International Journal of Energy, Information and Communications
Vol.8, Issue 1 (2017), pp.15-28
http://dx.doi.org/10.14257/ijeic.2017.8.1.02

Development of a Node Cooperation Method for Location Change of Sensor Data Processing

Shintaro Imai[1]*, Haruka Shimizu[1], Bingxi Zhao[1], Yoshikazu Arai[1] and Toshimitsu Inomata[1]

[1]*Graduate School of Software and Information Science,*
Iwate Prefectural University
s-imai@iwate-pu.ac.jp

Abstract

In sensor systems, sensor data processing locations are classifiable roughly as sensor nodes, a local server near sensor nodes, and an external server such as a cloud server. These processing locations have their respective benefits and shortcomings. It is possible to process sensor data without applying a large load to a specific node by selecting the data processing location and processing contents depending on the environment. This study was undertaken to realize a sensor system that can adapt to the environment. We propose a method to select the data processing location and processing content autonomously according to service characteristics, acquired data types, necessary data processing contents, and the environment. The system used with this method provides service using agent organization. When an environmental change occurs, the system performs agent re-organization to adapt to the environment. We implement a prototype system to evaluate this method and perform some initial experiments using this prototype system.

Keywords: *Sensor system, Sensor network, Multi-agent system*

1. Introduction

Recently, many proposed services use large amounts of data from a human or object in real space using sensor nodes. For instance, a service acquires a wearer's biological data from a wearable device and provides health advice. Sensor systems are also provided for measuring the home, the community, and the corporate environment. These services will continue to increase in the future, necessitating the continued increase of sensor data acquired from terminals.

The transmission of large amounts of data to a particular server adds a large load to the communication path and storage. Therefore, if necessary, the system must process the data in the neighborhood of the sensor node before sending data to the network.

Given such circumstances, Edge-Heavy Data [1] and Fog Computing [2] were proposed. Edge-Heavy Data describes many data that are managed in the sensor and in its vicinity (the network edge). Fog Computing is aimed at reducing communication costs by placing a processing server near the network termination.

Service characteristics, resources of the device, and data processing methods required for the service are regarded as selection criteria of the data processing location. For example, data processing in a server located near the sensor nodes is effective when a service requires high response speed. For a system that acquires data from a wide range and integrates them, data processing on a server on the internet is effective.

In addition, handling data during collaboration among different systems is expected to occur as the sensor system is extended. For this reason, if the data processing location is fixed, then the system cannot be sufficiently adapted.

In this paper, we propose a sensor system that selects the data processing contents and processing location autonomously to improve adaptability of the system to the environment. These are determined by the services, the data characteristics, the terminal and network environment, *etc*. This system can continue services by adapting to a changing device and network environment, and avoid load concentration of a particular node or network. In this study, the function of the system is realized by cooperation among mobile agents. They are disposed in the sensor node and sensor system server.

First, we describe related works in Section 2. Next, we describe organizational and behavioral design of the proposed method in Section 3. In Section 4, we show the result of experiment by prototype system and its evaluation. Finally, we conclude in Section 5.

2. Related Works

2.1. Target Environment

This study specifically examines a system comprising various sensor nodes and servers connected through a network. They operate cooperatively. The system provides various services to users. The target environment of this study is presented in Figure 1.

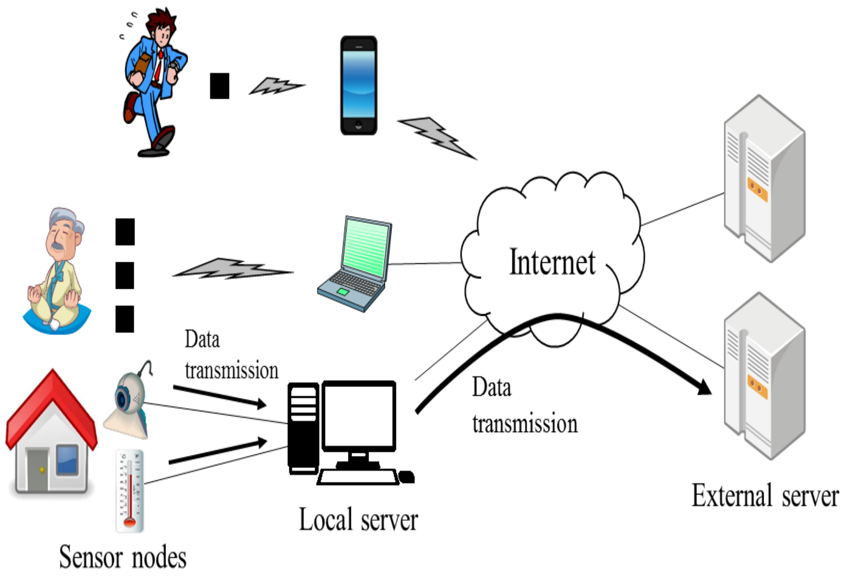


Figure 1. Intended Environment of this Study

Numerous sensor nodes that are established in the environment acquire various data. If necessary, the sensor node processes the acquired data. A sensor node sends the acquired data and the processing result to the server existing in its vicinity. This server is called a local server in this study. Not only sensor data dedicated server but also PCs and smartphones are treated as a local server. The smartphone device acts as a local server and a sensor node. We designate the network between the sensor nodes and the local server as an inside network. Wired and wireless LAN, Bluetooth and Zigbee, etc. are used in an inside network. A local server sends the received data to a server on the internet, such as a cloud server.

