6-2 Distributed and Cooperative Service Platforms for Home Network Services

YAMAZAKI Tatsuya, SAWADA Atsushi, NISHIMURA Toshikazu,
TAKAOKA Masanori, TAJIKA Yosuke, and MINOH Michihiko

In the UKARI (Universal Knowledgeable Architecture for Real-lIfe appliances) project, we studied and developed distributed and cooperative service platforms for home network services, which provide services by cooperating functions from networked appliances (NAs). The NAs include consumer appliances as well as sensors, robots, and so on. Main research results were "UKARI-Core" and "UKARI-Kernel", which constitute the platforms. UKARI-Core is a middleware with must and core modules of distributed function cooperation. UKARI-Kernel is, moreover, another middleware as an extension of UKARI-Core with additional functions for flexible user adaptation. We implemented the middleware into various kinds of real appliances and constructed home network services in a real environment to evaluate the efficiency of the proposed service platforms.

Keywords
Networked appliance, Home network, Function cooperation, Middleware

1 Introduction

With recent advances in hardware and software technologies, a variety of appliances have grown increasingly multi-functional and sophisticated. The value of each of these units has thus increased dramatically. On the other hand, the digitalization and networking of home electronic appliances is also progressing by the year, and efforts are presently underway in the construction of a home network environment based on the networking of home information appliances. Services currently provided by home networks include the inter-appliance exchange of content and remote control of the appliances. However, practical issues remain: most services, for example, are only operable between products from a single manufacturer, or may only be applicable to a specific type of system (such as white goods or audio visual [AV] appliances). Standardization is thus eagerly awaited.

Given this situation, users are presently enjoying the benefits of technological innovation as they adapt to and assimilate the new technologies. However, very few people use all newly available functions in any given appliance, and in some cases, users of an appliance may wish to make use of a function only offered by another appliance. Further, when multiple functions of similar types exist within a single household, it would be convenient if such functions could be shared as temporary alternatives. In the UKARI Project[1], R&D efforts were devoted to the production of a platform for the construction of distributed and cooperative home network services that would enable each function of an appliance to be treated as a unit over the network, by dividing the appliances into clusters of single, component functional units. The "UKARI-Core" was developed as the first step in the construction of a platform for such inter-appliance functional cooperation. As an extension of the UKARI-Core, the UKARI-Kernel offers additional functions that allow more flexible user adaptation. The present paper will describe these two home network platforms. A notable feature of both the UKARI-Core and UKARI-Kernel is found in the ways in which both systems target a wide variety of appliances to be connected to the network. These appliances are collectively referred to as the NA (Networked Appliances) below. Similarly, when a function connected to the external network is completely independent of other functions and operated through a preset interface, the function is referred to as an FE (Functional Element).

2 Related work and the significance of the proposed method

There are a range of ongoing studies of various types of systems aimed at the coordination of functions distributed over a network[2]-[7], as well as continuing R&D on a home-network oriented system[8] and associated standardization activities[9]. With advances in distributed object technology and Web service technologies, we can conclude that the platform for interoperable use of components via the network is being released. However, in order to construct services using such components, it will be necessary to determine in advance the details of the component interface specifications. For example, technologies providing service solutions such as UPnP (Universal Plug and Play)[6] and ECHONET (Energy Conservation and Homecare Network)[9] are capable of searching for the necessary service objects in response to a request, but since only connections with real appliances as the operative units are defined, it will be necessary to acquire detailed information on the unit interface separately for each service instance discovered by the search before constructing an application by cooperating appliances in each case. Conversely, on the component developer side, there are cases in which there is no choice but to design an interface based on a scenario for a specific application, and this factor inhibits reuse of the software. In order to overcome these problems, the UKARI-Core and the UKARI-Kernel provide a network link platform that enables cooperation among individual functions belonging to an appliance, instead of appliance-based cooperation.

In the future, not only home electronic appliances but sensors and robots as well will be connected to the home network, providing services tailored to the home environment. In order to ensure compatibility with such potential future appliances, the UKARI-Core and the UKARI-Kernel are fitted with a framework for function discovery and similar tasks, a framework that ingeniously combines the typology and abstraction of appliance functions with individualized description of detailed specifications. This framework will be introduced below.

