Repository-centered Agent Organization Method among Heterogeneous Agent Platforms

Takahiro Uchiya[1], Yuta Nakashima[1], Ichi Takumi[1], Tetsuo Kinoshita[2], Hideki Hara[3], Kenji Sugawara[3]

[1]Nagoya Institute of Technology, [2] Tohoku University, [3] Chiba Institute of Technology
t-uchiya@nitech.ac.jp, yuta@uchiya.nitech.ac.jp, takumi@nitech.ac.jp, kino@riec.tohoku.ac.jp, hara@net.it-chiba.ac.jp, suga@net.it-chiba.ac.jp

Abstract

Various agents have been designed and developed using agent platforms of many kinds aiming at cooperative problem solving, surrogation, and support of human activities. Generally, it remains as a difficult problem to realize cooperation or construction of agent organization using agents of different kinds running on heterogeneous agent platforms instead of similar agents running on homogeneous platforms. To overcome this problem, a systematic method to construct an organization of agent of different kinds on heterogeneous agent platforms is proposed in this paper. Then an experimental system is implemented and demonstrated to underscore the effectiveness of the proposed method.

Keywords: *Agent organization method, Agent repository, Heterogeneous agent platforms*

1. Introduction

Generally, software with new characteristics such as autonomy and sociality is called an "Agent." And an information system that uses agents as its components is called an "Agent System." Nowadays, various agents and agent systems have come to be developed and operated from the viewpoints of supporting a user's work, cooperative problem solving, etc. [1][2]. We designate the structure forming a base environment of these agents as an "Agent Platform." A function supporting cooperation and the connection among agents on the same platform is usually equipped with an agent platform. Using this function, if some agents operate on a same platform, we can carry out distributed cooperative processing that assumes communication among agents in a distributed surrounding. In fact, various developers are producing original agent platforms according to each purpose and usage now, with closed agent systems formed on respective agent platforms. However, future intelligent agent systems are expected to offer components as mutually connected, thereby improving the user's convenience.

Therefore, a genuine need exists to have agents that can perform on heterogeneous agent platforms and which can thereby take advantage of each agent platform's merits. Consequently, we can apply different agent platforms supplementary. As described herein, to achieve agent interoperability, we propose a new method to organize agent systems on heterogeneous agent platforms using an agent repository mechanism. Then we describe the design and implementation of our proposed method. Furthermore, we evaluate the effectiveness of our method through experimentation.

2. Related Work

2.1. Agent Platforms

An agent platform provides an execution environment for agents. It supports agent activities of all kinds, such as inter-agent communication functions and coordination functions. Agent platforms are approximately classifiable into two types: FIPA-compliant agent platforms and original specification-based (non-FIPA-compliant) agent platforms.

[FIPA-compliant Agent Platform]

An FIPA-compliant agent platform is implemented in accordance with the specification suggested by FIPA [3], which is a consortium of agent technology aimed mainly at agent interoperability. Some FIPA-compliant agent platforms are SAGE [4], JADE [5], LEAP [6], and FIPA-OS [7]. Based on the agent management reference model that FIPA defines, a platform comprises three mechanisms: an Agent Management System (AMS) controls the agent lifecycle, a Directory Facilitator (DF) sponsors the Yellow Page service, and a Message Transport System (MTS) provides communication mechanisms based on the FIPA agent communication language (FIPA-ACL).

Interconnectivity is fundamentally guaranteed among agent platforms implemented in accordance with the FIPA specifications. Consequently, mutual operations can be conducted on these agent platforms. A project to conduct a mutual operation experiment, designated as the "Agentcities Project" [8], is underway on an international level.

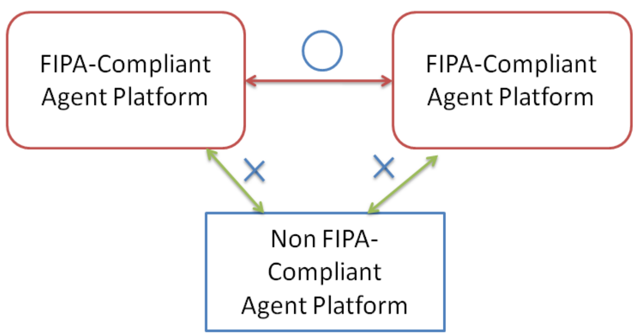

Figure 1. Interoperability among FIPA-compliant Agent Platforms and non-FIPA-compliant Agent Platforms

Figure 1 shows that interoperability among FIPA-compliant agent platforms is realized, but interoperability between an FIPA-compliant agent platform and a non-FIPA-compliant agent platform has not been realized.

[Original Specification-based Agent Platform (Non-FIPA-compliant Agent Platform)]

An original specification-based agent platform is designed for a certain problem domain with special functionality. The platform is classifiable as a mobile agent platform, intelligent agent platform, or dynamic self-organization platform.

• Mobile Agent Platform

A platform supplies a support function to execute a mobile agent. This category includes Telescript [9], picoPlangent [10], MobileSpaces [11], and Aglets [12].

• **Intelligent Agent Platform**

A platform holds knowledge processing, knowledge representation, a reasoning algorithm, and a learning algorithm applied to boost agent intelligence. Systems such as OMAS [13], ABLE [14], and AgentBuilder [15] have been developed.

• **Dynamic Self-organization Platform**

In this category, we first define "Agent Organization" as a process forming an agent group by assigning roles of agents based on specifications of the problem. Agent organization is a key technology used to perform agent-based cooperative or collaborative support of human activities.

This type platform can render an agent system capable of self-organization and reorganization of agents. An agent-repository-based multiagent framework called ADIPS/DASH [16] [17] [18] has been developed as a dynamic self-organization platform. An agent on this platform holds rule-based behavior knowledge written by the agent developers; it can support communication among agents using the KQML [19] based ACL message. Furthermore, agents situated in the agent repository are organized autonomously to satisfy the requirements of the user request. Then, an instance of the organization is created on the user's environment as a multi-agent system. The entire process is conducted according to the organizational protocols provided by the ADIPS/DASH agent platform.

These original specification-based agent platforms do not fundamentally share interoperability with platforms of other kinds. These systems might function independently. Figure 2 depicts the interoperability's relation among original specification-based agent platforms. Interoperability of the respective agent platforms is not guaranteed.

