Special Articles on Multimedia Information Processing

Service Navigation Technology based on Task Knowledge

Takefumi Naganuma and Shoji Kurakake

As a part of the many services available to users in a mobile environment, DoCoMo is researching an intelligent information platform for presenting services that fit the circumstances and objectives of individual users in the real-world. This article describes the study of a service navigation system using a knowledge base that stores user task models extracted from service-usage cases to simplify the discovery of services applicable to real-world problem solving.

New Technology Reports

1. Introduction

Mobile Internet technology is making rapid progress especially in the field of mobile terminals. The number of mobile terminals capable of Internet Protocol (IP) connections in Japan alone has grown to 69 million units (as of March 2004) [1]. In background of this high penetration rate, the volume of Web content is ever increasing that user can access a wide variety of information services from mobile terminals at anytime and anywhere. For the future, realization of ubiquitous services based on short-range wireless communications such as Wireless Local Area Network (WLAN), Bluetooth[*1], and Radio Frequency IDentification (RFID) are expected, and these would make more information services via mobile terminals available for users. With variety of information services available, the ability to select those services that would fit one's personal objectives should make it possible to construct an environment where many of the problems that a user faces in daily life can be solved regardless of location.

To provide optimal information services to users is to develop an intelligent service-provision infrastructure for selecting

*1 Bluetooth is a registered trademark of Bluetooth SIG. Inc., USA.

and presenting optimal information services taking user circumstances and objectives into account. The current service-provision infrastructure, however, is dominated by directory-based search menus and keyword-based full-text search engines. Using such a search system, a user selects some information services as candidates and then evaluates each of them using hints from index descriptions and the like. In general, an information service is used to solve a problem of some kind. This problem-solving process can be defined as a six-step process model [2] as shown in **Figure 1**. In this model, the support range of a conventional search engine corresponds to the third step (Access to information sources). However, the users should be able to decide the neccessary information in second step (Information Searching Strategies), it is unuseful for the user that has no information searching skills to select the proper information service.

Against the above background, we propose a service navigation system that aims to enable users including those with no information-searching skills to discover information services for problem-solving purposes using a mobile terminal. For this system, we construct a knowledge base that stores user task models extracted from service-usage cases and assign a correspondence between user tasks and information services. In this article, we present a technique for constructing such a knowledge base and describe a query matching system. We also describe a prototype system that we constructed and present the results of a subjective evaluation.

2. Background

2.1 Related Research

Recent years have seen much research activity on the semantic identification of information resources by attaching metadata[*2] to information resources on the Web and clarifying the semantic relationship between different instances of metadata [3] [4]. This capability is expected to improve the accuracy of searches and facilitate the integration of information based on semantic information. The semantic information addressed by these research studies is limited, however, to information for uniquely identifying content (i.e. metadata related to corporate names, personal names, product information, etc.). Such general-purpose semantic information by itself cannot be easily applied to identifying information resources that are needed for problem-solving purposes.

There is also research on using a knowledge base to infer the search goal from search text input by the user and then creating better search text with the aim of enabling even users unfamiliar with information systems to make optimal information searches. The GOOSE system [5], for example, adopts a large amount of knowledge (common sense knowledge) collected from general users on the Internet to infer a user's search goal and to present search results for achieving that goal. This system, though, infers a search goal by first analyzing the user's natural-language text input and then comparing that input with a

*2 Metadata describes the attributes and other characteristics of target data.