In the first step, the FE-type is categorized based on two axes — the type of media handled by the FE and the operations executed through the media (Table 1). Here, the media types consist of logical true/false structures (boolean), numeric, text, audio, image, and movie media. Operation types are as follows: generate, consume, mix, transform, and store. The adoption of this general FE classification scheme based on simple notations enables swift detection of the FE according to category.

Table 1 shows how the FE type is expressed

Table 1 *Classification of FE Types*

Media types	Operation types				
	generate (g)	consume (c)	mix (m)	transform (t)	store (s)
Boolean (B)	Bg	Bc	Bmk	BXt	Bs
Numeric (N)	Ng	Nc	Nmk	NXt	Ns
Text (T)	Tg	Tc	Tmk	TXt	Ts
Audio (A)	Ag	Ac	Amk	AXt	As
Image (I)	Ig	Ic	Imk	IXt	Is
Movie (M)	Mg	Mc	Mmk	MXt	Ms

by a combination of two or three alphanumeric characters. For all combinations presented in Table 1, the first character that is a capital letter indicates the media type. The generation, consume, and storage operations are indicated by the second character, which is a small letter. The mix operation is indicated by an "m" in the second character position, followed by a third character representing the number of media to be mixed. For example, "Am2" will indicate that the FE will mix two audio media. In the notation for the transform operation, the second character indicates the media type after transformation, which is followed by the third character, the letter "t", representing the operation type. For example, "ATt" will indicate that this FE will transform audio media into text media.

In order to coordinate cooperation between FEs, the classification scheme presented in Table 1 is too rough, and so in the second step, detailed information must be exchanged to facilitate such cooperation. This detailed information is described in the FECAP. For example, the FECAP for image media type contains information on coding method and size. In other words, the FE types in Table 1 are narrowed down to functional groups that can potentially serve as service elements. Detailed information is then traded by mutual exchange of the FECAP, to effect the final selection of functions that will actually be used in the construction of a service. Let us assume, for example, that there are several tens of NAs on the home network, and that each NA is capable of providing five FEs on average. In this case, the number of FEs that will be targeted in the search during the construction of a functional cooperation service will be on the order of several hundreds; therefore, it will not be feasible to retrieve and assess the massive volume of FECAPs for all FEs. Thus, a more efficient and effective method will be to determine the available FE resources for service construction using detailed information in the FECAP, after screening and selection of FE according to FE type based on our proposed classification scheme.

3 The UKARI-Core

3.1 General configuration and basic modules

Figure 1 shows the general configuration of the system for cooperation between monotypic functions, in which the NAs are interconnected via the home network. Each NA has several functions that correspond to the functions of real hardware, and for each func-

