Location-Mediated Service Coordination in Ubiquitous Computing

Akio Sashima
CARC, AIST / CREST, JST
2-41-6 Aomi, Koto-ku, Tokyo, 135-0064, Japan
+81-3-3599-8227
sashima@carc.aist.go.jp

Noriaki Izumi
CARC, AIST / CREST, JST
2-41-6 Aomi, Koto-ku, Tokyo, 135-0064, Japan
+81-3-3599-8297
niz@ni.aist.go.jp

Koichi Kurumatani
CARC, AIST / CREST, JST
2-41-6 Aomi, Koto-ku, Tokyo, 135-0064, Japan
+81-3-3599-8223
kurumatani@socsys.org

ABSTRACT

We propose location-mediated service coordination in ubiquitous computing. In the coordination, middle agents determine best-matched services for a user by considering the user's location based on location-ontology for ubiquitous computing. Introducing such location-aware middle agents and location-ontology into ubiquitous computing, we can extend application areas of software agents from the Internet to the real world. In this paper, we first illustrate the idea of location-aware middle agents and location-ontology. Second, we describe a multiagent architecture, called CONSORTS, as an implementation of the agents. In order to bridge the gap between device-oriented physical information in the real world and web-based conceptual information in the digital world, CONSORTS agents can translate sensor-based raw representation of the locations into conceptual one. Finally, we describe two applications of CONSORTS, an intelligent information assist system at a museum and wireless-LAN based location system.

Keywords

Web Agents, Semantic Web, Ubiquitous Computing, Service Coordination

1. INTRODUCTION

Two intelligent service frameworks, Ubiquitous (or Pervasive) computing [25] and Semantic Web[2], received much interest in both research and application in the past years. In order to improve the current web-based service frameworks, the Ubiquitous computing handles device-oriented physical information; the Semantic Web handles ontological information.

Ubiquitous computing enables computers to be aware of the context of users in the real world. The ubiquitous computing environment has numerous sensor devices and computers connecting to the networks. In the framework, the computers manage not only explicit information of the user interaction but also physical information from sensors in the real world. For example, the intelligent guide systems in a museum should manage both location and commands of the users in order to interactively offer information about the exhibitions there.

On the other hand, the Semantic Web enables computers to read the web contents by adding meta-data to the documents. The meta-data are developed on XML. One of such meta-data frameworks is RDF (Resource Description Framework) [18] and RDFS (RDF Schema) [19]. In this framework, the computers manage not only information explicitly described in the content but also ontological information of the content, which are shared with the computers on the Internet. For example, the computers handle the meta-data and the domain ontologies to understand the contents meanings, such as identifying "my father's father" with "my grand father" in a document.

Although there are many researches involved in the Semantic Web and the Ubiquitous computing, still few researches [8] [20] try to bridge the gap between device-oriented physical information in the Ubiquitous computing and web-based ontological information in the Semantic Web. In order to realize intelligent information services for our daily activities (e.g. shopping assistants, museum guide, etc.), we need a service framework that can seamlessly manage both of the information because our activities are comprehensible within the physical and ontological context.

For such seamless service framework, we have been developing a multiagent architecture named CONSORTS (Architecture for cognitive resource management with physically-grounding agents) [5]. In the architecture, middle agents, *location-aware middle agents,* can handle physical and conceptual information of environments, and determine the best-matched service according to the user's location. Bridging the gap between device-oriented physical information in the real world and web-based conceptual information in the digital world, CONSORTS agents can translate sensor-based raw representation of the locations into conceptual one.

In this paper, we describe a framework of service coordination mechanism in Ubiquitous computing. We first illustrate the idea of location-aware middle agents and location-ontology. Second, we describe an implementation of CONSORTS architecture. Finally, we describe two applications of CONSORTS, an intelligent information assist system at a museum and wireless-LAN based location system.

2. LOCATION-MEDIATED SERVICE COORDINATION

Although the ubiquitous computing is a promising framework of the intelligent services, it is still premature and has many problems. One of the problems is about coordination among services. In a ubiquitous computing environment, a lot of services are usually co-located (e.g. navigation, guide, advertising, information retrieval, controlling devices, etc.). When numerous services can be simultaneously provided to a user, a service coordination is required to provide proper services then and there. How can we realize such a service coordination in ubiquitous computing?