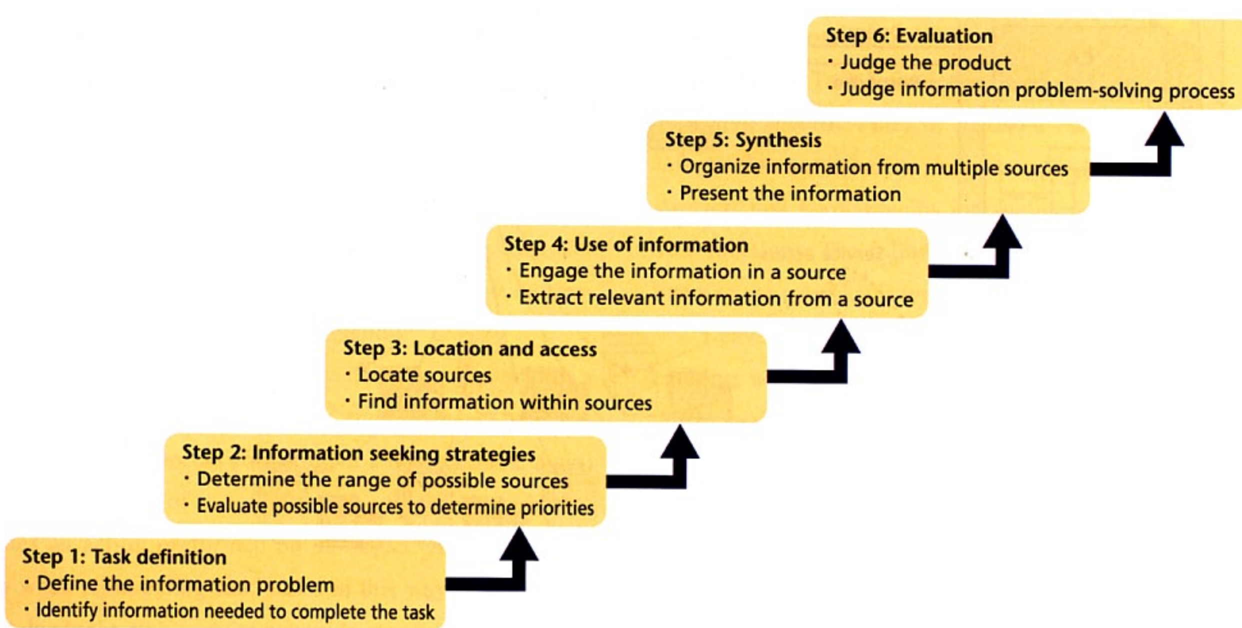


Figure 1 Information problem-solving process (big6 model)

knowledge base, which means that a short search string would probably not be sufficient for making an appropriate inference. In other words, it would not be easy to apply such a system to an environment where character input is difficult as in the case of mobile terminals.

2.2 Approach

The behavior that most people follow in everyday problem solving is to divide a problem into several sub-problems and to then solve those sub-problems to reach one's objective. For example, human daily life is driven by "proximal goals" (short-term goals), which are derived from "distal goals" (long-term goals). In the area of using information services to solve a problem, the problems which can be solved directly by using an information service corresponds to "proximal goal" and a "distal goal" is achieved by solving sub-problems corresponding to "proximal goals" derived from the "distal goal" [6]. If we apply this idea to information searching in relation to problem solving where information services are used as a means of finding solutions, we can imagine a user dividing a long-term goal into appropriate short-term goals based on functions provided by information services.

The following explains this concept using a specific example. A user that thinks "I'd like to go to a amusement park this weekend—what kind of information service would be available?" is posing a real-world problem. The information-seeking action generated by this problem may take the following form:

"go to amusement park ⇒ decide one's schedule ⇒ decide on ways to destination ⇒ decide on route for destination."

In this scenario, "go to amusement park" corresponds to the long-term goal, and the sub-problems of "decide one's schedule," "decide on ways to destination," and "decide on route for destination" that must be solved for achieving that long-term goal correspond to short-term goals.

To realize the above approach, we deal problems as "tasks" and construct a task knowledge base that stores the knowledge needed for dividing an upper-level task into lower-level sub-tasks as "task knowledge." We also construct a service knowledge base that stores the correspondences between tasks and information services as "service knowledge."

3. Service Navigation System

Figure 2 shows the architecture of the proposed service navigation system. This system consists of Task Knowledge Base (TKB) that stores task knowledge; Service Knowledge Base (SKB) that associates TKB with the Uniform Resource

