Construction and Use of Role-ontology for Task-based Service Navigation System

Yusuke Fukazawa, Takefumi Naganuma, Kunihiro Fujii and Shoji Kurakake

Service & Solution Development Department, NTT DoCoMo, Inc.
NTT DoCoMo R&D Center, 3-5 Hikari-no-oka, Yokosuka, Kanagawa, 239-8536 Japan
{fukazawayuu, naganuma, fujiiku, kurakake}@nttdocomo.co.jp

Abstract. We have been developing a task-based service navigation system that offers to the user for his selected services relevant to the task the user wants to perform. We observed that the tasks likely to be performed in a given situation depend on the user's role such as businessman or father. To further our research, we constructed a role-ontology and utilized it to improve the usability of task-based service navigation. We have enhanced a basic task-model by associating tasks with role-concepts defined in the new role-ontology. We can generate a task-list that is precisely tuned to the user's current role. In addition, we can generate a personalized task-list from the task-model based on the user's task selection history. Because services are associated with tasks, our approach makes it much easier to navigate a user to the most appropriate services. In this paper, we describe the construction of our role-ontology and the task-based service navigation system based on the role-ontology.

1   Introduction

The mobile Internet is expanding dramatically from various viewpoints, such as the number of subscribers and the volume of mobile contents[2]. As the mobile Internet gains in popularity, information retrieval must be made easier and more efficient. Towards this goal, we proposed a task-based service navigation system[1][3] that supports the user in finding appropriate services. Naganuma et al. proposed a method for constructing a rich task-model that represents a wide variety of user activities in the real world. Part of the task-model is shown in Fig.1. The connection between tasks is expressed by the *is-achieved-by* relation. The upper nodes of the task-model have generic tasks, while the lower nodes have more concrete tasks; the end nodes provide associations to services or contents via their URI. To use the task-model for service navigation, the user enters a task-oriented query such as "Go to theme park" and a list of tasks that match the query is sent to the mobile device. The user selects the most appropriate task and, in turn, the corresponding detailed sub-tasks are shown to the user. By repeatedly selecting a task and its sub-tasks, the user can clarify the demand or problem, and when the user reaches an end-node task, appropriate services associated with the selected task in the service DB are shown; a service is invoked by clicking its URI link.
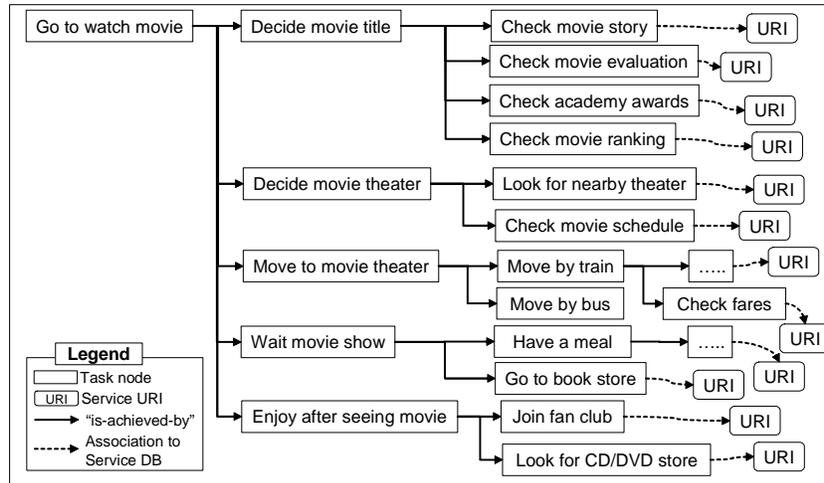
Fig. 1. View of a part of task-model[1]

The above task-model aims at modeling general real world activities that could be performed by the average mobile user. In order for an individual user to access the task-model more effectively, we need to generate a task-list appropriate for the individual user, which would make the user's task-selection process easier. Achieving this goal is particularly important when the user uses a mobile phone since mobile phones have small displays and poor input devices.

Our proposal for generating the task-list appropriate for the individual user is to use the role-concept as the most important factor in selecting tasks or services from the task-model and service DB. In order to use role-concept appropriately, we must categorize the many different roles user can play in the real world and make their relations clear. For this purpose, we have constructed a role-ontology. We have also designed an enhanced service navigation system that uses the role-ontology.

