

## 第 XXI 部

### 実空間ネットワーク環境



## 第21部 実空間ネットワーク環境

### 第1章 はじめに

SPEARS WG は、家電や日用品、ペットや人間などあらゆる物体を仮想的にネットワークに接続するネットワークアーキテクチャである、実空間ネットワークを構築するための諸問題について議論している。

当面の研究目的として、WIDE 研究会 / 合宿等の会議支援に係る研究開発に注力している。動作検証のため、2003年3月、9月のWIDE合宿において実験に参加した。実験の報告については、「実空間ネットワークを利用したコミュニケーション支援」として本報告書に収録されている。

また、上記を含む研究成果を2本の論文として執筆した。“Rendezvous enhancement for conference support system based on RFID” は、SAINT2004で発表され、会議におけるユーザ間のランデブー提供モデルの提案とプロトタイプ実装について述べている。

“インターネット上で実空間情報を収集・管理するフレームワーク (FIRSII) の提案と実装” は、奈良先端科学技術大学院大学の修士論文として発表されたものをまとめたもので、センサ及びセンサネットワーク間で実空間情報の共有、再帰的な利用が可能な環境を構築するためのフレームワークの提案とプロトタイプの実装について述べている。

本報告では、第2章でSPEARS WGの2003年の活動履歴を掲載し、第3章では“Rendezvous enhancement for conference support system based on RFID”を、第4章では“インターネット上で実空間情報を収集・管理するフレームワーク (FIRSII) の提案と実装”を掲載する。

### 第2章 SPEARS WG 活動履歴

本章では、1年間のミーティングで共有された情報を整理する。本WGでは、WIDE合宿を含め13回のミーティングが開催され、以下のような議論がなされた。ミーティングが開催されない月もML上で議論を以下のように行った。

- 2003/3
  - 3/4: WIDE 春合宿にてミーティング開催
    - 合宿実験に関する議論
    - 今後の研究の方向性について
    - 仮想的ネットワークに接続するネットワークアーキテクチャ、アプリケーションについて
  - ディスカッション
- 2003/4
  - MLで春合宿での問題点をブラッシュアップ
- 2003/5
  - 5/4: WG ミーティング開催
    - WIDE 研究会に向けて
    - 今後の方針について
  - 5/21, 27: WG ミーティング開催
    - 研究会での発表内容、実験について議論
  - 5/30: 5月研究会にてミーティング開催
    - 合宿での実験について議論
- 2003/6
  - MLで実験や関連記事について議論
- 2003/7
  - MLで実験や関連記事について議論
- 2003/8
  - 8/2, 10, 17: WG ミーティング開催
    - WIDE 合宿で行う実験の確認、情報共有
    - 開発
- 2003/9
  - WIDE 合宿
    - 合宿実験の詳細を説明し改善点などを議論

- 2003/10
  - 10/4, 18, 31: WG ミーティング開催
    - 9 月合宿の反省
    - 今後の活動方針
    - WIDE memo について
    - 12 月研究会および次回の WIDE 合宿に向けて目標の設定
- 2003/11
  - 関連情報について ML で議論
- 2003/12
  - 12/14: WG ミーティング開催
    - 12 月研究会についてブラッシュアップ
    - Spears ID について議論
    - 3 月 WIDE 合宿について

---

### 第 3 章 Rendezvous enhancement for conference support system based on RFID

---

#### Abstract

*GPS is popular to be used to construct a service based on location. RFID attracts attention as a useful device in similar application. SPEARS WG of WIDE Project had already developed a system[320], which uses RFID devices as a sensor of human's location. Purpose for this system is to provide the service based on a user's location in a conference. In this system, the model in which location of user is gathered and provided per each user was adopted. This report describes several areas of problems and improvements in our latest implementation. The following functions were implemented by the addition of a system called "AreaD": 1) provide a service tied with a location, and 2) provides users' communication rendezvous service. These functions are able to add to the existing system without altering the existing model. This report describes a location-oriented data processing model and applications.*

#### 3.1 Introduction and Background

Generally, ubiquitous computing means an environment where computer resources are

omnipresent. To realize this environment, following two achievements are indispensable: One is that connectivity to the Internet should be spread widely. The other is that huge number of computers with network connectivity should exist anywhere. About the former, according to statistics, it is said that broadband connectivity to the home or office are almost realized. In 2003, about 20% of household (about 9 million home) in Japan has the broadband Internet connectivity. The capacity for the broadband Internet connectivity is about 74 million for home. The capacity means the number of perpetrated facilities to connect broadband Internet to a home at the Internet service providers. Almost all offices already have broadband connectivity (according to statistics of Ministry of Telecommunications Japan).

In the latter case, although the portable computer has spread by its miniaturization or low pricing, it cannot be said that computer resources exist at anyplace ubiquitously. Under such a situation, a non-contact identification technology called RFID attracts attention. It is the technology that computers can recognize non-networked object and/or human by attached RFID. In Auto-ID Center[8], such research and development are doing in order to realize next-generation supply chain management.

SPEARS WG of a WIDE Project started in 2001, is a research group aiming to provide services based on users' location in the real space. With assumption that a user carries RFID, the activity of this group is to propose the way to abstract RFID readers and its event, developing the techniques for managing data, and investigating applications. As an application example for this technology, a conference support system was developed and field tests were conducted.

We conducted a field trial at biannual meeting of the WIDE project (WIDE Camp). There are over 200 participants, and 4 or 5 parallel meetings are held at the place. A system that provides a user's location information by using WEB page was developed. This system stores

the location information where the RFID has been identified and offers the information through WEB interface. In the database, the profile of participants is accumulated beforehand. The other application detects a user standing in front of the microphone and displays his or her profile on a projector.

Section 3.2 of this report overviews activities, which were done by SPEARS WG, and its assumptions. Section 3.3 proposes existing communication or information model. Before designing the system, we studied a model in which information of user is gathered and provided per each user. Section 3.4 describes AreaD design and implementation. Section 3.5 introduces sample applications and results of field test. Section 3.6 discusses issues and tentative solutions. Section 3.7 concludes of this report.

### 3.2 Assumptions and Field Test

Several experimental tests were conducted in advance of this research. WIDE SPEARS WG operated these experiments. The main purpose of SPEARS WG is to support user who participates in a meeting or conference. These experiments were carried out with the following assumptions.

A user carries RFID and communication equipments, such as Laptop PC, and uses several services on the computer. Services are regarded as applications on the Internet. In addition to that, information required for communication such as user's name, RFID, an e-mail address, etc. is registered beforehand.

First implementation was started at a field test on 2001, which displays information which room a user is located at. Concurrently, statistical data was also acquired. The graph that displayed the time-series number of the attendants for each conference room is shown in Figure 3.1. It can be recognized in this graph how many participants were in each room.

