International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:8, 2007

Context Modeling and Reasoning Approach in Context-Aware Middleware for URC System

Chung-Seong Hong, Hyung-Sun Kim, Joonmyun Cho, Hyun Kyu Cho, and Hyun-Chan Lee

***Abstract*—**To realize the vision of ubiquitous computing, it is important to develop a context-aware infrastructure which can help ubiquitous agents, services, and devices become aware of their contexts because such computational entities need to adapt themselves to changing situations. A context-aware infrastructure manages the context model representing contextual information and provides appropriate information. In this paper, we introduce Context-Aware Middleware for URC System (hereafter CAMUS) as a context-aware infrastructure for a network-based intelligent robot system and discuss the ontology-based context modeling and reasoning approach which is used in that infrastructure.

***Keywords*—**CAMUS, Context-Aware, Context Model, Ontology.

I. Introduction

THE general trend of computing is progressing towards the vision of ubiquitous computing, in which devices are seamlessly integrated into the life of everyday users, and services are readily available to users anywhere they go at any time [1]. In the vision of ubiquitous computing environment, computer system understands their situational contexts and provides appropriate services to users [2]. To realize this vision, it is important to develop a context-aware infrastructure which can help ubiquitous agents, services, and devices become aware of their contexts because such computational entities need to adapt themselves to changing situations [3].

Context is any information that can be used to characterize the situation of an entity. An entity includes a person, a device, a location and a computing application [4]. For example, environmental attributes such as noise level, light intensity, temperature, and motion, system and device capabilities, available services, and user's intention could be context. A context-aware system uses such contexts to provide relevant information and/or services to the user depending on the user's task [4].

In contrast to the traditional computing paradigm, context-aware computing progresses towards an open and dynamic environment [5]. In an open and dynamic environment, contexts are often distributed in heterogeneous sources and context-aware infrastructure is required to be capable of dynamically creating a customized application by composing the right components based on the context. To realize it, a context-aware infrastructure manages the context model representing contextual information and provides appropriate information based on this model when needed. This paper introduces CAMUS which is a context-aware infrastructure for a network-based intelligent robot system and the ontology-based context modeling and reasoning approach which is used in it.

The remainder of the paper is organized as follows. Section 2 discusses related researches about context-aware infrastructure and context modeling approach. Section 3 describes the concept of the Ubiquitous Robotic Companion (hereafter URC) system which is background of our research. In section 4, we introduce our developed context-aware infrastructure, CAMUS, and its components. Section 5 describes ontology based context modeling and reasoning approach in CAMUS and shows some implementation results. Finally, in section 6, we conclude this paper with some remarks.

II. Related Works

Building context-aware systems is a complex and time-consuming task due to lack of an appropriate infrastructure or middleware-level support. An appropriate infrastructure for a context-aware system should provide support for most of the tasks involved in dealing with contexts.

A number of research groups have developed context-aware infrastructures. Xerox's Palo Alto Research Center has been working on pervasive computing applications since the 1980s. Carnegie Mellon University's Human Computer Interaction Institute (HCII) worked Aura project whose goal is to provide each user with an invisible halo of computing and information services that persists regardless of location [6]. HP's Cooltown project has developed technology future where people, place, and things are first class citizens of the connected world, a place where e-services meet the physical world. In this place, humans are mobile, devices and services are federated and context-aware, and everything has a web presence [7]. The Massachusetts Institute of Technology (MIT) has a project called Oxygen [8]. The Oxygen system aims to bring an abundance of computation and communication to users through

Manuscript received October 14, 2007. This work was supported by the IT R&D program of MIC/IITA. [2006-S-026-02; Development of the URC Server Framework for Proactive Robotic Services]

Chung-Seong Hong is a corresponding author and with the Electronics and Telecommunications Research Institute, Daejeon, 305-700 South Korea (corresponding author; phone: 82-42-860-1876; fax: 82-42-860-1566; e-mail: cshong@etri.re.kr).