One possible answer is to introduce middle agents [21]. Middle agents are matchmaker agents that match service agents with service requester agents. The middle agents mediate the communication between the agents to achieve the good performance on the Internet. For example, Paolucci et al. [17] show the potential of the middle agents by the orchestration of Web Services [24].

This middle agents work efficiently on the Internet, but it is not sufficient for the middle agents in the Ubiquitous Computing. Most researches focus on the problem how the software agents properly serve the user who operates a computer terminal connected to the Internet [13][16]. However, the middle agents in the Ubiquitous Computing must handle the information of the location where service providers and service requesters communicate with each other. If a service is semantically matched to a request but the service resource is far removed from the requesting user, the service should not be provided in the context of Ubiquitous computing; the nearer service resource to the user is often better than any other services.

2.1 Location-Aware Middle Agents

In order to realize the service coordination suitable for ubiquitous computing, we propose *location-mediated service coordination*, and introduce new type of agent, *spatio-temporal reasoner*. The spatio-temporal reasoner manages sensor-based raw information and conceptual information of the environment, such as physical locations of objects, and reasons of their spatial relations1. Using the spatio-temporal reasoner, we extend the conventional middle agent framework to realize the location-mediated coordination in the Ubiquitous computing.

The location-mediated coordination process consists of following agents:

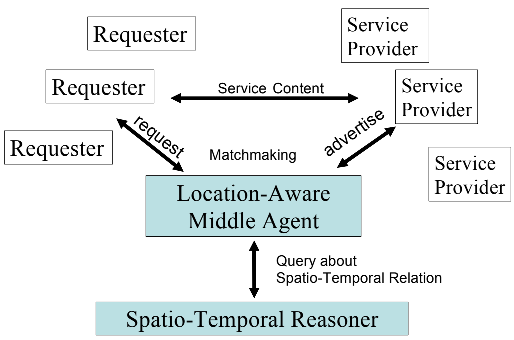

Figure 1. Location-mediated service coordination

[1] To track the object's location, we assume using sensor devices (e.g. RF-ID tags, GPS, etc.) and their information are available.

- **Service Provider Agent**: the agent that provides some types of service, such as problem solving, mediating Semantic Web services;
- **Service Requester Agent**: the agent that needs provider agents to perform some services for them;
- **Location-Aware Middle Agent**: the agent that helps locate other agents along with Spatio-Temporal Reasoner;
- **Spatio-Temporal Reasoner**: the agent that manages physical locations of Service Providers and Service Requesters, and reason about their spatial relation in cognitive way.

The coordination is as follows:

1) provider agents advertise their capabilities and physical service areas to middle agents;
2) middle agents store these advertisements, and inform their service areas to spatio-temporal reasoners;
3) spatio-temporal reasoners store these service areas;
4) a requester asks some middle agents whether they know providers with desired capabilities;
5) the middle agents ask some spatio-temporal reasoners whether the stored service areas cover the requester's location or not, and get the results, a subset of the stored service areas;
6) taking account of the results, the middle agent matches the request with the stored advertisements, and returns the result, a subset of the stored advertisements (see Figure 1.).

In this framework, the descriptions of location are separately managed in the spatio-temporal reasoners, so service contents modification is not required even if the service areas are changed. For example, the content for an information service in a museum can be used in exhibitions in other museums, by only adjusting the location information to the museum.

Additionally, this process is a natural extension of matching process proposed by Sycara et al.[21]. By integrating above agents with ordinal web agents on the Internet, we can extend application areas of Web agents (and Web Services) to ubiquitous computing environment.

3. MODELING SPACE IN COGNITIVE WAY

3.1 Conceptual Representation of Locations

To realize the location-mediated service coordination, the agents must share representations of locations (e.g. service area, the requester's location, etc.). Moreover, the representation must be human-friendly, conceptual ones. For example, to inform the user's location to others, the agents should understand "the user was in a museum on the morning", rather than "the user was at Longitude: 140.38.54 E, Latitude: 35.77.44 N at Mon Jan 13 07:47:06 2003 JST". Such human understanding of spatial concept is generally called mereological thinking [23], reasoning about part-of relation. To realize this kind of spatial reasoning as a basic function of our agent architecture, especially of Spatio-temporal reasoner, we model ubiquitous computing environment with a mereological, tree representation. The representation consists of *part-of* relations among 4-dimentional spatio-temporal regions in the real world.