This field test system consists of these three parts. 1) Physical sensor parts set on the each conference room. 2) Middle ware parts that

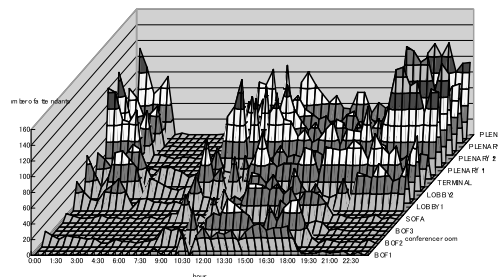


Fig. 3.1. The time-series number of attendants

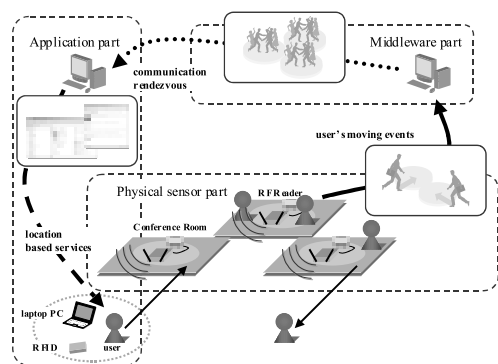


Fig. 3.2. Components of Experimental System

worked somewhere. 3) Application server and client on user's computer. Figure 3.2 shows relations between these parts on the field test system.

#### 1) Physical sensor part

Physical sensor part consists of a RF-Reader, small computer and software on that computer. Physical sensor parts placed at each conference room and these sensors recognize each user in that conference room. We use rf-code Spider Reader[240] as RF-Reader and two small computers. One is NEC Mobile Gear MC/R550[206] and the other is Internet Node RS6[130]. Spider Reader and two small computers tied each other by serial cable. RF-Reader acquires user's moving events and sends to small computers. Small computer that receive the events, normalize and send to middle ware part using UDP/IP.

#### 2) Middle ware part

Middle ware part controls each user's location information that changes dynamically and profiles that is used by communications on applications. Profile consists of static information. For example, users name, e-mail address and chat ID. In this field test, this middle ware part used

PC that have CPU Pentium4/1.6 Ghz, Memory 256 Mbytes, HDD 20 GB, NIC Fast Ethernet, OS FreeBSD 4.7-R. We will describe information management model on the middle ware part at section 3.3.

3) Application part

Application served two types of location search service. One service key is user name and the other key is conference room name. We used Web server as application server and Web browser as user interface.

**3.3 Real Space Information Management Models**

This section discusses the information management model for processing the real space information on the Internet. In order to provide the services based on a user’s existing location, it is necessary to manage the real space information somewhere. Real space information means the information that shows the state of objects, mainly the location of objects, in the real space. For example, it is the information in which room a certain user is located.

Before having a discussion about a model, it should be discussed what is the entity having the information, how is its information. The real-space information is the information tied with the objects in physical space. Therefore, there are two dimensions of real-space information. One is the entity itself, the other is the status related with the entity. To think with assumptions of this report, the entity means user, the status means user’s location.

Since many-to-many relationship is complicated, management is serious. In the real-space, certain man has relation with many places and certain area has relation to many humans, Figure 3.3 shows two informational spaces. An entity and its status are able to map to “User keyed Space” and “Location keyed Space”.

We divided models using these spaces. One uses User keyed Space, the other uses Location keyed Space. Models are shown as follows.

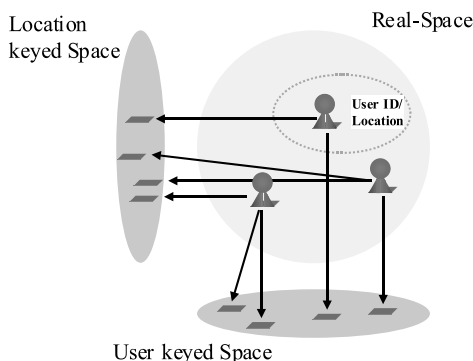


Fig. 3.3. Various Mapping from Real-Space to Informational Spaces

First, Personal Server Model proposed by SPEARS WG is mentioned and considered. In this report, Area Server Model is proposed with the result of considerations. Next, we will show that cooperation between both models is required for operation.

**3.3.1 Personal Server Model**

The Personal Server Model[311] is the processing model based on User keyed Space. In the model, each user owns his virtual host treating User keyed Space. We implemented system called Personal Server applying the processing model. Personal Server on the Internet is an expression of user in real space, stores communication identifiers, such as e-mail address.

It is not feasible to prepare every Personal Server as physical host. In our implementation, we attached many IPv6 Alias addresses to one host. Every Personal Server has different IPv6 address. These hosts have one database for all users and reply only certain user’s information when other application connects to certain Personal Server.

**3.3.2 Area Server Model**

The Area Server Model is also the model processing informational space. However, the difference is the viewpoint. This model is based on Location keyed Space. In this model, the locational entity store user’s identifications related

W I D E P R O J E C T 2 0 0 3 a n n u a l r e p o r t

with its location. The implementation applied this model is mentioned in Section 3.4.

### 3.3.3 Considerations for Management

#### Model

According to these two models, there are two informational spaces. One is about location information, and the other is about user or its ID. There are two requirements for real-space information processing system to realize real space service based on location. The system requires recognizing users related with certain location. Moreover, the system requires recognizing communication endpoint to provide service. The former means managing user ID on the namespace. The latter means that information depends on each application are necessary when these users are served actually. For example, mail delivery service needs that user's e-mail address, and automatic chat invite service needs user's Chat ID (like Jabber[140] ID).

Managing finite number of group that manages infinite number of entity is difficult. In contrast, managing infinite number of group that manages finite number of entity is feasible. The number of service type and identifier are infinite; but the number of service type that user uses is finite. To manage user's communication endpoint information, Personal Server Model is preferable. The number of User ID is infinite; but the number of user that tagged to certain location is finite. To manage users ID related with certain area, Area Server Model is preferable. Therefore, these two models have good and bad points, complement each other.

### 3.4 Design and Implementation of AreaD

This section describes the Area Server model compliant system design and implementation. The personal server manages communication endpoints each user has, to satisfy the requirement of the Real space service. To manage location-oriented information, we also designed and implemented a system named AreaD.

#### 3.4.1 Design

There are three requirements for conference support system. 1) Users in each conference room should be manageable and identifiable to the system in real time. The conference name of each conference is identifiable. 2) The name of the conference held in each room should be identifiable. 3) Time line for each conference on the system should be configurable to identifiable participants in each conference. The service, which is satisfying these requirements, will provide various location-aware-services.

Figure 3.4 shows the policy defined for managing information in order to define the data structure of this field test system. The data structure allows several conference rooms to exist at the same time. Each room is defined with start and end time of the conference, and the name of conference room. When a user enters or leaves a conference room, the system receives user's movement information as an event, and logs them to record which conference the user participated. Based on this information, the system maintains user ID for describing who participates the conference and a location information label for describing which conference room is used. These two information, user ID and location information label, are bound to the conference name.

Figure 3.5 shows an example of information management. In this example, there are two conference rooms, and two users as participants of the conference. Conference A and conference B were

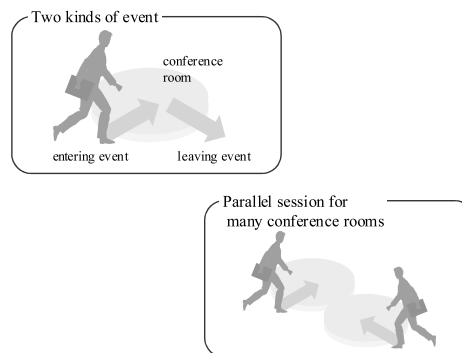


Fig. 3.4. Abstractions of User and Conference

held at conference room A, while conference C was held at conference room B. User 1 first participated in the conference A held at the conference room A, and then participated in the conference C held at the conference room B, and finally left the conference room. User2 only participate conference C held at the conference room B and then left the room.

