Journal of Web Engineering, Vol. 7, No.2 (2008) 133-157
© Rinton Press

OBSTACLES REVEAL THE NEEDS OF MOBILE INTERNET SERVICES -OOPS: ONTOLOGY-BASED OBSTACLE, PREVENTION AND SOLUTION MODELING FRAMEWORK-

MUNEHIKO SASAJIMA[1], YOSHINOBU KITANURA[1], TAKEFUMI NAGANUMA[2],
KUNIHIRO FUJII[2], SHOJI KURAKAKE[2] and RIICHIRO MIZOGUCHI[1]
[1]I.S.I.R, Osaka University, 8-1 Mihogaoka, Ibaraki-shi, Osaka, 567-0047, Japan
{msasa,kita,miz}@ei.sanken.osaka-u.ac.jp
[2]Research and Development Division, NTT DoCoMo, Inc., 3-5 Hikari-no-oka, Yokosuka-shi,
Kanagawa, 239-8536, Japan
{naganuma,fujiiku,kurakake}@nttdocomo.co.jp

Received May 1, 2007
Revised March 18, 2008

Growth in the mobile internet services industry has seen a marked increase in the number of mobile internet services provided, and this has made proper structuring and organization of the services difficult. Present methods of service provision have proven insufficient to guide users efficiently to the services they need. To solve this problem, a task-oriented menu, which enables users to search for services by what they want to do and not by category, has been proposed. Construction of the task-oriented menu is based on a task ontology modeling method which supports descriptions of user activities, such as task execution and defeating obstacles encountered during the task. This paper discusses a task ontology-based modeling method which supports descriptions of users' activities and related knowledge, such as how to solve problems that the users encounter and how to prevent or solve them on the spot. Models described by our method contribute to designing, testing, and improving mobile internet services.

Key words: Task ontology, mobile internet services, modeling services, semantic web
Communicated by: D. Lowe & N. Moira

1 Introduction

Today, mobile internet services are seeing remarkable growth worldwide. NTT DoCoMo, Japan's leading mobile communications company which manages mobile internet services and occupies about 58% of the market, provide a mobile internet service called i-mode in 19 countries/regions [1]. Boreum argues in [2] that, “Korea, Japan and Finland are considered mature or advanced markets for mobile internet services because of their early adoption of these services and the rapid increase in the number of mobile internet service users”. In fact, we have more than 96,000 service sites today in Japan. Services are varied, providing everything from entertainment (games, ring-tones, etc.) and information (train schedules, weather reports, etc.), to transactional services (online banking, reservations, etc.).

Rapid growth of the Japanese mobile internet service has led to various problems in the management and organization of services, and this has made it difficult for customers to search for and access services they need. Current menu systems for the services are organized from the viewpoint of domain, which we call domain-oriented menus. In such a menu system, each menu category is named for a domain, such as “Traffic info”, “Local info”, and “Hobbies”, but this does not adequately meet users’ needs. When a user faces a “problem”, such as “I have to get on a bus bound for the airport”, she/he first has to consider which category contains the proper internet service to solve the problem. In this case, she/he has to find that information about buses is served in the “Traffic” domain, and then she/he has to drill down through the menu hierarchy, like “Traffic > Public transportation > Buses > XYZ Bus Line”. The user is thus forced to consider two steps before she/he reaches the desired service. This example may seem like an easy one, but as we mentioned above, today there are too many services on numerous domains in Japan. Each layer of the menu hierarchy has several candidates to be selected. At the top level, for example, “Traffic”, “Local information”, and “Latest information” are contained, and all of them may have bus information. Because of such difficulty in finding the proper services, according to NTT DoCoMo’s annual report [1], services like “games” and “ring-tones” that have a clear correspondence between the name of the domain and the content of the service are most frequently accessed by mobile internet users.

To solve the problem, the authors proposed another type of menu system for finding services on mobile handsets, call a “task-oriented menu” [3]. The word task here means the user’s problem solving activity in the real world. In the task-oriented menu, the user searches for a mobile internet service by the name of a directory representing a task they are involved in, rather than the name of a category which might be unfamiliar to them. The user selects a menu that most resembles what they want to do, for example, “Go to station X”, “withdraw cash to buy a ticket”, or “get on the next bus”. The value of information depends on how well the information fits the needs of the user. The necessity of information lies in a task, not in a domain. You search for information when you face a problem that is difficult to overcome with knowledge at hand, on your way to achieving a task. Such a situation is the context and origin of the need for the information. Furthermore, task-oriented menus have the potential to provide useful information for mobile internet service users more quickly than today’s domain-oriented menus. Naganuma [3] evaluated a prototype task-oriented menu and showed that even novice users could reach their desired mobile internet services more rapidly than with the current domain-oriented menu systems or keyword searches.

The task-oriented menu evaluated by Naganuma [3], however, is an ad hoc one because they designed it based on an ad hoc model of users. Since there are neither effective resources nor guidelines for modeling mobile users, we have to design them from scratch. First, it is necessary to list as many user task models as possible. Referring to the models, providers of mobile internet services look for chances to support users by designing useful mobile internet services. To improve the design process, a vocabulary for the model should be carefully defined and systematized. An ontology for the users’ activities, designed from the list of task models, should satisfy certain requirements. Thus, we need to generalize and organize the task model to develop the ontology. Furthermore, since mobile internet services are often used when users face trouble, the ontology should contain such situations. At the same time, an effective modeling method to support descriptions of the situations is needed. According to the experts of the mobile services, currently there are no good tools for modeling mobile users' activity today.

To realize coherent mobile internet services from the top level (e.g., the front page menu of the mobile internet service) down to the concrete service level (e.g., a mobile banking service at "bank X"), we have to modify all of them. Since we have more than 96,000 service sites in Japan to be modified, effective support for technology transfer is critical. In other words, we have to design a new task modeling method and ontology to support the new structure.

In light of the background discussed above, in this article we propose a task-oriented mobile internet service navigation system with a task-ontology-based modeling framework. The research project described here was conducted by experts in ontology design, in cooperation with experts in mobile internet services. Although it is difficult to separate the various aspects of this research project with clear boundaries, this article mainly focuses on how to analyze users' activities, how to build models of them, and how to build necessary ontologies. For building and providing ontologies, we used "Hozo" [4, 5, 6], an ontology editor with a graphical user interface (GUI), which can export developed ontologies in various formats, such as OWL, RDF, XML, and so on. On the basis of task ontology theory [7] and experiences in the engineering domain [8, 9, 10, 11], we propose a methodology for modeling user activities and necessary ontologies with their design guidelines. The approach based on task ontology enables us to describe task models in terms of generic task vocabularies which are detached from the domain model. A model of the task "move", for example, can be applied to model movement in several domains like traveling, commuting, and so on. Based on the task ontology, our modeling method contributes to the design and description of users' activity models that are referred to by service providers when designing mobile internet services. Furthermore, specification of the modeling process based on categorization of users' activities provides the knowledge authors with guidelines.

We also conducted several experiments to evaluate the proposed framework. This paper reports preliminary experimental results in terms of (1) support for generation of ideas about mobile internet services, and (2) the effectiveness of our guidelines for modeling users.

This paper is an extended version of our former publication [12] and is structured as follows: Section 2 describes practical uses of the developed knowledge and its application framework for mobile internet services. Section 3 describes several issues in the modeling framework. In Section 4, we propose the Ontology-based Obstacle, Prevention and Solution (OOPS) modeling framework. Section 5 explains some ontologies we developed, and Section 6 shows experimental results. Section 7 discusses issues in related work, and Section 8 concludes with some description of future research topics.

2 Task-Oriented Menu for Mobile Internet Service Navigation

Fig. 1 shows the process of service selection using a task-oriented menu on the prototype system [3]. First, the most abstract task candidates are shown on the mobile phone (Fig. 1 left). A user selects one of them (e.g. "Go to a department store") to solve his/her current problem (e.g., "need to buy clothes"). Then, tasks and/or subtasks associated with each task are unfolded and displayed under the task nodes (Fig. 1 center). Finally, services associated with the task selected by the user are shown, and each of them leads to access to the actual service (Fig. 1 right).