3.1.1 Location and Region

First, we formularized locations of objects with the spatio-temporal regions using the formalization by Bittner [3]. The formalization is as follows.

The symbol o ($o \in O$) denotes an object, and the domain of objects, O. The symbol $\boldsymbol{x}$ ($x \in X$) denotes a spatio-temporal region, and the domain of the regions, X. The domain of the regions consists of 3-dimentional (spatial) regions, x^s, and 1-dimenational (temporal) region x^t. An object $\boldsymbol{o}$ is exactly located at the region $\boldsymbol{x}$ is formalized as follows;

$$\forall o \in O : \exists x^s \in X : L_\tau(o, x^s)$$

where the symbol τ denotes an instance of time. Weaker definition of object's location is as follows:

$$L(o,x^t,x^s) \equiv$$
$$\exists x' : x' \le x^t \wedge \forall \tau \in x' : \exists x : L_\tau(o,x) \wedge x \le x^s$$

where the symbol $\le$ denotes the part-of relation of the domains.

Hence, *L (John, his room, this morning)* means "*John was in his room in this morning*".

3.1.2 Tree Representation

We formularize the ubiquitous computing environments as a tree-structure of spatio-temporal regions based on Bittner [3]. A tree structure of spatio-temporal regions is denoted by G : $G = (R, \subseteq)$ where $R : (r \in R)$ denotes a set of spatio-temporal regions, $\subseteq$ denotes a part-of relation between two regions. The regions in G have following natures.

$$1)\ \exists r_i : r_i \subseteq r_i$$
$$2)\ \exists r_i, r_j : r_i \subseteq r_j \wedge r_j \subseteq r_i \rightarrow r_j = r_i$$
$$3)\ \exists r_i, r_j, r_k : r_i \subseteq r_j \wedge r_j \subseteq r_k \rightarrow r_i \subseteq r_k$$

where r_i, r_j, r_k is a region.

3.2 Mereological Reasoning

Now, we can infer and query the object's location using this notation. For example, given two statements: "John was in his room in this morning", and "His room is part of this building", we can infer "John was in this building in this morning", we can infer the "Jhon's location" as follows.

$$L(\text{John, his room, this morning}) \wedge$$
$$\text{his room} \subseteq \text{this building}$$
$$\rightarrow L(\text{John, this building, this morning})$$

Similarly, we can define the users that can receive the service in ubiquitous computing environment, denoted by C, as:

$$C =$$
$$\bigcup\{\exists Person : L(Person,\ anArea,\ aPeriod)\}$$

By using this kind of logical inference, Spatio-Temporal Reasoner can reason about their spatio-temporal relation. Replying results, a subset of the stored service areas, denoted by SR, Spatio-Temporal Reasoner can reason as

$$SR =$$
$$\bigcup\{\exists r_{service} : L(person, r_{service},\ aPeriod)\}$$

where $r_{service} : r_{service} \in R_{service}$ denotes a service area of a set of service areas in the environment.

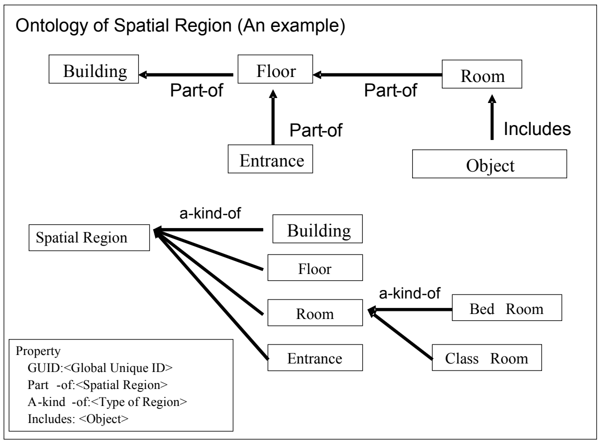

Figure 2 A part of location ontology

3.3 Location Ontology

Based on the above formalization, we define a location ontology for the ubiquitous computing. Figure 2 shows the location ontology for describing spatial region. We define similar type of ontology for describing temporal region. We define the relation of spatial regions, and their property in the ontology. Part-of relation in the figure is the relation denoted by $\subseteq$ in before section. The agents can describe object's location, and reason about spatial-relation based on the ontology.