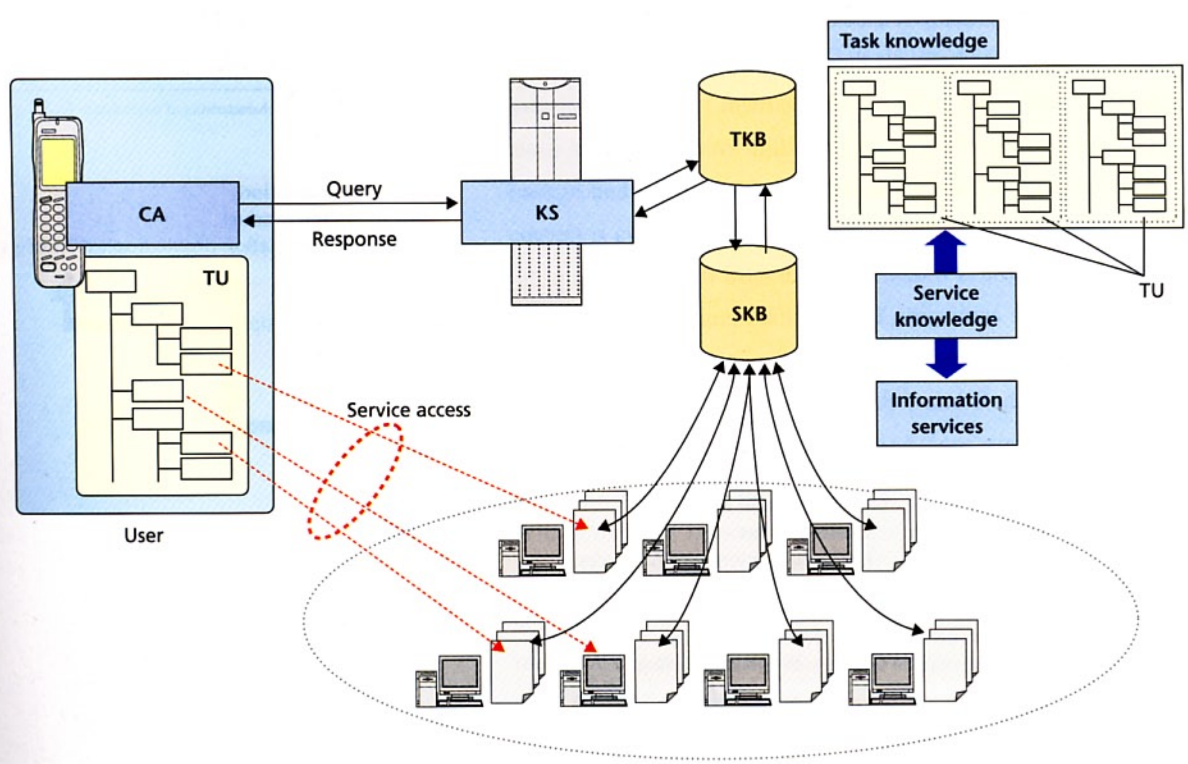


Figure 2 Architecture of proposed system

Identifiers (URIs) of information services; Knowledge Server (KS) which searches for appropriate task-nodes in TKB and SKB and returns a reply in response to the user's query from mobile terminal; and Client Application (CA) that provides a user interface, communicates with KS, and acquires and displays a Task Unit (TU), which is a set of related task nodes in TKB including a root task and its sub-tasks.

In this system, the user issues a TU request to KS via CA. The KS analyzes the request and selects a TU that matches that request by searching TKB. At this time, KS also searches SKB, selects the URIs of all information services associated with the selected TU, and returns them with the TU to CA. The user now searches the TU displayed on the CA and selects tasks to be executed, and then selects an information service from those associated with the tasks so selected.

3.1 Task Knowledge Base

Task knowledge has a hierarchical structure consisting of a root task node corresponding to a real-world problem perceived by the user and sub-tasks resulting from a division of that task node. Functions provided by information services (service functions) correspond to directly solvable task nodes from among those included in this hierarchical structure. It is therefore necessary to analyze information services that are being provided and extract service functions. It is then necessary to analyze specific cases (scenarios) up to the use of information services with the aim of integrating previously extracted service functions and structuring task knowledge.

When setting out to construct a knowledge base, in order to clarify the problem areas to be targeted by the system, and to this end, we have introduced a domain model based on categories of real-world locations (amusement parks, department stores, hotels, etc.). This model considers that one domain consists of the following three areas of activity: 1) Preparations for reaching the target destination; 2) Moving to the target destination; and 3) Behavior at the target destination.