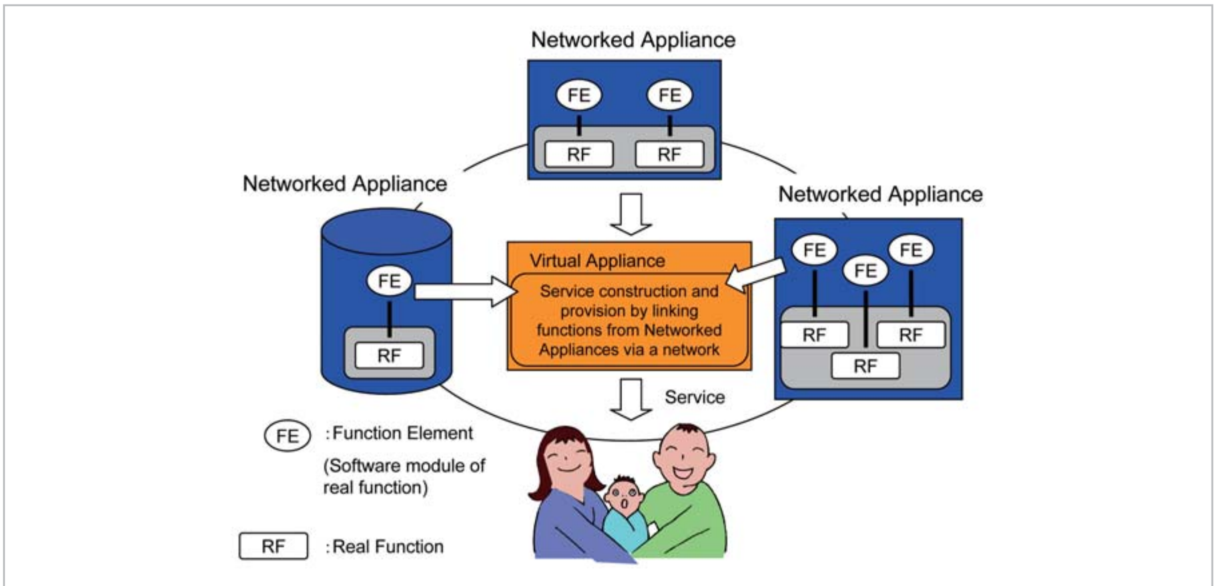

Fig.1 *General configuration of the UKARI-Core system*

tion, an FE exists as a software object that will act as the interface to the external network. In the home network platform proposed by our project, the NA is divided into an initiator, which carries out the setup and control of the service, and a responder, which provides the functions necessary for the service. However, the initiator does not have to be physically independent for each appliance, and so it is possible for this component to coexist with a responder inside a single appliance.

Figure 2 shows the basic modular configuration of the initiator. The main module inside the initiator effects service management, which consists largely of the service description parser, topology controller, and service scenario manager.

The service description parser is a module that interprets the service description written in XML (for “eXtensible Markup Language”), and the service description consists of the (service) scenario description and topology description. The topology controller searches for the required functions via the communication module based on the topology description and then establishes the network path to retrieve the function. The service scenario manager is a module that transmits a control message to each FE based on the scenario description and receives the returned event.

Figure 3 shows the basic module configuration of the responder. The responder mainly consists of the NA Manager, which controls the NA itself, and the FE Manager, which controls the FE provided by the NA. There is a one-to-one correspondence between the FE Manager and the FE provided by the NA.

The NA description parser within the NA Manager interprets the NA description written in XML, and the judgment unit determines whether there is an FE corresponding to the NA description in response to a request from the initiator.

The FEM Manager is a module that controls multiple FE Managers; the module configuration inside the FE Manager is shown in Fig. 4. The real function manager within an FE Manager consists of a module that proxies for the functional hardware of the real appliance, and the control path manager consists of a module for managing the network path over which the control messages (commands) are sent and received.

