

Ubiquitous Service Finder

Discovery of Services Semantically Derived from Metadata in Ubiquitous Computing

Takahiro Kawamura¹, Kouji Ueno¹, Shinichi Nagano¹,
Tetsuo Hasegawa¹, and Akihiko Ohsuga¹

Research and Development Center, Toshiba Corp.

Abstract. Metadata have been already given to most of the data and objects in the real world, such as books, foods, digital contents like movie, electric devices, and so forth. Further, they can be accumulated electronically by barcodes and RFIDs, which is expected to spread explosively in 2005. On the other hand, web services are getting popular in the internet, and UPnP services and ECHONET are penetrating into the home network. In our project, we propose a new handheld application called Ubiquitous Service Finder, in which user can intuitively browse as icons the metadata around him/her in a cellular phone, then invoke the services semantically related to the metadata by simple drag and drop operation.

1 Introduction

Objects and data are everywhere in the world, then most of them have their own descriptions, metadata. For example, industrial products have names and dates of manufacture and model numbers, and foods have production places and producers. As the already disseminated standards, there are EAN and UPC for barcodes, EPG for movie data, and ID3 for music data. Further, in near future RFIDs will lead to add PML and/or ucode to the products. Since food safety and recalls are common concerns these days, situations that people would need refer to such information will increase. On the other hand, there are lots of services and information in the internet, which can be related and used with such information. For example, manufacturers are providing their product catalog search service, and some public agencies are putting food safety information on the web. Further, map services, news search, and banking services, etc. became already quite popular. As their standards, there are RSS, FOAF, PICS, P3P for some kinds of web pages, and WSDL, OWL-S and WSMO for web services and their annotations. Services are not only in the internet, but also in the home network according to dissemination of home information appliances. For example, play and record functions of latest HDD recorders, temperature control functions of air conditioners, and surveillance cameras can be accessible via LAN. As their common standards, there are CC/PP for device profiles, UPnP and DLNA mainly for digital audiovisual players, and ECHONET for white

goods. Furthermore, the standards like RTMiddleware and RSi is now under consideration for coordination among home robots via the network near future.

That is, metadata and services are flooding around us, nevertheless, there is no simple and direct access method to them. You may have the following experiments: at inventory clearance users always need turn their PCs upside down to take a memo of small but long catalog numbers, and at stores need check tiny and illegible information on some food packages. Besides, you have no way to search the related information on the web at that moment. You first have to go home, and have a chair in front of your PC, then type that information by hand. In addition, we believe ubiquitous computing which is a product space of pervasive and mobile has two essentials: people can use the same services everywhere, and people can use the pinpoint services depending on that time and location.

Therefore, we have developed Ubiquitous Service Finder (USF) to provide the simple and direct access to ubiquitous metadata and services in a way of combining both essentials. Here, we took a cellular phone as a target device because of its ownership rate and adherence of younger people to bring it into their beds. Then, as the most intuitive and friendly interface for the people, we took an interface to represent metadata and services as icons. USF allows the user to browse a large variety of metadata and services surrounding him/her, and to invoke the services by icon-click and drag&drop operation. So that in cases that what specification does this PC have?, what songs are inside of this CD?, and when is the expiration date of this meat?, etc., you can just hold up your phone to them, and get displayed the objects as icons. Then, by simply clicking it you can see the content of their metadata, and store them locally. The user has no need to turn over something and take a memo by a pen. In addition to that, services in the internet and provided by the home information appliances are also displayed as icons in the phone. Then, by drag&dropping an object icon to a service icon you can just assign the metadata to the service inputs, and invoke it. The user has no need to start up your PC and type the keyboard.

In the rest of this paper, section 2 shows USF architecture and technical points inside, and section 3 illustrates three of typical usecases for USF. Then, we locate USF in the current ubiquitous computing research in section 4, and discuss the related activities in section 5. Finally, our status and future plan are summarized in section 6.

2 Proposal of Ubiquitous Service Finder

2.1 Architecture

In this paper, we have assumptions that metadata for the industrial products and foods can be looked up electronically, and web services in the internet and UPnP services in the office and home network can be accessed.