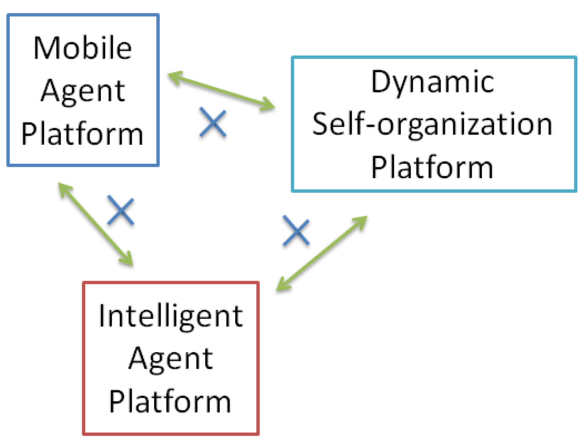

Figure 2. Interoperability among Original Specification-based Agent Platforms

2.2. Related Work on Interoperability

Studies have been conducted to realize agent cooperation and collaboration on different agent platforms. The Agentcities Project [8] specifically examines interoperability among FIPA-compliant agent platforms, but the original specification-based agent platform is not considered in the project. The incarnation agent [20] realizes interoperability among mobile agent platforms by transforming the mobile agent. An incarnation agent extracts the contents of the process and state of a certain agent. Then it translates them to an expression that is not dependent on a certain mobile agent platform. Subsequently, the agent translates it to an expression of the next target platform for movement. The agent platform protocol [21] is useful for realizing interoperability between the FIPA-compliant agent platform and the original specification-based agent platform, but its realization method is based on platform modification, and continual modification of the connection module is necessary to address system upgrades or changes that occur in the platform architecture. The gateway agent [22] provides interoperability between the FIPA-compliant agent platform and the original specification-based agent platform by conversion of agent mental states, agent communication language, and message transportation. Communication is achieved at a conversational level. Nevertheless, the absorption of differences in procedures in communication processes is not supported. Therefore, it remains insufficiently robust for actual applications. As described above, realizing interoperability from the viewpoint of agent cooperation and communication is being explored gradually.

In this research, we specifically examine "Agent Organization" methods that provide the concrete strategy and methodology to organize homogeneous or heterogeneous agents in a distributed environment. There are some researches for the support of agent organization on the homogeneous agent platform. An organization structure adaptation method that uses a central blackboard using self-diagnosis is proposed in one earlier report [23], an approach based on max flow networks to adapt organizational models dynamically to environmental fluctuation is proposed in another [24], and a coordination artifact in an open multi-agent system that is used to build and evolve a role taxonomy model over time is developed in yet another [25]. But there are hardly any researches for the support of agent organization on the heterogeneous agent platform.

As described herein, we newly propose an agent organization method not only for organizing homogeneous agents but also for organizing heterogeneous agents of different kinds flexibly using the ADIPS/DASH agent platform with an agent repository. The aim of our research is to realize an advanced organizing function necessary to address cooperative problem solving using agents which operate on heterogeneous agent platforms. Figure 3 presents our first-step goal of this research. We used SAGE and JADE as the FIPA-compliant agent platforms, and used OMAS as an intelligent agent platform. Here, ADIPS/DASH served as a base platform for interoperation of heterogeneous agent organization because the ADIPS/DASH agent platform has functions of constructing and reconstructing agent organization in a dynamic way.

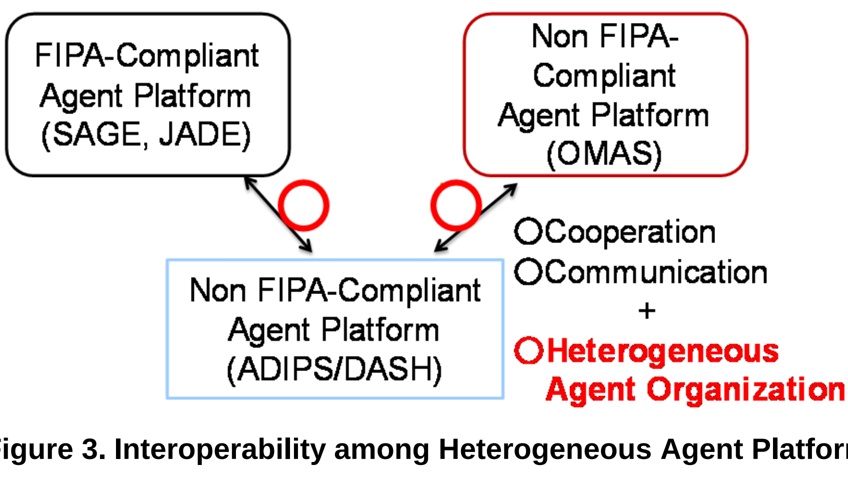

Figure 3. Interoperability among Heterogeneous Agent Platforms

3. Agent Organization Method on Homogeneous Agent Platform

3.1. Point of Focus

As described herein, agent organization methods serve an important function in the cooperative problem solving of agent system. In this section, we explain the reason why we chose the ADIPS/DASH platform as a base platform for interoperation of heterogeneous agents operating on the respective distributed platforms. Then we explain an agent organization method of the ADIPS/DASH platform.

3.2. Base Platform

In the ADIPS/DASH [9][10] agent platform, agent developers can construct and reconstruct the organization of agents merely by writing extremely little organization knowledge of each agent based on the agent repository, which has functions to search, organize and instantiate various agents based on the organization design knowledge given by the agent designer. This salient characteristic of the repository-based agent framework is not found in other agent platforms. If we adopt the other platforms as the base platform, then we must design and implement functions that were realized in the ADIPS/DASH platforms to construct and reconstruct agent organization.

In contrast, the agent repository of ADIPS/DASH platform can deal only with those agents which operate on the ADIPS/DASH platform. Therefore, we require some function to communicate with agents that operate on platforms of different types, to make an agent organization of heterogeneous agents. However the implementation of such communication functions is easier than the functions of an agent repository mechanism that can accommodate heterogeneous agents. For that reason, we adopt ADIPS/DASH as the base platform.

3.3. Traditional Organization Method of the ADIPS/DASH Agent Platform

We present an overview of the ADIPS/DASH agent platform consisting of the agent repository and the agent workplace as portrayed in Figure 4.

• **Agent Repository:** An agent repository stores agent-developer-produced agents that can act immediately according to the user’s request. Additionally, it is a mechanism to generate an agent organization depending on the user’s request. The concrete functions of the agent repository include acceptance of user requests, selection of agents, decisions related to agent organization, and generation of agent organizations operating on the agent workplace. The agent repository fully supports the agent system lifecycle through these processes.

• **Agent Workplace:** An agent workplace is an environment in which agent-repository-generated agent systems operate. An agent workplace offers a function by which an agent system exchanges messages with other agent groups and carries out cooperative problem solving based on the behavioral knowledge stored in each agent.

Next, we explain the organization method of agent system conducted in the agent repository. We show an organization procedure of the ADIPS/DASH agent platform in Figure 5.