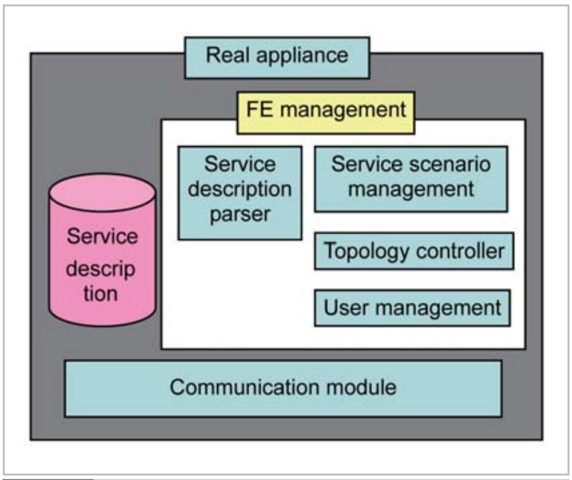

Fig.2 *Configuration of the initiator*

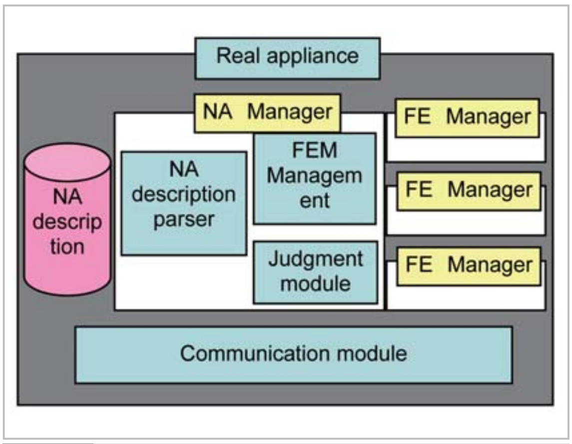

Fig.3 *Responder configuration*

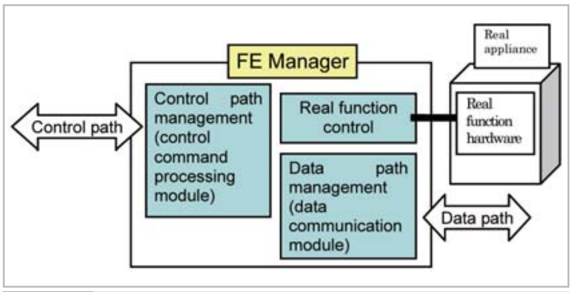

Fig.4 *FE Manager Configuration*

3.2 Operations protocol

This section will describe the protocol for operations. First, the initiator service must discover the functions that will constitute the service. The message sent over the network to discover these functions is called a “Bid”. A Bid is sent out to all responders present within a certain network area. When a responder that has received the Bid is equipped with the function in question, it will send a return message proposing to offer the function to the initiator. This message is called an “Offer”. The initiator then makes a request to the responder that has returned the Offer for a batch of description elements called the FECAP, which contains descriptions that include more detailed information. The responder will send the requested FECAP in response. In the UKARI-Core, the FECAP received by an initiator is retained (cached) for a certain length of time to eliminate any need for FECAP exchange. When an initiator receives multiple Offers, it will have to select one responder. Presently, this selection is assigned to the application implementation specification. Finally, when the selection of the responder providing the information is made, the initiator then returns a message of agreement to the responder, referred to as the “Agree” message. The initiator returns a Disagree message to those responders that were not selected.

When multiple functions are required to constitute a service, the above process is repeated until all of the required functions are discovered. The initiator then sends a message to each responder to establish the service path between the FEs for data transmission and reception. This is the general flow of the function discovery and control/service path construction processes for establishment of the service.

4 UKARI-Kernel

4.1 Overview

The UKARI-Core described in the previous section provides the basic mechanism for functional cooperation through the search and selection of the service-component FEs, based on service descriptions and the establishment of the ability to interlink FEs. However, in a mixed environment of home networks featuring a wide variety of appliances, the simple FE discovery and cooperation functions are not sufficient, particularly for platforms on which efficient user-adaptive service applications are developed and provided. The UKARI-Kernel[10] is a functional cooperation platform expanded from the UKARI-Core to incorporate the following three points.

(1) NA and FE access control based on user authorization
(2) Framework for describing not only the structure but also the behavior of applications
(3) Data format capable of detailed description of data from various sensors

For the access control function in (1), the UKARI-Core was expanded to make the NA Manager of the responder manage the link to each FE based on the Access Control List (ACL).

For the behavior description described in (2), the inter-FE relationship description — which had been called the service description — was renamed the topology description, and a scenario description was newly adopted to enable formal expression of the service application behavior. The initiator was also newly implemented with a function for interpreting the scenario description and controlling service execution. In the UKARI-Kernel, the topology, and the scenario descriptions are collectively referred to as the service description.

With respect to (3), the NA description specifications were revised so that the platform would be capable of additional description of composite data formats in addition to standard numerical expression under the SI unit system, in order to enable the expression of data output from a wide spectrum of sensors and also to ensure the selection of the most suitable sensor FE for the intended purpose of the service application once FE discovery has taken place.

The subchapters below will present brief overviews of the ACL and scenario description.

4.2 Access Control List (ACL)

In the UKARI-Kernel, the ACL, in which access authorization for NA and FE is issued for each user, was adopted so as to realize access authorization, both in view of the protection of privacy and to provide solutions in light of competition for resources. The ACL must satisfy the following requirements for providing access control on a home network service realized through the UKARI-Kernel.

