Task Knowledge Based Retrieval for Service Relevant to Mobile User's Activity

Takefumi Naganuma and Shoji Kurakake

Network Laboratories, NTT DoCoMo Inc. 3-5 Hikari-no-oka,
Yokosuka-shi, Kanagawa, 239-8536 Japan
{naganuma, kurakake}@netlab.nttdocomo.co.jp

Abstract. Since mobile Internet services are rapidly proliferating, finding the most appropriate service or services from among the many offered requires profound knowledge about the services which is becoming virtually impossible for ordinary mobile users. We propose a system that assists non-expert mobile users in finding the appropriate services that solve the real-world problems encountered by the user. Key components are a task knowledge base of tasks that a mobile user performs in daily life and a service knowledge base of services that can be used to accomplish user tasks. We present the architecture of the proposed system including a knowledge modeling framework, and a detailed description of a prototype system. We also show preliminary user test results; they indicate that the system allows a user to find appropriate services quicker with fewer loads than conventional commercial methods.

1 Introduction

NTT DoCoMo is the premier mobile communications company in Japan and is providing mobile Internet services to over 44 million subscribers. Currently, widely diverse contents such as entertainment services (Ring-tone download, Games, etc), transaction services (Money transfer, Airline reservation, etc) and information services (Weather forecast, Maps and local information, etc) are being offered through more than 89,000 mobile Internet sites [1]. The market volume of mobile Internet content exceeded one hundred thousand million yen in 2003, which is more than the market volume of Internet content accessed through PCs, 760 hundred million yen. Moreover, 36.2% of people accessing the Internet via mobile phones used for-fee Internet contents, a larger proportion than personal computer users for which it is only 9.5% [2]. 3rd Generation mobile network service (3G), which continues to spread, enables the transmission of more voluminous and richer content by extension of the communication bands, so the market is expected to increase greatly. However, those rich network environments will increase the service provider's cost for developing and maintaining content. Furthermore, the market has, up to now, been dominated by a few popular entertainment services such as character download and ring-tone download, but these markets have already become saturated. If we are to expand overall market volume, it is necessary to expand the market volume of non entertainment services by increasing the frequency with which users can benefit from the rich contents possible in the 3G mobile network.

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 959 – 973, 2005.

We believe that the key to the next step in mobile Internet market growth is realizing the intelligent service platform, which will mediate between services and users by interpreting the user's activity in the real world. Because mobile handsets are the most appropriate devices for problem solving in daily life as users always carry them, there will be many more occasions for providing services compared to Internet services accessed by the PC. By realizing the intelligent service platform, users will be able to access the most appropriate services to help their activities in the real world anytime, anywhere, and service providers will gain more chances for providing new services.

Current methods for accessing Internet services, such as directory-type search methods and keyword-type full-text search methods, do not suit mobile computing environments. This is because mobile handsets have strong resource limitations such as relatively weak input methods and small displays, and most users are non-experts in that they have no computer literacy, i.e. the young generation or the aged. Furthermore, most mobile Internet services are accessed for just a few minutes in daily life such as when waiting for a train. Novice users obviously cannot perform service retrieval efficiently because they do not understand the overall directory structure and can not come up with the keywords that will exactly identify the service needed.

Many research activities are targeting the realization of efficient service retrieval in the mobile Internet environment such as a service retrieval system based on user's location [3]. However, this system simply provides a mechanism for associating Internet content with the user's location according to the address information written in the target document, its does not take into account the meaning of the user's activity. The contents supported by the above approach are very limited, and content that has no address information can not be retrieved even if it is very useful to the user.

We treat user activity in the real world as the semantics of services by using Semantic Web technology. To realize this, we propose a knowledge modeling framework that makes it easy to describe various user tasks in the real world as the semantics of services. We define "task" as what the user wants to do in the real world. And we also propose the task knowledge based service retrieval system based on the task knowledge base developed by using the knowledge modeling framework. The task knowledge base contains a variety of structured tasks in the real world and their relations with the information services available. Our system makes it easy for non-expert users of mobile services to retrieve the service appropriate to solve the user's problem by just inputting her currently desired task as a problem-solving request to the system.

We will present our approach and the high-level architecture of the proposed system. We then discuss the knowledge modeling framework including our strategy for realizing network-wide knowledge sharing, semantic description of task knowledge with OWL-S [10], and a task knowledge modeling environment and service retrieval system. Details are provided of the design of the system sequence, server component and client interface, all of which are intended to be implemented on actual mobile handsets. We also describe a prototype system for each part. The paper finishes with an evaluation by user testing and our conclusions.

2 Overview of Proposed System

A general approach to problem-solving is to divide the large or abstract original problem into several small or concrete sub-problems. Paper [8] shows that human daily life is driven by "proximal goals" (short-term goals), which are derived from "distal goals" (long-term goals). In the area of problem-solving by using information services, a problem that corresponds to a "distal goal" is divided into sub-problems that correspond to "proximal goals" where each of the sub-problems can be solved directly by using one information service. Our idea is to divide the user's "distal goal" into a couple of "proximal goals" and lead the user to appropriate services that can solve the "proximal goals".