Task knowledge relates to these activity areas within a domain. Task nodes that make up task knowledge can be represented by a generic process consisting of a noun (generic noun) that acts as the object of an action and a verb (generic verb) that indicates an action performed on an object. A problem that occurs when representing task nodes in this way is descriptive variety due to the ambiguity of natural language. To give a simple example, "car" and "automobile" are variations of the same thing that can lead to multiple definitions of synonymous task nodes. To solve this problem, we have introduced a 220,000-word thesaurus providing semantic definitions for allowable vocabulary. Furthermore, when multiple operators are involved in constructing a knowledge base, generic levels for these words must be made uniform, and to this end, we have implemented a vocabulary-integration support tool in this research. This tool has a function for detecting the frequency that words related to input vocabulary (such as hyperonyms, synonyms, and hyponyms) are used in the knowledge base, which makes it possible to absorb any difference in generic levels between operators.

1) Extracting Service Functions

In order to define task nodes that can be directly solved through the use of information services, actual services must be collected and analyzed and service functions extracted. Service functions are generalizations of functions that provide Web content without losing the meaning of those original functions. They are described by a generic process the same as with task nodes. To begin with, we extract Web content for mobile terminals at random from various information sources. In this sampling, we exclude content such as ring tones, screen wallpaper, and games that is not applicable to user-behavior support. We next assign service functions to sampled Web content. For content having multiple functions, we assign multiple service functions. Finally, we merge service functions assigned to each service and integrate semantically equivalent service functions. Applying the above procedure has so far produced definitions for about 2000 service functions with respect to about 2700 items of Web content.

2) Structuring Task Knowledge Based on Cases

Using specific cases of behavior up to the actual use of information services, we extract upper-level task nodes and structure task knowledge by integrating previously defined service functions. The following scenario that includes a user's intent to use a service is typical of the cases that we used.

"I have come to a amusement park with my family. Before entering, we would like to get some lunch since we are hungry. After checking the parade starting time, I would like to search for a restaurant in the vicinity and then check its location using a map service."

Using cases such as this one enables us to extract user problem-solving requests that arise at various real-world locations as well as task knowledge associated with information services. Individual cases, moreover, can be easily collected using no more than the service-usage log, which prevents bottlenecks from occurring when acquiring this knowledge. About 500 cases have so far been collected targeting nine domains such as "amusement park" and "department store." These cases are being applied to the construction of a task knowledge base. **Figure 3** shows an example of a task knowledge base.

3.2 Query Matching

In response to a user's problem-solving request, a task node that can solve that request must be selected from the task knowledge base and presented to the user. "Query matching" refers to the mechanism for comparing a user's problem-solving request with task knowledge. This mechanism takes input from a mobile terminal into account and assumes that the problem-solving request expresses the problem using a short character string. For this reason, task-node selection actually presents not one but a set of task nodes, Task Unit (TU) related to the user's request and lets the user select the task node as an actual target best suited for the purpose at hand using that TU.

Task knowledge is represented as a hierarchical structure of task nodes configured using vocabulary included in the thesaurus. For each word in the user's problem-solving request that is included in the thesaurus, the mechanism creates a word set *W* consisting of that word and its synonyms and compares that set with task nodes. Evaluation value *val(T)* for task node *T*, is determined by the following equation, where *p(T, w)* is a function that returns a constant when task *T* and word *w* successfully match.

$$val(T) = \sum_{w \in W} p(T, w)$$

For each word in the user's problem-solving request that is not included in the thesaurus (such as a proper noun like a company name, person's name, or place name), the mechanism passes that word to a full-text search engine and uses a set of pages obtained from the list of results returned and extracts a feature-word set C. This procedure is summarized below.

- STEP1: Pass the word included in the problem-solving request to a full-text search engine targeting Web content for mobile terminals and treat the first ten documents as document-set *Doc*.
- STEP2: Extract from *Doc* text inserted between anchor tags and obtain noun-set *N* by performing morphological analysis on that text.