3.4 Spatio-Temporal Reasoner

Spatio-Temporal Reasoner manages "spatio-temporal inference engine" and "spatial information repository". The repository manages the device-oriented physical information of the ubiquitous computing environments, such as location of users and surrounding sensor devices. Location data are constantly updated, given by "device wrapper agents".

Spatio-temporal reasoner also manages conceptual representation of the environment, tree-structure (G in Section 3.1.2). The middle agent and various service agents, such as navigation planning, searching the nearest device to the user, use the tree-representation of environment, and do not use the device-oriented raw representation. It means that the reasoner hides such complex representations from other agents and humans.

Hence, one of important function of spatio-temporal reasoner is bridging the gap between sensor-based information in the real world and cognitively comprehensive annotated information in the digital world (see Figure 3.). For the translation between two representations, we define static mapping functions beforehand.

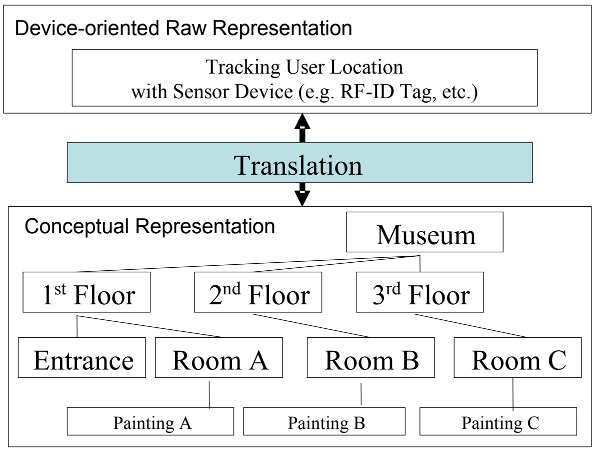

Figure 3 Data representations of spatio-temporal reasoner

4. LOCATION-AWARE MIDDLE AGENT

Location-aware middle agent is an agent that help locate other agents along with Spatio-Temporal Reasoner The middle agent orchestrates the various services of agents that manage sensor information of the physical environment and ontological information of the web services. Although the middle agents control the matchmaking process, the spatio-temporal reasoning is outsourced to Spatio-Temporal Reasoner.

4.1 Matchmaking

The location-aware middle agent matches the request against the stored advertisements with taking account of the results from Spatio-Temporal Reasoner. In our prototype system, we tentatively define the matchmaking process as following expression.

$$S_i = argMax_{i=1,2..n}\{u(S_i)\}$$
$$u(S_i) = \lambda f(SC_i) + (1.0 - \lambda) g(SR_i)$$
$$(0.0 \le \lambda \le 1.0)$$

Where S_i denotes one of available services in current situation. $u(S_i)$ represents a utility function of the service S_i . λ denotes a constant value for a normalized weight to matching degree of contents for utility function $u(S_i)$. $f(SC_i)$ represents a matching degree of a service content SC_i . $g(SR_i)$ represents a matching degree of service area SR_i . The middle agent selects the most suitable service which has the maximum utility value among the services. This coordination scheme is a basic function in We can implement complex services, such as service composition, as facilities of the middle agents.

Because the matchmaking process that we proposed is an extension of the ordinal matchmaking process among agents on the Internet, the process model of DAML-S[7] can be apply to the description of service content. Therefore, we omit the descriptions for service contents, and their matchmaking process $f(SC_i)$ in this paper. We concentrate to describe the location based matchmaking process $g(SR_i)$.

4.2 Location-based Matchmaking

Location-based matchmaking process has two methods: the method that calculates matching degree by *part-of* relations, and by *a-kind-of* relations. Each of two functions calculates the matching degrees between services and a user.

4.2.1 Matching Degrees by Part-of Relations

Each of services has a service area that corresponds to a node of the tree representation of the environment. This matching method calculates the matching degrees based on part-of relations of spatio-temporal regions, which include the service areas, or the requester are located at. At first, the agent filters the service areas where the requester is located, then, calculates matching degrees of each service areas.

The matching degrees of the service areas are discrete values corresponding to the path lengths from root node to the nodes of the service areas on a *part-of* relation tree. In the tree representation in Figure 2, if the user is located at "Entrance", the matching degree is 2. In other words matching degree of narrower region has high value (e.g. g(*Room A*) >g(*1st Floor*) in Figure 3).