USF shown in fig. 1 is an application to coordinate those metadata with any of services. In this figure, you can see several kinds of icons which are corresponding

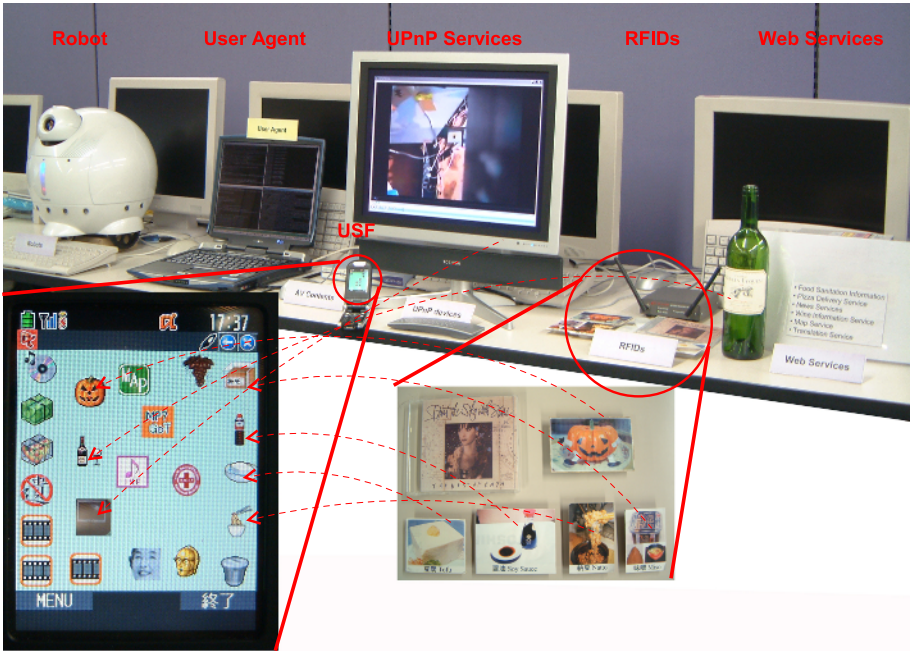


Fig. 1. Snapshot

to the real objects, data, and services. Note that the cellular phone is equipped with a touch panel as the screen.

Further, fig. 2 shows the internal architecture. Here we took SOA (Service Oriented Architecture) as our fundamental design, then services mean not only web and UPnP services, but also object and data (movie, music, photos, etc.) which only have outputs such as metadata and data themselves. In the following, we describe some of components in USF architecture.

WA2WS Getaway transforms web applications like cgis to web services, which have WSDL interfaces to be published and are accessible via SOAP. The detailed description can be found in [1]. Also, Annotator which is under development suggests the related ontology classes to the service. The user can just select some of them to make it semantic web service.

UPnP2WS Getaway is another gateway to transform UPnP services to web services, which is combined with UPnP Control Point (hub to manage UPnP devices), and automatically finds UPnP services within a sub-domain, then publishes them as web services accessible via SOAP to User Agent below.

User Agent is assumed to be in a home server (or its corresponding PC) in the home, and/or in a PC for personal use in the office. It gets metadata corresponding to tag ids detected by RFID readers from Metadata DB, and shows them as icons on the following USF device, as well as the available services in the internet and the home network. Then, according to the user's