2.2. Functional Requirements

This section summarizes the functional requirements of the sensor system that is capable of operating adapted to the environment. The system must be designed considering the following three properties:

(R1) Selection of the sensor data processing content and location response to the service characteristics: The amounts and kinds of data processing necessary for services differ. For example, the system response speed required for the service and a data collecting range affects selection of the sensor data processing location. If a system requires high response speed, as a person-monitoring system might, then processing data in the neighborhood of the sensor is suitable. If a system requires acquisition of widely various data, as a temperature measurement system might, then processing data on an external server is suitable.

(R2) Selection of the sensor data processing content and location response to environmental change: The system must determine the sensor data processing contents and location considering the resources of each sensor node and the network. Additionally, it is necessary to respond to environmental changes while providing service. For example, when the data processing terminal's CPU load increases, a system that changes the data processing location is suitable.

(R3) Expandability with addition of a service and a terminal: Services provided by the sensor system are expected to increase in the future. The terminal type used in the service will also increase. These services and terminals have different characteristics. Therefore, the system must enable flexible extension.

From these functional requirements, we place the mobile agent to the sensor nodes and the server in this study. Mobile agents provide services in a coordinated manner. Software agent's characteristic such as autonomy, social ability and reactivity is effective to satisfy the functional requirements. (R1) is realized by agent organization. Agents have characteristics of a service, characteristics of a data processing method, and device information. The system configures agent organization based on this characteristic and provides service. This agent organization realizes appropriate the sensor data processing content and location. (R2) is realized by agent re-organization. When agents observe an environmental change during providing a service, agent re-organization is performed. As a result, the system can be a new agent organization that responds to the new environment. (R3) is realized by adding a new agent.

2.3. Related Works

2.3.1. Classification of sensor systems based on the data processing location: This section presents comparison of a system for providing services using data acquired from sensor nodes based on the sensor data processing location. The following three locations might be used to process the sensor data: sensor nodes, an external server on the internet, and the neighborhood of the sensor nodes.

For processing data in a sensor node, one can reduce the data size before sending it to the network, thereby reducing the load to the inside network and the internet. This case offers excellent responsiveness. However, complicated data processing in a low-performance sensor node and service provision using the acquired data by plural nodes are difficult. Oracle proposed Oracle Event Processing for Oracle Java Embedded [3], which is a platform used to process the events and data in a sensor node.

For processing data in an external server, the system can process data using a high-performance server and provide a service using data from widely various sensor nodes. However, such a system adds a heavy network load and has low response speed. Many services that process data in an external server have been proposed.

Edge-Heavy Data [1] and Fog Computing [2] were proposed to realize both data processing of certain scale and network load reduction. These both offer the benefits of the previous two methods. Fog computing is designed to reduce the load on the network and an external server by processing small amounts of data, limited data, and real-time processing data.

Three approaches present benefits and shortcomings. Suitable data processing location differs depending on the service, devices, and the network environment. In addition, suitable data processing locations are changed by an environmental change during providing the service. Therefore, we propose a method to change the data processing location in response to an environmental change. This method is realized by the multi-agent system. Agents are placed in the server and sensor nodes of the sensor system.

2.3.2. Sensor Systems using a Multi-Agent Technology: Research that applies multi-agent technology to the sensor system to improve the adaptability and data analysis accuracy on the environment is performed. In addition, agent platforms for the sensor system are proposed.

Yoshida proposed PIAX to operate and search distributed sensor nodes in a wide area [4]. PIAX is the agent-based P2P platform. The agents operate actively in the PIAX platform. The system can search for an agent based on the geographical location of the user and characteristics of the information in the PIAX platform.

An agent-based Ambient Assisted Living (AAL) system was proposed [5]. The system uses the agent technology to get a function in response to an environmental change. Consequently, the system behaves as a self-managing system. The authors presented architectures and tasks for different devices that compose the system, and presented validation results of the system implementation. However, this system is only useful for an AAL system. Therefore, it is difficult to apply the proposed system to diverse services.