- The description must be capable of expressing usage policies such as priority sequencing and permission to use with respect to each appliance and service.
- Permission to use must be described per user to protect privacy.
- Since multiple functions within a single appliance will be utilized in different services, the description unit must not be based on type of appliance, but based on the individual function.
- Since the services realized by the UKARI-Kernel are constituted of a combination of functions, the priorities for each service must be prepared in advance.

In order to satisfy the above requirements, the ACL in the UKARI-Kernel allocates inherent identification data (ID) to each user, function, and service; this data is then combined to digitize and define permission-to-use services and service priorities. In other words, the ACL consists of a four-tuple set of user ID, service ID, function ID, and priority. Thus, it is possible to use a combination of user ID, service ID, and function ID as an inquiry key to obtain a reply on permission to use or priority.

There are several methods for carrying out priority control using the ACL. One method is to give priority to services that, in combination, have larger priority values compared to those with smaller priority values. Another is to impose extremely strong access control by regarding appliances forbidden for use as equivalent to extremely low-priority appliances. For more details on priority control, the user interface for using the ACL, and examples of conditions of practical application, readers are referred to[11].

4.3 Scenario description

FE cooperation in the UKARI-Core and UKARI-Kernel does not consist of tight coupling between specific objects, but instead consists of interactions within a loosely bound relationship formed by publication of an abstract interface. Accordingly, it was judged that a service-oriented approach would be most appropriate, and an XML-based scenario description method based on BPEL4WS[12] was designed as the framework of the behavior description[13]. Figure 5 presents a general scheme of the scenario description in the UKARI-Kernel. The shaded section in the central part of Fig. 5 corresponds to a scenario (expressed by process tag in BPEL4WS) for a single service application; this is a newly added framework in the UKARI-Kernel. The scenario description is interpreted and executed by the initiator, and the control transition procedure can be described using links expressing the active and interactive control dependencies. The data collected as a result of the data exchange between the scenario and the external network is also stored in the variable, which is then available for reference in other activities in the process.

The FE interface (FEI) on the right-hand side in Fig. 5 is the port for information exchange between the scenario and the FE. The FEI is defined with a one-to-one correspondence to the FE that serves as the topology description element, and provides the interface for FE control of the FE that will actually

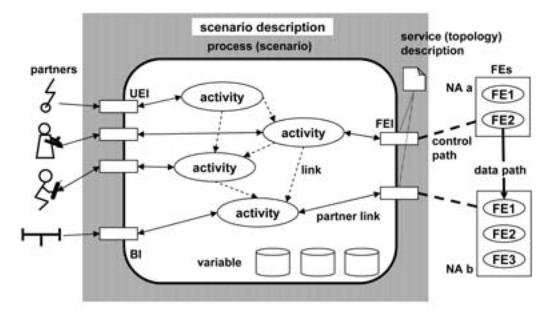

Fig.5 Scenario Description in the UKARI-Kernel

participate in the service at various stages: FE discovery and linking, FE function execution, inter-FE data transmission, etc. For example, a control path may be established by searching for an FE of the appropriate type within the topology description by issuing calls to operations for FE discovery and linking, as defined in the interface. The initiation, termination, and interruption of FE functions may be performed by issuing calls for operations to execute functions. Data transmission via the inter-FE data path may also be made by issuing calls for operations associated with data transmission and reception.

In contrast, the user and environment interface (UEI) on the left side of Fig. 5 acts as a port for exchanging the service elements that cannot be categorized into FE types. This UEI is mainly used for data exchange and control with users and external services. For the UEI, the interface for control and data exchange through the ports may be freely defined.

The broker interface (BI), also on the left side of Fig. 5, is a port for exchanging messages without specifying endpoints, and the initiator pre-registers the relevant message type on the broker in order to receive notifications that the corresponding type of message has passed through. The BI enables the provision of applications having configurations in which multiple, different scenarios are initiated by an event occurring at a single appliance or sensor.