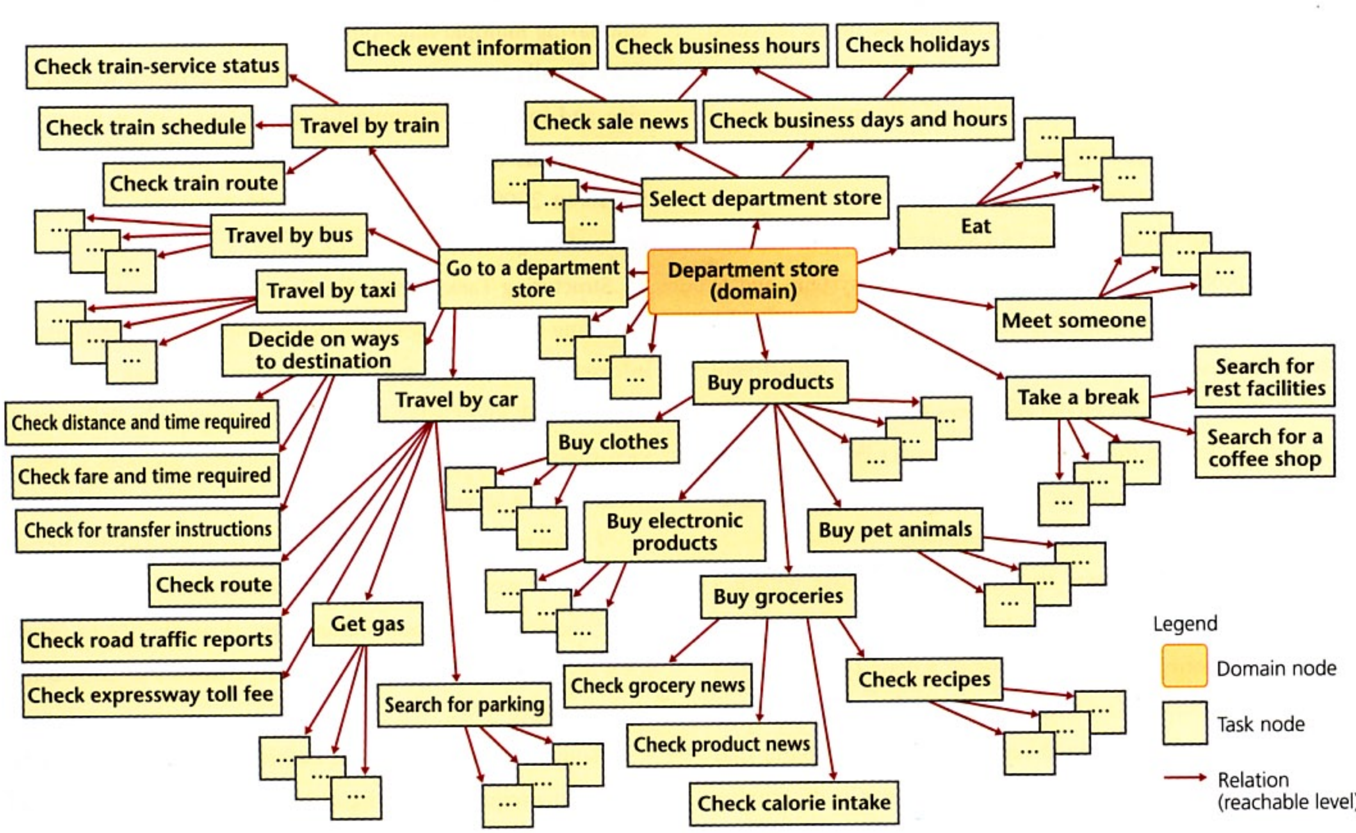


Figure 3 Example of constructing task knowledge

- STEP3: Compare noun-set *N* with the thesaurus and exclude words not included in the thesaurus. Also exclude words such as "date," "information," and "time" that contribute little to identifying a task and exclude their synonyms as well.

Now, for each word *c* included in feature-word set *C*, word-set *CR* consists of c and its synonyms, and KS compares *CR* with TKB. The evaluation value of task *T* *val(T)* can therefore be given by the following equation where *q(T, w)* is a function that returns a constant when task *T* and word *w* successfully match.

$$val\,(T) = \sum_{c \in C} \sum_{w \in CR} q\,(T, w)$$

The TU to apply is found as follows. All the words included in the user's problem-solving request are checked against all task nodes in TKB, and after summing the evaluation values determined as described above for each task node in each TU, the TU with the highest evaluation value for one task is selected.