To elucidate the above discussion we provide an example. Consider the real world problem: "I want to spend this weekend at amusement park XYZ, but I do not know how to get there". The information-seeking actions associated with this problem are represented by the following structure.

Decide how to get to amusement park XYZ
→ Check route
→ Check distance and time required
→ Check for transfer instructions

In the above example, the problem of "Decide how to get to amusement park XYZ" corresponds to the "distal goal", and the sub-problems to realize it, "Check route", " Check distance and time required" and "Check for transfer instructions", correspond to "proximal goals". The user is assisted in solving the original problem by invoking a sequence of services, each of which handles a corresponding sub-problem.

To realize the above approach, we treat "what a user wants to do" as the user's task and structure knowledge that can be used to divide the task into sub-tasks as task knowledge. We have built a knowledge base that stores such structured task knowledge. Furthermore, we have developed knowledge of service usage (for example, "This service provides functions for reserving movie tickets") as service knowledge, and have built a knowledge base that stores service knowledge. We developed a task oriented service retrieval system that uses these two knowledge bases. Each task is represented as an individual node that we call task node, and task and sub-task structures in task knowledge are represented as a graph structure among task nodes yielding parent tasks and their sub-tasks; service knowledge is expressed by associating each task node with the URI (Uniform Resource Identifier) of the appropriate concrete information service.

2.1 High-Level Architecture

Fig. 1 shows the high-level architecture of the proposed system. The system is composed of two parts; the Knowledge Modeling Framework (KMF), which provides a task knowledge modeling environment to support task knowledge modeling and stores the described model in the Task Knowledge Base (Task KB), and the Service Retrieval Framework (SRF), which provides a task knowledge based service retrieval system for actual mobile handsets through the use of the Task KB and the Service Knowledge Base (Service KB). KMF provides an environment that makes it easy to describe and acquire task knowledge and that also supports interoperability within task knowledge because it is to be shared by many network service providers and content service providers. SRF provides the server module needed to retrieve appropriate knowledge from the Task KB and the Service KB according to the user's request, and the client module, which actually runs on a mobile handset to provide an easy-to-use user interface.

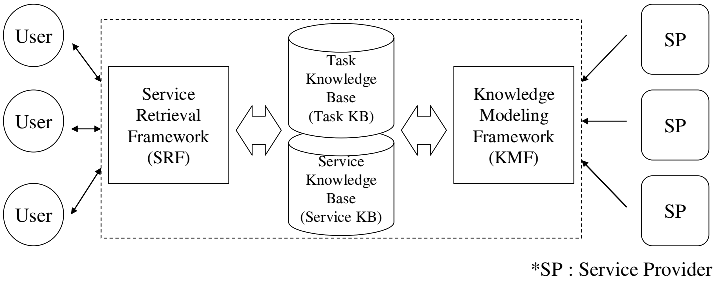

User
User
User
Service Retrieval Framework (SRF)
Task Knowledge Base (Task KB)
Service Knowledge Base (Service KB)
Knowledge Modeling Framework (KMF)
SP
SP
SP
*SP : Service Provider

Fig. 1. High-level architecture of proposed system. The system is composed of two parts; 1)KMF, which provides a task knowledge modeling environment, and 2)SRF, which provides a task knowledge based service retrieval system that can be used from a mobile handset.

2.2 Research Challenges

1. The *Knowledge Modeling Framework (KMF)* must be capable of describing all tasks likely to be performed by mobile users in a structured way as task knowledge. The framework must make it easy to describe knowledge and to extend existing knowledge. We note that task knowledge will be created by multiple entities.
2. The *Service Retrieval Framework (SRF)* must enable a mobile user to retrieve appropriate services by entering a simple problem-solving request. The environment must provide components that run on commercial mobile handsets and provide a user interface that is easy to use and to understand. The environment also must handle a wide range of user requests such as abstract or concrete requests.

3 Designing Knowledge Modeling Framework

The Knowledge Modeling Framework (KMF) is designed to provide the ability to describe many kinds of task knowledge, and to extend existing knowledge or add external knowledge from a third party. We show details of this framework below.

3.1 Basic Approach for Task Knowledge Modeling in the Real World

It’s important to manage and reuse already described knowledge because there are so many tasks in the real world. Much research has been conducted on the ontology for knowledge sharing in the engineering domain [12] [13], business domain [14] or problem-solving research area [15] [16]. The final goal of our research is to create large scale task ontology for modeling real world user activities. Towards this goal, first of all, we considered the strategy for extracting task knowledge that can be reused. Our approach is to extract task knowledge that depends on some specific place such as an amusement park or a department store at first, and next we extract the generic task knowledge from the place-dependent task knowledge. The advantage of this approach is that the extraction of the place-dependent task knowledge is easer to carry out by collecting and analyzing cases of service usage in the real world place. The category of the real-world place can be borrowed from commercial services such as a map service or car navigation service. In this research, we treat this category of place as a domain for classifying the task knowledge. Fig. 2 shows basic approach for task knowledge modeling in the real world.