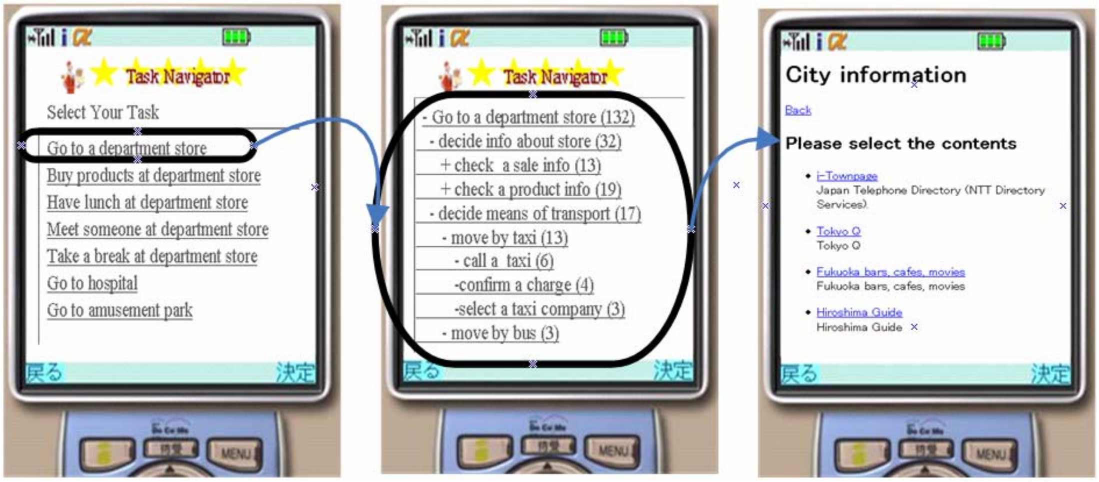
As shown in this example, the task-oriented menu is easy to use for novice users of mobile internet services. By just selecting what he/she wants to do in the real world from the menu, he/she will be led to a service for solving the current problem. Knowledge about the hierarchy of the domain-oriented menu is not necessary.

Although such a generic task hierarchy looks like the hierarchical structure of the category-based menus of today, there are fundamental differences. In certain cases, it is possible to label a concept with a noun instead of a verb or action. It is acceptable to label a mobile internet service that sells tickets as “Ticketing” or “Buy a ticket”, for example. In the same manner, abstract tasks can be labeled with nouns. Although it seems that any concept can be labeled by both verbs and nouns, it is a hasty generalization. Such a generalization may lead to the misunderstanding that we just followed the process used by the designer of the category-based menu in classifying the mobile internet services, thus introducing an abstract hierarchy of the tasks.

An important point is that the difference between “Ticketing” and “Buy a ticket” is just the expression of the label. The concepts are the same task. We focus on the concept and essential characteristics of the mobile user’s task. Comparing them at the conceptual level, a category-based classification of objects is totally different from a task-based one in terms of its structure.

First of all, the labels of the concepts are not always compatible each other. We cannot label a desk with a verb, for example. Next, the reason why a general category-based menu is used for mobile internet services today is that there are so many services that they had to be classified into smaller groups. On the other hand, we did not adopt a generic task hierarchy just for the same purpose. The category-based classification and the task-based classification are qualitatively different; the latter is more suitable for the classification of mobile internet services and for designing a menu to access them.

In the case of category-based classifications, generally speaking, the boundary or definition of


Task Navigator
Select Your Task
Go to a department store
Buy products at department store
Have lunch at department store
Meet someone at department store
Take a break at department store
Go to hospital
Go to amusement park
戻る
決定
Task Navigator
- Go to a department store (132)
- decide info about store (32)
+ check a sale info (13)
+ check a product info (19)
- decide means of transport (17)
- move by taxi (13)
- call a taxi (6)
-confirm a charge (4)
-select a taxi company (3)
- move by bus (3)
戻る
決定
City information
Back
Please select the contents
i-Townpage
Japan Telephone Directory (NTT Directory Services)
Tokyo Q
Tokyo Q
Fukuoka bars, cafes, movies
Fukuoka bars, cafes, movies
Hiroshima Guide
Hiroshima Guide
戻る
決定
MENU

(Left)Select from task lists (Center) Ways to achieve selected tasks
(Right) Services associated with the tasks
Fig. 1 Task-oriented menu system (prototype).

each category becomes vague or implicit. Classification of an object or a concept heavily depends on the intention of the designer who developed the menu. The categories “Hobbies” and “Shopping” are both located at the top level of the Japanese i-mode menu, for example. A mobile internet service that sells cars is classified in the former if the designer considers driving a car as a hobby. On the other hand, the service is classified in the shopping category if the designer focuses on the commercial aspect of the service rather than its object.

On the other hand, in classification of actions from the viewpoint of tasks, the boundary or definition of each category becomes more explicit. Since the basis for classification, such as pre-conditions, processes, and effects of the action, appear in both the label and the classified concepts, it is easy to find the location of a concept in a hierarchy classified based on task. For the same reason, it is easy to add a new concept to the task-based classification. A service that sells cars is classified in a sub-category of the task “Buy”, whether driving a car is a hobby or not.

For the reasons described so far, task-based categories are more suitable for the classification of mobile services. On this point, Naganuma [3] conducted a user test involving nine adult subjects to confirm the effectiveness of a task-oriented menu system and evaluate the process used to find services for problem-solving purposes in terms of process functionality. Subjects were divided into three groups according to their experience of mobile internet services: 1) subjects using mobile internet services every day, 2) subjects using mobile internet services a few days a week, and 3) subjects with no experience in using mobile internet services.

Subjects were asked to retrieve appropriate services to given problems by using the task-oriented menu system, a keyword-type full-text search system newly developed for the experiment, and a major commercial directory-type menu system. The test was designed based on ISO/IEC 9126 Part 4: Quality In Use Metrics. The items of evaluation were: 1) effectiveness, the ratio of users who could reach a service appropriate to the given problem, and 2) productivity, the time taken to reach the service appropriate to the given problem. The target page to be found was the same for each experiment and the total number of available pages was about 15,000.

Analyzing the results by user type, only the task-oriented menu system allowed non-expert users to find the appropriate services with the same success rate as experienced users. The results show that the task-oriented menu system is effective for mobile internet service navigation.

“Multiple classification” which folds possible services under several possible menu paths might occur in both task-oriented menu and category-type menu. The same service might be folded under the several different task menu items redundantly, which might cause increase of the services under one category and lowers usability. Since this shortcoming is common, we don’t have to compare them from the view point of multiple classification.

Nevertheless, task-oriented menu is better than category-based one. As mentioned above, task-based classification makes definition of each category explicit. A mobile user searches for a service that is executable and will solve a problem that he/she faces. Since such problem-solving contexts of the users appear on its label, the task-based menu is more suitable for mobile users than the category-based one. Thus task-based menu helps especially novice users, who cannot interpret their troublesome situation quickly to the name of the category, finding necessary services on the fly as experimental results by Naganuma[3] indicate.

Furthermore, task-oriented classification contributes to listing obstacles for the users' activities. Since mobile users search for services when they face a trouble, we have to model such troubles, that is, unsatisfied (pre-) conditions of the task. The design process of the task-oriented menu itself supports such listing of the obstacles, thus contributes to an improvement of the quality of mobile internet services.

Although we use the term "task-oriented", we do not intend to replace every menu item with task-oriented ones. The objects of a task "buy", for example, would be infinite, and menu items like "buy an item X", "buy an item Y", … would be unrealistic. We plan to complement our task-oriented menu system with conventional domain-oriented classifications and/or search engines if necessary.

3 Issues in Designing a Task-Oriented Menu

Since the experiment described above showed that a task-oriented menu system is a promising framework, the next step is to scale-up the prototype system to a real one. To this end, we have to support designers rather than end-users. Designing a task-oriented menu system is equivalent to designing a search space for mobile users to support their daily problem solving, at the abstract level. In order to construct a task-oriented menu system, therefore, we only have to enumerate solution methods and situations, that is, users' tasks and obstacles. Thus, in this paper, we focus on building a model of user behavior which contributes to the enumeration of tasks, and scaling up the prototype system to the scale of a real-world directory of services. For this purpose, it is more important to support designers for generating ideas about user activities, which are potential items in the task-oriented menu system.

First, we have to support user modeling with enough generality. As more activities are modeled, more needs for mobile internet services will be revealed. Ideally, all tasks that users want to achieve with a mobile handset must be listed and organized in a task-oriented menu. Of course, it is impossible to list every possible task with the limited number of members in our project. To achieve better results, we need a strategy and a framework to make our plan feasible. We therefore carefully designed task and domain ontologies with the guidelines described in Section 4. To enhance generality, we exploited an abstraction hierarchy in task organization. "Move", for example, can be a general task of "go to a restaurant", "go to a coffee shop", "travel", etc. At the same time, the quality of the ontologies and modeling framework was evaluated by the project members from NTT DoCoMo, who are experts in mobile internet services. In particular, NTT DoCoMo is a leading service provider in Japan whose mobile internet service is subscribed to by 46 million users. Many other service providers watch NTT DoCoMo's behavior and they are ready to follow it.

Second, the framework and modeling technology should be easy to transfer. As an application of our framework, we aim at supporting reorganization of mobile internet services from an abstract level like "move" to a concrete level like "get on a train to go from Osaka to Tokyo". As mentioned above, there are so many services that the limited number of project members cannot cover them all.

To cope with this issue, the ontology designer provides with not only a framework, ontologies, and models, but also their design principles and guidelines for usage. We plan to transfer our technology to let service providers modify their mobile internet services by themselves. Mobile internet service providers, the users of our framework, are neither experts in modeling nor experts in ontology engineering. We assume that they are not, in general, concerned about the design of ontologies. They utilize already designed ontologies and give feedback about their details. To reveal the difficulties involved in transferring our technology, we conducted several experiments. The results are described in Section 6.