Some studies have examined the use of multi-agent technology in the field of Internet of Things (IoT). One study [6] examines creation of a multi-agent system that suits the needs introduced by the IoT expansion, thereby being capable of overseeing Big Data collection and processing and also of maintaining the semantic links between the data sources and data consumers. One study [7] has specifically examined the optimization of service selection as the key operation in dynamic service composition using multi-agent architecture that achieves online generation and rescheduling of the optimal services execution workflow using a heuristic planning technique. Another study [8] examined a multi-agent web for the IoT and corresponding data management architecture. The proposed architecture provides autonomic characteristics for IoT, making the IoT manageable. However, these studies did not perform an evaluation in an actual environment.

An Agent Space [9] system consists of Personal Assistant Agents, Service Agents and Resource Connect Agents that control devices/data/programs by requests from Personal Agents and Service Agents. That system used multi-agents to find resources such as information, services, and devices.

An agent framework for IoT was proposed [10]. For it, a Concept Reply context awareness solution based on a hybrid framework was proposed. It integrates a semantic reasoning module and multiple processing agents for specialized / optimized processing tasks. This approach was evaluated by examination of two case studies.

A multi-agent system [11] was proposed to simulate the IoT environment populated with numerous web-service enabled devices. For the present study, the authors used a multi-agent system to realize functional requirements of a simulator, such as dynamic discovery of devices and services. A multi-agent system offers many benefits for an IoT environment.

Agent systems have been used in the discovery of services and devices as described above. We specifically examine sensor data processing contents and location response to environment in this study. Our proposed system responds to the environment through agent organization and re-organization based on observations of the environment. Results show that the system adaptability to the environment is improved.

3. Proposed method

3.1. Overview

Herein, we describe a sensor system that selects the data processing contents and processing location based autonomously on service characteristics, retrieved data, terminal, and the network environment. This system is constructed as a multi-agent system. Each of the agents responsible for data processing and service delivery operates cooperatively to respond to the environmental change. For example, the system changes the data processing location during provision of a service in case the load of the data processing node is increased. Then the load is added to the network. The remaining power of a node is lowered.

3.2. Agent organization design

The proposed method uses agents of the following three types.

Manager agent: Only one Manager agent exists in each server and sensor node. This agent holds the terminal resource information, which consists of the type of data which can be acquired in the terminal and the amount of available resources in the terminal. This agent also holds the information of other agents that are active in the terminal. Communication among agents is performed through Manager agents in the agent organization phase and the re-organization phase described in section 3.3. Given some information about the terminal characteristics, then the manager agent holds it too. Additionally, the Manager agent monitors its own terminal's resources such as the CPU load and the power consumption.

Service provider agent (SP agent): This agent provides a service to the user. A different SP agent is present for each service. This agent exists in an external server. The SP agent holds the type of data and the information necessary for providing its own service, and the resources necessary for data processing as the agent knowledge. For example, information related to the target motion and position is necessary to provide a watching service. This information is calculated from acceleration data and other resources. In turn, this calculation requires a certain amount of resources. The SP agent for a watching service holds this knowledge as the agent knowledge. The SP agent receives the necessary information from the DP agent, which is described below, and provides a service to the user. Additionally, this agent has the role of determining the data processing location in the sensor system. For this reason, this agent also holds service characteristics that should be considered, such as reactivity.

Data processing agent (DP agent): This agent processes data and extracts information. This agent exists as an external server when the system does not provide service. The DP agent exists in each type and processing method of data. For example, a DP agent estimates the subject behaviors from three-axis acceleration data using support vector machine. One DP agent might simply forward the received data. The proposed system moves the DP agent necessary for providing the service to the data processing terminal from the external server.

Table 1 presents the knowledge and information held by each agent of the proposed method.

Table 1. Knowledge and Information Held by each Agent

Manager agent	SP agent	DP agent
・ Addresses of other Manager agents ・ Addresses of other agents in the same terminal ・ Resource information of own terminal ・ Resource status of own terminal ・ Threshold of environmental change	・ Type of data and information necessary for providing services ・ Resources required for data processing ・ Address of the Manager agent in own terminal	・ Data processing method ・ Address of the SP agent (during providing service) ・ Address of the Manager agent in own terminal

Figure 2 presents the agent organizational design of the proposed method. When the system does not provide service, one Manager agent exists in each terminal; SP agents and DP agents exist in the external server. When the system starts providing the service, the agent organization described in Section 3.3 is performed. Results show that SP agent and DP agents that are necessary for providing the service are selected. If necessary, the DP agent is copied to another terminal. For instance, when service A is required and data processing method X is necessary for providing service A, the system determines that data processing is performed in the local server. The gray shaded agents provide service A cooperatively.