To build components that manages any location-related information, it should provide search and registration function of a conference detail, entering and leaving of attendees. The conference detail will be consist of conference name, location information label, and start and end time of the conference. The registration of the entering and leaving events are based on the user's information and the RFID reader of which the user was detected from.

The search function provides a search for conference names currently being held, a search for participating user ID based on conference name, and real-time user ID search based on the location information label.

Search based on conference name, is assumed to prove all user IDs who participated the conference including those who left during the conference. Time line based search using location information label, is assumed to provide only those who are actually at the conference room. In figure 3.5, conference A is returned when searched based on conference room at t1.

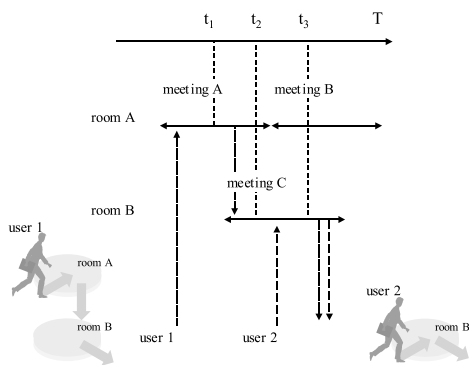


Fig. 3.5. Examples of Users' Behavior and its Operation

Similarly, ID of user1 is returned when searched with conference name as an argument for conference A.

When time line based search using location information label is used at t2, user ID can not be obtained with conference A as an argument, however ID of user1 is returned when searched with conference name as an argument for conference A.

3.4.2 Implementation

We have implemented AreaD as an information management system to support conferences. AreaD maintains binding between location information and user ID using information from the RFID-reader. AreaD itself is a middle-ware that does not directly provide real-space services to users, but is specialized for providing communication rendezvous. We have implemented the real-space services as other applications.

The field test system consists of the following 3 parts; protocol module for event triggered when entering or leaving a conference room, data management module for internal data base, and protocol handler for real-space services applications.

For the entering and leaving event protocol mode, we implemented a text-based protocol handler. The data management module maintains data structure as shown in Figure 3.6. The main data structure consists of data structure for real-time based search and data structure for conference name based search. We have also implemented a text-base event handler for search

Data Structure : Hash table for sessions / linked list for users

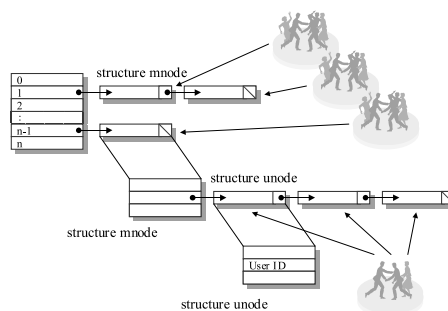


Fig. 3.6. Data Structure



protocol module as well.

Figure 3.7 shows screen image of AreaD. In this figure, it shows that some data structures are managed.

We implemented text-base protocol for the search. Figure 3.8 shows the search using TELNET. The main messages are MGET message, LGET message, and UGET message. When AreaD receives a MGET messages, it replies with the list of ongoing conference names. When AreaD receives a LGET message with location information label, it replies with the list of user ID of those at a certain conference room. A UGET message with conference name is used

for obtaining the list of user ID that is in the certain conference room.

AreaD can configure profiles of each conference beforehand. Configuration is described in configuration file read by the program. Figure 3.9 shows a sample configuration. Each conference described in a single line, each line consists with conference name, start time, end time, location information label and so on.

AreaD provides information of users in the conference room, and by using the search interface, location-aware services can be provided. The evaluation experiment was conducted at the WIDE camp held in March and September of 2003. The evaluation experiment system was built by adding the AreaD to the existing system. This allows using AreaD without any modification to the existing system. The real-space services are implemented as other application and described further in section 3.5.

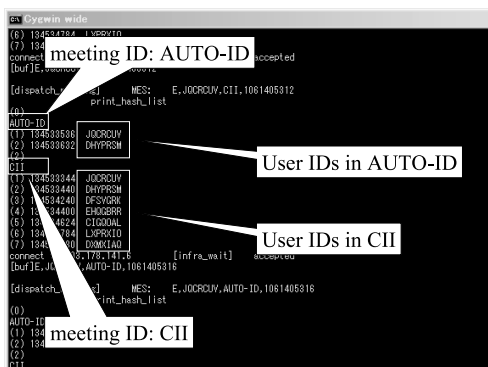


Fig. 3.7. The Screenshot of AreaD

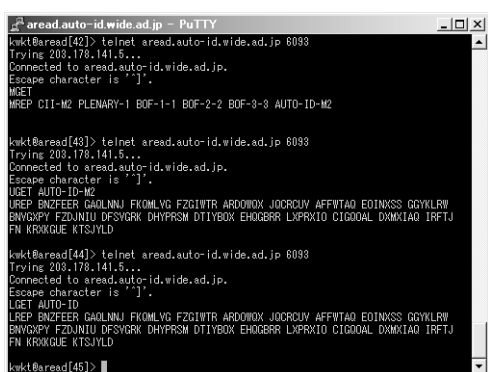


Fig. 3.8. Query and Reply Interface using TELNET

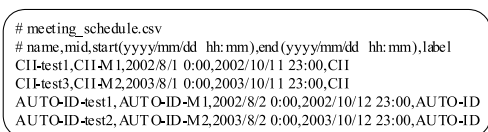


Fig. 3.9. Example of Configuration File

### 3.5 Result and Sample Applications

To establish the usefulness of AreaD based “Area Server Model” for conference support, we have implemented two kinds of sample applications; “Jabber Bot” and “WebDAV proxy”.

The “Jabber Bot” automatically creates a chat room and invite users who are in same physical space or participating same meeting. In addition, the “WebDAV Proxy” is a combination of the ad-hoc WebDAV directory for meeting participants to share their materials and a kind of annotation http proxy to add the WebDAV directory entry to all web pages that are participates accessed. Figure 3.10 and 3.11 shows the screenshots of these applications.

Figure 3.10 shows one of the Jabber-IM (Instant message) clients that is using “Jabber Bot” service. It shows that a user have been invited at Jabber conference named “bof1-1 room”. The Jabber Bot itself is one of the participants in this Jabber conference.

Figure 3.11 shows a typical action of the “WebDAV proxy” on the accessed web page. The

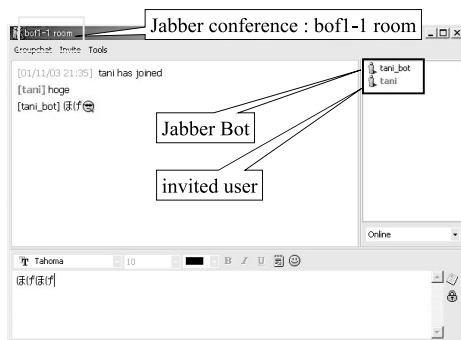


Fig. 3.10. The Screenshot of Jabber Application

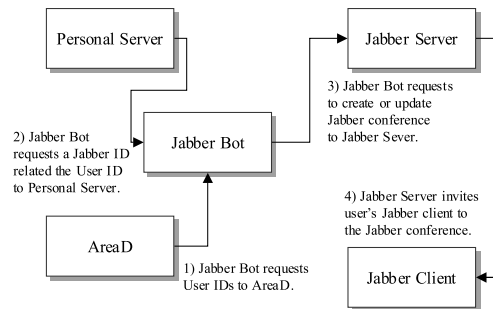


Fig. 3.12. Jabber Bot Components and its Relations



Fig. 3.11. The Screenshot of WebDAV Application

Conference is made, and when the meeting is over, Jabber Conference will be deactivated.