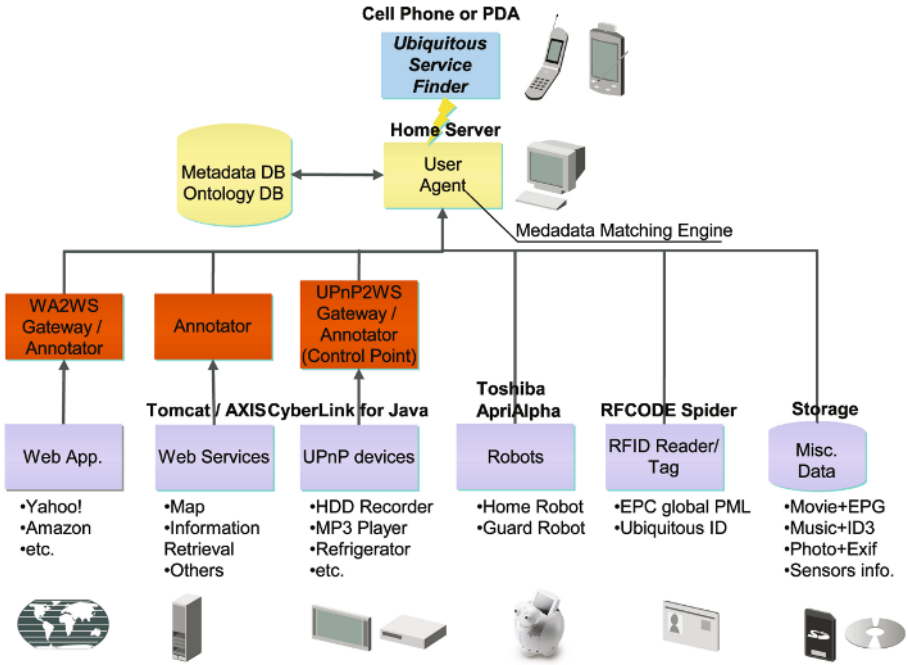


Fig. 2. USF Architecture

operation on USF device, it returns the content of metadata to USF device, and/or invokes web/UPnP services. User Agent includes a scripting system for sophisticated service flows, then if the user has some experiment on programming it is possible to add the user's own customization and batch processes. Further, Metadata Matching Engine described in the next section is included in User Agent.

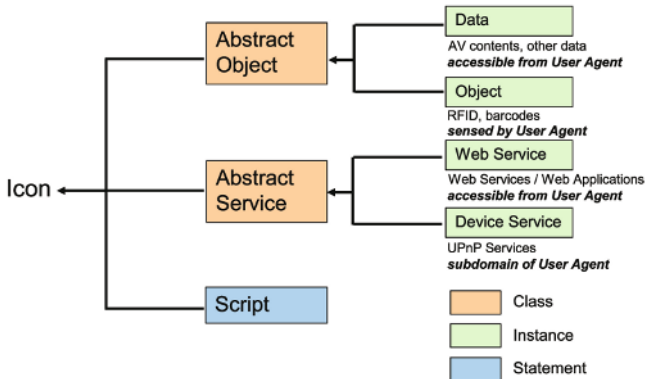


Fig. 3. Icons in USF

Metadata DB stores several kinds of metadata, and Ontology DB has a set of ontologies to find the related services to the metadata.

USF in the narrow sense is a Java application running on cellular phones, which displays the icons representing the objects, data, and services based on the information sent by User Agent. The user can check the metadata and invoke the services by familiar icon operations. Figure 3 lists the icons which can be seen on USF. Section 3 illustrates how to use those icons.

2.2 Metadata Matching Engine

In terms of HCI aspect, USF is characterized by its icon abstraction of objects, data and services in the real world, then by reflecting them on the mobile device giving the intuitive and familiar feeling as desktop PCs to the user in the ubiquitous computing environment. However, as you would easily imagine this approach has drawbacks that lots of icons flood in the small display, then it is confusing which metadata can be combined with which services. Also, the simple click and drag&drop operations do not have enough expressiveness to specify which value in the metadata to be inputted to which argument in the service if that service needs more than one inputs. Therefore, we have developed Metadata Matching Engine (MME) to the above problems, which calculates the relationship between metadata and services, then proposes the possible combinations. To this end, it has ontology (precisely, more like taxonomy) including 160,000 concepts, and semantic service matchmaker[2,3]. The following sections shows the feature of metadata matching.