Third, the framework should output models that represent obstacles for user activity. In many cases, users seek information when they come across some trouble on their way to achieving a goal or task, as we mentioned in Section 1. Mobile internet services are especially useful when users face some trouble or obstacle.

To satisfy this requirement, our framework explicitly includes models with such obstacles to be prevented or defeated by mobile internet services. Since such troublesome situations are the target of the mobile internet service, support for modeling them helps generate new ideas about mobile internet services.

4 Framework for Modeling

Fig. 2 depicts the framework of our system, where rectangles represent knowledge and circles represent people. The "ontology designer" designs ontologies for describing daily activities of mobile internet users, including tasks, domains, and obstacles. The "menu designer" cooperates with the ontology designer to exchange mutual expertise and builds the OOPS models for the upper menu design. By upper menu, we mean levels of the task-oriented menu system other than levels where concrete mobile internet services are described. Mobile users start from the top level of the menu, go through sub-menus (Fig. 2, upper right), and finally reach a concrete mobile internet service page (Fig.

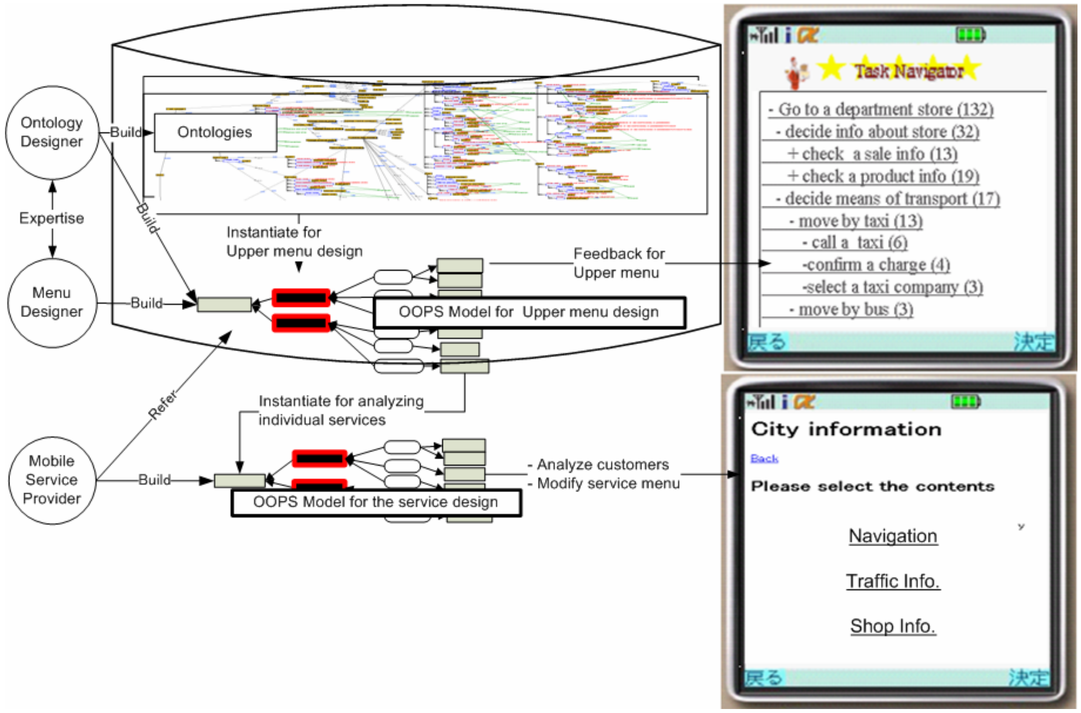


Fig. 2 Framework for modeling.

2, lower right). For this purpose, we now use Hozo and a graphic editor. We have designed other software which supports access to the ontologies and construction of the OOPS models by instantiating concepts in the ontologies. For ontology building, which is an important backbone of the OOPS models, we use Hozo.

Mobile service providers refer to the OOPS models developed for the upper menu design, instantiate them for analyzing their mobile internet services, and modify their service pages accordingly. In the process of instantiation, the service providers reflect their expertise about their business and customers in the OOPS model output. Also, they can listen to feedback from their customers to further refine the OOPS model; therefore, our framework involves customer desires indirectly.

Although the service providers usually have implicit business models for their own mobile internet services, they do not have generic task models for representing users' activities. The generic models and task/domain ontologies prepared by the ontology designer provide support in this respect.

4.1. Characteristics of the mobile internet service problem

A hierarchical description of the users' activities appearing on the screen of a task-oriented menu system (Fig. 1) looks like a hierarchical task planning model [13]. The problems which mobile internet services aim to solve, however, are essentially different from those problems dealt with by conventional expert systems, such as planning, diagnosis, and consultation. Conventional expert systems, including planning systems, evaluate input information and make a decision for the next search process to seek a solution. For this evaluation process, rich domain and problem solving knowledge is required, which makes building expert systems very difficult and/or task-bound.

On the other hand, a mobile internet service provides users with just "alternatives" which are different from the "solutions" given by conventional expert systems. The mobile user interprets the given information and utilizes it for solving a problem he/she faces on the spot. It does not require rich or exhaustive domain knowledge for evaluation of alternatives.

At the abstract level, designing a task-oriented menu system is equivalent to designing a search space for mobile users to support their daily problem solving. In order to compose a task-oriented menu system, we only have to enumerate solution methods and situations, that is, users' tasks and obstacles. Thus, we do not have to collect and organize exhaustive domain and problem solving knowledge.

The authors are specialists in building task ontologies [7], and have experience of their application to real-world problem solving [9]. Although there are huge numbers of "tasks" in the real world, the number that typically has to be solved by mobile handset users is not very large, since they are limited to daily-life tasks done outside the home. Furthermore, it is easier to organize task concepts than domain concepts because they are independent of domain, can be decomposed into subtasks, and have generality in the abstract space. For example, a task concept "buy a ticket for a movie" consists of two task concepts, "buy something" and "receive service" (including model of queuing). Both concepts can be applied to modeling similar tasks in various domains. Task concepts thus have generality by their nature, and hence, we can organize their structure at a high level of abstraction.

The authors are investigating reorganization of mobile internet services from the viewpoint of tasks. An approach based on task ontology enables mobile service providers to describe user activity models in terms of generic task vocabularies which are detached from the domain model. Furthermore, specification of the modeling process based on categorization of users' activities provides them with guidelines. Based on the task ontology, our method contributes to building homogeneous and generic models.

Fig. 3 shows an example of the OOPS model description process. Our framework has been designed to support four kinds of user models: (1) Usual activity based on a task decomposition model; (2) obstacles that hinder achievement of the activity; (3) how to prevent occurrence of trouble; and (4) how to solve trouble when it occurs.

4.2 Description of basic activity models

The OOPS modeling method is inspired by an ontology-based modeling framework for functionality of artifacts [8, 9]. It has been successfully deployed for engineering knowledge management in a manufacturing company [10]. It is summarized as follows: We define a function of a device as a result of teleological interpretation of its behavior when used to achieve a goal. One of the key ideas of the modeling framework is function decomposition based on the way a function is achieved (termed "way