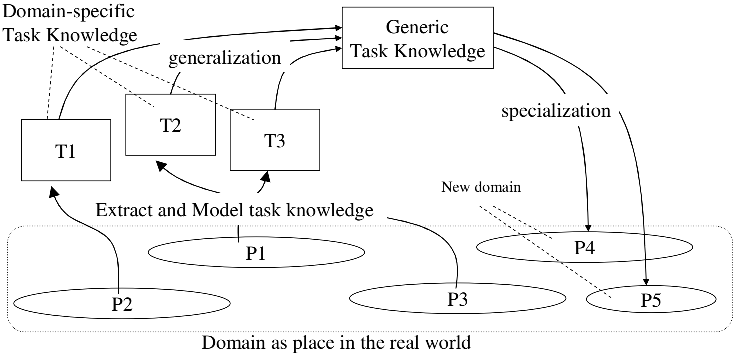


Fig. 2. Basic approach for task knowledge modeling in the real world

The generic task knowledge facilitates the reuse of knowledge. For instance, generic task knowledge such as "go to somewhere", which corresponds to the general process model to perform the activity of moving from a starting point to a destination, is common knowledge among specific task knowledge regarding going to specific places from specific places. The knowledge engineer can use such generic task knowledge to describe more specific task knowledge in a new domain which decreases the cost for expanding the coverage of task knowledge. We have constructed domain specific task knowledge for 12 domains so far.

3.2 Requirements for Semantic Description of Task Knowledge

Task knowledge consists of various kinds of task nodes and their relations. A good description of task knowledge is quite important for the management and reuse of knowledge. In addition, an interoperable standard format for knowledge representation is essential for network-wide knowledge sharing. By sharing the same description, we can combine different task knowledge such as knowledge created by a service provider and independent user groups of the service. We extracted the task knowledge description requirements which were based from the analysis of structured task knowledge developed in [4]. Fig. 3 shows an example of a generic task model and a specific task model and their relation based on the requirements.

1. The relations between task nodes must allow the discrimination of two relation types: “specialized” relation, which indicates that an *object* task node is more specific than a *subject* task node, and “achieved” relation, which indicates that an *object* task node is one of the divided task nodes that achieve a *subject* task node.
2. The achieved relation must be represented as a sequence of tasks by using some kind of control construct, such as Sequence, Exclusiveness, etc.
3. The achieved relation involves the role of each sub-task to achieve the task with which the sub-task is associated.
4. The specialized relation involves the viewpoint of the policy for categorization. For instance, “Buy products” and “Buy foods” are associated by a specialized relation from the viewpoint of identification of the object being bought.
5. The task nodes have to be discriminated based on context information such as location and/or time. For instance, when the user’s context is "midnight", the task of “travel by train” should not be associated with the user if there are no train services in midnight.
6. Task knowledge has to be associated with the widest possible variety of services, such as XML Web services, HTML-based Web content, etc.

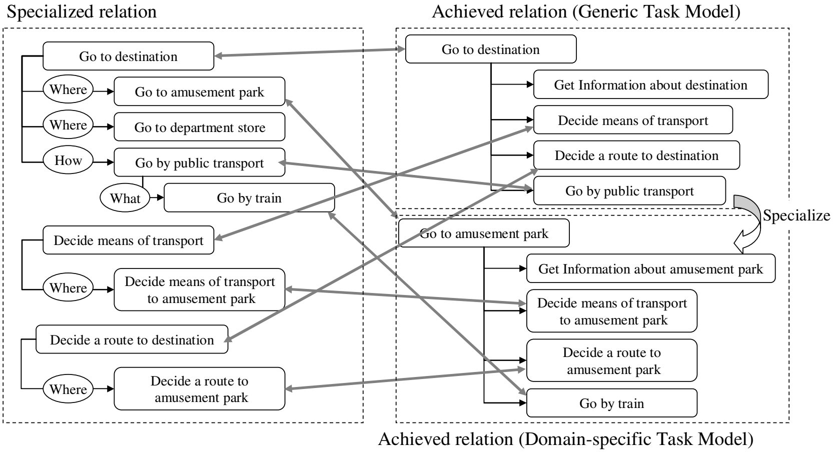

Specialized relation
Go to destination
Where
Go to amusement park
Where
Go to department store
How
Go by public transport
What
Go by train
Decide means of transport
Where
Decide means of transport to amusement park
Decide a route to destination
Where
Decide a route to amusement park
Achieved relation (Generic Task Model)
Go to destination
Get Information about destination
Decide means of transport
Decide a route to destination
Go by public transport
Specialize
Go to amusement park
Get Information about amusement park
Decide means of transport to amusement park
Decide a route to amusement park
Go by train
Achieved relation (Domain-specific Task Model)

Fig. 3. An Example of structured task knowledge described with achieved relation and specialized relation. All task nodes are defined in the specialized relation tree (left side) and some task nodes are associated with achieved relation (right side).