We briefly explain the idea of role-concept in the following. As per Masolo's definition[4], humans can play several roles simultaneously, "FamilyRole" and "ShoppingCustomerRole", and a role can change dynamically in the real world such as "ShoppingCustomerRole" to "TrainPassengerRole" when the user leaves the shop and takes a train. We use this idea of roles in the real world to realize task-based service navigation. That is, we assume that the user can play several roles simultaneously when selecting a task. We define two types of roles; **task-roles** depend on the task selected by the user (i.e. "ShoppingCustomerRole") while **social-roles** depend on the relationship to the surrounding people (i.e. "FamilyRole"). By storing the user's history of tasks selected under both social-roles and task-roles, we can generate a personalized task list and services that match the user's current social-role and task-role.

The remainder of this paper is organized as follows. Section 2 describes our constructed role-ontology and shows how the task-model can be enhanced by the roles defined in the role-ontology. Section 3 describes our task-based service navigation system that can generate a personalized task-list and services according to the current user's role. Section 4 concludes this paper.

2   Role-ontology for task-based service navigation system

In this chapter, we describe our role-ontology, which is used together with the task-model to guide the user to most appropriate services. Prior work is described in the next section. How we represent the role-concept is described in Section 2.2. The constructed role-ontology is described in Section 2.3 and the enhanced task-model is shown in Section 2.4.

2.1   Prior work

In the knowledge representation field, role-ontology has been studied in order to construct an accurate domain-ontology (conceptualization of knowledge domain) by strictly separating the role concept from the objective domain.

Sowa distinguished between natural types "that relate to the essence of the entities" and role types "that depend on an accidental relationship to some other entity"[5]. In his subsequent work, he asserts that role types are subtypes of natural types[6]. For example, the role types *Child*, *Pet* and *Quotient* are subtypes of the natural types *Person*, *Animal* and *Number*, respectively.

In developing Sowa's ideas further, Guarino makes an ontological distinction between role and natural types, and this has been adopted as the basic notion by other researchers in handling the role-concept in ontology[7]. In his notion, a role is a concept that is supported by other concepts while, on the other hand, a natural type is characterized by being free to stand independent from any relationship with others. For example, *Person* is a natural type since to be a person is independent of any relationships with other concepts and an individual person will always remain a person. On the other hand, *Student* is a role since to be a student requires entrance to university and a student stops being a student when he/she drops out or graduates.

Fan et al. noticed that role is heavily dependent on events, and stated that "the representation of role-concept is an extensional features of an entity that are due to its participation in some event"[8]. Sunagawa et al. noted that the relationship between context and role was decided by the context and proposed two relations, the *part-of* relation and the *participant-in* relation, and developed the ontology construction tool named *Hozo*, which can represent these relations[9].

Work to date has discussed how the role-concept and its relation to other concepts (such as natural type, instance, and other role concepts) must be represented in a knowledge-base or ontology. However, no approaches utilizing the role-ontology to generate a UI that can reduce the user's effort in reaching the objective resource have been published so far. In this paper, we construct a
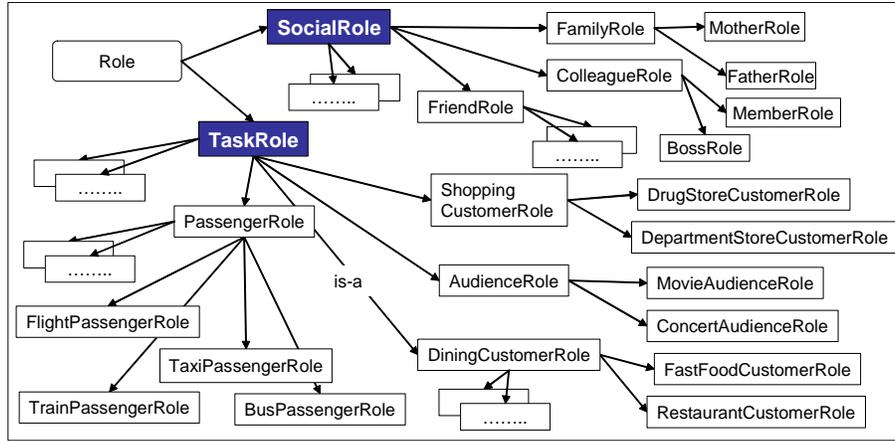
Fig. 2. Role-ontology for task-based service navigation system

role-ontology appropriate for the task-based service navigation systems explored in our previous work[1]. We then propose a new task-based service navigation system that can generate a task-list according to the current user's role and personalize it appropriately.

2.2 Representation of role-concept

The role-concept continues to attract researchers from different areas such as linguistics[10], knowledge representation[5], relational database[11], and access control[12]; a significant amount of research has been done to date.