Agents that designed and stored in the agent repository are classified as “manager agents” or “contractor agents” with respect to the task decomposition knowledge, which specifies the organization design knowledge given by designers.

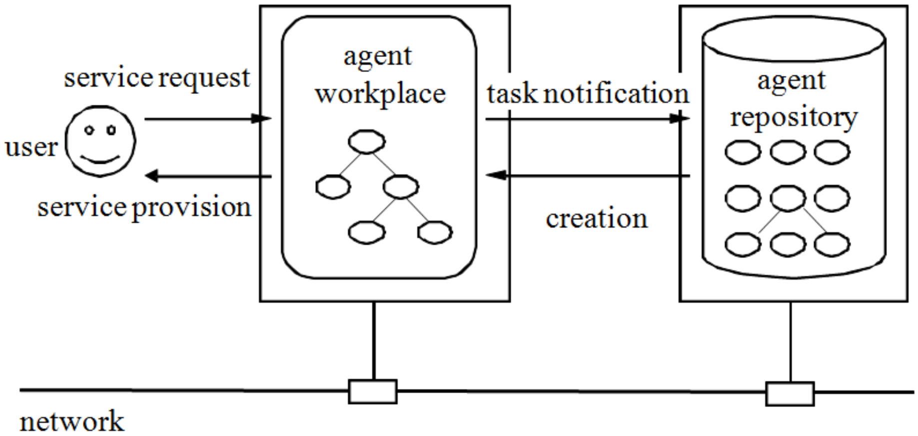

Figure 4. Overview of the ADIPS/DASH Agent Platform

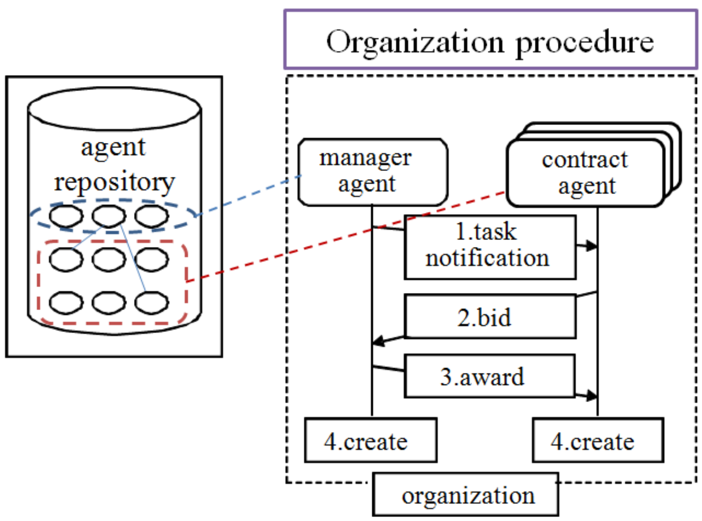

Figure 5. Organization Procedure in the Agent Repository on the ADIPS/DASH

• **Manager Agent:** A manager agent becomes a parent of an organization. The manager divides a certain task into several sub-tasks using its task decomposition knowledge.

• **Contractor Agent:** A contractor agent is an agent that becomes a child of an organization. It carries out a sub-task.

The agent repository receives a request from the agent workplace in the form of task notification. The repository delivers it to all agents in the repository. First, in the agent repository, an agent that can address the request is selected; this agent becomes a "manager agent"; then, using the task decomposition knowledge that is a part of the organization construction knowledge, the manager agent broadcasts a sub-task notification messages to other agents. Next, some "contractor agents" send the bid messages to the manager agent. Subsequently, the manager agent closes the bid at the specified deadline and sends an award message to the contractor agent whose bid is better than the other bids. The agent that makes a contract after the process described above is selected as a contractor agent, and becomes a member of the agent organization. Finally, the constructed organization is instantiated into the agent workplace to address a problem-solving task of the given request.

Using the procedure described above, agents on the ADIPS/DASH platform are reorganized dynamically with respect to the changes of user requests and the system's environment at the runtime of the agent system.

3.4. Problem

At the ADIPS/DASH agent platform described above, an agent organization is realized only on the ADIPS/DASH agent platform. However, considering the seamless mutual utilization of the heterogeneous agent platforms and agent systems, we must realize a function to organize the heterogeneous agent's organization on different platforms without redesign of these heterogeneous agents. Therefore, we propose a new organization method based on small extensions of the agent repository of ADIPS/DASH platform from the perspective of agent interoperability.

4. Repository-centered Agent Organization Method on Heterogeneous Agent Platform

4.1. Solution

We propose a new method to organize heterogeneous agents systematically without considering differences among heterogeneous platforms in the agent repository of ADIPS/DASH. In the proposed method, the following two mechanisms are introduced to extend the agent repository capabilities.

(S1) <u>A mechanism to accumulate service information of heterogeneous agents in the agent repository</u>

We newly introduce an interoperability mechanism to transmit a message between an ADIPS/DASH agent platform and the other agent platform. Additionally, we gather service information that a heterogeneous agent on a different kind platform offers into the agent repository from each platform. Herein, we develop a "**Collection Agent**" that carries out the collection work described above on a non-ADIPS/DASH agent platform. Thereby, we can gather service information from platforms of different kinds automatically, as shown in Figure 6.

(S2) <u>A mechanism to recognize a heterogeneous agent on the other platform as an available member of the agent organization</u>

To realize an agent organization, it is necessary that an agent with organization construction knowledge be generated in the agent repository. Therefore, using the gathered service information, a heterogeneous agent is defined virtually as an ADIPS/DASH contractor agent and accumulated as a "**Virtual Agent**" in the agent repository by the agent generation function of the repository. Using virtual agents, we carry out the construction of an organization of different kinds agents based on the ADIPS/DASH organization process. The concept of this mechanism is depicted in Figure 7.