4. Implementation

Figure 4 shows typical screen shots (user interface) of a prototype version of the proposed service navigation system. The CA on the mobile terminal is implemented using a Java[*3] emulator for mobile phones, and the KS is implemented as a servlet using the Tomcat application server running on Linux OS.

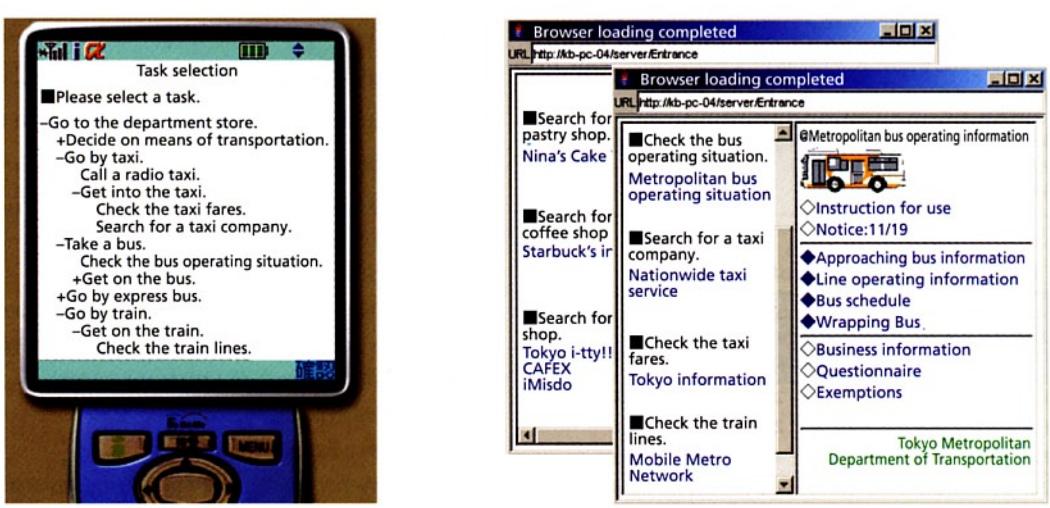
*3 Java is an object-oriented development environment for networks promoted by Sun Microsystems, USA.

In this implementation, the user enters a problem-solving request by text input. The system assumes this input to consist of words like "train" and "department store" and the problem to be expressed in natural sentence such as "I would like to go to a department store" and "I would like to buy a DVD player." On receiving input of such a natural-language sentence, the system divides it into separate words by morphological analysis and creates multiple word sets for processing.

The KS selects from TKB a TU conforming to the problem-solving request and passes it to CA. At this time, KS obtains from SKB the URIs of information services associated with the selected TU and passes them as well to CA. The CA then expands the received TU and displays it in tree form (Fig. 4(a)). The user inspects the displayed TU and selects one or several tasks that suit the current objective, and then selects a specific information service from those associated with the selected task(s). Now that an information service has been decided upon, CA displays that network information service in an interactive viewing format. The current prototype system displays this information service in frame-type HTML format using the emulator's browser function (Fig. 4(b)).

5. Evaluation

We performed a subjective evaluation using ten subjects to examine the effectiveness of the proposed system. This evaluation was limited to the scope of the current knowledge base, i.e., to the currently targeted domains. Subjects were asked to compare the proposed system with keyword-based full-text searching (A) and directory-based searching (B). The idea here was to



(a) (b)

Figure 4 Prototype screen shots

evaluate the process up to finding a service for problem-solving purposes in terms of process functionality and comprehension (ease of understanding). Two evaluation items were therefore used: 1) possibility of reaching an appropriate service; and 2) comprehension up to problem solution.

Subjects were presented with a definition of a problem and procedures for solving that problem using each of the three systems. (The defined problem could be solved using any of these systems.) Specifically, the procedure for entering keywords and searching through a list of search results was presented with respect to a keyword-based full-text search engine and that for searching through a directory was presented with respect to a directory-based search menu. Problem-solving procedures were presented beforehand to minimize the effects that user experience with any of these systems might have on evaluation results. On solving the defined problem according to the problem-solving procedures so presented, the subject was asked to score each system in the range from 1 (easy) to 5 (difficult).