Hyung-Sun Kim, Joonmyun Cho and Hyun-Kyu Cho are with the Electronics and Telecommunications Research Institute, Daejeon, 305-700 South Korea (e-mail: {kimhs, jmcho, hkcho}@etri.re.kr).

Hyun-Chan Lee is with the Hongik Univ., Seoul, 121-791, South Korea (e-mail: hclee@hongik.ac.kr).

natural spoken and visual interfaces, making it easy for them to collaborate, access knowledge, and automate repetitive tasks. The Gaia project [9] developed at the University of Illinois is a distributed middleware infrastructure that provides support for context aware agents in smart spaces. UMBC developed a broker-centric agent architecture (CoBrA) [10] to provide runtime supports for context-aware systems in an Intelligent Meeting Room environment. The design of the CoBrA architecture is aimed to enable distributed context-aware agents to contribute to and access a shared model of the context. Also the COBRA architecture allows users to control the access of their personal information in a context-aware environment.

Context modeling approaches are classified by the scheme of data structures which are used to exchange contextual information in the respective system [11]. In various kinds of approaches, ontology-based approach such as CONNON [12] and CoBrA system [13] has a number of advantages of expressiveness, knowledge sharing, logic inference, knowledge reuse, and extensibility.

CAMUS mainly focuses on software architecture for context-aware ubiquitous computing environments. We took into account general considerations for the development of the architecture, including flexibility, scalability and reusability. Especially, it was designed to overcome limitations of the ubiquity, context-awareness, and intelligence that existing mobile service robots have. Also, in CAMUS, we use OWL (Web Ontology Language) [14] and Jena [15] application development toolkit to represent and manipulate contexts. In this paper, we discuss several main technical issues on context-aware infrastructure development and ontology-based context modeling and reasoning approaches for the network-based robotic system.

III. URC System for the Network-Based Robot

Generally, the robot has three functional components of sensing, processing and acting. Recently, the Korean government has strategically promoted the development of a new concept of the network-based intelligent robot, which is called by URC. The main approach of URC is to distribute these functional components through the network and to fully utilize external sensors and processing servers as shown in Fig. 1.

The URC system requires not only the hardware infrastructure such as ubiquitous sensor networks and high-performance computing servers, but also the software infrastructure for context-aware applications development and execution. We developed a context-aware infrastructure which is called CAMUS.

In order to realize this concept of a network-based service robot system, it is essential for a robot which is minimally equipped with its internal abilities to give users intelligent services by utilizing central servers. When a user requests a service, it should be able to provide appropriate services by recognizing current contexts which are given by the central servers. Moreover, it should be able to proactively provide necessary information and services even though there are no requests from the user.

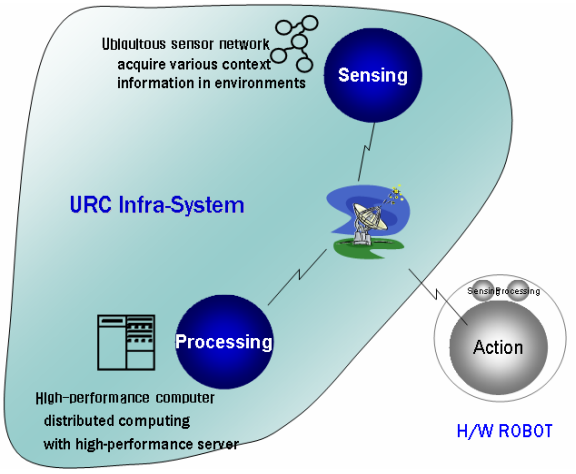

Fig. 1 The concept of the URC system

IV. A Context-Aware Infrastructure: CAMUS

CAMUS is a context-aware infrastructure for URC system [16]. Fig. 2 depicts the conceptual architecture of CAMUS.