Despite the many different ways in which the role-concept has been used, the number of ways roles have been represented is rather small. Steimann[13] examined more than 50 papers and identified three common ways of representing roles. In the following, we explain two role representations. First, the role can be represented by the relationships between roles. In the examples above, the relation *is-a* can represent the relationship between "PassengerRole" and "TrainPassengerRole". Second, the role can be expressed by the relationship between role-concept and another concept. For example, the relation *is-played-by* can represent the relationship between task-concept "Go to watch movie" and the role "MovieAudienceRole".

One aim of this paper is to construct a role-ontology and then enhance the task-model through the addition of the role-concept. We use the first way of representing role-concept to represent the relationship between role-concepts in role-ontology. We use the second way of representing role-concept to represent the relationship between task (defined in task-model) and role-concept.

2.3 Discussion of role-concept and constructed role-ontology

In this section, we discuss how the user can play several roles in the task-selection process, and then the constructed role-ontology is explained. We start by adapting the definition of roles given in[4]; 1) role is a property assigned to humans, roles can change dynamically, and 2) humans can have multiple roles simultaneously. We adapt the 1st property by supposing that the user changes a role when selecting a task, which we call the task-selection process. For example, the user plays "MovieAudienceRole" when the user selects "Go to watch movie", and if the user selects "Move to movie theater" from the child nodes of the previously selected task, the user's role changes to "PassengerRole". We define these roles that can change during the task-selection process as **task-roles**.

We adapt the 2nd property by supposing that user can play multiple roles during the task-selection process. In the task-selection process, the user can select just one task from the task-list; therefore the user has only one task-role as per the above definition. Other than the task-role, we define the **social-role**; it is decided by the people the user wants to do the task with, such as family or friends. The user can play both a task-role and a social-role simultaneously. For example, if the user selects the task for planning to go to watch movie at the weekend with Family, the user plays both "FamilyRole" and "MovieAudience-Role" simultaneously in the task-selection process.

We have constructed the role-ontology by using both of these defined role-concepts; task-role and social-role, and the relationships introduced in the previous section. The constructed role-ontology is shown in Fig.2. The concept "role" has two top-level role concepts: "social-role" and "task-role". As mentioned in the previous section, the relationships between role-concepts are expressed using *is-a* relation. For example, task-role has role-concepts such as "PassengerRole", "AudienceRole", "ShoppingCustomerRole" and "DiningCustomerRole" as its child nodes. On the other hand, social-role has role-concepts such as "FamilyRole", "FriendRole" and "ColleagueRole" as its child nodes.

2.4 Enhancing task-model with roles

In this section, we explain the enhancement of the task-model through application of the constructed role-ontology. As we described in the previous section, the task-role can change according to the task selected by the user. Therefore, in order to catch the change in task-roles and recommend the services appropriate for the current task-role, we need to express the relationship between the task and the task-role. For this we use the relation *is-played-by* introduced in Section 2.2.

Fig.3 shows a part of the task-model in which task-roles are related to task nodes. The node "Go to watch movie" has five children nodes, one of which is "Move to movie theater". The node "Go to watch movie" is associated with the task-role "MovieAudienceRole" by the *is-played-by* relation. This relation indicates that the task is performed by the user who plays the designated task-role. The node "Move to movie theater", on the other hand, is associated with
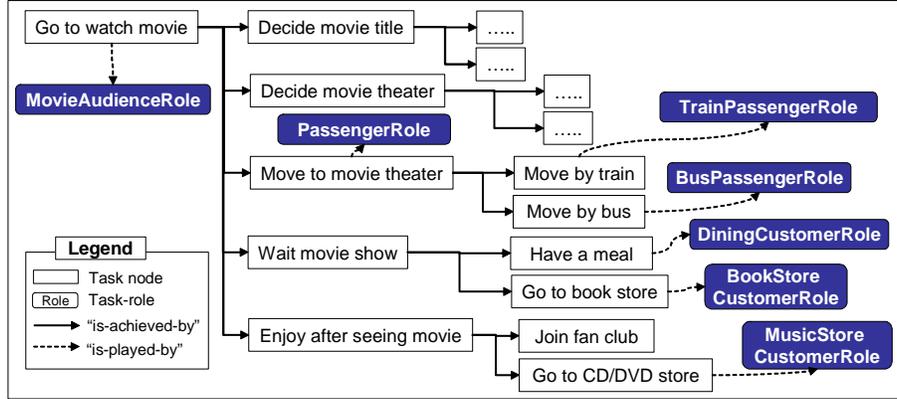
Fig. 3. Enhancement of task-model using role-concept defined in role-ontology

the task-role "PassengerRole". By associating the task with the task-role, we can recognize that the user plays "MovieAudienceRole" when the user selects "Go to watch movie". In addition, we can capture the change in user's role to "PassengerRole" if the user selects "Move to movie theater" from the child nodes of the previously selected task. Note that some nodes in Fig.3 have no corresponding task-roles; these nodes take the task-role of their parent.