Mapping from Metadata to Ontology. First of all, MME parses the target metadata, and by comparison with ontologies stored in DB retrieves a meaning (concept in DL sense) corresponding to each property in the metadata, such as producer, production place, expiration date, and so forth. For now metadata have several formats, and are under standardization by each industrial segment or organization. So that metadata parsers must be developed for each format, however, most of metadata have essentially tuple form where property and value(s) are pairwised. Thus, MME takes the property and compares it to the concepts in the ontologies. Ontology is a set of concepts that defines a domain with their properties and the relations between them. We currently have so-called lightweight ontologies, in the sense that they have no complicated relation and philosophical deliberation, but a huge number of concepts with some simple relations. Base on those ontologies, MME finds out the semantically corresponding concept to the property using regular expression technique and schema mappings. But, the properties defined in metadata are not arbitrary names, it is a set defined in advance. So in practice we are able to prepare most of mappings between a property and a concept. Then, since USF handles the objects and data as services, the concepts included in those metadata become the outputs of those services.

Discovery of the Related Services. As the next step, MME searches on services which can be used with the metadata. The services must be annotated as semantic

web services[4] in advance. Semantic web services is an attempt of putting meta-data referring ontology to web services and their exploitation for service discovery and composition. Currently, we transformed 50 kinds of web applications to web services by WA2WS Gateway, and annotated them as semantic web services for this project. We hope to have publicly accessible semantic web services in the internet near future. In the above semantic web services, service description language OWL-S[5] assigns any of concepts to a category and each input/output, etc. Thus, MME calculates the relationship between the concepts returned from metadata in the previous section and the concepts assigned to the services. Since the ontology is defined based on Description Logic, MME checks to see if there is a relation like subsumption, union between those concepts. We developed a web services matchmaker to find the similar services based on ontology[2,3], so here extended it to metadata for objects, data and UPnP services. In practice, MME firstly determines whether there is a certain degree of similarity between the most representative concept of the metadata and the concept for the service category. Secondly, it checks if all of the concepts for the service inputs can be supplied by any of the output concepts in the metadata. That is, the service discovery has two-step approach with a class hierarchy of service categories and IO Type matching. Then, if a service which can take the outputs of the metadata as inputs is found, the next service which can take the outputs of the service as inputs will be searched. After repeating this process a certain times, a sequence of possible combinations of services can be found. USF will show this sequence as a directed link of icons highlighted with red arrows to the user. This behavior means a kind of simple reactive planning where the services descriptions are operators. If there are more than one operator to be connected, MME sorts the possible candidates according to distance of the concepts and certainty calculated from user contexts below. If the user does not like the proposed combination, he/she can get the next combination displayed. Note that MME is just suggesting the possible combinations by planning once or several times, and the service invocation is not automatic. It will be done by the user's drag&drop operation of the first icon to one of the icons on the directed link. After that, User Agent will pass the values corresponding to the properties (concepts) in the metadata to the service, and the output values of the service to the next service for the sequential service invocation. We will show some examples in section 3.

Further, USF also allows the user to make service flows in advance as script programs. In our first observation we thought it had a limit to define in advance the service flows with specific service bindings, because there would be tons of services in the internet and the information appliances vary in each home. Therefore, we have developed MME to automatically suggest the combinations of services based on semantics. However, as an intermediate way we are now providing so-called latebinding scripts, where abstract services can be defined in the flow without specific bindings, and at the runtime the services to be invoked will be searched. The abstract services have OWL-S, and the above matchmaker searches for the possible services to invoke. The user can describe his/her own script, and User Agent will execute it.

3 Usecases

As described in section 1, USF has been developed for the purpose of providing an intuitive accessor to metadata of objects and data around us and services in the inter/home network. In this section, we introduce 3 useful cases: firstly the user browses metadata bound to the physical objects, secondly the user simply invokes a service with the data, lastly the user gets MME suggested a service sequence from metadata and executes the service sequence. Further, we summarize other features in USF.