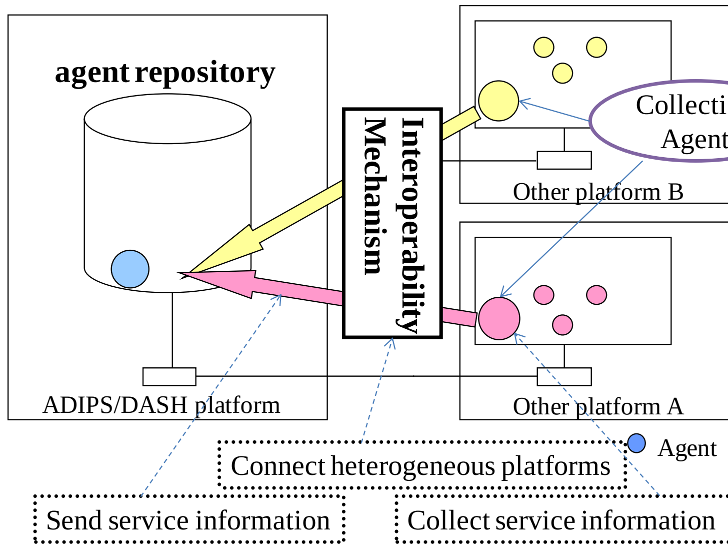

Figure 6. Concept of (S1) for Heterogeneous Agent Organization

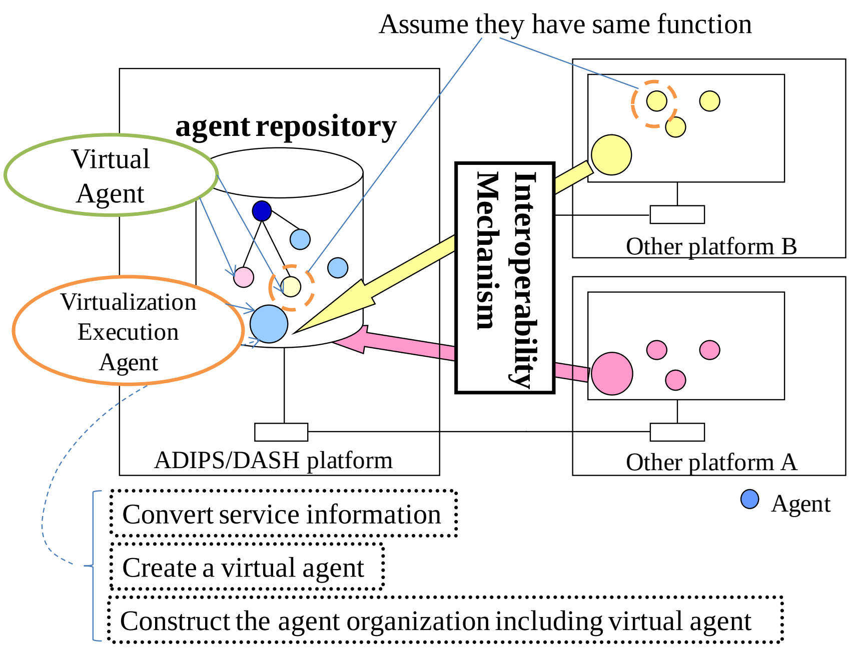

Figure 7. Concept of (S2) for Heterogeneous Agent Organization

4.2. Design

The mechanisms described above are designed and implemented by the following six functions.

(F1) Function of interconnect socket generation: This function generates the socket for mutual connections between an ADIPS/DASH platform and platforms of other kinds.

From the standpoint of the easy upgrade of the platform, we newly introduce an interoperability mechanism as mutual operative agents with a communication function.

If necessary, these agents generate a communication socket dynamically and interconnect two platforms. Detailed information of this interoperability mechanism is presented in the relevant literature [26].

(F2) Function of collection of service information on each platform: This function generates a "Collection Agent" in each non-ADIPS/DASH agent platform. A "Collection Agent" collects service information from a module sponsoring a Yellow Page function on the platform. For an FIPA-compliant agent platform, the Directory Facilitator functions as the Yellow Pages.

Using the agent description language provided with each agent platform, we realize the "Collection Agent" which carries out the acquisition, accumulation, and management of available service on each platform. For example, for an FIPA-compliant agent platform, this agent casts the "search" request into a Directory Facilitator (DF) module; then DF searches all services that the agent on this platform offers and acquires service information. From this service information acquired here, "agentname," "servicename," a set of "attributes," and a set of "values" are used as useful information for organization in a later function.

(F3) Function of message transmission among platforms: By this function, a "Collection Agent" on a different kind of platform transmits service information as an ACL message to an agent on an ADIPS/DASH platform.

For the "Collection Agent" realized in (F2), we add the behavior knowledge to transmit a message to an ADIPS/DASH agent platform through the interoperability mechanism. Furthermore, as an inside function of interoperability mechanism, we define a message conversion table based on ACL specifications that a heterogeneous agent uses. If necessary, agents refer to this message conversion table and convert the message.

(F4) Function of conversion of service information: This function converts the service information received by the (F3) function into a rule-based knowledge form automatically; it is a service description form of ADIPS/DASH.

For example, we can express the service information of an SAGE agent platform as the following list form.

```
(search-results
  :results
    (sequence
      (:df-agent-description
        :name (:agent-identifier :name realagentname :addresses (sequence IPaddress))
        :services
          (sequence
            (:service-description :ownership agentname :name servicename
              :properties (sequence (:property :name attribute1 :value value1 ..))
          :type global-search-target))
          :agent-ontologies (sequence ontologyname)
          :agent-protocols (sequence protocolname)
)))
```

To convert this information to the service description form of ADIPS/DASH, we introduce a description template. A description template is given in the following rule-based knowledge description form of ADIPS/DASH. Then, the conversion is performed by filling in some items on the description template. These items are agentname, realagentname, IPaddress, servicename, attribute, and value provided from the service information presented above.

```
(agent agentname
 (initial_facts
   (heterogeneous-agent :agentname agentname :realagentname realagentname
    :address IPaddress ..))
   ...
  (rule task-check
    (task-check :id ?id :task (task :name servicename :attribute1 value1 ...))
  ~(bid :id ?id)
    -->
    (make (bid :id ?id :content (task :name servicename :attribute1 value1 ...)
  ))))
```

An entity that executes the conversion described above automatically with service information and description template is the “Virtualization Execution Agent”, which is newly introduced into this paper.

The “Virtualization Execution Agent” has the following three roles.

(1) It receives service information from the “Collection Agent”.

(2) It carries out the conversion described above.

(3) It generates the “Virtual Agent” with function (F5), as described later.

The knowledge generated by this conversion is used in (F5). Particularly, the rule "task-check" becomes important for organization of construction knowledge for agent organization. The service that an agent offers is expressed in this “task-check” rule.

(F5) Function of virtual agent generation: This function considers a heterogeneous agent virtually as an ADIPS/DASH contract agent. It generates a “Virtual Agent” automatically in the agent repository.

A “Virtualization Execution Agent” generates a “Virtual Agent” based on the knowledge produced in (F4) using an agent generation function equipped with the agent repository. A generated “Virtual Agent” has a "task-check" rule as knowledge to carry out the bid action in the process of organization. Then the agent waits in the repository until the organization is actually conducted.

(F6) Function of a heterogeneous organization including virtual agents: This function constructs an agent organization by which a “Virtual Agent” can behave as a contractor agent of an organization.

When an agent repository receives a task notification message from the agent workplace, agents in the agent repository construct the agent organization for which processes obeyed the contract net protocol of ADIPS/DASH, for which the orders are "task notification," "bid," and "award," as described in subsection 3.3.