3.3 Semantic Description of Task Knowledge with OWL-S

For meeting the requirements in 3.2, we designed a description framework of task knowledge using OWL-S. OWL-S is an OWL [9] based Web service ontology for describing the properties and capabilities of Web services. OWL-S also includes process ontology for describing generic process.

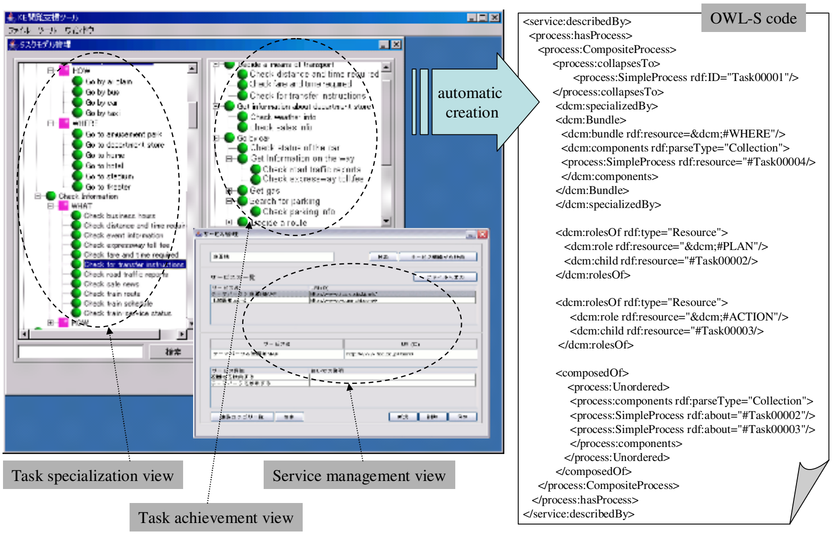
We describe the achieved relation by using the Process model and the control constructs defined in OWL-S, and define the specialized relation as an object property called “specializedBy”. In addition, we define a “View” class for describing the viewpoint of specialization. The value of the "view" property is either 1) WHAT; specialization of object, 2) WHERE; specialization of location, or 3) HOW; specialization of method.

Furthermore, we define a "Role" class for describing the role of a lower-level task with respect to an upper-level task in achieved relations. The "role" property can be either, 1) PLAN; the activity of planning to achieve upper-level task (e.g. decide a destination), 2) PREPARE; the activity of preparation before performing concrete action (e.g. collect information of destination in advance), 3) ACTION; the concrete activity to achieve upper-level task (e.g. move to destination), or 4) CONFIRM; the activity of checking the current status of situation (e.g. check traffic jam). Each task node is described as a Service class of OWL-S, and the context information that indicates the applicable condition of the task node is described by using the ServiceProfile.

For describing service knowledge that associates task node with not only Web services but also HTML-based Web content, we define an original class named "BookmarkAtomicProcessGrounding" (BAPG). BAPG has features that can associate multiple Web content with a single task node, and can handle service names and URIs.

3.4 Task Knowledge Modeling Environment

We developed a task knowledge modeling environment, named TEdit, to develop and manage the described task model. Fig. 4 shows the user interface of TEdit. TEdit is implemented in Java and has an SQL Database interface via JDBC. TEdit provides several functions, 1) Create task node and edit specialized relations, 2) Define and edit achieved relations, 3) Define control constructs in achieved relations, 4) Register services and edit association of services with task node, and so on. Furthermore, TEdit provides the feature of automatic creation of OWL-S codes.



```
<service:describedBy>
  <process:hasProcess>
    <process:CompositeProcess>
      <process:collapsesTo>
          <process:SimpleProcess rdf:ID="Task00001"/>
      </process:collapsesTo>
      <dcm:specializedBy>
      <dcm:Bundle>
        <dcm:bundle rdf:resource=&dcm;#WHERE"/>
        <dcm:components rdf:parseType="Collection">
        <process:SimpleProcess rdf:resource="#Task00004/>
        </dcm:components>
      </dcm:Bundle>
      </dcm:specializedBy>

      <dcm:rolesOf rdf:type="Resource">
        <dcm:role rdf:resource="&dcm;#PLAN"/>
        <dcm:child rdf:resource="#Task00002/>
      </dcm:rolesOf>

      <dcm:rolesOf rdf:type="Resource">
          <dcm:role rdf:resource="&dcm;#ACTION"/>
          <dcm:child rdf:resource="#Task00003/>
       </dcm:rolesOf>

      <composedOf>
         <process:Unordered>
          <process:components rdf:parseType="Collection">
          <process:SimpleProcess rdf:about="#Task00002"/>
          <process:SimpleProcess rdf:about="#Task00003"/>
          </process:components>
        </process:Unordered>
      </composedOf>
    </process:CompositeProcess>
  </process:hasProcess>
</service:describedBy>
```