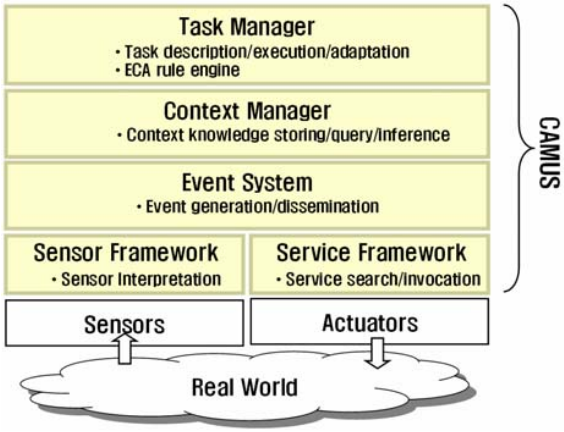

Fig. 2 The conceptual architecture of CAMUS

The Sensor Framework processes input data from various sources such as physical sensors, applications and user commands while considering the current situation, and transfers them to the Context Manager through the Event System. The Sensor Framework supports context-aware applications that are CAMUS task through mapping physical sensors to virtual computational space and extracting context information from sensors.

The Context Manager manages context information from the Sensor Framework. When context information in the environment is changed, the Context Manager transfers events to the Event System. Finally, events are delivered to the Task Manager to supply the necessary context information required for the task execution.

The Task Manager initiates individual tasks and manages on-going task processes. The Task Engine executes the actual task considering the situation. It has an inference engine to process facts and rules supplied by a task.

The Service Framework takes care of managing lifecycles of the Service Agents. The state transitions of the Service Agent are mainly caused not only by the explicit user's request, but also by the context change.

A. CAMUS Main Server

CAMUS is deployed to CAMUS Main Server and Service Agent Manager based on the installed physical locations and they communicate using its own communication framework PLANET.

As depicted Fig. 3, the CAMUS Main Server integrates information delivered from Service Agent Managers. It manages context information including user and environment contexts. It generates and disseminates appropriate events to the Task Engine according to the context change, which enables tasks to take actions required to the current context. It provides the framework to develop context-aware applications.

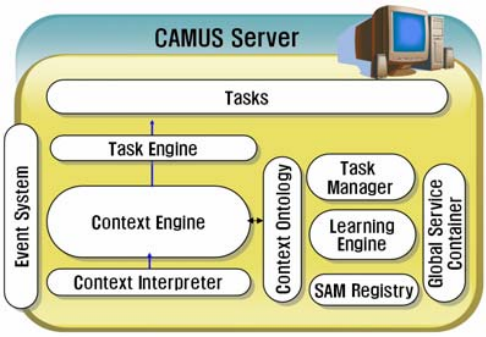

Fig. 3 The CAMUS Main Server

B. Service Agent Manager

Service Agent is a software module performed as a proxy to connect various external sensors and smart devices including robots to CAMUS. It delivers information of sensors in environment to CAMUS Main Server. Also, it receives control commands from CAMUS Main Server, controls devices in the environment, and conducts applications.

Service Agent is executed in the Service Container which Service Agent Manager provided. Service Agent Manger makes accesses to appropriate service agents out of the currently running service agents using a searching function. Fig. 4 shows the architecture of Service Agent Manager.

C. PLANET: Communication Framework

CAMUS is a distributed system which is composed of multi robot clients and a server group. In general, network communication of a distributed system is made by exchanging messages between the systems through communication middleware. This communication middleware was combined with a concept of object oriented technology and suggested a function of calling remote object. It has been used in various kinds of way such as DCOM, CORBA, RMI, XML Web Services, etc. But, they are unsatisfactory for the communication between various clients including robot clients and a server because the processor of the network based robots and smart devices may include embedded system whose computing resources are relatively limited. Therefore, the communication framework for CAMUS required the light-weight protocol to minimize the sending/receiving messages. It also should support the various operating systems and programming languages. Finally, because the network based robotic system may use the wireless network which is not guarantee for the perfect connection, we should consider the disconnected operations and asynchronous operations.

Under this background, we propose the communication framework which is called as PLANET. It has light-weight and fault-tolerant communication mechanism which is independent with operation systems and programming languages. It also supports the disconnected operations and asynchronous operations.