4.2.2 Matching Degrees by a-kind-of Relations

This matching method enables the requester to access the information generally related to facilities of the place. For example, when a user is located at a library, the user would like to access sorts of book guidance services. This service is not related to the physical location of the library but related to the cognitive facilities of the library. This kind of location-mediated coordination is suitable for assist services in a lot of places where human activities take place (e.g. station, library, museum, etc.).

The matching degrees are calculated based on *a-kind-of* relations. When the service providers and the requester are located at the different place, a-kind-of relations are used. At first, the agent looks at the a-kind-of property of the region where requester is located, and pick up the services that have the same a-kind-of property. Then, the agent calculates the matching degree of each service provider.

The matching degrees of the service areas are discrete values corresponding to the path lengths from root node to the nodes of the service areas on a *a-kind-of* relation tree. In the tree representation in Figure 2**,** if the user is located at "Bedroom", the matching degree is 2. Matching degree of narrower regions have high values (e.g. g(*Bedroom*) >g(*Room*) in Figure 2).So the user receives the information services related to bedrooms rather than rooms.

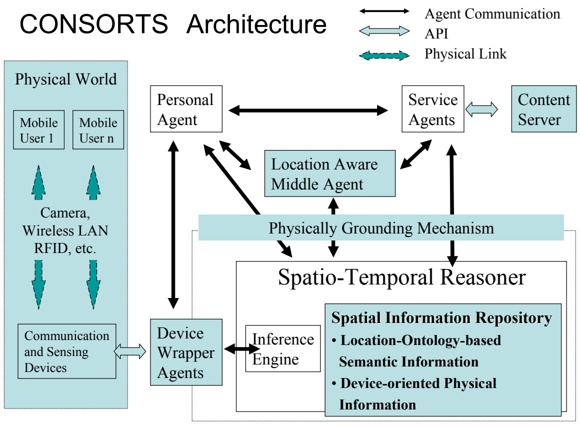

Figure 4 Outline of CONSORTS architecture

5. IMPLEMENTATION

In order to realize human-centered intelligent information services in ubiquitous computing, we have been developing CONSORTS (Architecture for cognitive resource management with physically-grounding agents), an agent-based software architecture that mediates the real world and a digital world, for assisting human in accessing intelligent information services. In this section, we describe the CONSORTS architecture as an implementation of the location-aware middle agents, and confirm the functionality of the middle agents in context-aware information assist service in a museum.

5.1 CONSORTS Architecture

The key concepts of CONSORTS are "physically-grounding" and "cognitive resources"[15]. By using sensory information brought by ubiquitous environment, agents have a grounding in physical world, and manage physical resources (especially spatio-temporal resources) in a cognitive way, i.e., they can recognize, reorganize, and operate raw physical resources as conceptual resources.

Figure 4 shows an outline of CONSORTS Architecture, which consists of following type of agents.

- ***Services Agents*** (Service Provider Agent) provide services, such as tour guide, navigation, and information presentation. Services agents realize the flexible service by using semantic information through the content servers. The content servers are resources on the Internet, especially Semantic Web and Web Services.
- ***Personal Agents*** (Service Requester Agent) request service behalf of a user. The agents communicate with Device wrapper agents, and manage them as user interface devices.
- ***Location-aware Middle Agents*** manage to interaction to mediate between agents and a user. The agent sends the request to the middle agents, and controls the interaction between service agents and the user. In current version, "Matching by *a-kind-of* Relation" is not implemented.
- ***Spatio-Temporal Reasoners*** manage physical locations of Service Providers and Service Requesters, and reason about their spatial relation in cognitive way. The agents manage both device-oriented physical information in the real world and cognitively comprehensive information in the digital world
- ***Device Wrapper Agents*** hide the diversity of physical device and legacy application. Device Wrapper Agents realize device-independent applications by wrapping the raw information, which is derived from legacy devices and applications, with the standardized agent communication language.

5.2 Context-Aware Information Assist Service in a Museum

Based on the CONSORTS architecture, we have implemented context-aware information assist service in a museum. In this system, the agents are aware of the distance between a user and paintings in the museum. The service system is that when the user approaches a painting, the agents automatically provide information of the painting via the user's portable display device.