Fig. 4. User Interface of task knowledge modeling environment named TEdit

The left side of the user interface provides a specialization view in which the user can define each task node hierarchically. The viewpoint of specialization can be set by selecting the appropriate view such as WHAT, WHERE and HOW on a context menu shown by clicking on a task node. The selected viewpoint is shown as a special node. The right-upper side of the user interface provides an achievement view in which the user can define the achieved relations between task nodes. The role of achieved relations can also be assigned by selecting appropriate sub-task roles such as PLAN, PREPARE, ACTION, and CONFIRM. The control construct of the set of sub tasks such as Exclusive, Sequence, or IF-then-else, and the context information which indicates applicable condition of the task can also be set in the achievement view.

4 Designing Service Retrieval Framework

We designed and implemented the service retrieval framework based on task knowledge that consists of server module and client module. The client module can run on commercial mobile handsets as a Java program specially designed for mobile handsets. We show the system architecture and the user interface in detail below.

4.1 System Architecture and Execution Sequence

Fig. 5 shows the system architecture of Service Retrieval Framework (SRF) based on task knowledge. The system is composed of several parts: Task Knowledge Base (Task KB) stores description for task nodes and the relationships between task nodes as Semantic Task Descriptions; Service Knowledge Base (Service KB) stores descriptions about the associations between a service (URI, service name and service explanation) with a proper task node in Task KB as Semantic Service Descriptions; Task Selector (TSE) locates the most appropriate task nodes in TKB according to a user request and context information; Service Selector (SSE) locates appropriate services in Service KB associated with task node IDs; Task Navigator (TNA) provides the user interface to let a user input a request and communicates with TSE to obtain the task nodes matched with the user's request; Web browser, which displays HTML-based Web content and Context Notifier (CNO) notifies TSE of the context information, such as user's current location and time.

A user sends a problem-solving request to TSE via the user interface presented by TNA. At the same time, CNO sends the user's context information to TSE. TSE analyzes the user's request and the context information, selects tasks that match the user's request and context information by searching the Task KB, and sends the results to TNA. The user selects an appropriate task from a list of the tasks displayed on TNA. This task may be very high level and abstract. TNA sends the task ID of the user selected task to TSE. TSE sends back all sub-tasks related via the achieved relation to the task. Here, some of the sub-tasks may be very detailed and concrete enough to be directly associated with concrete services. The user can brows all subtasks on TNA and selects one or more task nodes to execute and sends the IDs of the task nodes to SSE. SSE searches Service KB and retrieves the service associated with the task nodes that the user selected, and creates a summary HTML page listing the services deemed to be appropriate. The user uses the summary page displayed by the web browser to access the services.

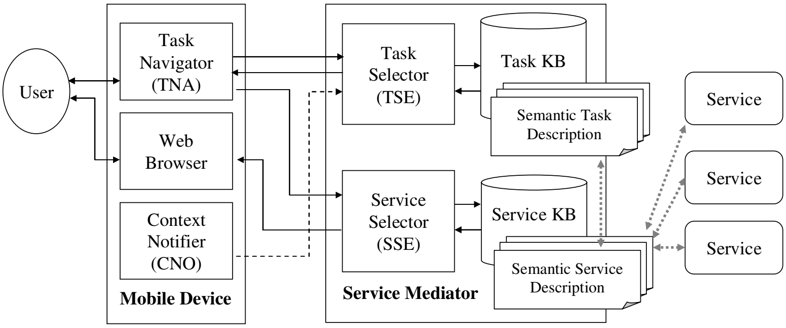
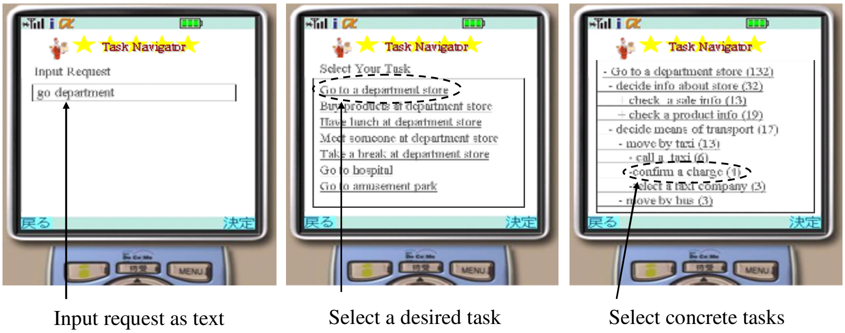


Fig. 5. The system architecture of Service Retrieval Framework (SRF). SRF consists of two parts; 1) Service Mediator, which provides the server module running as Servlet, and 2) Mobile Device, which provides the client module running on the mobile handset.