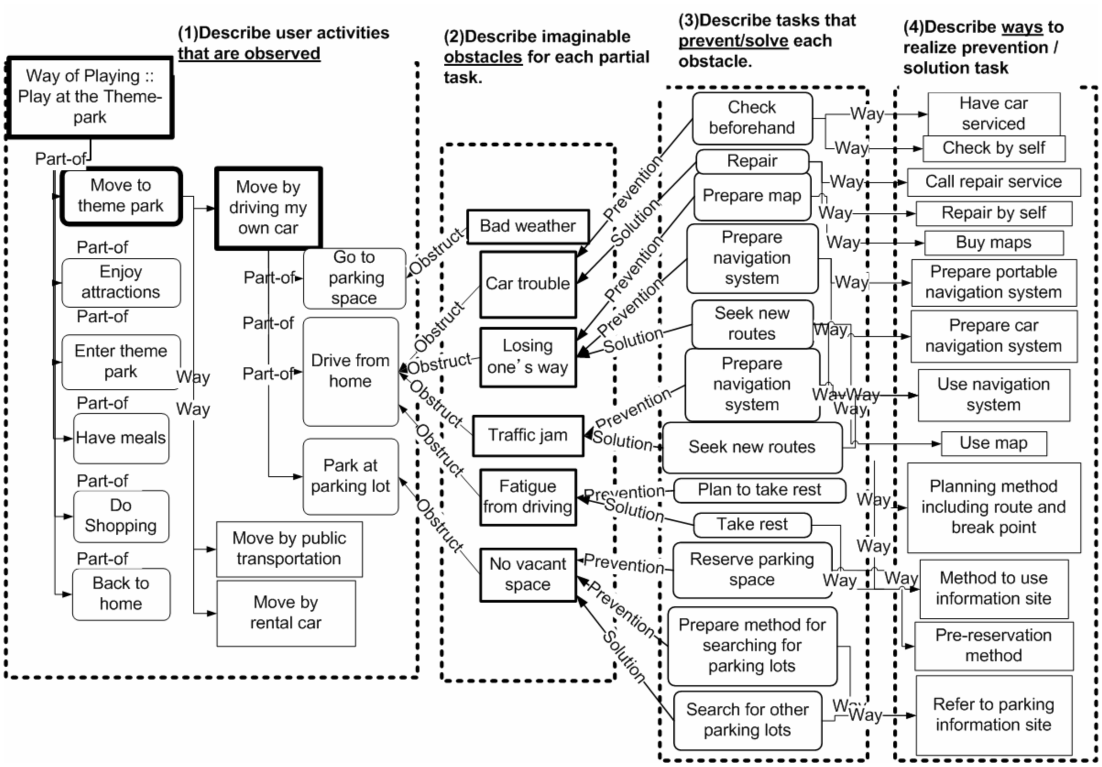

(1)Describe user activities that are observed
(2)Describe imaginable obstacles for each partial task.
(3)Describe tasks that prevent/solve each obstacle.
(4)Describe ways to realize prevention / solution task
Way of Playing :: Play at the Theme-park
Part-of
Move to theme park
Enjoy attractions
Enter theme park
Have meals
Do Shopping
Back to home
Way
Move by driving my own car
Go to parking space
Drive from home
Park at parking lot
Move by public transportation
Move by rental car
Obstruct
Bad weather
Car trouble
Losing one's way
Traffic jam
Fatigue from driving
No vacant space
Prevention
Solution
Check beforehand
Repair
Prepare map
Prepare navigation system
Seek new routes
Prepare navigation system
Seek new routes
Plan to take rest
Take rest
Reserve parking space
Prepare method for searching for parking lots
Search for other parking lots
Have car serviced
Check by self
Call repair service
Repair by self
Buy maps
Prepare portable navigation system
Prepare car navigation system
Use navigation system
Use map
Planning method including route and break point
Method to use information site
Pre-reservation method
Refer to parking information site

Fig. 3 The process of building OOPS models

of function achievement” below). The function of the whole system can be decomposed into sub-functions of sub-systems or components. The sub-functions are further decomposed into finer-grained sub-functions. Such a functional structure represents how the components achieve the overall goal-function of the whole system (i.e., how things work). We introduce the concept of way of function achievement as a conceptualization of background knowledge of functional decomposition, such as the physical principles serving as the basis of the achievement. For example, a cooling-function for a CPU can be achieved by the sub-functions “to provide air”, “to transfer heat”, and “to move air”. The *way of functional achievement* of the cooling function can be conceptualized as “heat conduction to moving air”. In general, there are other possible ways to achieve the same goal-function in this example, for instance, “heat conduction to coolant”. Conceptualization of the way of function achievement helps us detach “how to achieve” (way) from “what is intended to achieve” (function). We call this model the “function-achieved-by-ways model”.

The method is applicable to modeling users' activities. Since users' activities can be modeled at various sizes of granularity, we can decompose activities with larger granularity into smaller ones. Each and every activity has a goal and several ways to achieve it. We call such activity a “task”, and we can thus build a “task-achieved-by-ways model” by applying the same modeling method as that used for the “function-achieved-by-ways model”.

Fig. 3 represents a process of building an OOPS model. The dotted rectangle labeled (1) corresponds to the basic model of users' activities. It is described by instantiating generic models and/or ontologies. Description of the OOPS model starts from the task at the level of large granularity.

Next, ways to achieve the task are linked, and each of the ways is decomposed into a sequence of sub-tasks. Our “way” is similar to the “method” of CommonKADS [14] and “how to bundle” of the Business Process Handbook [15].

Following this process, the task of large granularity is decomposed into sub-tasks via ways. Area (1) in Fig. 3 represents that a task “Move to a theme park” is achieved by three ways. Among them, the way “Move by driving my own car” is decomposed into three sub-tasks, “Go to the parking space”, “Drive from home” and “Park the car at the parking lot”.

An important guideline in this framework is that the model of daily activity is described based on the observation of physical activity on the spot. Cognitive activities such as “plan to move more efficiently” or “learn traffic information beforehand” are not described in the model.

The guidelines and modeling process based on decomposition of the task contribute to making the modeling process easier and output more objective models. As described in Section 1, transfer of technology is important to realize a coherent task-oriented menu structure. If we allow modeling of non-observable activities, the quality of the output models will differ according to the skill of each model builder. The less knowledge about the task and/or domain he/she has, the worse the output model will become. In such a case, the process of modeling becomes implicit (not formalizable) and the transferability of the method decreases.

On the contrary, our guidelines allow modeling of physical activities only. The modeling process is thus simple: just observe physical activities and model them referring to the ontologies. The model does not contain cognitive activities such as “planning”, and is thus objective. The simplicity and objectiveness will contribute to the transferability of our technology in future.

4.3 Description of obstacles

Models of how to prevent/solve problems are described in three steps. First, the designer describes plausible obstacles for each partial task. For example, the task “Drive from home” has four kinds of obstacles: “Car trouble”, “Losing one’s way”, “Traffic jam” and “Fatigue from driving”.

Building models of possible obstacles is a unique feature compared to previous research like [3, 7, 11]. Since the most valuable mobile internet services provide information to solve such problems, our modeling method contributes to coping with obstacles that occur on the spot.

As mentioned in Section 3.1, the basic model of the users' activities is described by the decomposition of tasks. It is easier to foresee obstacles for a task with small granularity than for a task with large granularity. With our example in Fig. 3, it is easier to enumerate plausible obstacles for the task “Drive from home” than for the task “Go to the theme park”. Since we have more experience of driving than going to theme parks, we can enumerate more obstacles which might occur while driving. This characteristic meets our requirement that the framework should support the generation of new ideas. As described in Section 6, the results of our preliminary experiment also support this point.

4.4 Description of prevention/solution tasks

When we go out on a holiday, we generally make a plan and anticipate possible troubles beforehand. Then, we prepare to prevent the occurrence of the troubles. In the example in Fig. 3 (3), one of the tasks “Drive from home” may be obstructed by “Car trouble”. We can prevent occurrence of this obstacle by the task “Inspect the car beforehand”, for example. As described in Fig. 3 (3), the model of an obstacle and its preventive tasks are linked by “Prevention” links.

The most useful mobile internet services are ones that provide solutions for troubles that occur outside of the home. To deal with troubles that users come across and solve them, our method supports modeling of solution tasks as well as preventive tasks. In Fig. 3 (3), those tasks linked to the obstacles with a “Solution” link are solution task models. For example, we can solve the obstacle “Car trouble” with the task “Repair”.

Since models of obstacles are described for tasks with small granularity, we expect that one can imagine more tasks for their prevention and/or solution than conventional modeling methods. In Fig. 3, for example, it is more difficult to generate preventive ideas for obstacles to the task “Move to theme park” than for the task “Park at the parking lot in the theme park”, because the former task is abstract and contains many possible obstacles according to its interpretation. Based on the decomposition of the task models, our method helps generation of new preventive and/or solution ideas.

4.5 Description of ways to achieve prevention/solution tasks

The most important goal of mobile internet services is to provide ways to prevent or solve problems for the users. After modeling the prevention/solution tasks, our method supports description of ways to achieve the tasks. Examples are shown in Fig. 3 (4). The prevention task, “Check beforehand”, can be achieved in two ways, “Have the car serviced” and “Check the car by oneself”.

An advantage of our method is that it has the ability to allow designers to come up with new mobile internet services. Referring to the model of prevention and solution, designers of mobile internet services can check whether the current services are adequate. If there is no way or no internet service for a plausible obstacle, it means another service is now needed to solve it for the users. If a new mobile internet service is invented here, the service has a possibility of becoming a new business. The next section describes a scenario to improve existing mobile internet services by finding missing services from our OOPS model. In that sense, we expect that the models output from our framework should help in identifying new business opportunities. We plan to evaluate this issue in future.

4.6 Usage of the ontology and OOPS models

As mentioned at the beginning of this section, the OOPS model is used in two ways. The first usage is design of the upper level menu. By building a task ontology, as well as an ontology of obstacles, we can design the upper level of the task-oriented menu (Fig. 2).

The second usage is support for the design of mobile internet services. Fig. 4 depicts a design scenario. Suppose a designer currently tries to deliver general information about a city, such as the locations of stores and events in the city, and tries to improve his/her mobile internet service. Of course, an ideal scenario would be that the designer builds an OOPS model from scratch for his/her purpose, though some designers would be reluctant to learn the building method. While further study of the transfer of our technology will help to solve this problem in the near future, here we describe a most likely scenario.

First, the designer describes a user activity model in his/her own words (Fig. 4, Step (1)). Next, he/she envisages scenes to provide services according to his knowledge (Fig. 4, Step (2)). In this example, he/she described the user activity model at the very first step, “collection of information about goods”. Since the designer has little ability to provide information about goods, he/she focuses the model on “move” and “shopping”.