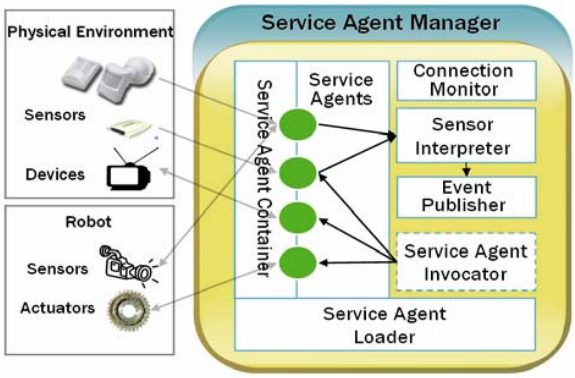

Fig. 4 The architecture of Service Agent Manager

V. Ontology-Based Context Modeling and Reasoning

A. Context Space in CAMUS

In CAMUS, context model simplifies context space as 2-dimensional space for the abstraction level and application domain segmentation. The abstraction level constitutes the vertical axis in the context space. Generally, highly abstracted knowledge can be easily reused by various applications. For example, various context-aware applications commonly use concepts such as place, user, and device. These concepts not only form the skeleton of context, but also act as indices of associated information because they constitute the upper level context knowledge.

The vertical axis is divided into three levels. The shared ontology is located at the highest level. The shared ontology is generally deployed when CAMUS is installed. In the shared ontology, the ontology concepts (i.e., classes and attributes) that are commonly used in various applications are modeled. The shared ontology provides the high level ontology knowledge to context-aware applications, and provides the

same aggregation and granularity to the subordinate domain ontologies which are located at the lower level. The domain ontology is generally built when a context-aware application is developed. In this ontology, domain specific ontology concepts are modeled. The domain ontology provides the domain specific knowledge to context-aware applications, and provides the metadata or schema to the subordinate instance-base which is located at the lowest level. In the instance-base, instances of the ontology concepts are represented. Instances correspond to real objects such as Alice (person), R210 (room), and 18ºC (temperature) of the physical world. The instance-base is generally created and updated when the context-aware applications are executed.

The horizontal axis of the context knowledge space is constituted by application domains such as an intelligent home application, a presentation helper application, and a weather service. In realistic ubiquitous computing environments, applications and services are usually grouped according their application domains because the effect of changes in the environment caused by an application reaches to the other applications.

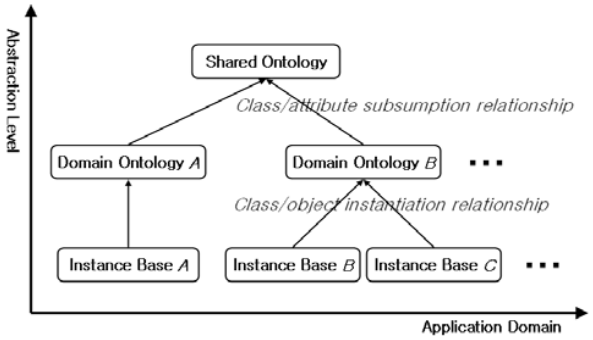

Fig. 5 The context space in CAMUS

B. Presentation Helper Scenario

The scenario is performed in an office environment, especially a meeting room, in which two RFID (Radio Frequency Identification) readers, a voice sensor (microphone), and a beam projector controlled by a notebook are installed. One RFID sensor covers the whole meeting room, and the other covers the presentation platform in a room. As a person enters a meeting room, the office RFID sensor informs the context knowledge management module about the person's ID and location. The context knowledge management module reasons out that the person is an attendant of the meeting which is scheduled at current time in that meeting room. When the meeting host starts the meeting through a voice command, the meeting state represented in the context model is updated as "started." After start of the meeting, if a person who was previously registered as a presenter enters the presentation platform, the platform RFID sensor will senses the presence of the presenter. The application (i.e., the presentation helper task) displays his/her presentation material to the screen using a beam projector, and then updates the state of his/her presentation as "started." When the presenter stops presentation through a voice command, the projection is ended and the presentation state is updated as "ended". This process is repeated until a meeting host stops the meeting with a voice command. User's voice commands and context changes such as presence of a person in the presentation platform are interpreted based on the context knowledge before any actions. For example, when a person who is not a meeting host speaks voice commands to start or stop the meeting, these commands will not be executed.