4.2 Implementation

Fig. 6 shows the user interface of the prototype system. TNA, which runs on a mobile device, was implemented as a Java application running on a mobile device. TSE and SSE were implemented as Servlet running on an application server. The current implementation does not support context-related features such as a Context Notifier.


Input Request
go department

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

- 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)

Fig. 6. User interface for the execution of task knowledge based service retrieval. Service selection procedure consists of 3 steps; 1) request input, 2) abstract task selection, 3) concrete task selection.

The user enters a problem-solving request in a text box on TNA (left side of Fig. 6). To respond to the user's problem-solving request, a task node that can solve that request must be selected from the Task KB and presented to the user. This mechanism considers that the input is from a small portable terminal and assumes that the problem-solving request is expressed as a short text message. For this reason, task node selection actually presents not one but a set of task nodes which are associated with achieved relations. For each word in the user's problem-solving request that is in the thesaurus, the mechanism creates word set *W* consisting of that word and its synonyms and compares that set to the task nodes. For task node *T*, evaluation value *val (T)* is determined by the following equation, where *p (T, w)* is a function that returns a constant value when task *T* and word *w* successfully match.

$$val(T) \ = \ \sum_{w \in W} p \ (T, w) \tag{1}$$

After all the words in the user's problem-solving request are checked against all task nodes in the Task KB, the sets of task nodes are listed in order of the average score of the evaluation value on TNA (center of Fig. 6). The user selects one desired task from the list and then selects more detailed and concrete tasks from the tree of the task nodes related via the achieved relations (right side of Fig. 6). The number of services associated with each task node is displayed on the right side of the task nodes.

5 Evaluations

We conducted a user test with 9 adult subjects to confirm the effectiveness of the proposed system. The purpose of this user test was to evaluate the process up to finding services for problem-solving purposes in terms of process functionality. Subjects were asked to retrieve appropriate services to given problem by using the proposed system, a keyword-type full-text search system developed by ourselves, and a major commercial directory-type search system [11].

This test was designed based on ISO/IEC 9126 Part4: Quality In Use Metrics. The evaluation items were 1) Effectiveness: the percentage of users who could reach the services appropriate to the given problem, 2) Productivity: the time taken to reach the services appropriate to the given problem.

5.1 Test Set

We designed a test set consisting of 4 different problem based on goal type as follows.

1. Service retrieval from a designated Web site: “You are at Tokyo station: Find a site that shows the location of a Karaoke shop near here. ”
2. Service retrieval for obtaining designated multiple information: “You are at Tokyo station: Find all of the following information, (a) a title of movie that is now being shown, (b)movie ranking of the title, (c) the location of a theater that is showing the movie near here, (d) the starting time of the movie. ”
3. Service retrieval for proper information to perform designated activity: “You are at Tokyo station and have a lot of luggage. Find a proper way to send it to your friend.”
4. Service retrieval for proper information to help user to plan and perform activity in the real world to achieve a designated goal: “You are at Tokyo station at midnight. Find a good way to spend the time until tomorrow morning.”

5.2 Test Environment

The test environment consisted of 3 different search systems, 1) S1: keyword-type full-text search system, 2) S2: directory-type search system, 3) S3: task knowledge based service retrieval system (proposed system). The target page set to be searched was the same for each system and the number of the page set was about 15,000. Subjects were divided into 3 groups according to their experience of mobile Internet service. 1) U1: subjects using mobile Internet service everyday, 2) U2: subjects using mobile Internet service a few days a week, 3) U3: subjects who have no experience in using mobile Internet service.

5.3 Test Results

Figures 7, 8, 9, and 10 show a test result. The horizontal axis shows the time taken (in seconds) to solve a given problem while the vertical axis shows the number of subjects who solved a given problem at the time. The right end of the horizontal axis, labeled 900, shows the number of subjects who could not solve the given problem.

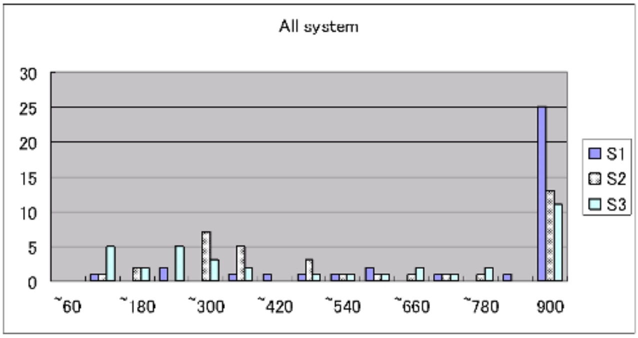


Fig. 7. Test results of all systems (*S1*: Keyword-type search system, *S2*: Directory-type search system, *S3:* Proposed system). The horizontal axis shows the time taken to solve a given problem by the second and the vertical axis shows the number of subjects who solved the problem.