Table 1 Possibility of reaching an appropriate service

	Task	Keyword-based Search A Average Score	Directory-based Search B Average Score	Proposed System Average Score
Amusement park domain	1.a	2.1	2.8	2.5
	1.b	1.5	2.3	2.6
	1.c	1.0	4.6	1.2
Department store domain	2.a	3.0	4.6	2.2
	2.b	1.9	3.0	1.6
	2.c	4.3	2.1	1.5
	Avg.	2.3	3.2	1.9

Legend (Tables 1 and 2)

Amusement park domain:
- 1.a Decide on destination, check attractions, confirm airplane reservation, reserve hotel room
- 1.b Check traffic reports, check expressway route
- 1.c Check postal code

Department store domain:
- 2.a Decide on destination, check for sales, check train route, check weather
- 2.b Check route by automobile, check toll fees
- 2.c Search for nearby shops

Table 2 Comprehension up to problem solution

	Task	Keyword-based Search A Average Score	Directory-based Search B Average Score	Proposed System Average Score
Amusement park domain	1.a	2.2	2.7	2.0
	1.b	1.8	2.9	3.6
	1.c	1.1	4.8	1.2
Department store domain	2.a	2.8	4.6	2.4
	2.b	2.3	3.0	1.4
	2.c	3.8	2.2	1.3
	Avg.	2.3	3.4	2.0

Three problems were prepared for each of the domains of "amusement park" and "department store" for a total of six problems. **Tables 1** and **2** show evaluation results with a legend defining these problems.

Examining these results, we see that the proposed system had the best scores for both evaluation items indicating that it is indeed effective. Furthermore, on comparing results for the proposed system between the two domains targeted by this evaluation, we see that the scores for the "department store" domain were somewhat better than those for the "amusement park" domain. This suggests that the proposed system can function effectively for domains like "department store" that have few sites (portals) that handle related information in a comprehensive manner.

6. Conclusion

In this article, we proposed a service navigation system based on task knowledge, and described a task-knowledge-base construction method and a query-matching system. We also described the implementation and evaluation of a prototype system and confirmed the effectiveness of the proposed system for a limited number of domains.

The proposed system organizes information services from the viewpoint of users and aims to enable even users unfamiliar with the use of information services to access and utilize needed services. This system features a scheme for dividing an upper-level task into subtasks and displaying that task and its subtasks. The user is therefore presented with means of solving problems that the user was not initially aware of, resulting in a "discovery" effect.

As a future topic of research, we aim to establish a system for efficiently expanding the knowledge base. We also plan to study a system for optimizing task knowledge by linking with various types of context information such as location information and sensor information from peripheral devices.

References

[1] Telecommunications Carriers Association: http://www.tca.or.jp/

[2] M. B. Eisenberg and R.E. Berkowitz: "Information Problem Solving: The Big Six Skills Approach to Library and Information Skills Instruction," Ablex Publishing Corp, Norwood, NJ, 1990.

[3] http://www.w3.org/2001/sw/

[4] Atanas Kiryakov, Borislav Popov, Damyan Ognyanoff, Dimitar Manov, Angel Kirilov and Miroslav Goranov: "Semantic Annotation, Indexing, and Retrieval," Proc. 2nd International Semantic Web Conference, LNCS 2870, pp.484–499, Sanibel Island, FL, USA, Oct. 2003.

[5] Hugo Liu, Henry Lieberman and Ted Selker: "GOOSE: A Goal-Oriented Search Engine With Commonsense," Proc. Adaptive Hypermedia and Adaptive Web-Based Systems 2nd International Conference, LNCS 2347, pp.253–263, Malaga, Spain, May 2002.

[6] A. Bandula: "Self-regulation of motivation and action through internal standards and goal systems," Goal Concepts in Personality and Social Psychology, A.P. Lawrence, PP.19–85, Lawrence Erlbaum Assoc, 1989.

ABBREVIATIONS

CA: Client Application
IP: Internet Protocol
KS: Knowledge Server
RFID: Radio Frequency IDentification
SKB: Service Knowledge Base
TKB: Task Knowledge Base
TU: Task Unit
URI: Uniform Resource Identifier
WLAN: Wireless Local Area Network