To recognize participants in target meeting room, Jabber Bot acquire the list from AreaD. Jabber Bot inquire of each user's Personal server to get one's Jabber ID. Then Jabber Bot invites the user to ad-hoc Jabber conference using the ID. Users who want to be invited from Jabber Bot just start a Jabber client application. Figure 3.12 shows these procedure and Jabber Bot components and its relations.

WebDAV proxy inserts a sub window (control panel), which has pointer to the ad-hoc WebDAV repository. The participants can access to common folder by just clicking this link.

### 3.5.1 Jabber Bot

Jabber is a set of streaming XML protocols and technologies that enable any two entities on the Internet to exchange messages and presence. The Instant Messaging (IM) service is major application of it.

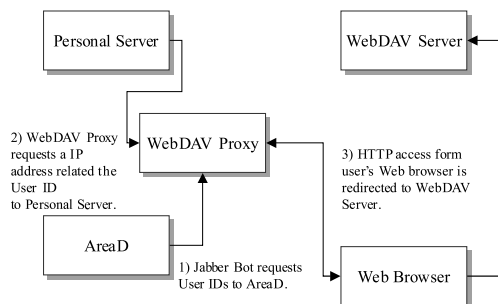
Since the IM users must exchange each ID (Jabber ID) before establish communication peer, it is difficult to organize ad-hoc group of meeting participants without full-mesh registration IDs among all of them. Jabber Bot provides automatic invite service for users who are in the same room using information served by AreaD. Thus this Jabber conference corresponds with an original meeting. When the meeting starts, Jabber

### 3.5.2 WebDAV Proxy

WebDAV is a kind of remote file system using web standard protocols. The repository location is represented as URL form.

To moderates a meeting efficiently, WebDAV proxy provides ad-hoc shared repository to the meeting participants. When a meeting is held, WebDAV Proxy creates ad-hoc WebDAV repository for the meeting and advertises the location to each participant via web page.

To advertise the location of ad-hoc repository, WebDAV Proxy works as HTTP proxy (to use this application, all participants should use it as HTTP proxy). The combination of AreaD, Personal Server and some support application to collect IP address is able to resolve a list of IP address that is used by each participant's host. Therefore, WebDAV proxy can selective annotation only for the meeting participates using the list. When WebDAV proxy is accessed from the listed IP address, WebDAV Proxy returns



**Fig. 3.13.** WebDAV Proxy Components and its Relations

annotated web page that is added some javascript and the link to the repository. According to this procedure, it offers corresponding WebDAV folder automatically. Figure 3.13 shows these procedure and architecture of WebDAV Proxy.

### 3.6 Conclusion

We addressed to two information spaces to enable real-space oriented data processing in this report, location keyed space and user ID keyed space. Using the information spaces, we show two management models for real space oriented data. The “Personal Server Model” represents the method to handle user-keyed information. In contrast, the “Area Server Model” addresses handling method of location-keyed information.

This report shows these two models are both required to realize location-aware and real-space computing environment. We designed and implemented an Area Server compliance service named AreaD. The AreaD provides location-aware rendezvous service among users.

To more concretization the information spaces discussion, there are some issues which need more works.

In this report, information management model is divided to two models, Personal Server Model and Area Server Model. In other words, AreaD manages user’s location information by area. Two applications using AreaD, Jabber Bot and WebDAV Proxy, are also implemented. By using these two applications based on Area Server Model,

the location-aware conference support system is provided to users.

### Acknowledgment

We thank co-operation staff, Kouji Okada, Daisuke Naruse, Shun Hirose, Ryusaburo Tani, et al., from Keio University and Nara Institute of Science and Technology and other universities in carrying out the field test. They supported our idea and implemented part of the system such as applications. The members of WIDE Project helped to discuss the architecture. Especially, Michimune Kohno, Mitsunobu Kunishi, Keisuke Uehara gave us good comments. We thank Noritsu Kudo, Shoko Mikawa, Masafumi Watari, Shigeya Suzuki and Yojiro Uo.

Product names mentioned herein may be trademarks of their respective companies.

---

## 第4章 インターネット上で実空間情報を収集・管理するフレームワーク (FIRSII) の提案と実装

---

**概要:** センサネットワークの普及に伴い、現実世界の物理的な位置に依存した情報が提供されている。しかし、センサ情報以外の実空間情報の収集、センサネットワークを構成する様々なセンサへアクセスするための仕様の違い、センサへアクセスする方法の多様化という問題が存在する。そこで、センサ情報以外の実空間情報を生成し、センサへアクセスするための仕様を吸収する仮想センサを提案する。また、センサの発見、仮想センサを生成する機構を含んだフレームワークである Framework of Integrated Real Space Information on the Internet (FIRSII) を提案する。実空間情報の通信部分では、既存の技術を利用することによりセンサへアクセスする方法を統一する。これにより、様々なセンサネットワークの中に存在するセンサ情報を再帰的に利用することが可能となり、異なるセンサネットワーク間においても様々な実空間情報の収集・加工・提供が可能となる。

#### 4.1 はじめに

インターネット上におけるサービスのほとんどが、仮想空間上で提供されている。例えば、メールを相手に送信する場合、相手の物理的な位置を意識することなく送信している。サービスが仮想空間上で提供される事により、利用者は現実世界における場所の概念を意識せずサービスを利用できる。しかしながら、特定の場所に存在する相手にメッセージを送るなど、物理的な位置に依存したサービスを利用することが困難である。一方で、特定の部屋に存在する人を把握するシステムのように、現実世界の物理的な位置に依存した情報をも提供可能とするシステムやアプリケーションが求められるようになってきた。ここで生身の人間がその中に物理的に存在し、活動する空間を実空間とし、実空間の中に存在する人や物、空間が持つ物理的な位置に依存した情報を実空間情報と呼ぶことにする。

実空間情報を提供するシステムやアプリケーションを構築するには、まず利用者に提供する実空間情報を取得しなければならない。実空間情報を取得する方法の一つとして、実空間情報を検知する装置であるセンサが利用されている。また、センサで取得した実空間情報は、インターネット上で収集されるようになってきた。インターネット上で実空間情報を収集することにより、様々な場所に遍在する情報を、実際にセンサが設置されている場所に行かずとも、リアルタイムに収集することが可能となった。さらに収集した実空間情報に付加価値をつけた新たな実空間情報が様々な場所で利用されてきている。

しかし、センサを収集する手段がアプリケーションによって異なるため、センサで取得した情報以外の実空間情報の収集が困難、センサへアクセスするための仕様の違い、センサへのアクセス方法の多様化などの問題が残されている。

したがって、センサで取得したセンサ情報以外の実空間情報を生成し、センサへアクセスするための仕様の違いを吸収する仮想センサを提案する。仮想センサは、現実の世界に存在するセンサを抽象化し、抽出したセンサの機能をもち、センサで取得した情報を収集・加工・提供する。また、現実世界に存在するセンサだけでなく、仮想センサを加えた各センサの発見や仮想センサを生成する機構を含んだフレームワークである Framework of Integrated Real Space Information

on the Internet ( FIRSII ) を提案する。さらにセンサへのアクセスを統一するために、FIRSII では実空間情報の通信部分を Simple Network Management Protocol ( SNMP ) に対応させた。これにより、様々なセンサネットワークの中に存在する実空間情報を再帰的に利用、センサ情報以外の実空間情報を生成することが可能となり、様々な実空間情報の収集・加工・提供が可能となる。また、センサへアクセスするための仕様の違いを吸収し、センサへのアクセス方法を統一することにより、現実世界の情報を利用する者は他のセンサネットワークで管理されているセンサ情報を容易に利用、連携することが可能となる。