2.5 Reasoning functionality of role-ontology

Our proposed role-ontology is not only used for clearly distinguishing between the various kinds of roles, but also for supporting determining task candidates, their order, and recommended services appropriate for current user's social-role/task-role by using the *is-a* relation defined in role-ontology. Basically, these determinations can be done based on the relationship between task and current user's social-role/task-role, which can be found in enhanced task model shown in Fig.3 and user's task selection log, see Table 1. However, there is a possibility that we cannot find the task that has relationships to the current task-role or current social-role, and as a result, we cannot make the above determinations. We deal with this problem by using the *is-a* relation defined in role-ontology and make the above determination by reasoning from the relationship between the task and the child/parent role of current task-role or social-role. Details of these determination of the task candidates, their order, and recommended services for each task candidate are described in Sections 3.2, 3.3 and 3.4, respectively.

3 Enhanced task-based service navigation

In this chapter, we propose task-based service navigation as realized by the enhanced task-model and the role-ontology. The overall process of enhanced task-based service navigation is shown in Fig.4.
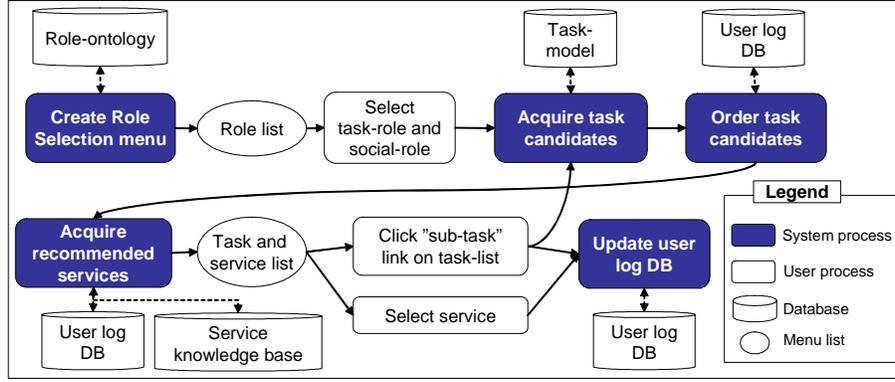
Fig. 4. Overall processes of enhanced task-based service navigation

In the following section, we explain the system processes in the figure, acquisition of user's current social-role and task-role in Section 3.1, acquisition of task candidates in Section 3.2, acquisition of recommended services for each task candidates in Section 3.3, and personalization of task-list including updating user log DB and ordering task candidates in Section 3.4.

3.1 Acquisition of user's current social-role and task-role

The role-selection menu is generated in order to acquire the user's current social-role and task-role. A screenshot of a generated social-role selection menu is shown in the upper part of Fig.5(a). The social-role selection menu consists of the child nodes of the social-role as defined in the role-ontology such as "FamilyRole", "FriendRole" and "ColleagueRole". "None" is not a role-concept but means that the user is considering to do the task by himself. A screenshot of a generated task-role selection menu is shown in the lower part of Fig.5(a). This role-selection menu is customizable. By selecting "edit this menu" link on the role selection menu, social-roles and task-roles can be added to the role-selection menu by selecting additional roles from the role list.

3.2 Acquisition of task candidates

This section describes the generation of task-candidates. The "Determine task candidates" step in Fig.4 has two inputs from other processes; 1)"select task-role and social-role", and 2)"click sub-tasks", as explained below.