A “Virtual Agent” sends the bid message in the contract process using its own organization construction knowledge. When it receives the award message, it concludes a contract as a member of an organization. Then the agent will be in charge of a part of cooperative problem solving.

5. Experiment and Evaluation

5.1. Implementation

We chose the following platforms in our implementation of a prototype system, i.e., ADIPS/DASH platform, SAGE Platform, JADE platform, and OMAS platform.

We implemented two mechanisms. The first accumulates service information of heterogeneous agents in the agent repository. This mechanism has three functions: (F1), (F2), and (F3). To realize (F1) and (F3), we introduced and designed the "Interoperability Mechanism." Then we implemented this mechanism as mutual operative agents. For that reason, it was unnecessary to change the inner implementation of ADIPS/DASH and another agent platform. Subsequently to realize (F2), we introduced and implemented the "Collection Agent" in the SAGE, JADE, and OMAS side.

The other mechanism is that which recognizes a heterogeneous agent on the other platform as an available member of the agent organization. This mechanism has three functions: (F4), (F5), and (F6). To realize (F4) and (F5), we introduced and implemented the "Virtualization Execution Agent" in the ADIPS/DASH side using a rule-based language. This agent generates a "Virtual Agent" based on the service information that was transmitted to this agent about heterogeneous agents on SAGE, JADE, and OMAS platforms. Furthermore, the function of (F6) is provided by agent repository on ADIPS/DASH platform.

5.2. Experiment 1: Heterogeneous Organization between ADIPS/DASH and SAGE

We applied the proposed method to the agent system, which focused on the health management domain. The agent system offers the user some advice related to health management by performing information exchange and cooperative problem solving in the ubiquitous environment. We implemented two agents on an ADIPS/DASH platform. Then we implemented one agent on a SAGE platform. Roles of the respective agents are the following.

(A) (ADIPS/DASH) The "Health Advice Manager" receives health advice requests from the user. Then, after dividing the "health advice" task into the two sub-tasks of "acquisition of body info" and "inference of health advice," it sends them to the contractor.

We show a part of organization construction knowledge of "Health Advice Manager" in Figure 8. The rule named "decompose-check" is rule-based knowledge used to carry out task decomposition, and rule "bid-check1" is rule-based knowledge used to carry out the award process.

(B) (ADIPS/DASH) The "Advice Inference Contractor" infers health advice using body information.

We show part of the organization construction knowledge of the "Advice Inference Contractor" in Figure 9. The rule named "task-check" is rule-based knowledge to carry out the bid process.

(C) (SAGE) "Bodyinfo Acquisition Contractor", which is called BodyInfoAgent, acquires user body information from the inner database.

```
(agent HealthAdviceManager
 (property (create :author "t-uchiya@nitech.ac.jp"))
 (initial_facts)(include :file Dash-Org.rset)
 ...
 (rule-set Dash-Org
  (property)(initial_facts)
  (rule decompose-check
   (task-check :id ?id :task (task :name "healthadvice-service"))
  ~(bid :id ?id)
  -->
  (make (bid :id ?id :content (task :name "healthadvice-service")))
  (make (decompose :id ?id :to _broadcast  :env "null" :wait 10000
   :wp "null" :task (task :name "provide-healthadvice")))
  (make (decompose :id ?id :to _broadcast :env "null" :wait 10000
   :wp "null" :task (task :name "provide-bodyinfo"))))

  (rule bid-check1
   (award-check :award "null") = ?ac
   (bid-check :id ?ac:id :bid (task :name "provide-healthadvice") :no ?no)
   -->
   (modify ?ac:award ?no))
```

Figure 8. Part of the Organization Knowledge of the Health Advice Manager

```
(agent AdviceInferenceContractor
 (property (create :author "t-uchiya@nitech.ac.jp"))
 (initial_facts)
 (include :file Dash-Org.rset)
  ...
 (rule-set Dash-Org
  (property)(initial_facts)
  (rule task-check
   (task-check :id ?id :task (task :name "provide-healthadvice"))
   ~(bid :id ?id)
   -->
   (make (bid :id ?id :content (task :name "provide-healthadvice")))
)))
```

Figure 9. Part of the Organization Knowledge of the Advice Inference Contractor

We describe the organization process of this agent system in Figure 10.

(1) First, we started an ADIPS/DASH platform and a SAGE platform separately on two PCs (Windows XP). On an ADIPS/DASH platform, "Health Advice Manager (A)" and "Advice Inference Contractor" were generated in the repository. On a SAGE platform, BodyInfoAgent was generated (Figure 10(1)).

(2) When we started the interoperability mechanism realized as mutual cooperative agents, a "function of interconnect socket generation (F1)" provided by

"Interoperability Mechanism" operates; two platforms were mutually connected. In addition, the "Virtualization Execution Agent" was generated in the repository on the ADIPS/DASH platform (Figure 10(2)).

(3) Based on the "function of service information acquisition on each platform (F2)", the "Collection Agent" was generated in the SAGE platform, which acquired service information from the Directory Facilitator. The information acquired from the Directory Facilitator is the following. Here, we confirmed that the "Collection Agent" obtains service information related to BodyInfoAgent (Figure 10(3)).

(search-results :results (sequence (:df-agent-description :name (:agent-identifier :name BodyInfoAgent:90202271@giraffe :addresses (sequence 172.20.20.9)) :services (sequence (:service-description :ownership BodyInfoAgent :name provide-bodyinfo :type global-search-target)) :agent-ontologies (sequence ontology1 ontology2) :agent-protocols (sequence request))))

(4) The "Collection Agent" on the SAGE platform transmitted service information to the "Virtualization Execution Agent" on the ADIPS/DASH platform using the "function of message transmission among platforms (F3)" provided by "Interoperability Mechanism" (Figure 10(4)).

(5) The "Virtualization Execution Agent" creates agent knowledge for agent organization using the "function of conversion of service information (F4)". (Figure 10(5))

We show this knowledge description in Figure 11.

(6) The "Virtualization Execution Agent" generates one "Virtual Agent" which can participate in the agent organization on behalf of the "BodyInfoAgent" using the "function of virtual agent generation (F5)" (Figure 10(6)).

Here, we confirmed that a series of some preparations for agent organization functions successfully.

(7) When we sent a health advice request to the repository, the platform worked as follows. In the repository, the agent organization process was performed by the "function of heterogeneous organization includes virtual agents (F6)", and one organization was created. In this organization, the manager is the "Health Advice Manager (A)", and the contractors are an "Advice Inference Contractor (B)" and BodyInfoAgent (C) (Figure 10(7)).

(8) After creation of the agent organization, they were generated in the workplace on the ADIPS/DASH platform (Figure 10(8)).