C. Implementation

Context models for the presentation helper scenario are built by using the context modeling scheme described in Section 5. A context-aware application for this scenario is also developed. Fig. 6 shows parts of OWL based context model in CAMUS. The classes and attributes which are prefixed with the letters "camus" come from the shared ontology. In the shared ontology, concepts that can be commonly used in various applications i.e., highly abstracted concepts such as Activity, Device, Environment, Information Source, and Person are modeled as classes. In the domain ontology, the concepts which are specific to the presentation helper task are modeled. The domain ontology imports the shared ontology to reuse and share it.

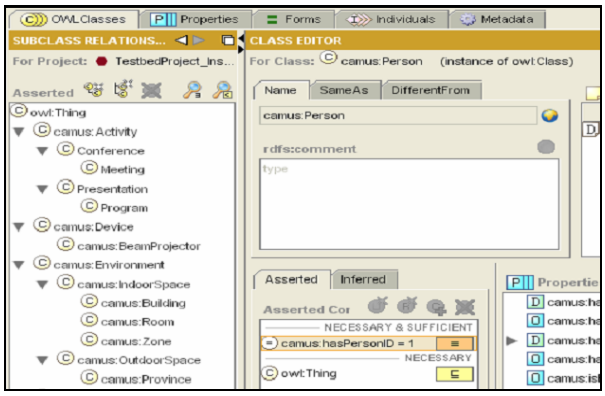

Fig. 6 An example of context model used in CAMUS

Fig. 7 shows the query statement which searches for the meeting object which a person attend by comparing the person's location information with the meeting's venue information. This query is specified with Jena knowledge query language, RDQL. The constituent which is prefixed with '?' means a variable.

```
SELECT  ?meeting
WHERE   (?user,   <rdf:type>,          camus:Person),
        (?user,   <camus:hasLocation>, ?room),
        (?room,   <test:isVenueOf>,    ?meeting),
        (?user,   <test:isAttendantOf>, ?meeting)
USING   camus  FOR  <http://etri.re.kr/URC/CAMUS#>,
        test   FOR  <http://etri.re.kr/URC/TestProject#>,
        rdf    FOR  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

Fig. 7 an example of query statement used in the presentation helper scenario

In order to manage the context knowledge, it is need to build not only the context model but also user-defined inference rules. For example, in the case that a person enters a room where a meeting is being held, the person object and the meeting object should be connected through a relation (e.g., isAttendantOf relation) in order to explicitly reflect the fact that the person is an attendant to the meeting. Such knowledge cannot be un-redundantly represented only by OWL language, and cannot be automatically reasoned out by DL reasoner. Fig. 8 shows this user-defined inference rule. Fig. 9 depicts the structure of CAMUS context reasoning in Context Engine that is implemented based on OWL and Jena application development toolkit.

```
@prefix camus: <http://etri.re.kr/URC/CAMUS#>
@prefix test: <http://etri.re.kr/URC/TestProject#>