After the user selects task-role, tasks associated with either the selected task-role or child node role-concept of the selected task-role, are acquired from the enhanced task-model. For example, if the user selects "Audience" from the task-role menu, the user's current task-role is judged as "AudienceRole". We acquire
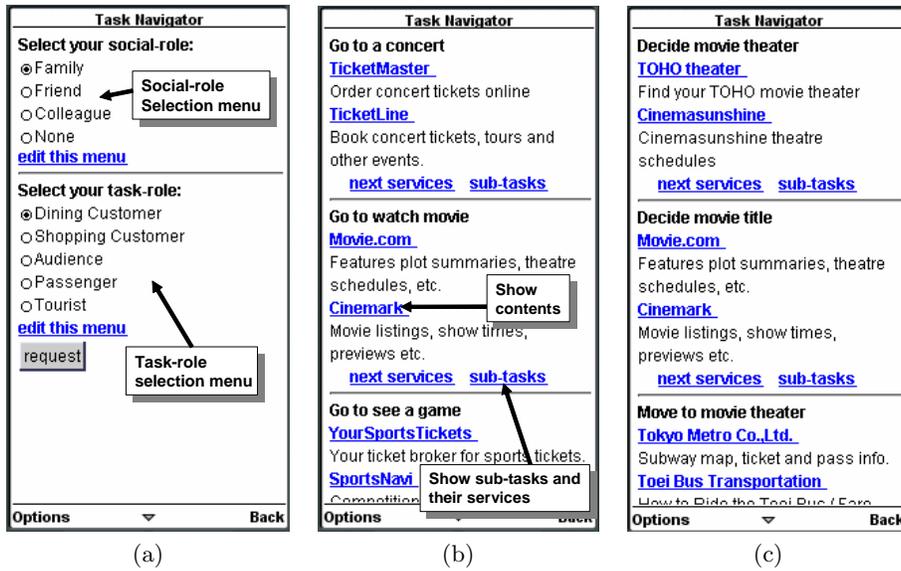
Fig. 5. Screenshots of service navigation;(a)role-selection menu, (b) task-list and recommended services when user selected "Audience Role" from the role-selection menu shown in (a), (c)task-list and recommended services when user clicks "sub-tasks" link of "Go to watch movie" from the task-list shown in (b)

the task candidates; "Go to watch movie", "Go to see a play", "Go to a concert" and "Go to see a game", which are associated with the child roles of "AudienceRole" such as "MovieAudienceRole", "PlayAudienceRole", "ConcertAudienceRole", and "GameAudienceRole", respectively.

We create the task selection menu list (task-list) as shown in Fig.5(b). Each task candidate in the task-list has two kinds of links to the other task-list; the links named "sub-tasks", and the links to recommended services for each task candidate. The links to the recommended services take the user to the mobile contents, which user can use to accomplish the user's task. The links named "sub-tasks" can refine the user's demand by showing a list of the sub-tasks of the task candidate and their services. If user clicks "sub-tasks", sub-tasks associated with the task of clicked "sub-task" are extracted from the enhanced task-model as new task candidates. For example, if the user selects "Go to watch movie" from the task-list shown in Fig.5(b), the new task candidates; "Decide movie title", "Decide movie theater", "Move to movie theater" and "Enjoy after seeing movie", which are associated with the task "Go to watch movie", are acquired. We create the task-list using the acquired task candidates, and their recommended services as shown in Fig.5(c).

3.3 Determining services for each task candidate

This section presents the process of determining the recommended services for each task candidate. In the system proposed by Naganuma et al., the user first selects the generic task, which is an upper node of the task-mode such as "Go to watch movie", and gradually refines the demand by repeatedly selecting the specific task associated with the generic task in a tree structure. When user reaches the end-node task, user can find the services to accomplish the task. In our method, we recommend the typical services together with generic tasks (task candidates shown at the beginning) which can reduce the number of operations the user must perform in finding and selecting the desired task.

While refining the task candidates, the user can play not only the task-role associated with the task-candidates but also other task-roles associated with the sub-tasks of the task-candidate. For example, the user can play not only "MovieAudienceRole" but also task-roles such as "PassengerRole", "MusicStore-CustomerRole", "DiningCustomerRole" while refining the task candidate "Go to watch movie" as shown in Fig.3.

It is clear that the task-role played most frequently while refining the task candidates will depend on the individual. Therefore, we realize the personalization of recommended services by determining services according to the task-role history, which is different among individual users. Our method of using the user's history of task-selection to acquire this task-role is described in the next section. After determining the task-role, we acquire the services associated with the end-node task, which is associated with the determined task-role, from the end-node sub-tasks of the task candidate. If there are no such end-node tasks, we use *is-a* relations defined in the role-ontology to determine the recommended services. Concretely, we search for end-node tasks whose associated task-role is a child task-role of the determined task-role.

We show here an example of acquiring the recommended services for the task candidate "Go to watch movie" assuming that the determined task-role played most frequently while refining the task candidate is "MovieAudience-Role". At first, the end-node sub-tasks of the task candidate "Go to watch movie", are acquired such as "Check movie evaluation", "Check movie schedule", "Check fares", "Look for CD/ DVD store" etc. from the enhanced task-model shown in Fig.3. Next, we acquire the tasks associated with the determined task-role "MovieAudienceRole" from the acquired end-node tasks. In this case, both "Check movie evaluation" and "Check movie schedule", which inherit the task-role of "MovieAudienceRole" from their parent task node, are acquired. Next, we acquire the services associated with these acquired end-node sub-tasks and present the services as recommended services for the task candidate "Go to watch movie" such as "Movie.com" and "Cinemark". We show the recommended services below the links of each task candidate as shown in Fig.5(b).