An agent in this organization has the environment name, the name of each agent, and a function name that each agent offers. Therefore, we confirmed that the service provisioning preparations are well done. After completion of the process described above, we confirmed that the organization of heterogeneous agents was conducted properly.

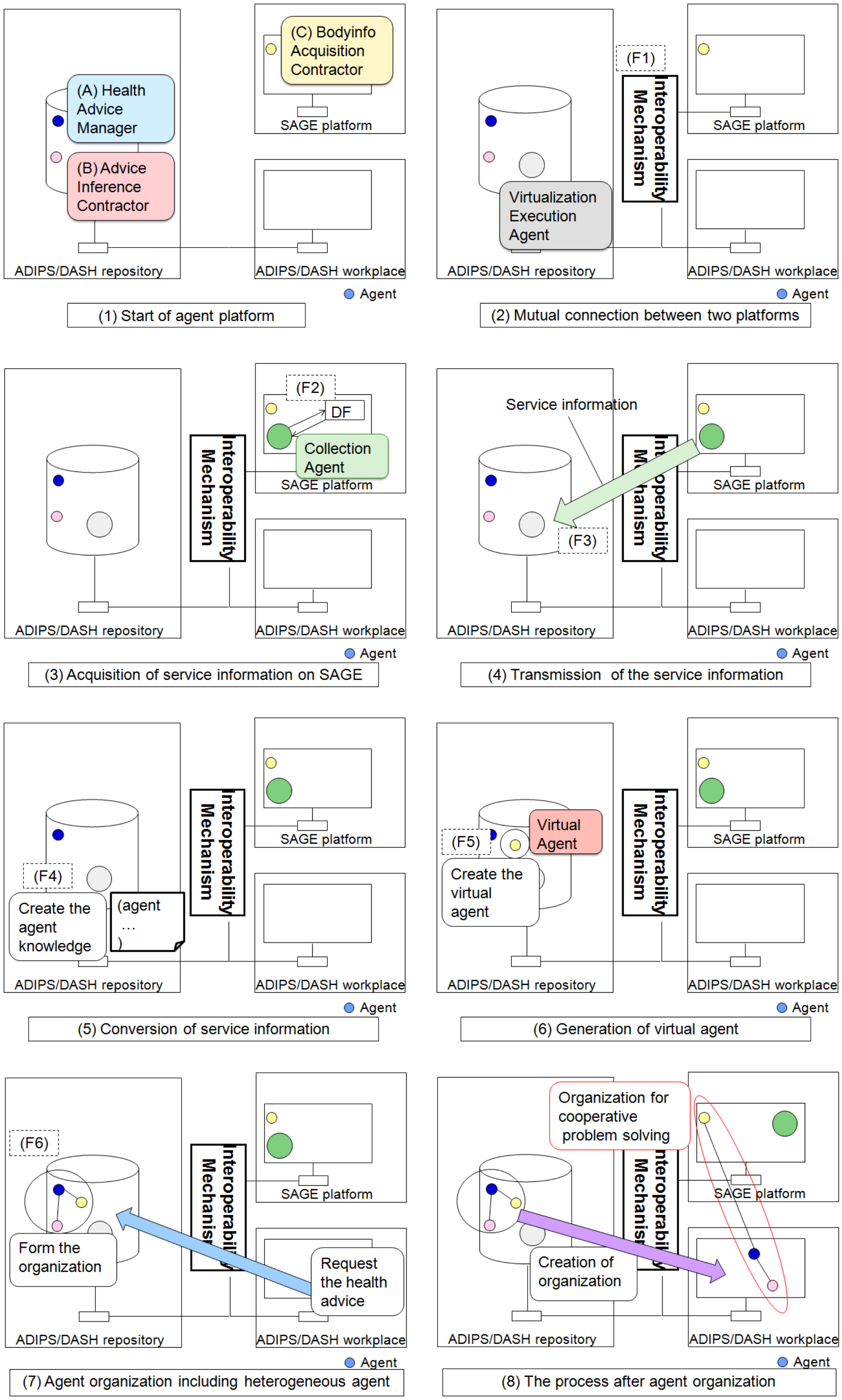

Figure 10. Process of Heterogeneous Agent Organization

```
(agent BodyInfoAgent
 (property (create :author "auto_creator"))
 (initial_facts (heterogeneous-agent :agentname BodyInfoAgent
 :realagentname BodyInfoAgent:90202271@giraffe
 :address 172.20.20.9 :platformname SAGE))
 (include :file Dash-Org.rset)
 ...
 (rule-set Dash-Org
  (property)(initial_facts)
  (rule task-check
   (task-check :id ?id :task (task :name "provide-bodyinfo"))
  ~(bid :id ?id)
   -->
   (make (bid :id ?id :content (task :name "provide-bodyinfo")))))
```

Figure 11. Part of Auto-created Knowledge of Virtual Agent

5.3. Experiment 2: Heterogeneous Organization between ADIPS/DASH and JADE

We applied the proposed method to implement an agent system that provides hot-spring information. Using this system, we verified the virtual agent creation behavior.

• Verification of virtual agent creation

The "Virtualization Execution Agent" successfully generated one "Virtual Agent" named "AichiOnsen_yuta-PC" which can participate in the agent organization on behalf of the JADE agent. We were able to check the existence of this agent and verify this agent's properties such as agentname, servicename, servicetype, and other attributes and values using GUI presented in Figure 12.

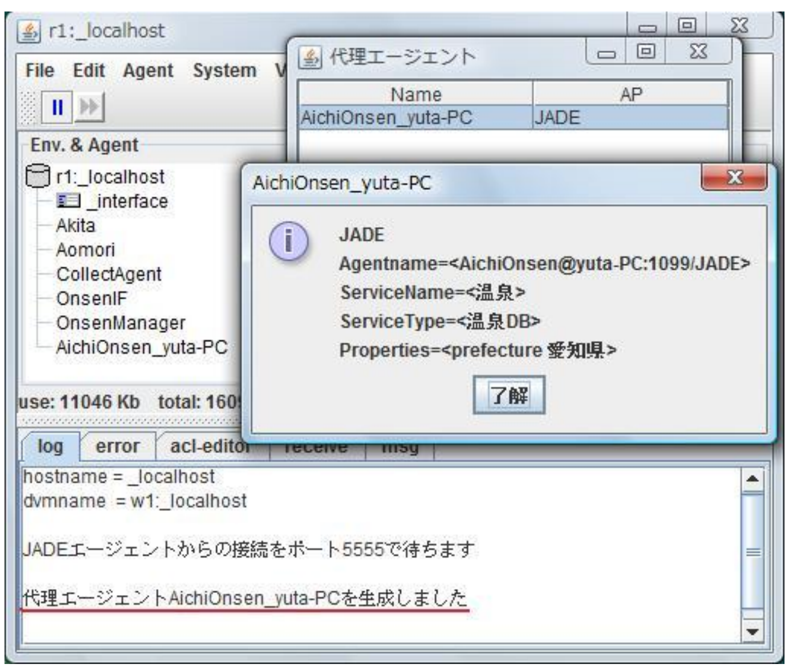

Figure 12. Confirmation Window of Virtual Agent's Properties of JADE

Next, we confirmed the relation between the number of agents and the processing time of agent creation. The results are presented in Table 1.