In the museum, user can access general museum information based on the procedure of *matching degrees by a-kind-of Relations*. In addition, when a user is located near a painting, the user can receive the information of the painting via the user's portable display device from the agents. This service is based on the procedure of *matching degrees by part-of Relation.* Since, the information of the painting is derived from the Internet resources (e.g. Web services, Search Engine, etc.).For example, if the user needs more information of the painting, the user should push the "tell me" button of the portable device. The CONSORTS agents notice the user's request, make a search query about the painting with the necessary information they've already had, e.g. the painting's name and user's preference, and access the internet search engine, e.g. Google Web Services [10], behalf of the user.

Figure 5 shows a snapshot of the monitor display of the system. You can see a museum map in a main window, two information windows with some pictures, and a message window displaying agent communication. In the map, human icons represent current locations of the users; the yellow zones represent service zones of the museum service agents; the blue lines represent users' trajectories. The information windows correspond to the screen of users' portable devices. In this demonstration program, the users randomly roam in the museum. If a user enters the yellow zone, an information window pops up in the screen to display the picture information.

Currently, users, users' behavior, and a museum in this system are simulated ones. However, we have designed the device wrapper agent to hide the diversity of the devices API, we can integrate the physical devices into the system on the CONSORTS.

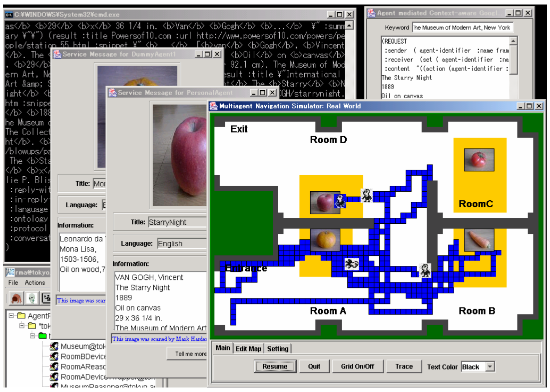

Figure 5 Snapshot of the assistance system at the museum

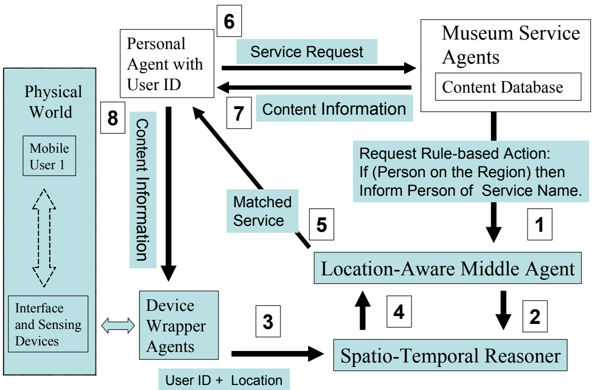

Figure 6 Message flows in the assistance system at the museum

Figure 6 outlines the agent communication in the museum system. First, the service agent requests the middle agent to continuously infer whether users are located on a service region associated with a service name ([1] in Figure 6). The middle agent requests the Spatio-Temporal Reasoner to request to inform when users are located on a service region ([2] in Figure 6). On the other hand, the device wrapper agent informs the spatio-temporal reasoner where a user with unique id is located ([3] in Figure 6). When the spatio-temporal reasoner notice a user is located on a service region, the reasoner informs the middle agent of the user id ([4] in Figure 6). The middle agents tell the personal agents of the user available service name ([5] in Figure 6). Then, the personal agent requests a service based on the service name ([6] [7] in Figure 6). Finally, the personal agent requests the device wrapper agent to show the service content on a user's portable display device ([8] in Figure 6).

Table 1 A message in the assistance system at the museum

```
(REQUEST
 :sender      PersonalAgent
 :receiver    MuseumAgent
 :content     "((action
                (agent-identifier
                        :name MuseumAgent)
               (provide
               (person :name PersonalAgent)
               (service :name Monalisa
                   :provider MuseumAgent)))"
  :language   fipa-sl
  :ontology   location-ontology
)
```