We show here another example of acquiring the recommended services "Move to movie theater" whose task-role is "PassengerRole" in Fig.5(c), we must find the end-node tasks whose task-role is "PassengerRole" and acquire the services associated with the tasks. However, there are no such end-node tasks whose
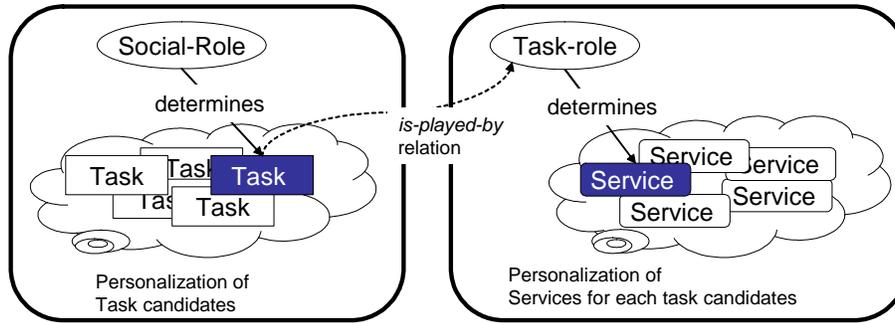
Fig. 6. Personalization of task and services in terms of role-concept

task-role is "PassengerRole" as shown in Fig.3. Instead, we search for end-node tasks whose task-role is a child task-role of "PassengerRole" such as "TaxiPassengerRole" and "TrainPassengerRole", and acquire the services associated with these end-node tasks.

3.4 Personalization based on user's task-selection log

We observe that the social-role the user plays determines the user's task, which is socially limited, and the task-role, on the other hand, determines the services or lower level tasks that the user needs to accomplish the task determined by the social-role (Fig.6). For example, when the user is with family, the user sometimes go to see a movie, and always uses the services "Tokyo metro" to check train timetable and "Cinemasunshine" to check movie schedule. In this case, the user is prone to executing the task "Go to see a movie" while playing "FamilyRole" and is prone to use the services of the lower level task "Check train timetable" and "Check movie schedule" while playing "MovieAudienceRole", which is the task-role of "Go to see a movie". We use this observation to personalize the task-list and services. We use social-role to determine priority order between multiple task candidates, and use task-role to determine the recommended services. In the following, we first describe the method of storing user's log of task-selection, and then propose the personalization of the task-list and recommended services; examples are provided.

Storing user's log of task-selection This section describes how the task-selection actions are stored and updated. The "Update user log DB" step in Fig.4 has two kinds of input from other processes; 1)"Click sub-tasks", and 2)"Select services", as explained below. In the following, we call the task candidate of the links and services user has clicked or selected the target-task.

If the user clicks a "sub-tasks" entry, the target-task is stored in the user log DB together with the user's social-role. On the other hand, if user selects service,

Table 1. An example of user log DB

Social-role	Selected task	Frequency
FriendRole	Go to watch movie	4
	Enjoy after seeing movie	4
	Look for CD/DVD store	4
FamilyRole	Go to a concert	2
	Move to a concert hall	2
Task-role	**Selected task**	**Frequency**
MovieAudienceRole	Look for CD/DVD store	4
ConcertAudienceRole	Move to a concert hall	2

the target-task is stored in the user log DB together with both user's social-role and task-role since selection of the service includes both selection of target-task from task candidates and selection of service of target-task simultaneously. This process is a reflection of the facts that social-role determines the task and task-role determines services to accomplish the task (Fig.6). We show an example below. The stored log for the following explanation is shown in Table 1.

The user selects "AudienceRole" as task-role and "FriendRole" as social-role from the role selection menu yielding Fig.5(a). If the user then clicks the link "sub-tasks" of "Go to watch movie", see Fig.5(b), the target-task "Go to watch movie" is stored together with current user's social-role "FriendRole". The user's current task-role changes to "MovieAudienceRole". Next, if the user selects "sub-tasks" of "Enjoy after seeing movie" from the task-list shown in Fig.5(c), "Enjoy after seeing movie" is stored together with user's social role "FriendRole". Next, if user selects service "HMV.com" of the task "Look for CD/DVD store" from the newly shown task-list, "Look for CD/DVD store" is stored together with both user's task-role "MovieAudienceRole" and social-role "FriendRole".