3.1 Usecase 1: Metadata Scouter

Firstly, we show the simplest example that the user browses metadata surrounding him/her via USF (fig. 4(a)). In this example, we use RFIDs to find the objects near the user carrying USF. There are already some mobile devices attached with RFID readers, such as Ubiquitous Communicator[6], a cellular phone[7]. Although we expect USF will be ported to those devices near future, the current implementation is a simple but versatile one where a Java-enabled cellular phone has a RFID in the back. RFID readers are connected to a PC

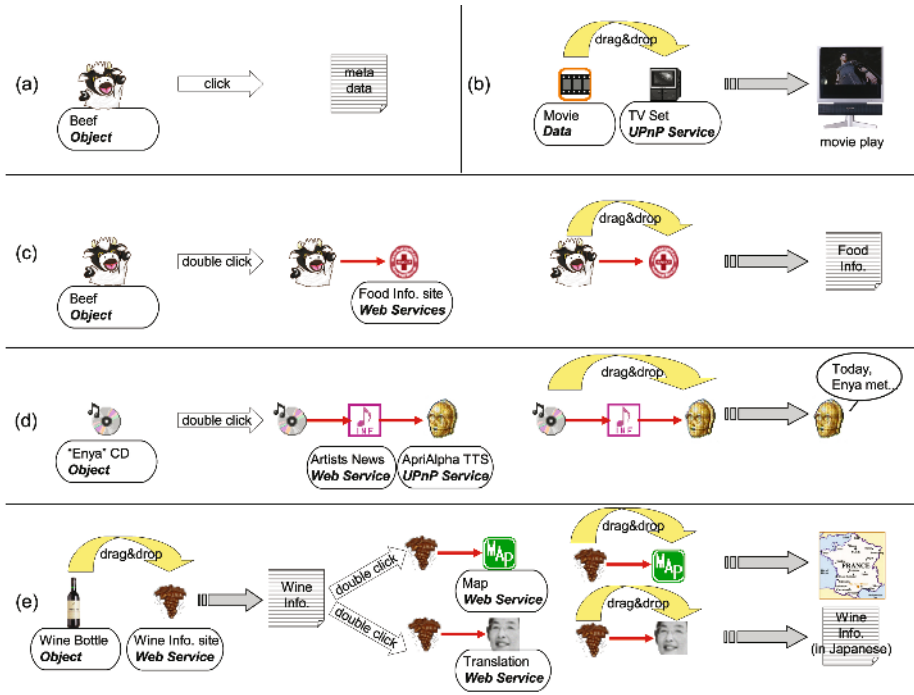


Fig. 4. Usecases

running User Agent, and controlled by the Agent. If User Agent detects the tag id assigned to the cellular phone, it determines other ids detected by the same reader as the user's neighborhood, and get icons corresponding the ids displayed on USF application in the phone.

For instance, when the user goes into the kitchen with USF, a tag attached to a beef pack bought at a store is detected, and a beef icon will appear on USF. Then, if the user clicks on the icon, User Agent accesses the metadata DB with the tag id to get the description about the cow's birth place and the expiration date, etc. and get that text information displayed on the user's USF. For the other use, it is useful for checking the specification of PCs and electronic equipment purchased some time ago, or for copying metadata of books and DVDs in the real shops into USF, then comparing their prices with the net shops in the home ¹.

3.2 Usecase 2: Ubiquitous Remote

As the next step to just browsing metadata, the user will want to pour the information into any service, and invoke it. In this section, we show an example that the user operates home appliances with USF (fig. 4(b)). It is possible for USF to handle not only metadata for the real object, but also ones for electric data ². For example, if movie or music data are stored at a file server and they are accessible from User Agent, those icons will appear on USF. Also, if web and UPnP services are accessible from User Agent, those will appear as icons on USF. In practice, since UPnP services are available within a sub-domain, the accessible services are UPnP services detected by an UPnP Control Point in that sub-domain and web services in the internet. UPnP2WS Gateway also has the Control Point function as mentioned before. So if User Agent determines from the information of the RFID reader that the user who has USF is approaching a sub-domain, UPnP services detected by UPnP2WS Gateway will appear on USF.

For instance, when the user goes into the living room, movie data stored at a file server in the home network and a replay service provided by an UPnP-compliant HDD recorder will appear on USF. Then, if the user drag&drops a movie icon to the replay service icon, the actual movie is played by the HDD recorder. Further, if the user retires to his/her room, USF will show another playable service like a PC in the room, and the icon for the HDD recorder in the living room will disappear. Then, if the user drag&drops the movie icon to the new playable service, this time the movie is played by his/her PC ³.