本提案の基となる提案は、2003 年 11 月 18 日に、情報処理学会ユビキタスシステムコンピューティング第二回研究会で発表した。

#### 4.2 センサの抽象化による仮想センサの提案

センサは、実空間情報の中のある特化した情報を取得するために設計されている。このため、センサ情報以外の実空間情報を収集することが困難である。しかし、実空間情報の中には、既設されているセンサで取得不可能な情報も含まれているため、新たなセンサを設置しなければならない。さらに、センサへアクセスするための仕様の違いや、取得する実空間情報の種類が同じであっても単位系が異なる問題がある。このため、仕様の異なるセンサから収集した実空間情報を、アプリケーション毎で吸収しなければならない。

したがって、センサで取得可能な情報以外の実空間情報を生成し、各センサの単位系の仕様を吸収する機能を持つ仮想センサを提案する。仮想センサは、実際に存在するセンサと同じ機能を持ち、同じプロトコルでアクセスすることが可能である。ここで、実空間上に存在するセンサと仮想的に存在するセンサとを区別するために、実空間上に存在するセンサを Real Sensor、仮想的に存在するセンサを Virtual Sensor と呼ぶことにする。Real Sensor と Virtual Sensor の関係を図 4.1 に示す。

図 4.1 の実空間 ( Real Space ) には複数の Real Sensor が設置されており、Real Sensor は Sensor Network を形成している。Virtual Sensor は、仮想空間 ( Virtual Space ) に抽象化した Real Sensor や他の Virtual Sensor から実空間情報を収集する。さ

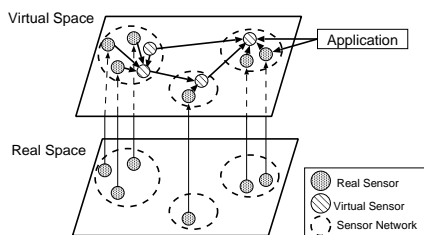


図 4.1. Real Sensor と Virtual Sensor の関係

らに、センサへアクセスするための仕様の違いを吸収、新たな実空間情報を生成する。Application は実空間情報を利用するアプリケーションであり、Real Sensor や Virtual Sensor より実空間情報を取得する。実空間情報をやり取りする場合、既存のプロトコルを利用することによりセンサへのアクセス方法を統一する。

#### 4.3 FIRSII の提案

Real Sensor や Virtual Sensor へアクセスし、センサが取得した実空間情報を利用するためには、各センサの物理的な位置およびネットワーク上の位置を把握する必要がある。また、仮想センサを登録し、生成する機構が必要である。そこで、インターネット上の Real Sensor や Virtual Sensor から実空間情報を収集し、システムやアプリケーションに実空間情報を提供するためのフレームワークである FIRSII を提案する。これにより、生成した Virtual Sensor から取得可能な実空間情報が再帰的に利用可能になる。また、センサネットワーク内や各センサネットワーク間で実空間情報のやり取りが容易となる。

次に想定環境について述べる。FIRSII では、複数のセンサネットワークが構築されており、センサネットワーク間で実空間情報がやり取り可能な環境を想定している。やり取りされる実空間情報は、インターネット上で収集することを前提としている。そこで、現実世界に存在するセンサ自身が、ネットワークに接続されている、もしくはネットワークに接続可能な他の機器を介してネットワークに接続されている環境を前提とする。

#### 4.4 FIRSII の設計

本節では、FIRSII の機能要件を述べ、FIRSII の構成を述べた後に、各構成要素について詳述する。

##### 4.4.1 FIRSII の機能

本項では、センサ情報以外の実空間情報の収集が困難、センサへアクセスするための仕様の違い、センサへのアクセス方法の多様化というインターネット上で実空間情報を収集・加工・提供する場合における 3 つの問題を解決するための FIRSII 全体の機能について述べる。これらの 3 つの問題を解決するための機能要件に、実空間情報を管理するための機能要件を含めた 4 つの機能要件を以下に示す。

1. センサ情報以外の実空間情報の生成
  2. センサへアクセスするための仕様の違いの吸収
  3. センサへのアクセス方法の統一
  4. 実空間情報の物理的な位置の管理
- 以下に 4 つの機能要件を詳述する。

##### 1. センサ情報以外の実空間情報の生成

本提案において、実空間に存在する情報の取得には、センサを利用することを前提としている。センサは、実空間情報の中にある特定の情報のみを提供する。本提案においてセンサで取得不可能な情報に対しては、Virtual Sensor を利用することにより、Real Sensor で取得した情報以外の実空間情報を Virtual Sensor より生成することが可能となる。

##### 2. センサへアクセスするための仕様の違いの吸収

アプリケーションが実空間情報を収集する場合、各センサの様々な仕様に応じてセンサへアクセスする必要がある。また、各センサが取得する実空間情報が同じであっても、単位系が異なる場合が存在する。例えば、同じ温度という実空間情報を取得する温度計がある場合、温度計が取得した温度の単位が摂氏であったり華氏であったりする場合がある。そこで Virtual Sensor の構成要素の一つである Function にセンサへアクセスするための仕様、センサが取得する実空間情報の単位の違いを吸収する加工規則を記述する。これにより、センサへアクセスするための仕様の違いを吸収し、アプリケーションは、センサの仕様を意識することなく実空間情報を収集することが可能となる。

##### 3. センサへのアクセス方法の統一

センサ情報以外の実空間情報の生成、センサへアクセスするための仕様の違いを吸収するために Virtual Sensor を提案した。しかし、Virtual Sensor から実空間情報を収集する場合において、

既存のシステムやプロトコルで実空間情報が取得できなければ、汎用性を損なう恐れがある。そこで FIRSII では、実空間情報の収集する部分を既存のシステムやプロトコルに対応させる。

4. 実空間情報の物理的な位置の管理

実空間情報を利用するアプリケーションへ実空間情報を提供する場合、実空間情報が存在する物理的な場所が重要となる。ここで実空間情報が存在する物理的な場所は、実空間情報を収集した Real Sensor や Virtual Sensor の物理的な場所となる。そこで FIRSII は、Real Space Manager (RSM) を利用し、実空間情報が収集された物理的な場所および、実空間情報を収集した各センサのネットワーク上の位置を管理する。これにより、実空間情報が取得された物理的な場所を知ることが可能となる。

4.4.2 全体の構成

FIRSII は、Real Sensor、Pseudo Sensor、Virtual Sensor、Real Space Manager、Application で構成される。図 4.2 に FIRSII を利用したアプリケーションを含めた全体構成を示す。また、図 4.2 に記述している数字は、構成要素間の処理の流れを示したものである。処理の流れについては後述する。

以下に各構成要素の機能を詳述する。

• Real Sensor

Real Sensor は、実空間に存在する情報を取得するセンサである。センサの管理者は、公開してもよい Real Sensor を RSM に登録する。Real Sensor を登録する際に必要な情報は、Real Sensor が存在する物理的な位置、Real Sensor がネットワーク上に存在する場所を表す IP ア