As another example, the user selects "AudienceRole" as task-role and "FamilyRole" as social-role from the role selection menu. If the user then clicks "sub-tasks" of "Go to a concert" from the task-list, the task "Go to a concert" is stored together with "FamilyRole". The user's current task-role changes to "ConcertAudienceRole". Next, if the user clicks the link of the service of "Move to a concert hall" from the task-list, the target-task "Move to a concert hall" is stored together with both user's task-role "ConcertAudienceRole" and social-role "FamilyRole".

Personalization of task-list and their services We realize personalization of the task-list by ordering the task candidates according to the frequency with which the user selected the tasks under the current user's social-role. We first acquire task candidates according to the flow described in Section 3.2. Next, the user log DB is checked and if a task candidate is stored under the **user's current social-role**, we acquire the number of times the user selected the task. If no such prior information is held in the DB, we use *is-a* relations defined in the role-ontology to order task candidates. Concretely, task candidates are ordered

by referring to the task selection log of parent or child social-role of the current social-role. If no task selection log of both parent or child social-role is held in the DB, we treat the number of times that the task candidate was selected as 0. We acquire this data for all task candidates and use it to order the task candidates in descending order of the number of times selected.

Next, the recommended services for each task candidate are determined. As per Section 3.3, we determine the recommended services by acquiring the task-role the user has played most frequently while refining the objective task, which we call target task-role in the below. We determine the target task-role by acquiring the task-role of the task user has selected most frequently while playing the **task-role of each task candidates** from the user log DB. If there is no task selection log stored together with the task-role of the task candidate, we determine the target task-role as the task-role of the task candidate. Next, the services associated with all end-node sub-tasks whose associated task-role is the same as the target task-role or the child role-concept of the acquired task-role is determined. Detail of the acquisition of recommended services after target task-role is determined is given in Section 3.3. Finally the ordered task candidates and their recommended services are shown to the user.

Example of generating personalized task-list We show an example of generating a personalized task-list. If the user selects "FriendRole" as social-role and "AudienceRole" as task-role from the role selection menu, the task candidates; "Go to watch movie", "Go to see a play", "Go to a concert" and "Go to see a game", all associated with the child nodes of "AudienceRole", are selected from the task-model. After acquiring the task candidates, the system refers to the user log DB shown in Table 1, and finds that the most frequent task candidate associated with "FriendRole" is "Go to watch movie". Accordingly, "Go to watch movie" is given highest priority when ordering the task candidates.

Next, the recommended services for each task candidate are determined. We explain here the determination of services for "Go to watch movie". The system refers to the user log DB shown in Table 1, and finds that "Look for CD/DVD store" is the most frequently selected task under "MovieAudience-Role", which is the task-role associated with "Go to watch movie". Therefore, the task-role user played most frequently while refining the task candidate "Go to watch movie" is judged to be "MusicStoreCustomerRole", which is associated to "Look for CD/DVD store". Next, the services associated with all end-node sub-tasks whose task-role is "MusicStoreCustomerRole" are acquired such as "HMV.com" and "iTunes music store". Finally the ordered task candidates and their recommended services are shown to the user as shown in Fig.7(left).

As another example, if the user selects "FamilyRole" as social-role and "AudienceRole" as task-role from the role selection menu, the same task candidates as described in above are acquired. After acquiring the task candidates, the system refers to the user log DB shown in Table 1, and finds that the most frequent task candidate associated with "FamilyRole" is "Go to a concert". Accordingly, "Go to a concert" is given the highest priority when ordering the task candidates.
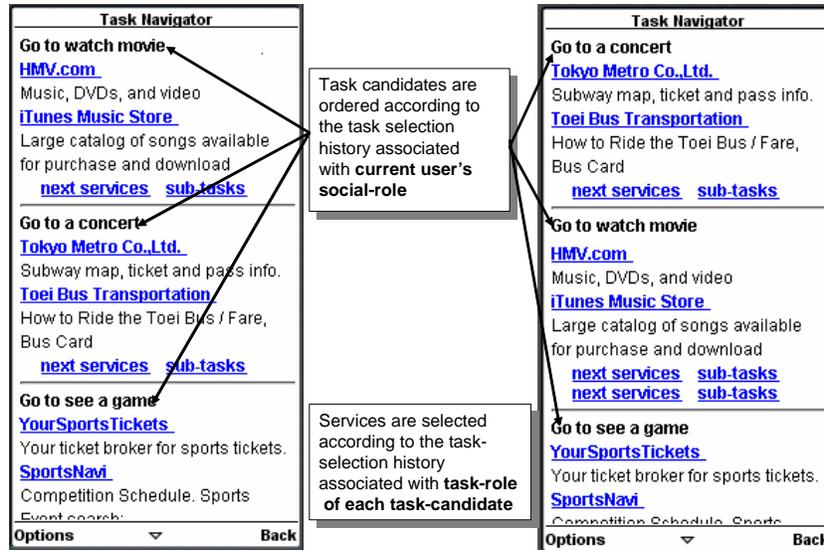
Fig. 7. Screenshots of personalized task-list and services; left: task-list and services tuned for a user with friend, right: task-list and services tuned for a user with family