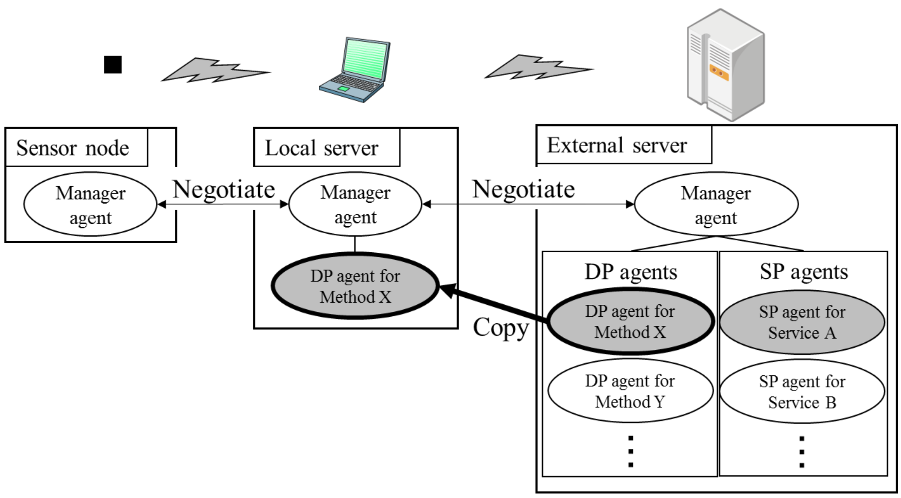


Figure 2. Agent Organization before Service Provision

3.3. Agent Behavior Design

Agent behavior design of the proposed method consists of the following three phases.

3.3.1. Agent Organization Phase: In the agent organization phase, the following (i)–(iii) are executed. The processing flow of the proposed method is presented in *Figure 3*.

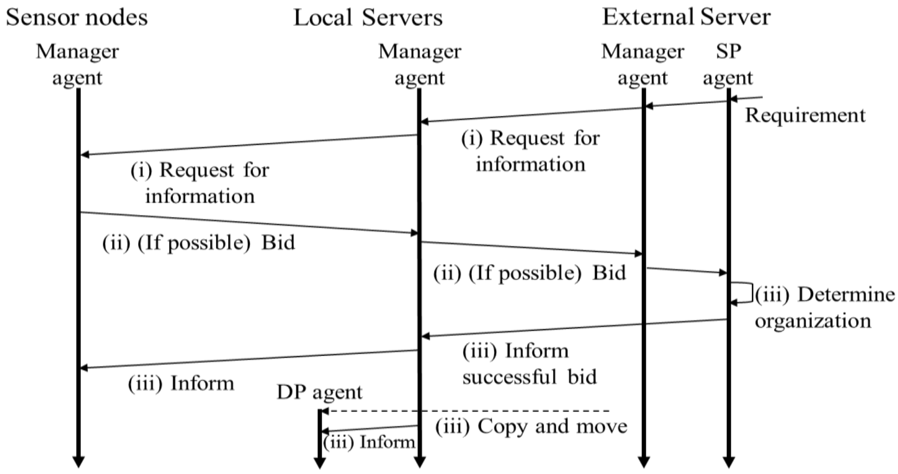


Figure 3. Flow of Agent Organization Phase

(i) When an SP agent received service requirement from a user, it requests the provision of the information to Manager agents in local servers through the Manager agent in the external server. For example, if information about the motion of the observed person is necessary, then SP agent requests information about the motion of the observed person or the acceleration data to derive this information. Contract net protocol (CNP) [12] is used for this communication. CNP is a task-sharing protocol in multi-agent systems. In this protocol, when a task is presented from the client agent, an agent that can perform its task bids the task. The client agent compares bids and determines the request destination agent. As a result, the client agent can assign the task to the most suitable agent. The Manager agent in a local server forwards the received request to the Manager agent in the sensor node.

(ii) If the Manager agent in the sensor node determines that its own terminal can acquire necessary information and data, then this agent replies to the bid. At this time, specification information of the terminal is added to the reply. Similarly, if the Manager agent in the local server can provide necessary information and data based on the data received from sensor nodes, then this agent replies to the bid, along with specification information of its own terminal to the Manager agent in the external server.

(iii) If the Manager agent in the external server receives a bid, then this agent sends all bids to the SP agent. The SP agent determines the organization of the local server and sensor nodes to provide the service using the evaluation method described in section 3.4. The SP agent informs success of the bid to the Manager agent in the local server through the Manager agent in the external server. If the SP agent chooses to instruct the local server or the sensor node to process data, then the SP agent presents that result to the DP agent. The DP agent copies itself. The copied DP agent moves to the local server or the sensor node. *Figure 3* depicts a case in which the SP agent chooses to make the local server process data. Additionally, the SP agent informs threshold of environmental change that triggers the re-organization. For example, this threshold is a value of the residual capacity of the node's battery power. The Manager agent in the local server transfers the received successful bid notification to the Manager agent in the sensor node.

By this procedure, the system starts providing the service using organized agents.