¹ Note that there is an assumption here that the RFID tags are still active after purchases, and the tag information mainly annotated for SCM (Supply Chain Management) by manufacturers or distributors can be accessible from consumers. However, barcode DBs like UPC or EAN are accessible in fact.

² Needless to say, they are the metadata in its literal meaning. In this paper, we call explanatory data for the real object also as metadata.

³ In the current implementation, the actual movie data is copied by User Agent and directly sent to the playable service which is published as an UPnP service. If DLNA[8] will disseminate in the digital audiovisual appliances near future, streaming would also be possible.

3.3 Usecase 3: Service Finder

The above two cases illustrated introduction of USF as a browser and a remote. However, as the number of icons increase, the icons flood in the display of USF and the user will be confused the possible combination of the icons. Further, in the previous example data themselves were inputted into a service, but metadata can also be inputted to a service. In that case, the user needs to indicate which property of the metadata is inputted to which argument for service inputs. Therefore, the above mentioned Metadata Matching Engine will become necessary. In the followings, we illustrate: discovery and invocation of web services, coordination of web services and services provided by home appliances, and interactive combination of data and services.

In the previous example, the user intuitively found the combination of the movie data and the playable service, but it would not be so easy to find which service is combined with the beef icon. In such a case, if the user double-clicks the beef icon, USF will pick, for example, a food safety service from the crowd of services, and link the beef icon to it with a red arrow. To this end, firstly MME gets a pair of the property and the value like *name: US beef* in the beef metadata, then according to the mapping from metadata to ontology described in section 2.2 specifies a concept representing Meat in a food ontology stored in the ontology DB. In the same manner, metadata like *production region: Pennsylvania* and *process day: 29/12/2004* are used to specify Location and Time concept. Secondly, according to the discovery of the related services in section 2.2, MME determines that the semantical distance between the Meat and the Food which is a category concept given to the food safety service are close enough. Then, the beef icon and the food service are linked after checking if the inputs of the food service are filled by the beef metadata. That is, USF are digging and proposing the possible services combined with the beef on behalf of the user. Finally, when the user drag&drops the beef icon to the food service icon, the information of the beef will be given to a search form for the food service, and the detailed safety information will be displayed. In fact, MME determines the information about Location, Time corresponds to inputs concepts for the food search service. Then, each value (Pennsylvania, 29/12/2004, etc.) are given to the search form, and the result is shown in a text dialog on USF. Figure 4(c) shows the above flow, and fig. 5(1)-(4) shows the actual screen shots, although a beef is replaced with a pumpkin.

For the other example, when the user double-clicks a CD icon, USF finds the search service on Yahoo! Music via Music concept, then further the Text-to-Speech service provided by a home robot via Text Information concept (see fig. 5(5)). This is an example of the services sequence composed by repeating the discovery of the services possibly combined twice. The length of the sequence can be set at User Agent, normally it should be 2 or 3 due to computational time. Figure 4(d) shows this flow.

On the contrary, it is possible to combine the services step by step. For example, after the user gets the information about a wine by drag&dropping a wine icon to a wine information service, then the user can search for the



Fig. 5. Screenshots

next service combined with the wine information by double-clicking on the wine service. Further, if the user gets a map service from the wine service via Location concept and does not like it, double-clicking on the wine service again shows the next service such as a translation service and the previous TTS service. Figure. 4(e) explains this sequence.

Although all the above examples are for generic metadata for the object and the data, USF can handle metadata changing over time like MPEG-7, and find the different service depending on timing (scene) of the double-click during the play.

3.4 Additional Use

In this section, we show some of USF functions not mentioned before.

Metadata Search. In the previous section, we showed the examples that the services possibly combined with the object and the data are semantically searched. However, in the first place there is a case that the user wants to find icons for some specific objects, data, or services. For example, in a case of finding a particular author's book at a big bookstore or a library, it is impractical to check all the icons one by one. For such a case, we have provided the abstract icon. The abstract icon is a kind of folder predefined with search conditions.

By double-clicking it, the user can search metadata which satisfy the conditions from all the metadata currently detected. Setting of the conditions to the abstract icon is written in User Agent. In the above case, the user can prepare the abstract icon with the condition for the author and genre, and easily check existence and location of desired books by double-clicking it at the library.