Next, the recommended services for each task candidate are determined. We explain the determination of services for "Go to a concert" below. The system refers to the user log DB shown in Table 1, and finds that "Move to a concert hall" is the most frequently selected task under "ConcertAudienceRole", which is the task-role of "Go to a concert". Therefore, the task-role the user played most frequently while refining the task candidate "Go to a concert" is "PassengerRole", which is the task-role associated with "Move to a concert hall". Next, the services associated with all end-node sub-tasks whose task-role is child role of "PassengerRole" are acquired, since there are no end-node sub-tasks whose task-role is "PassengerRole", such as "Tokyo metro" and "Toei bus". Finally the ordered task candidates and their recommended services are shown to the user as shown in Fig.7(right).

We show an example when the social-role is first designated and there is no associated task selection log entry. Here, we consider that the user customizes his role selection menu using the "edit this menu" link on the role selection menu shown in Fig.5(a) and designates the social-role as "Brother". In this case, we cannot order task candidates using the task-selection log since there are no such log entries (See Table 1). Instead of using the task selection log associated with "Brother", we use the task selection log associated with "Family", which is the parent node of "Brother". As a result, even if the user manually designates the "Brother" role, the user is presented with the same order of task candidates as for "Family" as shown in Fig.7(right). Here, if user clicks a "sub-task" link on the task-list, the new user task selection log entry shows "Brother", not "Family".

4   Conclusion

In this paper, we constructed a role-ontology and applied it to realize improved task-based service navigation. For this application, we enhanced the basic task-model by associating tasks with role-concepts defined in a role-ontology. By acquiring the user's current role and watching the role changes, we can generate a personalized task-list that allows the user to perform the task-selection process more efficiently. The system can also recommend services for each task candidate appropriate for the task-role associated with each task candidate, which provides the user with more chances of reaching the desired service as soon as possible. In future work, we will construct a rich role-ontology that can express the relation to other concepts; such as situation and context (place, time, etc.) so as to reason the user's current social-role and task-role from the user's situation.

References

1. T. Naganuma and S. Kurakake. Task knowledge based retrieval for service relevant to mobile user's activity. In Y. Gil et al., editor, *4th International Semantic Web Conference: ISWC 2005, LNCS 3729*, pages 959–973, 2005.
2. A. Serenko and N. Bontis. A model of user adoption of mobile portals. *Quarterly Journal of Electronic Commerce*, 4(1):69–98, 2004.
3. Y. Fukazawa, T. Naganuma, K. Fujii, and S. Kurakake. A framework for task retrieval in task-oriented service navigation system. In R. Meersman et al., editor, *Int. Workshop on Web Semantics, LNCS 3762*, pages 876–885, 2005.
4. C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In *Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning*, pages 267–277, 2004.
5. J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, New York, 1984.
6. J.F. Sowa. Using a lexicon of canonical graphs in a semantic interpreter. In M.W. Evens, editor, *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, pages 113–137. Cambridge University Press, 1988.
7. N. Guarino. Concepts, attributes and arbitrary relations. *Data & Knowledge Engineering*, 8:249–261, 1992.
8. J. Fan, K. Barker, B. Porter, and P. Clark. Representing roles and purpose. In *Proc. of the Int. Conf. on Knowledge Capture*, pages 38–43, 2001.
9. E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi. Organizing role-concepts in ontology development environment: Hozo. In *AI Technical Report*, volume 4 of *1*, pages 453–468. Artificial Intelligence Research Group, I.S.I.R., Osaka Univ., 2004.
10. C.J. Fillmore. Types of lexical information, ausztige abgedruckt. In R. Dirven and G. Radden, editors, *Fillmore's Case Grammar: A Reader*. Julius Groos Verlag, 1987.
11. E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
12. R. S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
13. F. Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data and Knowledge Engineering*, 35:83–106, 2000.