Table 1. Processing Time to Create Virtual Agents

Number of agents	1	2	5	10	20	50
Time (ms)	25.0	25.2	47.0	90.2	149.8	305.8

Results showed that the processing time increases in direct relation to the number of agents, but the maximum processing time to create 50 heterogeneous agents is acceptable for us because it is less than one second. This result demonstrates that our organization method has scalability in terms of the number of agents.

5.4. Experiment 3: Heterogeneous Organization between ADIPS/DASH and OMAS

We applied the proposed method to implement an agent system, which provides information about hot springs. The agents on the ADIPS/DASH platform are the same as those in Experiment 2. We newly developed one OMAS agent named "SHIZUOKA_ONSEN" which has local data related to hot springs. OMAS is a non-FIPA-compliant platform, and its communication module is implemented using LISP programming language, although the communication module of ADIPS/DASH platform is implemented using Java. Consequently, some difficulties arise in realizing their mutual communication. To overcome these difficulties, we designed and implemented an OMAS-dedicated Interoperability Mechanism that resolves differences of communication modes between ADIPS/DASH and OMAS.

Results of the experiments related to heterogeneous organization confirmed that a series of agent organization functioned successfully. "Virtualization Execution Agent" successfully generated one "Virtual Agent" named "SHIZUOKA_ONSEN_OMAS" which can participate in the agent organization on behalf of the OMAS agent. We were able to confirm the existence of this agent and verify this agent's properties such as agentname, servicename, and other attributes and values using GUI presented in Figure 13.

We then verified whether communication between the ADIPS/DASH agent and OMAS agent was executed properly or not.

• Verification of the communication after organization of heterogeneous agents.

Figure 14 shows that the ADIPS/DASH agent named DASH.201110191705254 and the OMAS agent named SHIZUOKA_ONSEN were able to communicate with each other. The ADIPS/DASH agent sent the OMAS agent the message related to the hot spring search request and the OMAS agent replied with the hot spring information to the ADIPS/DASH agent. This result demonstrates that our organization method is applicable to the heterogeneous organization between the ADIPS/DASH platform and the non-FIPA-compliant platform.

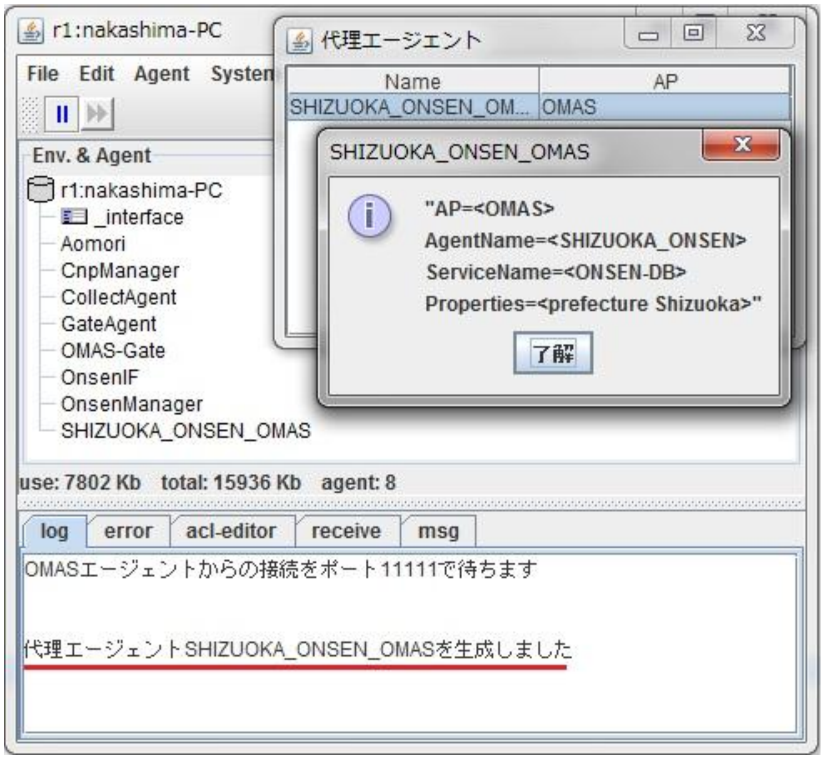

Figure 13. Confirmation Window of Virtual Agent's Properties of OMAS

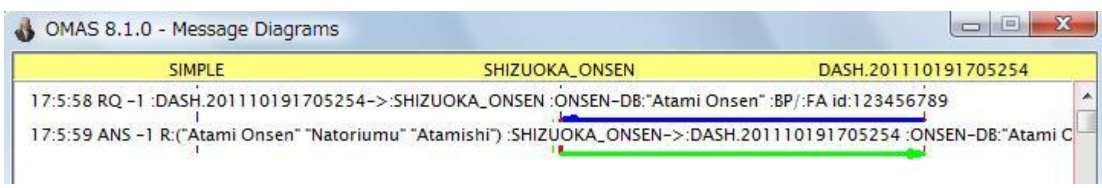

Figure 14. Message Diagrams between ADIPS/DASH and OMAS.

5.5. Evaluation

We confirmed through the three experiments described above that our proposed method has the following practical characteristics.

• Dynamic organization of heterogeneous agents

Our proposed method can organize an agent system that includes heterogeneous agents. Although some other methods described in the literature can support the dynamic organization of homogeneous agents, it is difficult to produce an appropriate heterogeneous agent system according to changes of user requests and system environments. Therefore, our proposed method has sufficient novelty from a standpoint of organizing agents of different kinds on heterogeneous agent platforms.

• Expansion of coverage of cooperative problem-solving processing

For our proposed method, we can use a non ADIPS/DASH agent as the contractor agent, which is an organization member for problem solving. Even if a non-ADIPS/DASH agent has no knowledge related to organization, the organization construction knowledge is given automatically using the proposed method. After heterogeneous agent organization, using some functions provided by non ADIPS/DASH agents, we can expand the coverage of cooperative problem-solving processing.