ドレス、Real Sensor を識別するための識別子、Real Sensor が取得可能な実空間情報の名前、取得した値、取得した値の単位、Real Sensor が実空間情報を取得するための更新時間、モード情報である。モード情報には、パッシブモードとアクティブモードの 2 つがある。パッシブモードとは、センサが実空間情報を要求する信号を受信した場合、取得した実空間情報を送信するモードのことである。次にアクティブモードとは、センサが取得した情報を要求する信号を受信することなく、センサ自身がセンサ情報を発信するモードのことである。

• Pseudo Sensor

Pseudo Sensor は、Real Sensor がネットワークに接続する機能を持たない場合、Real Sensor の代わりにネットワークに接続する通信機能を持つ機器である。

• Virtual Sensor

Virtual Sensor は、Real Sensor や他の Virtual Sensor の値を収集・加工し、Real Sensor と同じ機能を持つセンサである。ここで Virtual Sensor で加工する部分を Function と呼ぶ。この Function には、加工規則が定義されており、Real Sensor や他の Virtual Sensor から既存のプロトコルを用いて収集した実空間情報の値を加工する。実空間情報の通信部分を既存のプロトコルに対応させることにより、既存の技術を用いて Real Sensor から実空間情報を収集することが可能となる。さらに Virtual Sensor は、自動的に消滅する機能を持つ。Virtual Sensor も、Real Sensor と同様に FIRSII の構成要素である RSM により登録されるが、登録時に Release Time として消滅時間を設定する。つまり Virtual Sensor が生成された後、登録された消滅時間が経つと、自動的に Real Sensor や他の Virtual Sensor へ、実空間情報を要求することを止め、データベースより消滅し、公開されなくなる。

Virtual Sensor は、Function を持ち、Function に単位系を吸収する加工規則を記述すれば、センサが取得する実空間情報の単位系の仕様を吸収することが可能となる。また、Application が実空間情報を収集する場合、センサの仕様を意識することなく実空間情報が取得可能となる。

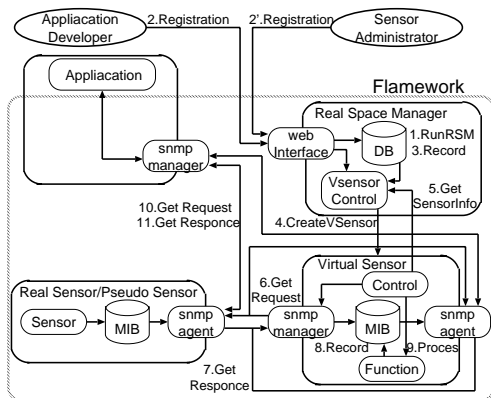


図 4.2. FIRSII

- Real Space Manager (RSM)

RSM は、Real Sensor、Pseudo Sensor、Virtual Sensor の物理的な位置および仮想的な位置を把握する機能をもつ。この機能により Application や Virtual Sensor は、Real Sensor や Virtual Sensor へ実空間情報の送信要求を出すことが可能となる。また RSM は、Virtual Sensor の登録要求を受けた場合、登録された内容を検査し、登録、生成する機能を持つ。

- Application

実空間情報を利用するアプリケーションであり、RSM から Real Sensor や Virtual Sensor の物理的な位置情報とネットワーク上の位置を検索する。Application は、Real Sensor や Virtual Sensor のネットワーク上の位置情報を RSM から取得した後、Real Sensor および Virtual Sensor から実空間情報を収集する。

#### 4.4.3 処理の流れ

FIRSII での処理の流れは、実空間情報の登録、収集、加工、提供の大きく分けて 4 つにわかれる。以下に処理の流れを詳述する。

##### < 実空間情報の登録 >

##### 1. RSM の起動

RSM は、Sensor および Virtual Sensor のセンサ情報が登録されるのを待つ。

##### 2. Virtual Sensor の登録

センサ情報の登録には、Web による登録方法とコマンドによる登録方法の 2 つの方法を利用することが可能である。

- Web による登録

登録の際にはセンサ情報を入力し、送信ボタンを押すと入力したセンサ情報が RSM に送信される。入力内容のチェックを受け、RSM が正しいと判断すれば登録することが可能である。

- コマンドによる登録

登録用コマンドは、RSM のネットワーク上のアドレスおよび、登録に必要な項目を入力したファイル名を引数として実行し、登録する。

##### 3. RSM によるセンサ登録情報の検査

RSM は、センサ登録時に入力される登録項目に記述される書式、他のセンサ識別子との重複

を検査する。RSM が検査し、センサ情報を正しいと判断した場合、センサ情報をデータベースへ登録する。

##### < 実空間情報の収集 >

##### 4. Virtual Sensor の生成

Virtual Sensor が RSM に登録されると、Virtual Sensor が生成される。

##### 5. 実空間情報を収集する対象であるセンサにおけるネットワーク上のアドレスの取得

Virtual Sensor は、収集するセンサの存在するネットワークの位置情報を表すアドレスを RSM から取得する。

##### 6. 実空間情報取得要求の送信

Virtual Sensor は、Real Sensor、Virtual Sensor との通信プロトコルについて Simple Network Management Protocol (SNMP) に対応し、Get Next Request を送信する。

##### 7. 実空間情報取得要求の返信

Get Next Request を受信した Real Sensor および Virtual Sensor は、Get Responce と共に実空間情報を送信する。但し、アクティブセンサの場合には、Trap を用いて Virtual Sensor に実空間情報を送信する。

##### 8. 実空間情報の保存

Virtual Sensor が取得した実空間情報は、データベースへ登録される。

##### < 実空間情報の加工 >

##### 9. 実空間情報の加工

Virtual Sensor は、Virtual Sensor がもつ加工規則である Function に、取得した実空間情報の値を渡す。値を渡された Function は、登録された加工規則に基づき実空間情報を加工する。

##### < 実空間情報の提供 >

##### 10. 実空間情報取得要求の送信

Application は、取得したい実空間情報を持つ Real Sensor、Virtual Sensor に対して Get Request もしくは Get Next Request を送信する。

##### 11. 実空間情報取得要求の返信

Get Request もしくは Get Next Request を受けた Real Sensor、Virtual Sensor は、指定された Object ID を基に、Get Responce を Virtual

Sensor、Application に返信する。

これにより Application は、公開された Real Sensor の情報から Virtual Sensor を生成し、新たな実空間情報を収集することが可能となる。また、Application は既存のプロトコルを用いることにより、Real Sensor、Virtual Sensor の違いを意識せず、Virtual Sensor は、あたかもその空間に存在しているかのように Real Sensor と同様な方法で実空間情報を取得することが可能となる。

#### 4.4.4 実装

実装環境として、Real Sensor、Pseudo Sensor、Virtual Sensor、Real Space Manager を Linux Redhat 9 上の java 1.4.2 を用いて実装した。Application に関しては、FreeBSD5.1-RELEASE 上の java 1.4.2 を利用して実装した。また、Real Sensor、Pseudo Sensor、Virtual Sensor、Application において実空間情報を収集するための SNMP Agent、SNMP Manager、MIB は、java 1.4.2 を用いて実装した。センサ登録には、Web サーバとして Redhat 9 上で Apache 1.3.29 を用いた。センサ情報を登録するデータベースには、PostgreSQL 7.4 を用い、データベースとの通信部分では PHP 4.3.4 を用いた。