Next, to support the design process with the OOPS models and the ontologies, the original terms in a model are translated into ontological terms. Since several methods have already been proposed for mapping of terms and models (e.g., the ontology mapping method of Kitamura [16]), we do not focus on this point here. For example, “Go to store” is translated into “Move from home to store” by mapping the verb and supplementing missing nouns.

Now the designer can refer to the ontology and OOPS model using the ontological terms (Fig. 4, Step (3)). For designers, such concepts as “buy” or “move” are too simple to consider deeply and precisely. For this reason, they sometimes overlook important activities of mobile users and lose the chance for supporting them. Looking at the ontological definition of “Buy”, he/she may find that his/her service has to guide users to the location of a store, for example. Referring to the precise definition in the ontology prevents this.

Referring to OOPS models, the designer can learn others’ ideas about obstacles and solutions. For example, finding an OOPS model “Way to buy at a store” in the case database, he may notice that his user may come across a traffic jam on the way to the store. Since he notices such a new obstacle, he can add new services to his mobile internet service, such as “City map” for route guidance and “Route Navi” to predict and avoid traffic jams (Fig. 4, Step (4)). Now the designer can charge for use of the navigation service, for example. Like this, an OOPS model is able to work as a finder of new business opportunities.

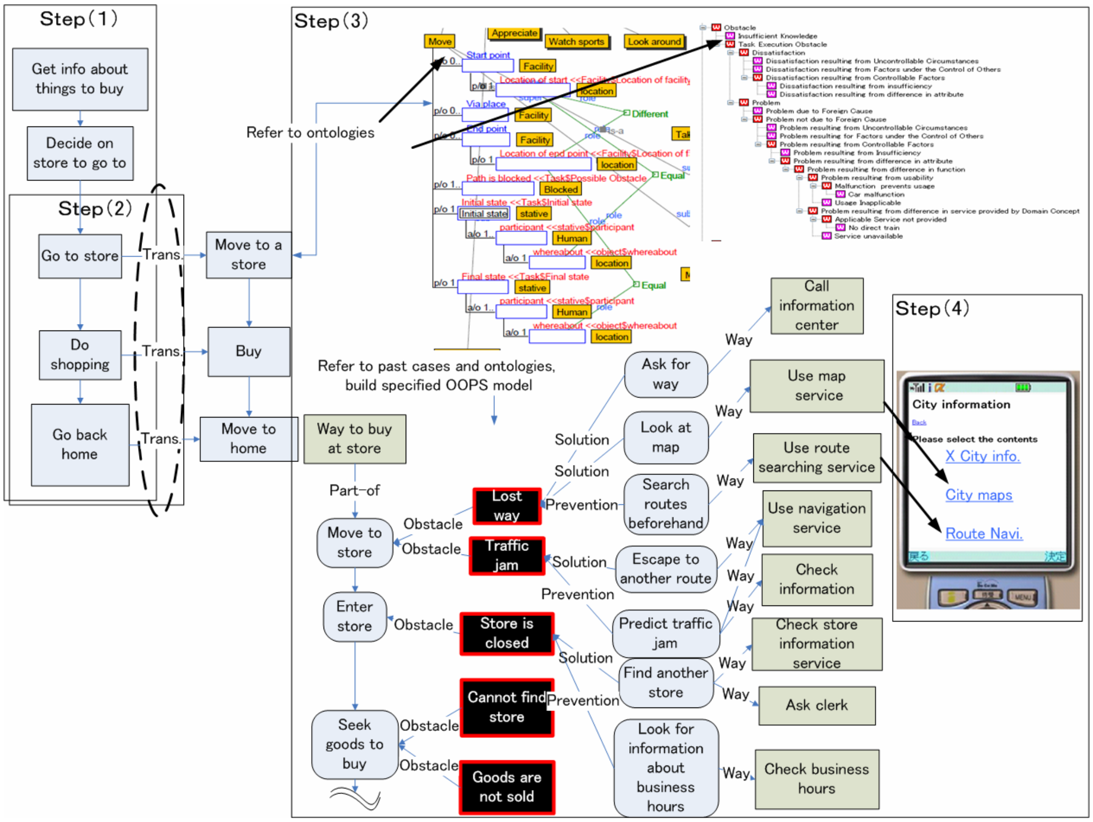


Fig. 4 A design scenario: improvement of a mobile service site.

5 Ontologies for the Users' Activities Model

Like Gruber's definition in [17, 18], we use the term ontology to mean a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. What is important is what an ontology is used for. With the purpose of enabling knowledge sharing and reuse, an ontology is a specification used for making ontological commitments. In practice, an ontological commitment is an agreement to use a vocabulary (i.e., make queries and assertions) in a way that is consistent (but not complete) with respect to the theory specified by the ontology.

For building OOPS models of mobile internet service users, ontologies that are referred to for describing the model should be scalable and general. This makes the OOPS modeling framework scalable and general, too. Although this requirement should be satisfied, it is difficult to build such ontologies at this early stage of research when experience of mobile internet services is still lacking. Therefore, we first build ontologies concerning mobile users' activities within limited situations. Then we expand the target domain and tasks. Furthermore, we exploit an abstraction hierarchy in the task ontology and reuse it when expanding the target tasks and domain. “Receive a service”, for example, can be a general action for buying a book, having a meal at a restaurant, etc.

We use Hozo [5, 6] as a tool for building ontologies. It supports export of the described ontologies to other standard formats, such as XML, RDF(S), OWL(-S), and so on. The latest version and example of ontologies are available at [4].

The process of building an ontology was supervised by experts in mobile internet services. The ontology designer designed it from the top level with about 200 concepts, and the experts supplemented these with about another 200 concepts by a bottom-up analysis in which they corrected terms and definitions extracted from current mobile internet services which reflect the viewpoint of actual users. As a whole, we built ontologies with Hozo consisting of about 400 basic concepts and 121 relational concepts based on assumptions about mobile users’ activities. Our ontologies consist of a task ontology, a domain ontology, and generic ways ontology. This section describes our ontologies, together with their design principles.

5.1 Task ontology

A task ontology is a system of concepts/terms for describing a problem solving structure domain-independently [7]. In building a task ontology, we describe each task as a sequence of states. When a user of a mobile handset performs a task, certain states around the user change. Fig. 5 is a graphical representation of “task” at the top level of the task ontology to be inherited by other task concepts (e.g., move, receive service, etc.). Rectangles linked to task concepts with right-angled links in Fig. 5 represent slots for attributes (denoted by a/o) or parts (denoted by p/o) of the tasks. The value of each slot is constrained to be an instance of the class described in a rectangle to the right of the slot. Lastly, the short text description in each rectangle represents a role concept [19].

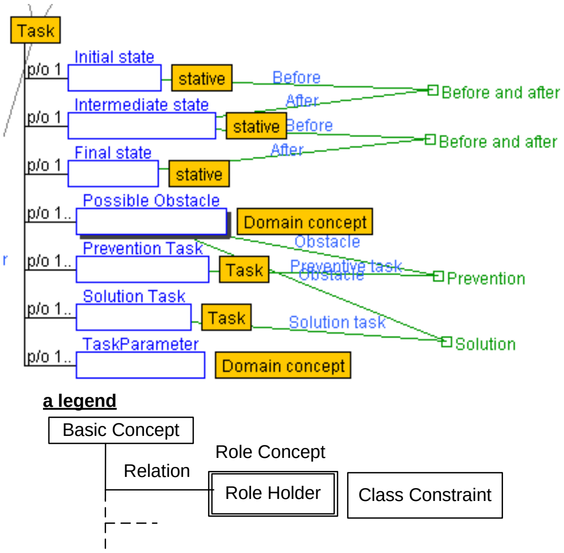


Fig. 5 Definition of “Task” (Screen shot of Hozo).

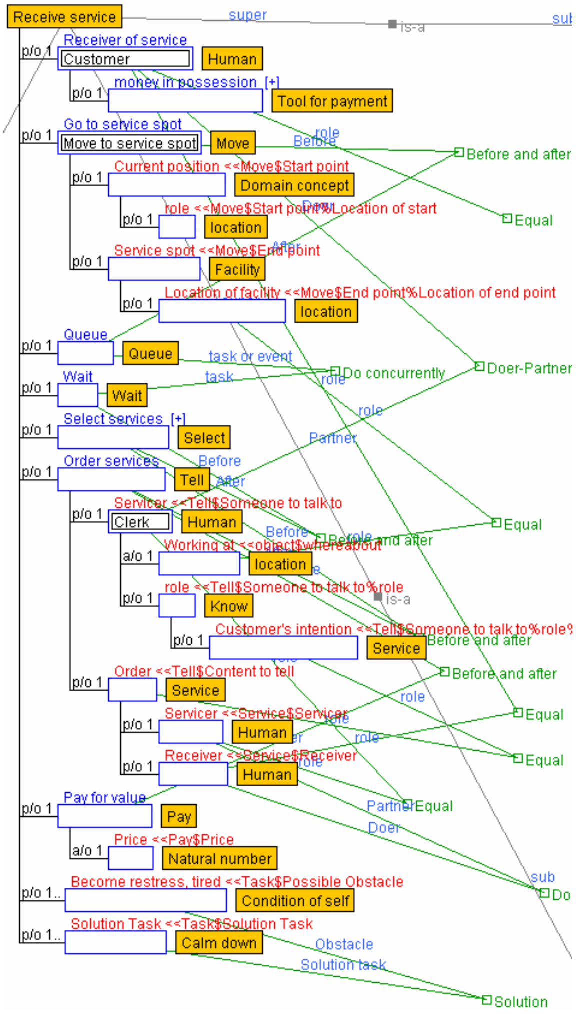


Fig. 6 Example of a task concept “Receive service”.