@include <file:D:/ContextManager-Prototype/RULE/camus_rule.ckr>


[meeting_attendant:
   (?meeting   rdf:type            test:Meeting)
   (?meeting   test:hasVenue       ?room)
   (?room      rdf:type            camus:Room)
   (?person    camus:hasLocation   ?room)
   (?person    rdf:type            camus:Person)
  ->
   (?person    test:isAttendantOf  ?meeting)]
```

Fig. 8 An example of user-defined inference rule used in the presentation helper Scenario

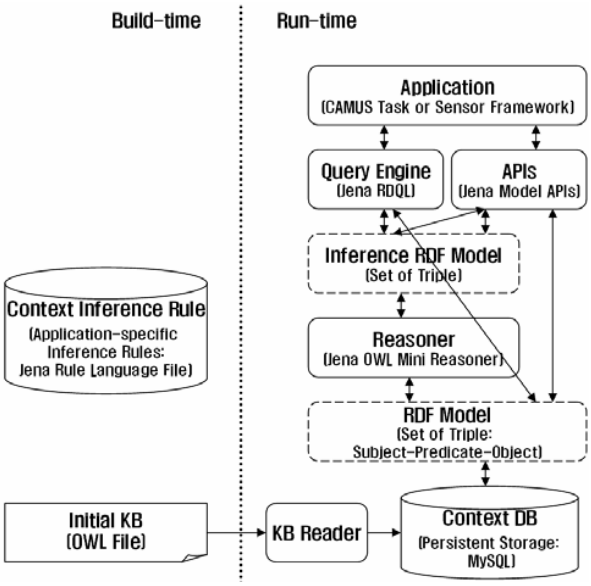

Fig. 9 The structure of CAMUS Context Engine

VI. Conclusion

In the vision of ubiquitous computing environment, computer system understands their situational contexts and provides appropriate services to users. URC is a new concept for a network-based service robot system and needs a context-aware infrastructure for supplying intelligent services to users through robots and environmental devices.

In this paper, we introduced CAMUS which is a server platform and a context-aware infrastructure and discussed technical issues and structures for developing infrastructure. Also, we discussed the ontology-based context modeling and reasoning approach and validated it with the presentation helper scenario.

CAMUS has applied to the u-Dream Hall [17] that is established by Ministry of Information and Communication and new-Technology Exhibition Center of SAMSUNG Electronics in Republic of Korea.

References

[1] M. Weiser, "The Computer of the 21st Century," *Scientific American*, vol. 265, pp. 66-75, 1991.
[2] T. Kindberg and J. Barton, "A Web-based nomadic computing system," *Computer Networks*, vol. 35, no. 4, 2001.
[3] K. Henricksen, J. Indulska, and A. Rakotonirainy, "Modeling context information in pervasive computing systems," *Proceedings of the First International Conference on Pervasive Computing (LNCS 2414)*, August 2002.
[4] A. K. Dey, G. D. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context -Aware Applications," *Human-Computer Interaction (HCI) Journal*, vol. 16, 2001.
[5] W. N. Schilit, "A System Architecture for Context-Aware Mobile Computing," Ph.D thesis, Columbia University, 1995.
[6] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project Aura: Toward Distraction-Free Pervasive Computing," *IEEE Pervasive Computing*, 2002.
[7] The Cooltown project. Available: http://www.cooltown.com/cooltown/.
[8] MIT Project OXYGEN, Available: http://oxygen.lcs.mit.edu/.
[9] A. Ranganathan and R. H. Campbell, "A Middleware for Context -Aware Agents in Ubiquitous Computing Environments," *Proceedings of the ACM/IFIP/USENIX International Middleware Conference*, 2003.
[10] H. Chen, S. Tolia, C. Sayers, T. Finin, and A. Joshi, "Creating context-aware software agents," *Proceedings of the First GSFC/JPL Workshop on Radical Agent Concepts*, 2001.
[11] T. Strang, and C. Linnhoff-Popien, "A context modeling survey," *Proceedings of UbiComp: 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.
[12] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using OWL", *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004.
[13] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 18, no. 3, 2003.
[14] OWL (Web Ontology Language) Guide, Available: http://www.w3.org/TR/owl-guide/.
[15] Jena – A Semantic Web Framework for Java, Available: http://jena.sourceforge.net/
[16] H. Kim, Y.-J. Cho and S.-R. Oh, "CAMUS: A middleware supporting. context-aware services for network-based robots," *IEEE Workshop on. Advanced Robotics and its Social Impacts*, 2005.
[17] u-Dream Hall, Available: http://u-dream.or.kr/.