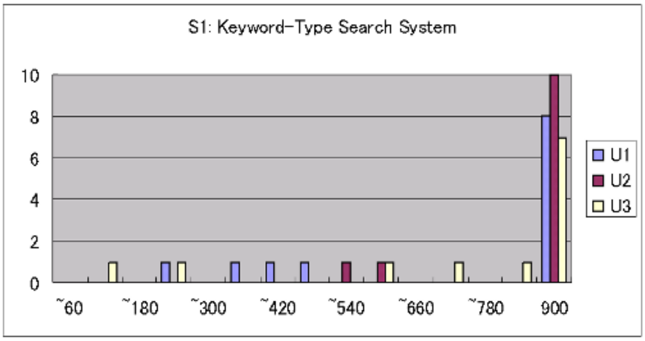


Fig. 8. Test results of Keyword-type search system (*S1*) for each user type (*U1* use mobile Internet everyday, *U2* use mobile Internet a few days a week, and *U3* has no experience in using mobile Internet). The horizontal axis shows the time taken (in seconds) to solve a given problem while the vertical axis shows the number of subjects who solved a given problem.

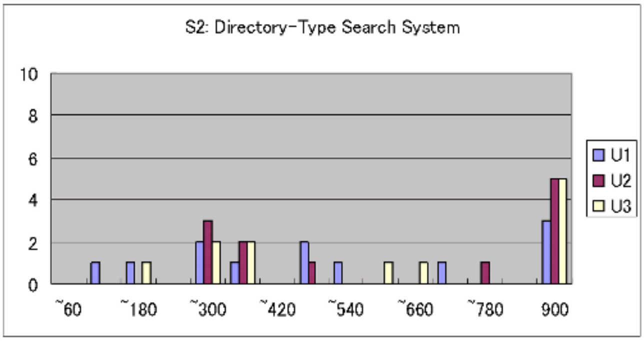


Fig. 9. Test results of Directory-type search system (*S2*) for each user type (*U1* use mobile Internet everyday, *U2* use mobile Internet a few days a week, and *U3* has no experience in using mobile Internet). The horizontal axis shows the time taken to solve a given problem by the second and the vertical axis shows the number of subjects who solved a given problem.

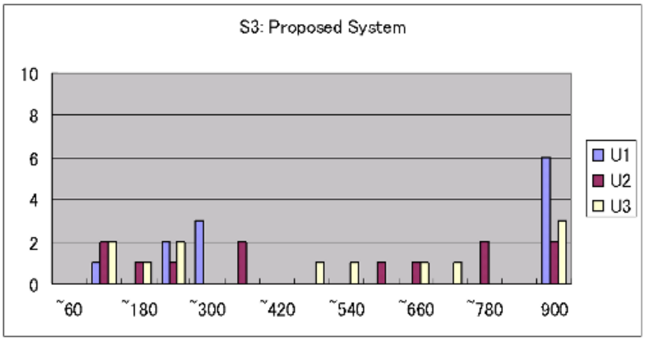


Fig. 10. Test results of proposed system (*S3*) for each user type (*U1* use mobile Internet service everyday, *U2* use mobile Internet service a few days a week, and *U3* has no experience in using mobile Internet service). The horizontal axis shows the time taken (in seconds) to solve a given problem while the vertical axis shows the number of subjects who solved a given problem.

The test results show that the proposed system offers superior performance. The rate of users who could reach the appropriate services was about 63%, which was higher than the result of the keyword-type search system (16%) and the result of the directory-type search system (56%). In particular, 50% subjects reached the appropriate services within 300 seconds which is an acceptable value in actual mobile Internet use, which was higher than directory-type search system (33%) and the keyword-type search system (10%). In this test, the number of subjects was not large enough and the test set was not comprehensive, but the test results actually show the effectiveness of our approach in an actual mobile Internet environment.

With regard to the test results analyzed by user type, only the proposed system allowed non-expert users grouped as U2 and U1 to achieve the same level of success as experienced users, U1. One observed drawback of the proposed system was that performance of U1 was not enhanced much by our system. We imagine the reason is that users in U1 were accustomed to conventional search systems and did not have enough time to get accustomed to the different search strategy of the proposed system. To resolve the above problem, we will consider the idea of changing the search system automatically according to the user's response.

6 Related Work

Task Computing Environment (TCE) [6] is a DAML-S [5] based service discovery and composition framework for pervasive computing environments. TCE can reduce the cost of performing a task by displaying the tasks that could be performed by executing pervasive services located around the user. However, TCE can not associate a user's task with services on the Internet that are not physically adjacent to the user, and does not support abstract tasks that are not solved directly by using those services, because the support range of TCE is limited to just those tasks that are solved directly by pervasive services.

The process-based service retrieval approach [17] is a sophisticated service retrieval method that offers better precision and recall than the conventional service retrieval method. In this research, service models are indexed into a process ontology, such as MIT Process Handbook [14], and the process query language called PQL is introduced. A preliminary evaluation showed that the process-based service retrieval approach offers qualitatively higher retrieval precision than any other existing approach.