#### 4.4.5 サンプルアプリケーション

本項では、作成したサンプルアプリケーションである不快指数および熱指数表示アプリケーションについて述べる。

温度計、湿度計が設置されている実空間において、温度計からは温度が、湿度計から湿度が取得可能である。但し、温度計、湿度計が設置されている実空間では、温度と湿度のみ取得することが可能である。そこで FIRSII の一部である Virtual Sensor を利用し、不快指数センサを生成する。不快指数センサは、取得した温度および湿度を不快指数へ加工する。不快指数へ加工するための規則は、Virtual Sensor が持つ Function に記述されている。またサンプルアプリケーションでは、温度計から取得する温度の単位の仕様を吸収する仮想センサを作成し、単位を統一した温度と湿度から加工した不快指数の値を持つ不快指数センサ、熱指数を値に持つ熱指数センサを作成する。サンプルアプリケーションを作成することにより、既に設置されている Real Sensor から取得する実空間情報から、新たな実空間情報が Virtual

Sensor により取得可能なことを示す。また、様々な仕様のセンサが設置された場合、設置されたセンサの仕様を吸収し、仮想センサが再帰的に利用可能であることを示す。

Virtual Sensor である不快指数センサを Web 上で登録すると、登録したセンサ情報を基に、不快指数センサオブジェクトが生成される。不快指数センサオブジェクトが生成されると、不快指数センサオブジェクトは、java で実装された snmpwalk クラスを呼び出し、SNMP を利用して温度計および湿度計からそれぞれ温度および湿度を収集する。収集する間隔は Virtual Sensor が登録される際に UpdateTime として登録されており、その UpdateTime 毎に、温度、湿度を収集する。Virtual Sensor は、温度、湿度を収集すると、Function メソッドを呼び出し、温度、湿度を加工し不快指数を生成する。また熱指数センサは、要素として温度と湿度を用いるが、温度は摂氏ではなく華氏を用いる。そこで、単位が摂氏である温度を華氏に変換する華氏温度センサを Virtual Sensor として新たに登録した。これにより熱指数センサは、不快指数センサと同様に、華氏温度センサおよび湿度センサからそれぞれ温度、湿度を収集し、加工することにより、熱指数という値を持つことが可能となった。このように FIRSII では、各センサの仕様の違いを吸収することが確認できた。

本アプリケーションは、2003 年 12 月に東京大学で開催された WIDE Project 主催の 12 月研究会においてデモを行った。FIRSII を利用して作成したサンプルアプリケーションである不快指数および熱指数アプリケーションの表示画面を図 4.3 に示す。表示

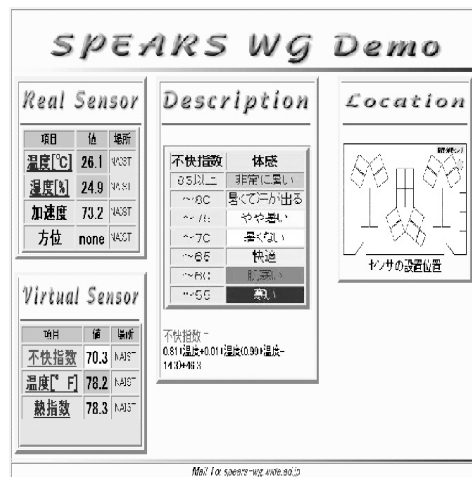


図 4.3. 不快指数、熱指数の表示



画面は4つの要素にわかれている。具体的には、Real Sensorの値を示すReal Sensor、Virtual Sensorの値を示すVirtual Sensor、Real SensorおよびVirtual Sensorの詳細を示すDescription、Real SensorおよびVirtual Sensorの設置場所を示すLocationに分かれている。

#### 4.5 FIRSIIの評価と考察

本節では、FIRSIIを定性的に評価および定量的に評価し、考察する。

##### 4.5.1 定性評価

FIRSIIを定性的に評価するために、以下3つの機能が実装されていることにより、既存の問題点が解決することを示す。以下に評価項目を示す。

- センサ情報以外の実空間情報の生成
 

提案したVirtual Sensorが、Real Sensorや他のVirtual Sensorの実空間情報を収集・加工することにより、センサ情報以外の新たな実空間情報を生成可能なことが確認できた。また、サンプルアプリケーションにより、Virtual Sensorが持つFunctionで加工、生成された新たな実空間情報が、アプリケーションに提供可能であることを示すことができた。
- センサへアクセスするための仕様の違いの吸収
 

新たに単位や仕様の異なる温度計を設置し、提案したVirtual Sensorが持つFunctionに、仕様を吸収するための関数を記述した。これにより、温度計の仕様、特に単位の違いを吸収することが確認できた。
- センサへのアクセス方法の統一
 

実空間情報をやり取りする通信プロトコルとしてSNMPを利用し、センサ同士およびセンサアプリケーションとの間で、実空間情報をやり取り可能であることが確認できた。これにより既存のプロトコルによりセンサへのアクセス方法を統一することを示すことができた。

しかし、FIRSIIを用いて実空間情報を利用するアプリケーションを作成する場合、アプリケーションで実空間情報を収集・加工・提供する従来のシステムに比べて、RSMを設置しなければならない。また、実空間情報を加工する部分と、提供する部分を分離したVirtual Sensorを介するだけの通信コストがかかる。したがって、仕様が統一されているセン

サが設置されている環境や、設置されているセンサを再帰的に利用せず、アプリケーションのみで実空間情報を提供する場合には、アプリケーションで実空間情報を収集・加工・提供の方が望ましい。そこで、アプリケーションで収集・加工・提供する従来のシステムを基準とし、FIRSIIを利用したシステムの性能の低下率を定量的に評価する必要がある。

##### 4.5.2 定量評価

FIRSIIの定量的な評価では、アプリケーションで収集・加工・提供する従来のシステムを基準とし、FIRSIIを利用したシステムの性能の低下率を定量的に評価する。

具体的には、Real Sensor、Virtual Sensor、Applicationがそれぞれ増加した場合において、従来のシステムで実空間情報を収集・加工・提供する時間を測定する。従来のシステムとは、実空間情報をアプリケーション毎に収集・加工・提供するシステムである。測定後FIRSIIを用いたシステムで実空間情報を収集・加工・提供する時間を測定し、従来のシステムを用いて要した時間に対し、FIRSIIを利用したシステムで要した時間の割合をFIRSIIの性能として測定する。

実空間情報を収集・加工・提供に要する時間は、従来のシステムおよびFIRSIIを利用したシステムにおいて、それぞれ1000回測定した平均の時間とする。FIRSIIの性能を測定することにより、アプリケーションに応じて、RSMの連携およびServerを用いないVirtual Sensorを分散させる指標とすることが可能となる。特にリアルタイムな実空間情報が必要とするアプリケーションを作成する場合の重要な指標となる。

図4.4、図4.5にReal Sensorが増加した場合における測定結果を示す。

図4.6、図4.7にVirtual Sensorが増加した場合における測定結果を示す。

図4.8、図4.9にApplicationが増加した場合における測定結果を示す。

##### 4.5.3 考察

FIRSIIを利用したシステムにより、定性的な評価から従来のシステムで問題であったセンサ情報以外の実空間情報の収集が困難、センサへアクセスするための仕様の違い、センサへのアクセス方法の多様化とい