In addition to this, it is possible to automatically add the user's context to the condition. The current implementation is only providing macros to get the user's current location and time as the context information, but using this information would be useful for finding the nearest shop to the current location, and so forth.

Script Definition. As already described, we also provides the script icon to execute the predefined flow. In a case that the user repeatedly uses the same services on a daily basis, the sequence of the services should be described in a script program at User Agent. In the script, a service definition can be written with a specific URL (binding), but it is also possible to define it with necessary metadata and get discovered the actual service at the invocation as described in section 2.2. In the internet, there are lots of web services which require the number of credit cards at the purchase. So that it would be useful for the user who occasionally buys something at a particular site to prepare a script which executes the necessary sequence by just drag&dropping the product metadata. Although the user can define my profile icon including name, address, phone numbers, etc. in USF, it would be safer to fill the important information like the card number into the script, and make it not readable.

Agent Mobility. In USF, User Agent is managing the user's conditions, scripts, and other preferences like history. So we have a plan to make User Agent mobile and follow the user to run on a home server when the user is in home, and migrate to the office when the user is at work. We have already developed lightweight mobile script system[9], and now merging it to MME.

4 Discussion

First of all, we locate USF in the ubiquitous computing research. We believe the user's icon choice in USF device represents the user's current interest, and it is like a snapshot of the user context. Then, USF is providing a mechanism to recommend the services according to the context. So that USF is regarded as so-called high-level service discovery, one of typical problems in this research area. Methodologies for service discovery in the ubiquitous computing have already been classified, however, at the same time it is well known that service discovery based on the context, that is, the high-level service discovery is still unsolved[10]. As a distinct feature of USF, it is exploiting ontologies rather than rules in lots of other systems to link the context to any service. Consequently, USF has some advantages like that reasoning process gets visible and it is easier to keep knowledge consistency, and recommendation gets faster, although the rules

are still necessary to make eccentric combination. As the other feature, USF is adopting a forward chaining style in the sense that the proposed service is what the user is inspired by something just attracted at that moment. We believe it is more casual case for people than purposive behavior (backward chaining style) like service composition by planning techniques. Also, this mechanism can be seen as tracking associative relations with ontology. So the user might be able to find surprising combinations of metadata and services, and use it not for pragmatic use.

Also, USF has adopted Service Oriented Architecture as its basic design. In contrast that CS system assumes the other party to connect from the design phase, SOA system in its principle is that service providers provide services to a public space, then service requesters search for and try to connect the services if necessary. In USF, when the user enters a particular area, he/she tries to utilize the objects and data available in that area combining the accessible services. Therefore, it would be sort of a right design decision to take SOA as its base. Then, web services has been taken as the richest and semantic web friendly framework.

On the other hand, some people might wonder from practical standpoint due to its semantic approach. However, we are not aiming at the very intelligent task, but rather simple task where hard-coding is impractical. The recent web browsers have a function to automatically fill the user's name and address in the corresponding part of html form. In fact, USF is extending it with ontology for more general service invocation. Further, semantic web is now actively investigated in the world, and application of metadata and ontology on the web is growing faster. Semantic web services is one of such activities, and WSDL 2.0 which is a de-fact standard for web services has already included mapping to RDF[11]. Also, the next version of UDDI is considering to adopt OWL[12] for category description. These movement will support the semantic use of USF.

Finally, we mentioned why we took a cellular phone as the device in section 1, but we do not adhere to it in the implementation, because the technical strengths in our system are converged at User Agent. So we have some plan to port the front-end application to other devices like TV, PC, car navigation system, and so forth.

5 Related Works

Sakamura et al.[6] and KDDI[7] have already developed mobile devices with RFID readers. However, either one is only for tag detection, and has no mean to connect web/UPnP services on the network. We will consider to port USF to those devices.

Aura (Advanced User Resource Annotation System)[13] project in MS Research is providing an application for a mobile device with a barcode reader, by which the user can scan the barcode on products, and search the related data in UPC database or Google and eBay. Besides, the individual user can annotate it on the web, and share with other users. However, there is no semantic aspect, and service discovery and combination are not considered.