All messages of the agent communication are described with FIPA-ACL (Foundation for Intelligent Physical Agents Agent Communication Language)[9]. The message content is described with FIPA-SL. Table 1 shows a message which represents "Personal Agent request to Museum Service Agent to provide the service named Mona Lisa (showing information about the painting of the Mona Lisa)." This message flows at arrow-line [6] in Figure 6. Although we have already implemented RDF-based descriptions as a content part of the message, we adopt FIPA-SL as the content language for interoperability with other agent platform in this system. We are also planning to adopt DAML families [11][6] in the near future. We have implemented the CONSORTS agents using JADE [12], a software framework to develop the agent system that conforms to FIPA specifications. Using standardized ACL like FIPA-ACL, the system can connect to open agent systems, such as AgentCities [1]. Hence, using the CONSORTS architecture, FIPA-agents can share the physical information and devices in the museum, and open the possibility of new services which use the physical resources over the world.

In this demonstration system, we have shown the potential of the distributed open agent systems that can bridge device-oriented physical information in the real world and web-based conceptual information in the digital world.

5.3 Wireless-LAN based Location System

Based on the CONSORTS architecture, we have implemented Wireless-LAN (IEEE 802.11b) based Location System. In this system, the agents are aware of user location by watching the status of the wireless-LAN stations. The service system is that when the PC or PDA with a user connects with one of the stations, the system detects the connection, and store a MAC address of the network card of the PDA with physical location of the station. Because a MAC addresses is a globally unique, the system can track the location of the card (with the user) globally.

Figure 7 shows an outline of the location system. In the system, Web services, which provide location information like sensor devices, are integrated with a FIPA-agent platform, and the location information is shared with FIPA-agents.

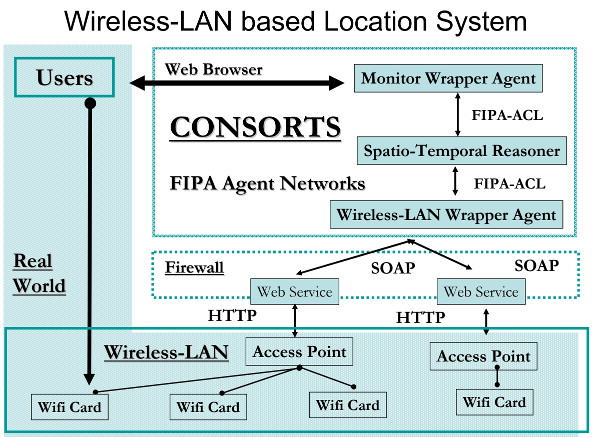

Figure 7 Outline of wireless-LAN based location system

We have experimentally confirmed the functionality with real wireless-LAN stations that located on various cities, (e.g. Barcelona, Tokyo, etc.) in the world. Figure 8 shows a snapshot of the monitor window of the location system in that experiment.

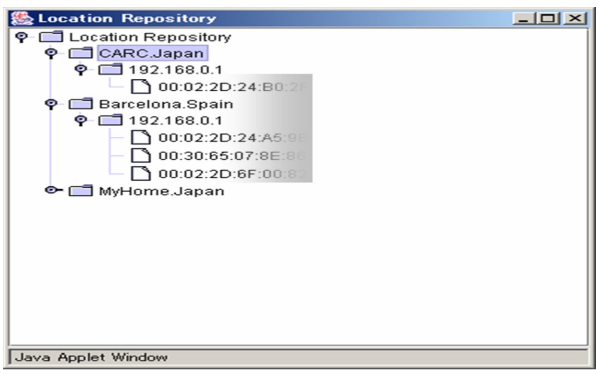

Figure 8 Snapshot of the location repository

In this experiment, we have shown the potential of the distributed open agent systems that can access and unify the real world information covering all over the world.

6. RELATED WORK

CoolAgent [4] is a context-aware software agent system. The agents can share the context information described by RDF that can be an ontology description framework for the Semantic Web. However, CoolAgent system does not provide the design framework of ubiquitous agents, such as location-aware middle agent. In this paper, we formally describe the design pattern of the coordination of ubiquitous agents.

There are some researches for middle agents[21][22][17]. As we described before, the work in this paper is extension of their works in ubiquitous computing. Hence, their results, such as integration of DAML-S, must be useful for our future research.