• Easy expansion of the target platform

When we want to use another heterogeneous agent on the FIPA-compliant agent platform, we can reuse the source codes of the proposed mechanism. Our tasks to realize heterogeneous agent organization including a new target platform are "development of Interoperability Mechanism as mutual operative agents of a target platform" and "development of a Collection Agent of a target platform". It is easy to implement these components through referencing of existing components' source codes.

6. Conclusion

To realize a heterogeneous agent organization based on agent interoperability, we proposed a new method to design and implement an agent system using agents of different kinds operating on heterogeneous agent platforms. Furthermore, we demonstrated the effectiveness of the proposed method through experimentation. Future work shall include application of the method to other agent platforms and examination of the scalability of interoperability mechanisms related to the number of platforms.

Acknowledgement

This work was supported by Grant-in-Aid for Young Scientists (B) 22700068.

References

[1] T. Suganuma, K. Sugawara, T. Kinoshita, F. Hattori, and N. Shiratori, "Concept of Symbiotic Computing and its Agent-based Application to a Ubiquitous Care-Support Service," in Y. Wang (Ed.), Transdisciplinary Advancements in Cognitive Mechanisms and Human Information Processing, Ch.3, pp.38-59, IGI Global, ISBN 9781609605537, EISBN13: 9781609605544, 2011.

[2] H. Kim, and T. Kinoshita, "A Multiagent System for Microgrid Operation in the Grid-interconnected Mode," Journal of Electrical Engineering & Technology, vol. 5, no. 2, 2010, pp. 246-254.

[3] FIPA – Foundation for Intelligent Physical Agent Specification, http://www.fipa.org/

[4] SAGE – Scalable Fault Tolerant Agent Grooming Environment, http://sage.niit.edu.pk/

[5] JADE – Java Agent DEvelopment Framework, http://jade.cselt.it/

[6] LEAP – Lightweight Extensible Agent Platform, http://leap.crm-paris.com/

[7] FIPA-OS, http://fipa-os.sourceforge.net/

[8] Agentcities project, http://www.agentcities.org/

[9] Telescript, http://citeseer.ist.psu.edu/context/776334/0

[10] picoPlangent, Intelligent Mobile Agent Designed For Ubiquitous Environment, http://www2.toshiba.co.jp/rdc/plangent/

[11] MobileSpaces, the Second Generation Adaptive Mobile Agent System, http://research.nii.ac.jp/ichiro/mobilespaces/jp/

[12] Aglets, http://www.trl.ibm.com/aglets/

[13] OMAS – Open Multi-Agent System –, http://www.utc.fr/barthes/OMAS/

[14] ABLE – Agent Building and Learning Environment. http://researchweb.watson.ibm.com/able/

[15] AgentBuilder, http://www.agentbuilder.com/

[16] T. Kinoshita, K. Sugawara and N. Shiratori, "Agent-based Framework for Developing Distributed Systems," in *Proc. of Workshop on Intelligent Information Agents (CIKM'95-IIAW), ACM-SIGART*, 1995.

[17] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, "Agent-Based Design Model of Adaptive Distributed System," *Applied Intelligence*, 1998, vol. 9, no. 1, pp. 57-70.

[18] DASH – Distributed Agent System based on Hybrid architecture, http://www.agent-town.com/dash/

[19] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," in *Software Agents*, Cambridge, MA: MIT Press, 1997, pp. 291-316.

[20] T. Hasegawa, K. Cho, F. Kumeno, S. Nakajima, A. Ohsuga, and S. Honiden, "Interoperability for Mobile Agents by Incarnation Agents," in *Proc. of AAMAS 2003*, 2003, pp. 1006-1007.

[21] K. Takahashi, Z. Guoqiang, S. Amamiya, T. Mine, and M. Amamiya, "Message communication protocol for interoperability between agent platforms," *IEEJ Transactions on Electronics, Information and Systems*, 2003, vol. 123-C, no. 8, pp. 1503-1510 (in Japanese).

[22] H. Suguri, E. Kodama, M. Miyazaki, and I. Kaji, "Assuring Interoperability between Heterogeneous Multi-Agent Systems with a Gateway Agent," in *Proc. IEEE International Symposium on High Assurance Systems Engineering*, 2002, pp. 167-170.

[23] B. Horling, B. Benyo, and V. Lesser, "Using Self-diagnosis to Adapt Organizational Structures," in *Proc. of AGENTS 2001*, 2001, pp. 529-536.

[24] M. Hoogendoorn, "Adaptation of Organizational Models for Multi-agent Systems based on Max Flow Networks," in *Proc. of IJCAI'07*, 2007, pp. 1321-1326.

[25] R. Hermoso, H. Billhardt, and S. Ossowski, "Role Evolution in Open Multi-agent Systems as an Information Source for Trust," in *Proc. of AAMAS* 2010, 2010, pp. 217-224.

[26] T. Uchiya, T. Maemura, X. Li, S. Konno, and T. Kinoshita, "Agent Interoperability Mechanism among Heterogeneous Agent Platforms for Symbiotic Computing," in *Proc. 7th IEEE International Conference on Cognitive Informatics (ICCI2008)*, 2008, pp. 286-293.

Authors

Takahiro Uchiya is an Associate Professor of the Information Technology Center, Nagoya Institute of Technology, Nagoya, Japan. He received a Ph.D. degree from Tohoku University in 2004. His research interests include knowledge engineering and design methodologies of agent system. Dr. Uchiya is a member of IEICE and IPSJ.

Yuta Nakashima is a Master course student of the Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Japan. His research interests include agent-oriented software computing.

Ichi Takumi is a Professor of the Information Technology Center, Nagoya Institute of Technology, Nagoya, Japan. His research interests include knowledge computer system networks, fundamental informatics, intelligent informatics, measurement engineering, communication/ network engineering, and natural disaster science.

Tetsuo Kinoshita is a Professor at the Research Institute of Electrical Communication, Tohoku University, Japan. He received a Dr. Eng. degree in information engineering from Tohoku University in 1993. His research interests include agent engineering, knowledge engineering, knowledge-based systems and agent-based systems. He received the IPSJ Research Award, the IPSJ Best Paper Award, and the IEICE Achievement Award respectively in 1989, 1997, and 2001. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, IPSJ, and JSAI.

Hideki Hara is an Associate Professor of the Department of Information and Network Science, Chiba Institute of Technology, Chiba, Japan. He received his Ph.D. degree from Chiba Institute of Technology in 1999. His research interests include computer network and software agent. Dr. Hara is a member of IEICE and IPSJ.

Kenji Sugawara is a Professor of the Department of Information and Network Science, Chiba Institute of Technology, Japan. He received his doctorate degree in Engineering from Tohoku University in 1983. His research interests include agent-oriented computing and web computing. Prof. Sugawara is a member of IEEE, ACM, IEICE, and IPSJ.