The first three slots in Fig. 5 represent changes of states as the task is performed. The values of each slot are constrained to be an instance of a “stative” class which is at an upper level of concepts about the states of things in the target domain. The fourth slot represents possible obstacles which correspond to “obstacles” in the OOPS models (see Section 4.3). This slot shows that a domain concept plays the role of a “Possible Obstacle” (e.g. “rain” obstructs “playing at theme park”).

The fifth (sixth) slot represents prevention (solution) tasks for the obstacles. It corresponds to “Prevention task” (“Solution task”) in the OOPS models. Although there are three kinds of tasks in our OOPS models, we build a task ontology for “Task”, but we do not build ontologies for “Prevention task” or “Solution task” because they are merely examples of the “Task” role. A task concept “Withdraw cash” for example, prevents the occurrence of an undesirable state “short of cash”. It also solves the undesirable state “short of cash” if it has already occurred. As this example shows, any task can play the role of “prevention” or “solution”. The ontology of a normal “task” is enough, and we model “Prevention” and “Solution” tasks as normal tasks, as shown in the 5th and 6th slots in Fig. 5.

Fig. 6 shows an example of a task concept “Receive service”. It consists of a sequence of several subtasks. As the first slot indicates, a person plays the role of “Customer”, and the customer first goes to the service spot. After that, he/she joins a queue, waits for a service, selects a service, asks the clerk, and pays. The domain concept “condition of self” plays the role of an obstacle for those people waiting for services. The obstacle is solved by the task “calm down”. The task concept “Receive Service” is inherited by other task concepts such as “Buy”, “have a meal”, and so on.

5.2 Domain ontology

As described in the beginning of this section, the domain concept contains a tremendous number of concepts and we cannot cover them all. To cope with this, we adopt the Activity First Method (AFM) [11], a methodology of building a task ontology with a necessary domain ontology from a task-specific point of view. First, we limit the target tasks and domain (e.g., to activities in a theme park). Within this limited situation, we analyze the activities first to develop a task ontology. Along with the development of the task ontology, we develop a domain ontology which contains at least necessary concepts for defining each task concept. Next, we expand the target tasks and domain (e.g. activities in daily life). AFM thus provides us with task-specific guidelines for articulating and organizing domain concepts.

Referring to the domain concepts, we define task concepts like Fig. 6. As the example in Fig. 6

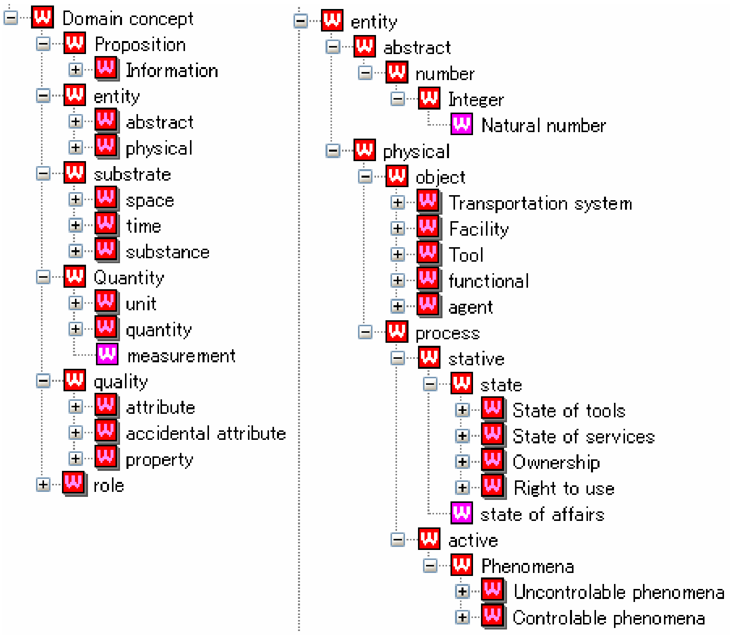


Fig. 7 (left) Top level of the domain concept, (right) hierarchy of a domain concept “entity”.

shows, the meaning of a domain concept depends on the task concept which refers to it. To make the domain ontology general and scalable, we separate the interpretation of the result from the domain ontology. As a kind of weather, “rain”, for example, is bad weather for people playing at a theme park outside. On the other hand, this weather is good for a farmer waiting for water for his/her field. So we define “rain” or “sunny” in our domain ontology, but we do not define, for instance, “bad weather” or “good weather for driving”, which contains a task-dependent interpretation of the result, which would make the domain modeling less general.

Furthermore, even if we build a domain ontology assuming only limited situations of the users’ activities, there are still many ways of defining domain concepts. To make the domain ontology coherent, we apply a “top-level ontology” to the upper level of our domain ontology to organize it. The left part of Fig. 7 represents our top-level ontology. The domain concept consists of “Proposition”, “entity”, “substrate”, “Quantity”, “quality”, and “role”. Since all of these concepts are defined explicitly, ontology designers can search for or add domain concepts to define task concepts as they wish. If a designer wants to refer to the domain concept “weather”, for example, he/she can look into the concept “entity” which contains concepts about “Uncontrollable phenomena”, including weather (see the right part of Fig. 7).

5.3 Generic way concepts

Although the concept of *way* is a key one for building our OOPS models, it strongly depends on the task and domain concepts. Thus, its explicit separation is important to keep generality of the task and domain ontologies.

For describing our OOPS models which represent various methods to achieve task goals, we developed a hierarchy of generic way concepts separately from the task and domain ontologies. For a task concept “move”, for example, we have several ways, such as “move by self”, “move by train”, “move by rental car”, etc. to achieve the goal (see Fig. 8).

6 Evaluation of OOPS Modeling Method

We conducted several experiments with 25 subjects in their twenties (11 subjects), thirties (12 subjects), and forties (2 subjects). Fourteen of them use mobile internet services every day, and 3 of them use them for 3 or 4 days per week. All subjects had experience of using computers, ranging from

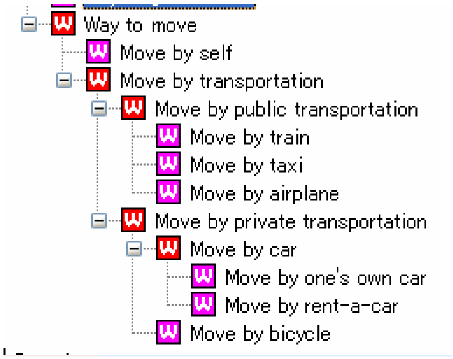


Fig. 8 Part of conceptual hierarchy of generic way “Way to move”.

5 to 18 years (average: 10.96 years), but no experience of user modeling or ontology. In the experiments, we interviewed the subjects to determine their actual views.

6.1 Understandability of OOPS models

For two groups of subjects, we conducted an experiment to examine the understandability of our OOPS models. In this experiment, we gave both groups (G1 and G2, 5 subjects randomly selected for each) a list of 100 existing mobile internet services, with their titles and brief explanations, such as “JR Travel Navigator” and “Using the latest timetables, you can search for connections among the different transportation systems.” We gave a task to each of the subjects to identify, using the list, possible usage scenarios of the mobile internet services in given situations. The situations were (a) at a theme park, and (b) at a train station. We also asked the subjects not to describe the usage scenarios with the same meaning as in the brief explanation itself, but in terms of interpretation results. For example, in the case of the “JR Travel Navigator”, we treated “Search for the last train from the theme park bound for Tokyo” as a valid answer, but not “Seek for connections among the different transportation systems”.

To each subject in G2, we gave correct OOPS models (Fig. 2) for the two situations described above, and to subjects in G1, we gave nothing. We allowed one hour for each group, including time for instructions, and after the experiment, we collected the subjects' answers.

Table 1 shows the results for the number of valid answers. Since we selected the list of 100 service sites at random, it included several sites which had no relation to the situations given, like “Information about driving school”. As Table 1 shows, with reference to the OOPS models, subjects in G2 described more usage scenarios. Although the difference of the result in the situation “Theme park” was small, subjects of G2 described 1.5 times as many scenarios for the situation of “Station” than those of G1; thus, there is a discernable difference as a whole. Furthermore, subjects in G2 also gave rather positive comments on this point, such as “The OOPS model was useful to find the boundary of a given situation” and “The model was useful to some extent when I was stuck for new ideas for using it.” The results show that the OOPS model can support the imagination of users when using the mobile internet services in various scenarios.