3.3.2. Service provision phase: During the system’s service provision, the sensor node acquires the data. The DP agent processes the received data. For example, when the system processes data in the local server as shown in Figure 2, the system processes acquired data as portrayed in Figure 4.

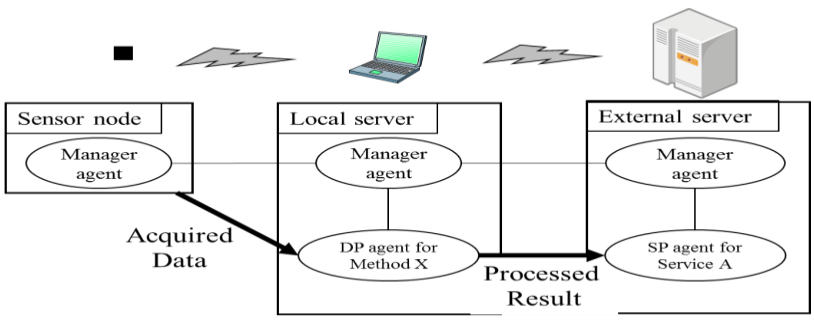


Figure 4. Agent Organization during Service Providing

3.3.3. Service re-organization phase: When the sensor node or the network environment changes, the system responds to the environment changes by agent re-organization. The agent re-organization procedure is the following.

(i) If the notification criteria that are notified in the agent organization configuration phase (iii) are satisfied, then the DP agent or the Manager agent in the local server or in the sensor node notifies the environmental change to the Manager agent in its own terminal. The Manager agent then informs the Manager agent in the external server. The Manager agent in the external server then informs the SP agent.

(ii) The SP agent in the external server receives the notification. Then it performs agent organization similar to that in the agent organization phase. The system provides the service using this organization.

3.4. Evaluation and selection method of the node

This section presents description of the evaluation method for determining the local server or the sensor node that provides the service. This method is used in agent organization and the re-organization phase described in sections 3.3.1 and 3.3.3. Processing locations of the acquired sensor data are also determined using this method.

In this method, the following elements are considered.
(E1) The amount of resources necessary for the sensor data processing
(E2) The residual power of the terminal
(E3) The usable amount of resources in the terminal
(E4) The network environment between the local server and the external server
(E5) The network environment between the local server and the sensor nodes

The SP agent knows (E1) of its own service, as described in Section 3.3.1. Additionally, the SP agent can get the local server's and the sensor node's (E2), (E3), and (E5) by bidding in the agent organization phase, as described in Section 3.3.1. The SP agent gets (E4) by communication in CNP.

From these elements, the SP agent determines the local server and sensor nodes that comprise the service. The SP agent evaluates bids using the following equation if there are plural bids. In the following, v_i stands for the evaluation value, r_i denotes the value of (E1), and e_i signifies the value of (E2), (E3), and (E4).

$$v_i = \frac{e_i}{r_i} \tag{1}$$

The SP agent calculates the minimum v_i of each bid. The bid with the largest value of minimum v_i becomes the successful bid. For example, in a case in which (E1) is "Data storage capacity 1 Mbyte, Remaining power 50%", local server A's (E2) and (E3) are "Data storage capacity 2 Mbyte, Remaining power 40%", and local server B's (E2) and (E3) are "Data storage capacity 1 Mbyte, Remaining power 60%", minimum v_i of local server A is 0.8, minimum v_i of local server B is 1.0. Consequently, the SP agent informs the local server B of the successful bid result.

Next, the SP agent determines the data processing node using the following procedures.

(i) The SP agent estimates the node at which the necessary data are aggregated. A node that is more distant than this node from the external server is excluded from candidates.

(ii) The SP agent evaluates v_i of the candidate node using equation (1). The node which has the largest minimum v_i becomes the data processing node.

4. Evaluation

4.1. Experimental Environment

We implemented a prototype system to evaluate the proposed method. The prototype system configuration is shown in Figure 5. We used a self-made device and a smartphone as a sensor node. The self-made device includes a sensor module mounted with an acceleration sensor (KXR94-2050; Kionix Inc.) and a communication module (Zigbee XBee; Digi International K.K.). The smartphone can become a local server. The local server and the external server are common PCs. The link between the self-made sensor node and the local server is connected with a single-hop ZigBee network. The link among the smartphone, the local server, and the external server is connected with a wired and wireless LAN. We implemented the prototype system using Android SDK 16 (Alphabet Inc.), which is used for the smartphone, and Java 1.7.0. We used PIAX [4] as the agent framework. The agent in PIAX is the Java class. PIAX provides an agent search function, a communication function, and other functions.