Another similar approach to USF is Task Computing of Fujitsu[14]. This allows the user to selectively combine and invoke a sequence of web and UPnP

services on desktop PCs, and uses ontology to determine whether the combination is possible or not. Further, it is also possible from an arbitrary document to make a service which outputs the document. This is similar to our handling for metadata of the object and the data. However, mainly due to their different purposes, USF is focusing on the dynamic discovery of available services based on metadata retrieved in the ubiquitous environment, but Task Computing seems to emphasize to make the user build a sequence from a list of services on PCs in the office environment.

Further, CALI[15] of Nokia is a reasoning engine to semantically combine the user context written in DL with any services. They are using SIP for exchange of the context, and have a plan to implement on a cellular phone near future.

Finally, especially in Japan there are some internet services, which makes the user read QR code or barcode with cellular phones or Pocket PCs, and searches for its price in Amazon, or public reviews with RSS over the internet. However, those are also not considering semantics and web services invocation.

6 Conclusion and Future Works

In an industry segment it is expected that coordination among networked appliances in the home will become a big movement near future. Also, web services in the internet was hype, but is growing constantly. On the other hand, SCM by RFID is considered to be popular triggered by the admission of UHF tag. Also, as HDD recorders and digital music players like iPod get popular, annotation to digital data would make rapid progress.

In this circumstance, we developed USF aiming at simple coordination of objects, data, and services, and took up two drawbacks of our approach: discovery of services and inputs to the services. Then, we proposed metadata mapping and service combination using Metadata Matching Engine. Our current status is that we have just developed the prototype, and now have evaluation on precision ratio on the proposed services based on our ontologies and matching strategy response time at multiple accesses. As future works, we have a plan of public experiment with some target products within this year. We hope we will provide a value-added ubiquitous solution with semantics based on the result.

References

1. H. P. Huy, T. Kawamura, T. Hasegawa, How to make Web sites talk together - Web Service Gateway Solution, Proceedings of 14th International World Wide Web Conference (WWW 2005), 2005.
2. M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara, Semantic Matching of Web Services Capabilities, Proceedings of First International Semantic Web Conference (ISWC 2002), LNCS No. 2342, pp. 333-347, 2002.
3. T. Kawamura, J. D. Blasio, T. Hasegawa, M. Paolucci, K. Sycara, Public Deployment of Semantic Service Matchmaker with UDDI Business Registry, Proceedings of 3rd International Semantic Web Conference (ISWC 2004), LNCS 3298, pp. 752-766, 2004.

4. D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, K. Sycara, Bringing Semantics to Web Services: The OWL-S Approach, Proceedings of International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), 2004.
5. OWL Services, <http://www.daml.org/services/owl-s/>.
6. Ubiquitous Communicator, <http://www.ubin.jp/press/pdf/TEP040915-u01e.pdf>
7. http://www.kddi.com/corporate/r_and_d/kaihatsu/ (in Japanese)
8. Digital Living Network Alliance, <http://www.dlna.org>
9. K. Ueno, T. Kawamura, T. Hasegawa, A. Ohsuga, M. Doi, Cooperation between Robots and Ubiquitous Devices with Network Script "Flipcast", Proceedings of Network Robot System: Toward intelligent robotic systems integrated with environments (IROS 2004 Workshop), 2004.
10. Hetal S., Standards for service discovery and delivery, IEEE Pervasive Computing vol. 1, no. 3, 2002.
11. Resource Description Framework, <http://www.w3.org/RDF/>.
12. Web-Ontology Working Group, <http://www.w3.org/2001/sw/WebOnt/>.
13. Annotate the Planet, <http://aura.research.microsoft.com>
14. R. Masuoka, B. Parsia, Y. Labrou, E. Sirin, Ontology-Enabled Pervasive Computing Applications, IEEE Intelligent Systems, vol. 18, no. 5, pp. 68-72, 2003.
15. D. Khushraj, O. Lassila, CALI: Context-Awareness via Logical Inference, Proceedings of Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications, 2004.