6.2 Support for modeling users’ tasks

For three groups of subjects, we examined whether or not our framework supports modeling users’ tasks. In this experiment, we gave two other groups (G3 and G4, 5 subjects each), different from the groups in the experiment described in Section 6.1, a task to imagine and describe as many activities related to shopping in a department store as possible. To one group (G3), we gave no instructions, and the subjects described activities by natural language like “Search for information about the department store.” To the other group (G4), we gave an instruction manual about our modeling method, modeling tool, libraries based on ontologies for modeling, and some examples. Subjects in G4 described a graphical model for the given task. After the experiment, we counted the numbers of tasks described by the subjects.

For further comparison, we conducted another experiment. We formed 9 groups from 41 university students from the Graduate School of Computer Science. Each group consisted of 4 or 5

Table 1. Total number of the valid usage scenarios described by subjects.

Situation	G1 (No reference)	G2 (With reference)
Theme park	159	177
Station	67	100

Table 2. Number of task models described by subjects.

Number of tasks described by subjects	G3 (No instruction)	G4 (With instruction)	G4' (G4 without obstacles)	G5 (KJ method used by 9 groups)
Average	24.6	56.2	30.4	32.7
Minimum	18	39	19	23
Maximum	34	72	37	54

members. We asked them to perform the same task by the KJ (Kawakita Jiro) method [20], which is a well-known method for facilitating generation of ideas by group discussion. We instructed the students about the method for 10 minutes, and allowed them 20 minutes for description of the task model by natural language. During the instruction, we encouraged them to describe not only normal scenarios but also problematic ones, such as losing one's way to the department store. Although the KJ method requires a discussion process, which sometimes becomes a bottleneck for the generation of ideas, there seemed to be no communication overhead for it since the subjects were all from the same faculty and therefore knew each other.

Table 2 shows the results. The second row in the table shows average numbers of tasks imagined by the subjects. Compared to G3, subjects in G4 imagined about twice the number of tasks with our modeling framework. Furthermore, even the minimum number of the tasks described by one of the subjects in G4 exceeded the maximum number of the G3 subjects.

Analyzing the data, we found that subjects in G4 described many preventive tasks or solution tasks for obstacles in our modeling framework. For comparison, we counted the number of the tasks identified by G4, excluding prevention or solution tasks connected to nodes of obstacles (G4’ in Table 2). The average number of task models in G4’ was about 54% of that in G4. Comparing tasks contained in G3 and G4’, both of them mainly consisted of normal activities like “buy a ticket”, and only a few prevention tasks or solution tasks were included. This shows that the description of obstacles revealed alternative situations for subjects and stimulated new ideas.

Comparing G3, G4, and G5, the average number of tasks described by the subjects satisfied

G4 (OOPS model) > G5 (KJ method) > G3 (No instruction).

This result shows that the OOPS modeling method facilitated the generation of ideas more readily than the conventional method. As we described in Section 4, we need to transfer our technology to providers of mobile internet services who can describe more domain-specific activity models. This characteristic of the proposed method, namely, supporting the generation of new ideas, meets this requirement; the method thus has the potential to contribute to the realization of task-oriented menus.

6.3 Similarity of the described models and subjective evaluation by experts

We evaluated the similarity of the task models described by the subjects from the viewpoint of structure, in comparison with “correct” examples, as defined by the authors. The results show that the similarity of task models described by 5 subjects using the ontologies was 2.5 times higher than those without ontologies.

As described in Sections 6.1 and 6.2, our OOPS modeling framework with ontologies supports the generation of many ideas about users’ activities. Furthermore, experimental results show the similarity of the described models. As described in Section 3, we have to transfer the OOPS modeling technology to many service providers to realize coherent task-oriented mobile internet services. The experimental results show that our method is promising for supporting better analysis of users’ activities, as well as description of homogeneous models, regardless of differences among modelers.

Furthermore, for about one year we transferred our ontology and OOPS modeling method to a group of experts from mobile service company. They applied OOPS modeling method for modeling customers who visit a department store, and developed an application which provides information about the store to the customers on the spot. They claimed positively that the ontology makes their user models clear and helps understanding of the models built by other members. Also they claimed that OOPS modeling method facilitated generation of ideas about users' activities in the store. These two points support that our framework contributes to building mobile internet applications by experts.

7 Related work

Boreum *et al.* investigated which factors of mobile internet services are important for users [2]. They interviewed people from three countries, Japan, Korea, and Finland, which have mature mobile internet service markets. According to their analysis, both the “logical order of the menu” and “meaningful classification of the contents” are considered to be important by many subjects from the three countries. The results validate our approach for improving the menu system and classification of the contents, which should contribute to user satisfaction.

In the field of semantic web research, there are several mainstream approaches aiming at achieving high quality of web services for consumers. Although the goal of our research is the same, our framework centers on a description of users' activities, whereas many other research focuses on description of web services. For example, OWL-S [21], SWSO/FLOWS [22], and WSMO [23] describe models of web services first, and define ontologies necessary for describing those services. Lastly, they translate the service models into executable forms by referring to the ontologies. They have been submitted to the W3C for standardization, and several guidelines and tools have been released.

These research projects mainly focus on how to utilize existing web resources, not how to build or improve the quality of the resources or ontologies. Research related to WSMO [23], a mainstream approach in Europe, has resulted in the release of several tools for designing and developing semantic web services, including WSMX [24]. WSMX provides support at various stages, from finding services, to mediation of the services and execution of the composed services.

To satisfy users' needs, many researchers today focus on better composition of existing mobile internet services. Our modeling method, which focuses on better analysis of users' needs, is able to strengthen the research explained in this section. Hierarchical Task Network planning (a general explanation is given in [13], and applications for web services are described in [25]) supports how to divide and conquer a web user's “problem”, which resembles our task decomposition process in OOPS modeling. In the process of composing web services, Motahari-Nexhad [26] proposes how to identify mismatches of the interfaces and protocols between two services to be composed. Domingue [27] describes how to cope with heterogeneous interaction patterns with the framework of IRS-III, and Ashri [28] discusses the interaction protocols in their experience of IRS-II. In such an organization process, alignment of the ontologies behind the services is necessary. Omelayenko [29] proposes a method for mapping meta-ontologies among web services, and Ehrig [30] describes a machine-learning method for an initial stage of ontology alignment. Tsz-Chiu Au [31] points out that it is unrealistic to assume that the information provided by the web services is static in many cases. They propose another framework to deal with volatile information, taking a ticket reservation service problem as an example.

These studies, however, do not consider the contents of the mobile internet services. In contrast, our approach starts from analyzing users' activities, including problematic situations which require mobile internet services. We then design mobile internet services based upon the user model. Most research on web services implicitly assumes that web browsing is done on desktop computers; thus, the time and cost involved in searching and evaluating the answers are not of much concern. On the other hand, in the case of our mobile internet service problem, users need prompt answers. Thus, we pay attention to quickly navigating users to the proper service which is the source of the answer. We leave evaluation of the answers to the users themselves.

Masuoka proposed a Task Computing framework and built a ubiquitous environment which provides more than 100 web services [32]. The web services are described by OWL-S, and the environment changes dynamically. The ubiquitous environment is unique because it deals with dynamic changes such as sudden appearance/disappearance of clients/services, like the real world.

MIT's Process Handbook Project [15] deals with knowledge models about businesses. It focuses on modeling business activities and has a taxonomy of basic business activities. However, the method for building the model is implicit, and confusion of task concepts with way concepts occurs with some models. One of the models, “buy in a store”, consists of a task concept “buy” and a way concept “in a store”, for example. Such confusion lowers the generality of the model, and does not meet our requirements.

In the field of the human–computer interactions, although there are many studies about web interfaces, there are not so many studies specific to mobile phones. James [33] compares the efficiency of two text input methods used on mobile phone: multitap and prediction. Kamvar [34] analyzed search patterns of a search engine specifically designed for mobile internet services on a large scale. The search patterns resembled those of desktop search engines. The results show that mobile internet services are still not organized well for mobile users. The users rely on search engines, as they do on desktop computers, since they cannot reach the necessary services.

Together with such conventional technologies, we plan to design an interface for a task-oriented menu system. It is unrealistic to replace everything with a task-oriented style; rather, integration of a search engine and/or domain-oriented classification will be necessary for some tasks. For example, a task "shopping" deals with millions of items which require conventional search technologies.

8 Conclusion and future work

To improve mobile internet services based on semantic web technology, in this paper we proposed the OOPS modeling method, a new framework for analyzing and building models of users' activities. Since our past research demonstrated that a new type of menu system, which we call a "task-oriented menu", has greater potential to provide useful information for mobile internet service users, in the present research, we aimed at realizing such a task-oriented menu system.

In designing the new framework, we focused on three important issues: generality and scalability of the framework, ease of transfer of the technology, and modeling of obstacles under the mobile environment for sparking the generation of new ideas.