Other research has considered the use of a knowledge base to infer what the user really wants to do. The GOOSE system [7], for example, employs a large amount of knowledge (common sense knowledge) collected from general Internet users to infer the user's search goal and to present search results that achieve that goal. This system, though, requires relatively long text input because it applies natural language processing technology to infer the search goal. It would not be easy to apply such a system to an environment where text input is cumbersome as is the case with mobile terminals.

7 Conclusion and Future Work

This paper proposed a task knowledge based service retrieval framework for non-expert mobile users that makes it easy to retrieve services appropriate for tackling the user's problem in the real world. The system features a task knowledge base that contains knowledge about which services will solve the problems that a user faces in daily life. Details of the prototype system including knowledge modeling framework were described. While the prototype system has only limited coverage, the results of a user test confirmed that it lowers the difficulty of service access. In addition, the system allows the user to recognize ancillary tasks that were not initially thought of by the user since the system shows tasks related to the task of solving the problem directly extracted from the user's initial request.

The next step in our research is to consider and develop a task-ontology-based knowledge modelling environment. Task ontology provides a common vocabulary and common viewpoint for describing task knowledge, and enables knowledge authors to create reusable knowledge models. Furthermore, we will extend the scope of the knowledge to be able to describe relationships between obstructive events, such as “missing route” or “no vacant spaces”, and the task for preventing and resolving those obstructive events.

We also plan on improving the user interface on the mobile handset. The current prototype system provides just one text box interface to input the user’s request. This interface is simple but some users are not certain about what kinds of request can be interpreted. Actually, some users input long sentences into the text box such as “find restaurant now available” or “find hotel near Tokyo station” in our test, but the current prototype system can not interpret conditions that contain time or location information. We will consider more functionality to help the user input his request with minimal load.

Acknowledgement

We would like to acknowledge Prof. Riichiro Mizoguchi, Prof. Yoshinobu Kitamura, and Dr. Munehiko Sasajima at Osaka University for their useful discussions and comments on this work.

References

1. NTT DoCoMo web site.: http://www.nttdocomo.com/
2. Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan.: Information and communications in Japan, Chapter2, Section5. (2004)
3. Hiramatsu, K., Akahani, J., Satoh, T.: Querying Real World Services Through the Semantic Web. In: Proceedings of The Third International Semantic Web Conference (ISWC 2004) (2004) 741-751
4. Naganuma, T., Kurakake, S.: A task oriented approach to service retrieval in mobile computing environment. In: Proceedings of IASTED International Conference on Artificial Intelligence and Applications (AIA 2005) (2005)
5. The DAML Services Coalition (Anupriya Ankolenkar, et al).: DAML-S: Web Service Description for the Semantic Web. In: Proceedings of The First International Semantic Web Conf. (ISWC), Sardinia, Italy (2002) 348-363
6. Masuoka, R., Parsia, B., Labrou, Y.: Task Computing - the Semantic Web meets Pervasive Computing -. In: Proceedings of The Third International Semantic Web Conference (ISWC 2003) (2003) 865-881.
7. Hugo Liu, Henry Lieberman, and Ted Selker.: GOOSE: A Goal-Oriented Search Engine With Commonsense. In: Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems Second International Conf., Malaga, Spain (2002) 253-263.
8. Bandula, A.: Self–regulation of motivation and action through internal standards and goal systems, Goal Concepts in Personality and Social Psychology (Hillsdale, NJ: A.P.Lawrence). (1989) 19-85.
9. Web Ontology Language (OWL).: http://www.w3.org/2004/OWL/
10. OWL-S 1.0 Release.: http://www.daml.org/services/owl-s/1.0/
11. Yahoo Mobile web site.: http://mobile.yahoo.co.jp/index.html
12. Kitamura, Y., Kashiwase, M., Fuse, M. Mizoguchi, R.: Deployment of an ontological framework of functional design knowledge. Advanced Engineering Informatics, 18(2), (2004) 115-127

13. Horváth I, Vergeest JSM, Kuczogi G. Development and Application of Design Concept Ontologies for Contextual Conceptualization. In: Proceedings of 1998 ASME Design Engineering Technical Conferences DETC (1998)
14. Herman, G. A., Malone, T. W.: What is in the process handbook?, Organizing Business Knowledge: The MIT Process Handbook, MIT Press (2003) 221-258
15. Mizoguchi, R., Ikeda, M., Seta, K., and Vanwelkenhuysen, J.: Ontology for Modeling the World from Problem Solving Perspectives, in IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
16. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B.: Knowledge Engineering and Management - The Common-KADS Methodology, The MIT Press, Cambridge, MA (2000)
17. Bernstein, A., Klein, M. Towards High-Precision Service Retrieval, In: Proceedings of The First International Semantic Web Conf. (ISWC), Sardinia, Italy (2002)