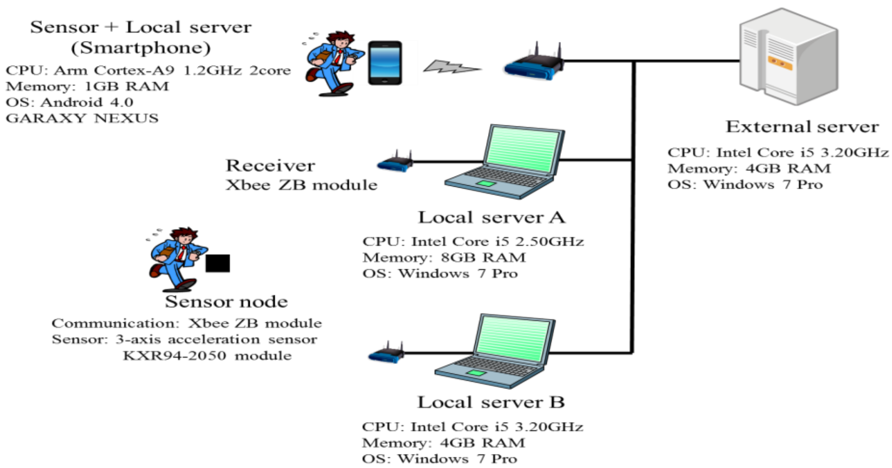


Figure 5. Configuration of the Experimental System

We implemented one SP agent and two DP agents. The SP agent provides a watching service using motion estimation. One DP agent processes acceleration data and estimates motions. The other DP agent only forwards received data. In this prototype system, the DP does not have a moving function. The Manager agent and two DP agents are placed in the external server, the local server, and the smartphone. The Manager agent is assigned the address of the other Manager agent and the amount of resources of own device in advance.

In this prototype system, the system considers the CPU usage rate, storage capacity, and remaining power as a resource. The SP agent requires the necessary kinds of information and data written in strings, and the resource values of three kinds above to the local server in agent organization phase described in Section 3.3.1. If the Manager agent in the local server bids, then this agent sends providable information and data written in strings, with resources values of three kinds.

4.2. Experiment

This experiment examines the effectiveness of the proposed method related to functional requirements (R2) selection of the sensor data processing content and location response to environmental change and (R3) expandability to the addition of a service and a terminal described in Section 2.2. The system provides a watching service using motion estimation.

4.2.1. Experiment 1: In this experiment, we verify operation of the proposed method when a sensor node moves through local servers. Moving a sensor node is the environmental change in (R2). In this case, the system must detect a change in the network configuration and perform agent re-organization. We use the external server and local servers in Figure 5 in this experiment. An observed person who wears a sensor node is within the range of communication with local server A at the start of the experiment. The observed person moves to the range of communication with local server B after the experiment starts. The sensor node acquires the acceleration data and sends it at regular intervals (100 ms). The motion is estimated every 3000 ms.

The agent organization of the prototype system performs the following operation based on the behavior design described in Section 3.3.1.

(i) The SP agent placed in the external server requires the observed person's motion or the acceleration data. A necessary amount of resources is informed simultaneously. In this experiment, both local servers can provide this resource.

(ii) Only the Manager agent in the local server A bids because the local server B cannot provide the observed person's motions or acceleration data.

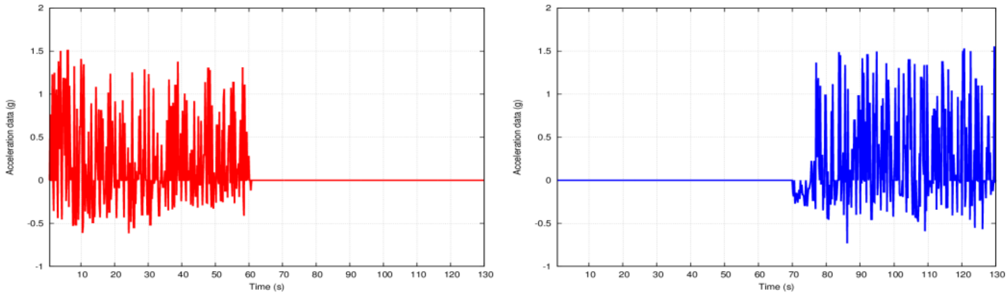
(iii) To determine the data processing location, the SP agent compares itself and the local server A based on equation (1) described in Section 3.4. In this experiment, the SP agent selects the external server as the data processing location because the external server has more resources than the local server A. The SP agent requires the transmission of acceleration data to the Manager agent in the local server A based on this result. In the local server, the Manager agent places the DP agent that can only forward received data.

(iv) If the DP agent in the local server A detects that the data are not sent from the sensor node, then it informs the SP agent.

(v) The SP agent requires the observed person's motion or the acceleration data again in the agent re-organization phase.

(vi) Only the Manager agent in local server B bids for the contract. Later, the SP agent and agents in the local server B perform in the same way as described for (iii).