To enhance generality and scalability, we first analyzed the character of mobile internet services. The problems which mobile internet services are provided to solve are essentially different from the problems dealt with by Web Service composition or conventional expert systems. Conventional expert systems evaluate input information and make a decision for the next search process to seek solutions. These processes require rich domain and problem solving knowledge. This makes building expert systems very difficult and/or task-bound.

On the other hand, a mobile internet service typically provides users with just "alternatives", which does not require such knowledge, unlike the "solutions" given by conventional expert systems. At the abstract level, designing a task-oriented menu system is equivalent to designing a search space for mobile users to support their daily problem solving. In order to compose a task-oriented menu system, we only have to enumerate solution methods and situations, that is, users' tasks and obstacles.

For generality and scalability, we carefully designed task and domain ontologies with guidelines. In designing the task ontology, we adopted the Activity First Method [11] and separated task concepts from domain concepts to keep the domain ontology general and scalable. Defining each task concept, we applied our theory of role concepts [19] to keep basic task concepts general. Although our OOPS model contains three kinds of tasks, we build only normal task concepts, and represent prevention or solution tasks as special roles played by normal tasks.

In designing the domain ontology, we excluded interpretation results, such as the effect of natural phenomena like weather conditions, to keep its generality. Furthermore, we adopted a top-level ontology for its organization to make it coherent. Lastly, our modeling framework explicitly represents obstacles, and our experimental results showed that description of obstacles enhances task models.

Our OOPS modeling framework and ontologies were designed to support descriptions of homogeneous models regardless of differences among modelers, which in turn facilitates the transfer of our technology. Experimental results in Section 6 showed the potential of our proposed framework on this point. As described in Section 1, we conducted this research in corporation with NTT DoCoMo, whose moves are closely watched by many other service providers. We can expect feedback from not only NTT DoCoMo but also these other providers, which should lead to refinement of our work. To support OOPS model building, we currently use Hozo and a graphic editor. We have designed another tool for this purpose and plan to develop it to cooperatively work with Hozo, our ontology editor, to support description.

Since our proposed framework satisfied these specifications and worked well with the limited set of tasks and domains considered in the study, we are now aiming at the next goal. To realize task-oriented mobile internet services, our next step is to expand the target domain. Our next target will be activities in other domains, like towns, streets, and so on.

We expect that we will be able to reuse the task models already built for expansion of the target domain, since the task concepts are common, regardless of differences among various domains, as described in Section 1. Of course, further effort must be devoted for this expansion; to realize task-oriented menus in future, we are planning to repeat the cycle of building ontologies and having them reviewed by specialists.

References

1. NTT DoCoMo annual report
   http://www.nttdocomo.co.jp/english/corporate/ir/finance/annual/index.html
2. B. Choi, I. Lee, J. Kim, and Y. Jeon (2005), *A Qualitative Cross-National Study of Cultural Influences on Mobile Data Service Design*, In Proc. of the SIGCHI conference on Human factors in computing systems 2005 (CHI2005), pp.661-670.
3. T. Naganuma and S. Kurakake (2005), *Task Knowledge Based Retrieval for Services Relevant to Mobile User's Activity*, In Proc. of the ISWC2005, pp. 959-973.
4. Hozo web site
   http://www.hozo.jp/
5. K. Kozaki, Y. Kitamura, M. Ikeda and R. Mizoguchi (2000), *Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role"*, In Proc. of PKAW2000, pp.205-221.
6. K. Kozaki, Y. Kitamura, M. Ikeda and R. Mizoguchi (2002), *Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship"*, In Proc. of EKAW2002, pp. 213-218.
7. R. Mizoguchi, J. Vanwelkenhuysen and M. Ikeda (1995), *Task Ontology for Reuse of Problem Solving Knowledge*, In Proc. of KB&KS '95, The Netherlands, pp. 46-59.
8. Y. Kitamura and R. Mizoguchi (2004), *Ontology-based Functional Knowledge Modeling Methodology and its Deployment*, In Proc. of the EKAW 2004, pp. 99-115.
9. Y. Kitamura and R. Mizoguchi (2003), *Ontology-based description of functional design knowledge and its use in a functional way server,* Expert Systems with Application Vol.24(2), pp.153-166.
10. Y. Kitamura, M. Kashiwase, M. Fuse, and R. Mizoguchi (2004), *Deployment of an Ontological Framework of Functional Design Knowledge,* Advanced Engineering Informatics, Vol.18(2), pp.115-127.
11. R. Mizoguchi, M. Ikeda, K. Seta and J. Vanwelkenhuysen (1995),*Ontology for Modeling the World from Problem Solving Perspectives*, Proc. of IJCAI-95 WS on Basic Ontological Issues in Knowledge Sharing, pp. 1-12.

12. M. Sasajima, Y. Kitamura, T. Naganuma, S. Kurakake, R. Mizoguchi (2006), *Task Ontology-Based Framework for Modeling Users' Activities for Mobile Service Navigation*, In Proc. of Posters and Demos of the 3rd European Semantic Web Conference (ESWC2006), pp. 71-72.
13. S. Kambhampati (1997), *Refinement Planning as a Unifying Framework for Plan Synthesis*, AI MAGAZINE, summer 1997, pp. 67-97.
14. G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W.V. de Velde, and B. Wielinga (2000), *Knowledge Engineering and Management - The CommonKADS Methodology*, MIT Press.
15. T.W. Malone, K Crowston, and G.A. Herman (2003), *Organizing Business Knowledge - The MIT Process Hand Book*, MIT Press.
16. Y. Kitamura, N. Washio, M.Ookubo, Y. Koji, M. Sasajima, S. Takafuji and R. Mizoguchi (2006), *Towards a Reference Ontology of Functionality for Interoperable Annotation for Engineering Documents*, In Proc. of Posters and Demos of the 3rd European Semantic Web Conference (ESWC2006), pp. 75-76.
17. T.R. Gruber (1993), *A translation approach to portable ontologies*, Knowledge Acquisition, Vol.5(2), pp.199-220.
18. T.R. Gruber (1995), *Toward principles for the design of ontologies used for knowledge sharing*, in International Journal of Human-Computer Studies, Vol. 43, Issues 4-5, pp. 907-928.
19. E. Sunagawa, K. Kozaki, Y. Kitamura and R. Mizoguchi (2005), *A Framework for Organizing Role Concepts in Ontology Development Tool: Hozo*, In Technical Report FS-05-08, Papers from the AAAI Fall Symposium, “Roles, an Interdisciplinary Perspective”, AAAI Press, pp. 136-143.
20. Jiro Kawakita (1967), *Hassouhou - Souzousei Kaihatsu no Tameni -*, ISBN 9784121001368, CHUOKORON-SHINSHA (in Japanese).
21. OWL-S: DAML Services (DAML-S/OWL-S) http://www.daml.org/services/owl-s
22. Semantic Web Services Ontology (SWSO) http://www.daml.org/services/swsf/1.0/swso
23. Web Service Model Ontology (WSMO), W3C Member Submission 3 June 2005, http://www.w3.org/Submission/WSMO
24. Web Service Execution Environment (WSMX), W3C Member Submission 3 June 2005, http://www.w3.org/Submission/WSMX
25. U. Kuter, E. Sirin, D. Nau, B. Parsia, and J. Hendler (2005), *Information gathering during planning for web service composition*, Journal of Web Semantics, Vol.3(2-3), pp. 183-205.
26. H.R. Motahari-Nexhad, A. Martens, F. Curbera, and F. Casati (2007), *Semi-Automated Adaptation of Service Interactions*, In Proc. of WWW2007, pp. 993-1002.
27. J. Domingue, S. Galizia, and L. Cabral (2005), *Choreography in IRS-III - Coping with Heterogeneous Interaction Patterns in Web Services*, In Proc. of ISWC2005, LNCS 3729, pp. 171-185.
28. R. Ashri, G. Denker, Darren Marvin, Mike Surridge and Terry Payne (2004), *Semantic Web Service Interaction Protocols: An Ontological Approach*, In Proc. of ISWC2004, LNCS 3298, pp.304-319.
29. B. Omelayenko (2003), *RDFT: A Mapping Meta-Ontology for Web Service Integration*, Knowledge Transformation for the Semantic Web, B. Omelayenko and M. Klein (Eds.), pp. 137-153, IOS Press.
30. M. Ehrig, S. Staab, and Y. Sure (2005), *Bootstrapping Ontology Alignment Methods with APFEL*, In Proc. of ISWC2005, Springer, LNCS 3729, pp.186-200.
31. T. Au, U. Kuter, and D. Nau (2005), *Web Service Composition with Volatile information*, In Proc. of ISWC2005, LNCS 3729, pp. 52-66.

32. R. Masuoka, B. Parsia and Y. Labrou (2003), *Task Computing - The Semantic Web Meets Pervasive Computing*, ISWC2003, Springer, LNCS 2870, pp. 866-881.
33. C.L. James and K.M. Reischel (2001), *Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance*, In Proc. of CHI2001, pp. 365-371.
34. M. Kamvar and S. Baluja (2006), *A Large Scale Study of Wireless Search Behavior: Google Mobile Search*, In Proc. of CHI2006, pp.701-709.