7. FUTURE WORK

As future work, we are planning to introduce the concept of "Mass User Support"[15]. Existing context-aware applications provide information for a single user. However, if the applications provide some users with the same information at the same time, serious conflict with users' behavior, such as a long queue formed in front of a tourist attraction, may be aroused. In that situation, the total performance to assist users' activities will decrease. Thus, we are planning to realize the service which has facility to resolve the conflict with multi users' requirements and to implicitly modify the providing information for each user in order to maintain the performance of the whole system. We are implementing the facility using market mechanism, called "user intention market."[14]

8. CONCLUSION

In this paper, we described a framework of service coordination mechanism in Ubiquitous computing. We first illustrated the idea of location-aware middle agents and location-ontology. Second, we described an implementation of CONSORTS architecture. Finally, we described two applications of CONSORTS, an intelligent information assist system at a museum and wireless-LAN based location system.

In our demonstration systems, we have shown the potential of the distributed open agents in the real world. In the Semantic Web[2], an "agent" is defined as '*the program that collects information from diverse sources, processes the information and exchanges the results with other programs.*' We have realized such agents in the context of the Ubiquitous computing, and extended application areas of the agents from the Internet to the real world.

9. REFERENCES

[1] AgentCities Web, http://www.agentcities.org/, 2003

[2] Berners-Lee, T., Hendler, J. and Lassila, O., the Semantic Web. *Scientific American*, 2001.

[3] Bittner, T., Reasoning about qualitative spatio-temporal relations at multiple levels of granularity. In F. van Harmelen (ed.): ECAI 2002. *Proceedings of the 15th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, 2002

[4] Chen, H. and Tolia, S., Steps towards creating a context-aware agent system. *TR-HPL-2001-231*, HP Labs, 2001.

[5] CONSORTS Architecture Web, http://consorts.carc.jp/, 2003

[6] DAML+OIL (March 2001) Reference Description, http://www.w3.org/TR/daml+oil-reference, 2002

[7] DAML Service, http://www.daml.org/services/, 2002

[8] Finin, T. and Perich, F., editor. *Proc. of the AAMAS Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices*, 2002

[9] The Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org/, 2002.

[10] Google Web APIs, http://www.google.com/apis/, 2003

[11] Hendler, J. and McGuinness, D. L., DARPA Agent Markup Language (DAML). *IEEE Intelligent Systems*, 15 (6), 72–73, 2001.

[12] Java Agent DEvelopment Framework (JADE), http://sharon.cselt.it/projects/jade/, 2002

[13] Jennings. N., An agent-based approach for building complex software systems. *Communications of the ACM*, 44 (4): 35-41, 2001.

[14] Kurumatani. K., User Intention Market for Multi-Agent Navigation - An Artificial Intelligent Problem in Engineering and Economic Context; Kurumatani, K., Chen, S.-H.,Ohuchi, A. (eds.), *Proceedings of The AAAI-02 Workshop on Multi-Agent Modeling and Simulation of Economic Systems, Technical Report WS-02-10*, pp.1-4, AAAI Press, 2002.

[15] Kurumatani. K., Social Coordination in Physically-Grounded Agent Architecture, to appear in *Proceedings of Landscape Frontier International Symposium*, 2002.

[16] Maes, P., Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7): pp.31-40, ACM Press, July 1994.

[17] Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K., Semantic Matching of Web Services Capabilities. In *Proceedings of the 1st International Semantic Web Conference (ISWC2002)*, 2002

[18] Resource Description Framework (RDF), http://www.w3.org/RDF/, 2002

[19] RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema/, 2002

[20] Sashima, A., Kurumatani, K., and Izumi, N., Physically-Grounding Agents in Ubiquitous Computing, *Proc. of Joint Agent Workshop* (JAWS2002) , pp.196- 203, 2002.

[21] Sycara, K., Decker, K., and Williamson, M., Middle-Agents for the Internet, *Proceedings of IJCAI-97*, January 1997.

[22] Sycara, K., Klusch, M., Widoff. S., and Lu, J., Dynamic Service Matchmaking among Agents in Open Information Environments, *SIGMOD Record (ACM Special Interests Group on Management of Data)*, Vol. 28, No. 1, 47-53. , March, 1999

[23] Varzi, A., C. and Casati, R., Parts and Places. The Structures of Spatial Representation, Cambridge, MA, and London: MIT Press, 1999.

[24] Web Services Activity, http://www.w3.org/2002/ws/, 2002.

[25] Weiser, M., The computer for the 21st century. *Scientific American*, 94-104, Sep 1991.