The experimentally obtained results are presented in Figure 6, which shows the acceleration data that are stored in local servers A and B. DP agent in the local server A became unable to receive the data because the observed person moved after about 60 s from the start of the experiment. Next, the DP agent in local server B began to receive data by agent re-organization. Therefore, the system was able to continue to provide service through autonomous response to environmental change.



(a) Local Server A (b) Local Server B

Figure 6. Acceleration Data Received by the Local Servers

4.2.2. Experiment 2: This experiment verifies operation of the proposed method when a CPU load on the local server that processes data is increased by processing of other services. Increasing the CPU load is the environmental change in (R2). We reproduce the load by loading the CPU utilization obtained at the time of video viewing. Additionally, we limit the resources of the external server. Therefore, the system has the local server process the data. We use the external server and the local server B in Figure 5 in this experiment. The sensor node acquires acceleration data and sends it at regular intervals (100 ms). The motion is estimated every 3000 ms.

The prototype system performs the following operation.

(i) The system performs the same operation as (i)–(iii) in Section 4.2.1. The data processing is allocated to local server B.

(ii) The Manager agent in local server B detects the CPU load increase and informs the SP agent.

(iii) The SP agent requires the observed person's motion or the acceleration data again in the agent re-organization phase.

(iv) The Manager agent in the local server B bids for the contract. However, the value of the equation (1) changes because the CPU load is increased. Consequently, the data processing is allocated to the external server.

In this experiment, the system detected the local server B's CPU load increase and performed agent re-organization. For that reason, the data processing location was moved from the local server B to the external server. Therefore, the system enabled reduction of the load of the terminal autonomously in response to the environmental change.

4.2.3. Experiment 3: In this experiment, we verify operation of the proposed method when the residual power of the smartphone that plays a sensor node and the local server role declines. The residual power is the environmental change in (R2). Additionally, we verify (R3) through response to the smartphone characteristics. We use the external server and the smartphone in Figure 5 in this experiment. The smartphone acquires the acceleration data and sends it at regular intervals (100 ms). The motion is estimated every 3000 ms. The environmental change threshold that is notified in the agent organization phase is set as 75% power remaining. Results of a preliminary experiment showed that the smartphone used in this experiment reduces the data transmission by processing data in its own terminal. The power consumption is lower than when the smartphone does not process data. For this reason, the element of equation (1) related to the remaining power energy is calculated as a reciprocal.

The prototype system performs the following operation.

(i) The system performs the same operation as (i)–(iii) in Section 4.2.1. Data processing is allocated to the external server.

(ii) The Manager agent in the smartphone detects the remaining power decrease and informs the SP agent.

(iii) The SP agent requires the observed person's motion or the acceleration data again in the agent re-organization phase.

(iv) The smartphone bids for the contract. The value of the equation (1) changes because the remaining power is decreased. For that reason, the data processing is allocated to the smartphone.

In case the system moved the data processing location, the time until the remaining power decreased from 80% to 70% was 28 min 2 s on an average of five times. This is the effect of agent re-organization. Otherwise, in case the system did not move the data processing location, this time was 26 min 26 s. Therefore, the system enabled power consumption reduction autonomously in response to environmental change.

4.3. Discussion

The experimentally obtained result explained in Section 4.2 showed that the system detected environmental change such as configuration of the sensor node, the available resources in the terminal, and the remaining power. Then it performed agent re-organization. Results show that the system realized service continuation, reducing the load and reducing power consumption by changing the data processing location. Therefore, (R2) selection of the sensor data processing content and location response to environmental change was realized. However, the agent organization and re-organization add load to the terminals and the network. These processes might reduce the responsiveness, such as at the start of the service. It is necessary to evaluate the influences of these processes.

Similarly, (R1) selection of the sensor data processing content and location response to the characteristics of the service will be realized by the agent organization. However, we implemented one SP agent and two DP agents in the prototype system. The proposed method assumes an environment in which various service and data processing techniques coexist. Therefore, additional experiments must be conducted using several SP agents and a DP agent.

(R3) Expandability to the addition of a service and a terminal response to the characteristics of the service were realized using agent technology. In experiment 3, equation (1) was determined based on characteristics of the terminal. This function was realized by the Manager agent, which holds its own terminal’s characteristics. We expect to improve expandability to the addition of a service and a terminal through addition of new agents to the external server.

Additionally, it is necessary to evaluate the proposed method in an environment in which large amounts of data are generated. For example, if an image sensor node sends unprocessed data, then it adds heavy data to the network and receiver nodes. The proposed method that enables distribution of the load will be effective in this environment. We will implement a simulator for evaluation in environments where large amounts of data are generated.