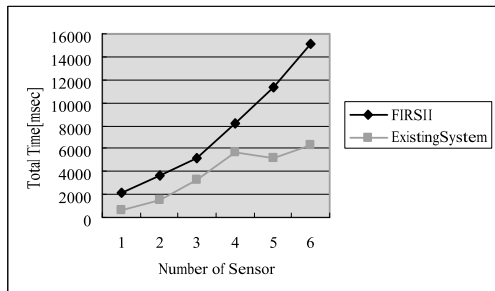


図 4.4. Real Sensor が増加した場合の実空間情報の収集・加工・提供時間

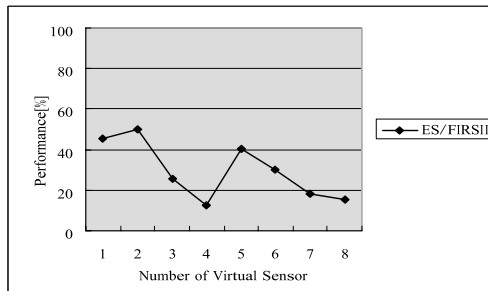


図 4.7. Virtual Sensor が増加した場合の FIRSII の性能

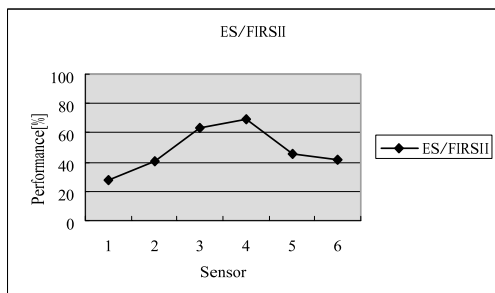


図 4.5. Real Sensor が増加した場合の FIRSII の性能

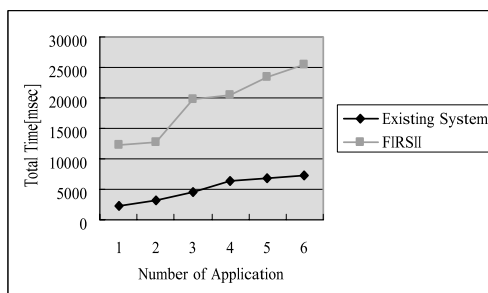


図 4.8. Application が増加した場合の実空間情報の収集・加工・提供時間

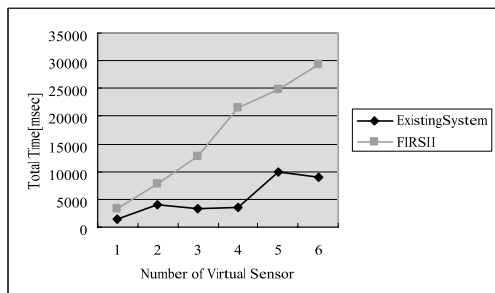


図 4.6. Virtual Sensor が増加した場合の実空間情報の収集・加工・提供時間

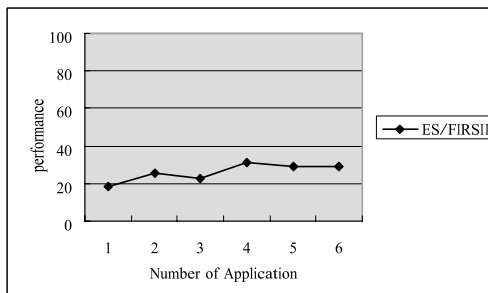


図 4.9. Application が増加した場合の FIRSII の性能

う問題を解決することを示せた。しかし、定量的な評価から Real Sensor を増加した場合における FIRSII の性能には規則性がなかった。Virtual Sensor が増加した場合には、Virtual Sensor が増加するほど、ほぼ性能が落ちていくが規則性がなかった。Real Sensor、Virtual Sensor が増加していくにつれて FIRSII の規則性が見られなかった原因は、Virtual Sensor の実装によるものと思われる。Virtual Sensor が生成される毎にスレッドが生成される実装になっている。このため Virtual Sensor の実空間情報の要求や返信が重なった場合、実空間情報の収集・加工が遅くなる。逆に処理が重ならない場合、実空間情報の収集・加工が早くなる。このような実装上の問題によ

り FIRSII の性能に規則性がなかったと考えられる。しかし、FIRSII の性能は 50% にも達していないことが確認できた。Application が増加した場合には、Application の増加に関わらず FIRSII の性能はほぼ 30% であった。

以上のような結果は、今回作成した不快指数や熱指数を表示するアプリケーションのような即時性が求められないアプリケーションでは有効であるが、即時性を求めるアプリケーションでは問題であり、システムを改善しなければならない。

## 4.6 おわりに

本節では、FIRSIIの今後の課題を述べ、まとめる。

### 4.6.1 今後の課題

本項では、今後の課題について述べる。

- 場所の名前管理

現在、センサを設置している場所の名前は、利用する側が既に知っているものと想定している。しかしアプリケーションにより、名前付けの方法や場所の範囲の粒度が異なる。したがって、実空間情報の場所を表す場所の表現方法を検討し、Virtual Sensorが要求するReal SensorやVirtual Sensorの適用および発見方法を検討していかなければならない。

- Real Space Manager (RSM) の発見

RSMの発見に関しては、既に発見されているものとしてフレームワークを設計している。しかし、センサの物理的に存在している場所や、ネットワーク上の位置を管理するRSMが発見できなければ、実空間情報を取得する各センサへアクセスすることが不可能である。今後は既存のサービス発見技術を参考としRSMの発見を検討しなければならない。

- RSMの分散化

FIRSIIでは、実空間情報を利用する者は、RSMが管理している場所を既知としている。今後、Virtual Sensorの連携が複数となり、複数のセンサネットワークで実空間情報がやり取りされる場合、他のRSMとの連携が必要となってくる。

- Virtual Sensorの分散化

FIRSIIでは、Serverを用意し、RSMを起動させておくことによりVirtual Sensorを管理している。しかし、定量評価にもあるように、従来のシステムの性能の50%を下回る性能であった。センサネットワークの連携を想定しているため、Serverを用いずVirtual Sensorを分散させることにより、FIRSIIの性能を向上させていく必要がある。

- 実空間情報の取り扱い

実空間情報の中には、プライバシーに関わる情報も含まれることがある。FIRSIIでは、SNMPに実装されているコミュニティ名およびIP Addressによりアクセス制限することが

可能である。しかし、IP addressの詐称などにより、プライバシーが侵害される恐れがある。したがって、実空間情報の取り扱いについては、さらに考慮しなければならない。

### 4.6.2 まとめ

本提案では、実際に存在しているReal Sensorを抽象化し、Real Sensorの機能をVirtual Sensorに継承することにより、新たに加工規則の記述およびReal Sensorや他のVirtual Sensorから実空間情報を収集することが可能となった。これにより、実空間情報を利用するアプリケーションは、既存のセンサだけでなく、公開されている他のセンサネットワークに設置されているReal Sensorや、Virtual Sensorからも実空間情報を収集することが可能になった。実空間情報を収集する際にも、既存のセンサの仕様や単位系の違いをVirtual Sensorで吸収することにより、仕様の違いを意識する必要がなくなった。さらに、収集した実空間情報を加工することにより、新たな付加価値を付け、再び公開することにより、Virtual Sensorが再帰的に利用することが可能となった。