Eight types of activities that will form the components of processes were defined in the scenario description language for the UKARI-Kernel, as shown in Table 2. There are several differences between the UKARI-Kernel and BPEL4WS: the interfaces required for FE discovery and linking are implicitly defined as FEI elements; communication formats with no specified endpoints are supported by the BI; and implementation in embedded systems with little power is attempted through simplification and lightweight design of the logics of the interpretation execution systems, by eliminating elements other than pick, from the scenarios.

Table 2 Types of Activities

receive	Input data through FEI, UEI, and BI
reply	Input data through FEI, EI, and BI
invoke	Start-up operation through FEI, EI, and BI
assign	Substitute variables by contents of message
terminate	Terminate scenario execution
pick	Transition to activity according to data type input through FEI, EI, and BI
findNA	Discover NA by specifying keywords, etc. through FEI
findFE	Discover FE on topology description through FEI

5 Examples of implementation

Our project has developed the UKARI-Core and UKARI-Kernel as middleware on a proposed home network platform, and we have performed validation experiments on service construction on real home networks through implementation in various real appliances such as home electronic appliances, sensors, and robots. As a result, we have confirmed that services that could not be provided by a single appliance may be realized through functional cooperation between multiple, existing appliances, through function discovery operations and functional cooperation. In the present paper, we present a representative example of a service realized using the UKARI-Core. Both the UKARI-Core and UKARI-Kernel have been widely publicized for R&D purposes as open software[14].

Figure 6 shows the configuration of the implementation system. The UKARI-Core is implemented on real appliances such as digital cameras, plasma display TVs, refrigerators, door phones, and pyroelectric (human detection) sensors. The door phone has an On/Off switch consisting simply of a button, and the On/Off event is accommodated directly in the adapter board of the UKARI-Core via the sensor network, using ZigBee (ZigBee is a registered trademark of Koninklijke Philips Electronics N.V.). The pyroelectric sensor is an infrared sensor that detects the presence of objects emitting infrared radiation within a 5-m radius and within a designated angle range. When such an object is detected, it is reported as an event to the adapter board of the UKARI-Core. An external view of the implementation system is shown in Fig. 7.

The digital camera, plasma display TV, and refrigerator correspond to responders in the present implementation system, and the configuration of the implemented execution modules is presented in Fig. 8. Moreover, the functional hardware of the real appliances has been divided into units according to function type, and modules for their control and notification are also implemented. For example, the function for displaying characters (text data) on a plasma display TV has been incorporated into a unit and coupled with the FE for an NA that provides functions. Real appliance functions were also expressed through service descriptions before operational validation testing on the implementation system. The programs themselves were written in C and C++ and were operated on the Linux platform on an IP network, including embedding of the parsers for XML interpretation.

We have been working on the construction of the real-life validation test bed known as the "Ubiquitous Home" since April 2004, to conduct validation experiments on the effectiveness of info-communication technologies, including validation of the developed middleware in real situations, all within the daily environment of a real home[15]. During the one-year period of FY2005, real-life validation experiments in this Ubiquitous Home were performed with a total of five sets of subjects — four families of different structures and one group of researchers. In the experiments, each group was asked to actually live inside the Ubiquitous Home for approximately two weeks, following the same daily life pattern of their usual lives, including activities such as eating, sleeping, and commuting. The daily services prepared were continuously used during the period for subsequent evaluation. We were able to obtain feedback on the conditions of service use from these real users in the form of responses to inquiries and answers on evaluation sheets, as

