user has trouble using a product, the trouble is expressed as words called indices. The user can reach an adequate page described as a solution through the contents or index pages of the instruction book. And also, the user searches useful pages from document file in a computer. However, the user does not always reach the required page due to the excess of pages and difficulty for the user to remind words as same as indices in the document.
We assumed that the novice user could describe the physical state for the PSM, because the elements of the PSM are independent of the real product and the user is familiar with the situation for using the product. Figure 1 shows a display image of the application. A user can describe the initial and goal states of the phone by selecting from the elements menu. The initial state expresses the physical situation around the phone. The goal state is the situation desired by the user, but the user cannot reach this state by operating the phone. The user selects the PSM elements for describing the initial state and goal state. Then, the user clicks the search button, and the page is listed as a search result. We implemented only instruction pages that have relevance to operations about sound and vibration adjustment of the phone.
We held an experiment to compare searching a page both by the application and by a PDF file of the instruction book. The subjects are 20 novice users who usually use cellular phones but have no experience of the phone used in the experiment. We asked the users to find a page which explains a specific operation. These results indicate the effectiveness of the pro-posed method. Figure 2 shows search time for each user in the experiment. The y-axis is time (seconds) and the x-axis represents the 20 users. The left bar shows the search time for the PDF file and the right bar shows the search time for the proposed application

3.2 Auto Operation

This section describes an application example of automatic execution with two different actuators from a document. In this example, the document is a coffee

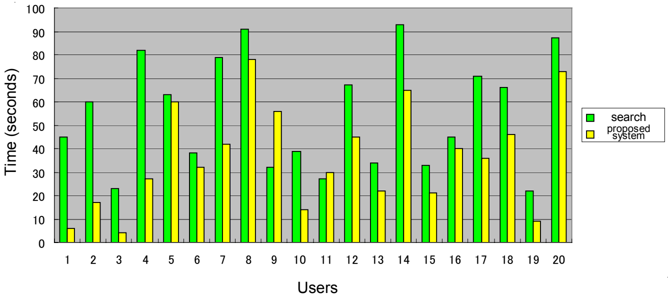

Figure 2. Search time of each subject in the experiment.

recipe and the actuators are a coffee vending machine and a robot hand (RV-E series, Mitsubishi Ltd. which has 6 axes.).
The application generates command sequences for the actuators. The coffee recipe is structured by metadata. To generate the execution sequence, we created an algorithm for map-ping the PSM to the actual product with forward planning [Weld, 1994].
We held an experiment of automatic execution using the coffee recipe. In the experiment, one coffeemakers simulates a coffee vending machine with three cylinders which contain coffee, hot water and milk, and with a bottom which opens or closes (see Figure 3). The other actuator is the robot hand which includes instant coffee, water, milk, mixing stick and a heater.
Giving an initial and goal state in PSM, then the algorithm finds mapping between the PSM and the product's operations based on a generate-and-test method. The algorithm checks all possible combinations for execution.
As a result, we confirmed that the both applications act adequately to make coffee from the recipe.

4 Conclusion

This paper describes two examples of semantic web applications describing the physical state transformation around a product. The Product State Model is used as metadata for the product instruction book. These examples show that the proposed application can assist a user in searching a page from the book. And, the application automatically executes two products from a document with the PSM metadata.
The development of a technique to automatically create the PSM for a document is our future work.

References

[Weld, 1994] Weld, D.: An Introduction to Least Commitment Planning, *AI Magazine*, 15(4), pages 27-61, 1994.

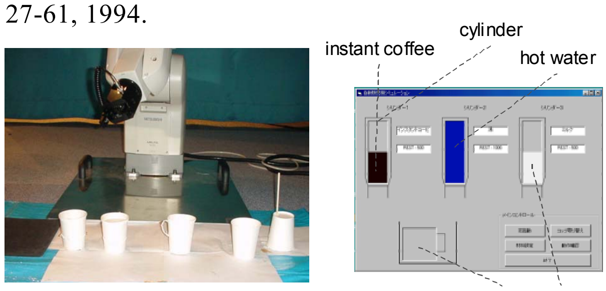

Figure 3. Application for auto operation.