5. Conclusion

As described herein, we proposed a method for selecting data processing contents and locations based on the service, the acquirable data, the required data processing, and the terminal and network environment. This method was realized by agents placed in system sensor nodes and servers, and by their cooperative operations. Subsequently, we conducted some initial experiments using a prototype system. Based on the experiment results, we concluded that improving the sensor system adaptability to the environment can be realized using the proposed scheme.

In our future work, we will conduct additional experiments using several SP agents and the DP agent. We will also implement a simulator to evaluate environments in which large amounts of data are generated.

Acknowledgments

This work was supported by KAKENHI (50510260).

References

[1] D. Okanohara, S. Hido, N. Kubota, Y. Unno and H. Maruhara, "Krill: An Architecture for Edge-Heavy Data", Proceedings of the Third Workshop on Architecture and Systems for Big Data, Tel-Aviv, Israel, **(2013)** June 24.

[2] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things", Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, (**2012)** August 13-17.

[3] Oracle, "Oracle Event Processing for Oracle Java Embedded", http://www.oracle.com/us/technologies/ java /embedded/event-processing/overview/index.html

[4] Y. Teranishi, "PIAX: Toward a Framework for Sensor Overlay Network", Proceedings of the Sixth IEEE Consumer Communications and Networking Conference, Las Vegas, USA, **(2009)** January 10-13.

[5] I. Ayala, M. Amor and F. Fuentes, "Self-StarMAS: A Multi-Agent System for the Self-Management of AAL Applications", Proceedings of the Sixth International Congress on Innovative and Internet Services in Ubiquitous Computing, Palermo, Italy, **(2012)** July 4-6.

[6] B. Manate, F. Fortis and P. Moore, "Applying the Prometheus methodology for an Internet of Things architecture", Proceedings of the IEEE/ACM Seventh International Conference on Utility and Cloud Computing, London, UK, **(2014)** December 8-11.

[7] A. Kouicen, A. Chibani, A. Tari, Y. Amirat and Z. Tari, "Dynamic services selection approach for the composition of complex services in the web of objects", Proceedings of the IEEE World Forum on Internet of Things, Seoul, Korea, **(2014)** March 6-8.

[8] P. Leong and L. Lu, "Multiagent Web for the Internet of Things", Proceedings of the International Conference on Information Science and Application, Seoul, Korea, **(2014)** May 6-9.

[9] K. Sugawara, Y. Kaeri, Y. Manabe, C. Moulin and J.-P. Barthes, "An application of an agent space connecting real space and digital space", Proceedings of the 13th International Conference on Cognitive Informatics & Cognitive Computing, London, UK, **(2014)** August 18-20.

[10] C. Barbero, P. D. Zovo and B. Gobbi, "A Flexible Context Aware Reasoning Approach for IoT Applications", Proceedings of the 12th IEEE International Conference on Mobile Data Management, Luleå, Sweden, **(2011)** June 6-9.

[11] S. Karnouskos and M. M. J. Triq, "Using Multi-Agent Systems to Simulate Dynamic Infrastructures Populated with Large Numbers of Web Service Enabled Devices", Proceedings of the International Symposium on Autonomous Decentralized Systems, Athens, Greece, **(2009)** March 23-25.

[12] R. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", IEEE Transaction of Computers, vol. C-29, no. 12, (**1980)**, pp. 1104–1113.

Authors

Shintaro Imai, is a lecturer of the Graduate School of Software and Information Science, Iwate Prefectural University, Iwate, Japan. He received PhD in information science from Tohoku University, Japan. His research interests include cognitive sensors and networks, agent-based computing, symbiotic computing, ubiquitous computing and flexible network. He is a member of Information Processing Society of Japan (IPSJ).

Haruka Shimizu, received B.S. degree in 2013 and M.S. degree in 2015 from Iwate Prefectural University, Japan. Her research interests include cognitive sensors and networks.

Bingxi Zhao, received dual degree in 2012 from Dalian Jiaotong University, China, and M.S. degree in 2014 from Iwate Prefectural University, Japan. Her research interests include cognitive sensors and networks.

Yoshikazu Arai, is associate professor of the Department of Software and Information Science, Iwate Prefectural University, Iwate, Japan. His research interests include distributed autonomous robotic systems, robot navigation and intelligent transport systems. He is the recipient of the Best Paper Award from the 4th International Symposium on Distributed Autonomous Robotic Systems (DARS'98). He is a member of Japan Society of Mechanical Engineering (JSME), Robotics Society of Japan (RSJ) and The Institute of Electronics, Information and Communication Engineers (IEICE).

Toshimitsu Inomata, is a professor of the Faculty of Software and Information Science, and a dean of Department of Software and Information Science, Iwate Prefectural University, Iwate, Japan. He received his doctorate degree in engineering from Toyohashi University of Technology, Japan. His research interests include real-time systems and embedded systems He is a member of Information Processing Society of Japan (IPSJ) and The Institute of Electronics, Information and Communication Engineers (IEICE).