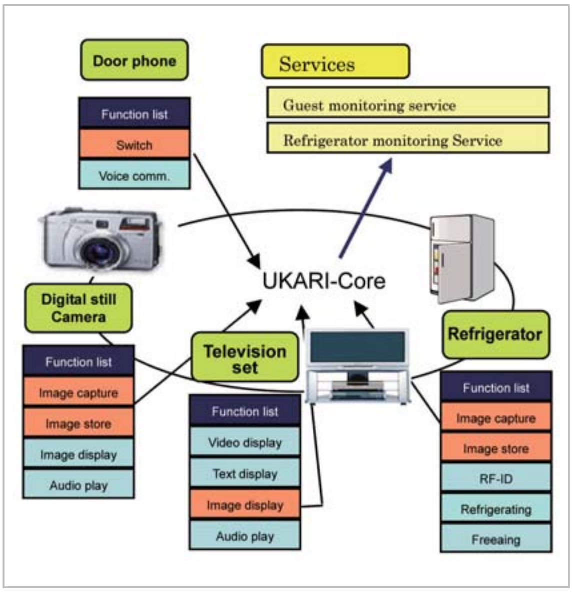

Fig.6 Implementation system configuration

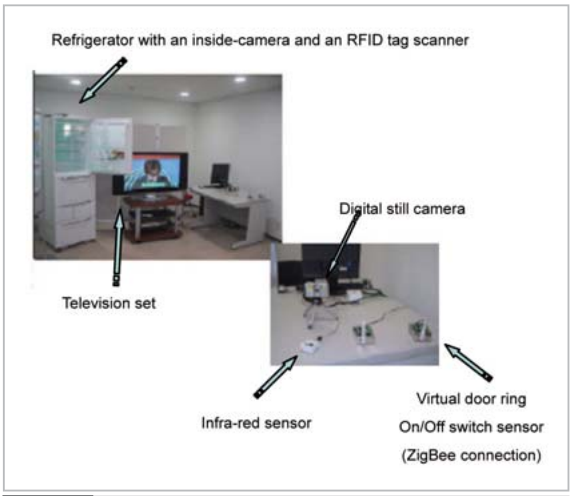

Fig.7 External view of implementation system

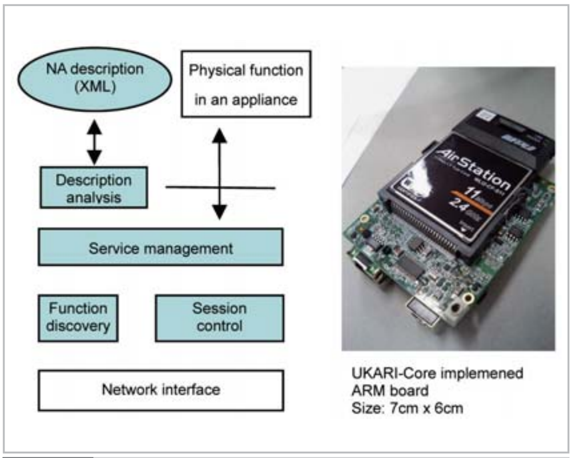

Fig.8 Implementation module; configuration and example of implementation

well as through interviews. We were also able to collect behavioral data under real life conditions by various sensors.

During this real-life validation experiment, a UKARI-Core service was provided that notifies the user that the washing machine has completed its cycle. With this service, wireless tags were used to track the positions of the users, and the voice media generation appliance (normally a speaker) installed nearest to that point is discovered to form an on-the-spot data path to use the speaker. During the two-week period of the experiment, this service was shown to operate stably, with users expressing their amazement on how “the house apparently knows where the user is” for notifications. The service was evaluated as highly convenient.

6 Conclusions

This paper has presented an overview on the distributed and cooperative service platforms for home network services developed through the R&D efforts of the UKARI Project. Through this platform, home network services are constructed through the cooperation of the individual functions of NAs as units in the network. The UKARI-Core is the middleware that holds the bare essentials of the mechanism for functional cooperation, and the UKARI-Kernel is an extension with functions of access control, scenario description, and sensor extension added to the UKARI-Core, enabling more flexible and user-adaptive service applications. This platform opens the door to the following possibilities:

- Framework for function description in an application-independent format
- Framework for formal description of the structure and behavior of a functional cooperation service
- Extraction and typifying of functions shared across a wide range of appliances
- A flexible function discovery method adapted to the real-life environment and real appliances

The UKARI-Core and UKARI-Kernel have been publicized as open source software for research and development purposes, and various modifications are underway — for example, to ensure compatibility with real-time streaming functions[16].

References

1 Minoh Michihiko, “6-1 Experiences in UKARI Project”, This Special Issue of NICT Journal.

2 The Oxygen Project Website: http://oxygen.lcs.mit.edu/.

3 The Ninja Project Website: http://ninja.cs.berkeley.edu/.

4 Masateru Minami, Kaoru Sugita, Hiroyuki Morikawa, and Tomonori Aoyama: “A Design of Internet Application Platform for Ubiquitous Computing Environment”, Trans. IEICE, Vol.J85-B, No.12, pp.2313-2330, 2002. (in Japanese)

5 Gupta, R., Talwar, S., and Agrawal, D. P.: “Jini home net-working: a step toward pervasive computing”, Computer, Vol.35, No.8, pp.32-40, 2002.

6 Universal Plug and Play Website: http://www.upnp.org/.

7 HAVi Website: http://www.havi.org/.

8 Nakazawa, J., Tobe, Y., and Tokuda, H., “VNA: A Serverless Distributed Architecture for Integrating Information Appliances”, IEICE Trans. Fundamentals of Electron., Commun. & Comput. Sci., Vol.E84-A, No.7, pp.1610-1623, 2001.

9 ECHONET Website: http://www.echonet.gr.jp/.

10 Atsushi Sawada, Toshikazu Nishimura, Tatsuya Yamazaki, and Michihiko Minoh, “Facilities for Access Control and Behavior Description in UKARI-kernel — a Middleware for Networked Home Appliances”, Technical Report of IEICE, MoMuC2006-60, Vol.106, No.409, pp.13-18, 2005. (in Japanese)

11 Toshikazu Nishimura, ”Facilities for privacy preference and access control in UKARI-kernel”, The 3rd International Workshop on Ubiquitous Home, pp.117-122, 2006. (in Japanese)

12 Andrews, T., Curbera, F.,Dholakia, H.,Goland, Y.,Klein, J.,Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickobic, I., and Weerawarana, S. “Businness Process Execution Language for Web Services”, Version 1.1, 2003.

13 Atsushi Sawada, Yosuke Tajika, Tatsuya Yamazaki, and Michihiko Minoh, “A Service Scenario Description Framework for Functional Cooperative Environment of Networked Appliances”, IPSJ SIG Technical Reports, 2004-SE-145, Vol.2004, No.87, pp.97-104, 2004. (in Japanese)

14 UKARI Project Open Website: http://open-ukari.nict.go.jp.

15 Tatsuya Yamazaki, “Ubiquitous Home: Real-life Testbed for Home Context-Aware Service”, Tridentcom 2005 (First In-ternational Conference on Testbeds and Reserch Infrastruc-tures for the DEvelopment of NeTworks and COMmunities), pp.54/59, 2005.

16 Yoshitaka Morimura, Tatsuya Yamazaki, and Michihiko Minoh, “A Streaming Service Control with QoS Management in Distributed and Cooperative Service Middleware”, Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO2007), 2007. (in Japanese)

***YAMAZAKI Tatsuya**, Ph.D.*

Research Manager, Universal City Group, Knowledge Creating Communication Research Center (Former: Senior Researcher, Distribute and Cooperative Media Group, Keihanna Human Info-Communication Research Center)

ICT for Multimedia Processing

***SAWADA Atsushi**, Dr. Eng.*

Professor, Department of Mathematical Sciences and Information Engineering, Nanzan University (Former: Expert Researcher, Distribute and Cooperative Media Group, Keihanna Human Info-Communication Research Center, Information and Network Systems Division)

Software Engineering

***NISHIMURA Toshikazu**, Ph.D.*

Associate Professor, College of Information Science and Engineering, Ritsumeikan University (Former: Expert Researcher, Distribute and Cooperative Media Group, Keihanna Human Info-Communication Research Center, Information and Network Systems Division)

Computer-Mediated Communication

TAKAOKA Masanori

Assistant Manager, NCOS Laboratory, NEC Communication Systems,Ltd.

Network

***TAJIKA Yosuke**, Ph.D.*

Former: Toshiba Corporation

Home Network

***MINOH Michihiko**, Dr. Eng.*

Professor, Academic Center for Computing and Media Studies, Kyoto University (Former: Group Leader, Distribute and Cooperative Media Group, Keihanna Human Info-Communication Research Center, Information and Network Systems Division)

Three Dimensional Modeling, Environmental Media, Communication Support System