

# Multiagent-based Autonomic Service Organization Scheme for the Internet of Things

Takumi Kato, Hideyuki Takahashi, Tetsuo Kinoshita

**Abstract**—As a massive number of devices are becoming part of the Internet, it is highly difficult to organize IoT devices as a system to provide desired services in our lives. It is difficult because the IoT devices are heterogeneous, and the situation often changes overtime which requires the complicated reconfigurations of IoT systems. In this paper, we propose multiagent-based autonomic service organization scheme for IoT systems, which provides IoT devices with the functionality of task-oriented and resilient service provisioning in IoT systems by its capability of agent-based IoT device organization/re-organization. As a proof of concept, we have implemented logistics application using simplified robots using the proposed scheme and conducted logistics experiments.

**Keywords**— Agent, Organization, Internet of Things (IoT), Autonomous Logistics.

## I. Introduction

Our daily lives involve a number of various computational devices connected to the Internet, e.g. smart phones, personal computers, information home appliances, building components, and even the electrical power system components. As such devices gain the ability to connect to the Internet based on their controlling software, there are diverse services that utilize them to benefit our lives. This paradigm is called the “Internet of Things (IoT),” in which heterogeneous devices are utilized by series of software through the Internet. The challenge that the number of IoT related studies share, is the effective utilization of several and diverse computational entities that operate in a physical space to realize diverse services for people.

As the number of IoT devices and its services grows, it becomes a tremendous burden for users and developers to organize individual IoT devices based on the requirements of services. In order to realize inter-service utilization of IoT devices, each device needs to be re-usable, flexible, and autonomous to communicate and perform multiple tasks with the other devices. There are investigations on realizing flexibility and autonomy of IoT as individuals [1]. Although there are investigations on smart functionality of IoT device as an individual, it is a remaining problem to autonomously organize a set of IoT devices to realize a service based on user requirements, as well as to re-configure the IoT system under changes.

To address the problem, we propose an agent-based Internet of Things (AIoT) architecture, and autonomic service organization scheme to realize resilient IoT device organization.

The rest of this paper is organized as follows: Section 2 describes the related work and the detail of underlying problem on realizing services in IoT systems. Section 3 explains the proposed scheme to resolve the underlying problems. Section 4 shows the implementation and experiments to show the autonomic and resilient service provisioning capability based on the proposal. We conclude this paper in Section 5.

## II. Related Work

In order to examine the underlying problem in the existing schemes on IoT device organization for service provisioning in IoT systems, this section looks into the existing works to realize such capability.

### A. Service Provisioning based on Automatic Mash-up and Enabling Architecture of IoT devices

To utilize heterogeneous IoT devices together, Device Ensemble System [2] investigates a way to enable sensors and smart home appliances to work together by connecting them into a centralized webserver, and control these devices based on the requests sent from a smartphone of a user. There is also a study on organizing IoT devices registered in REST based web servers, called the Web of Things (WoT) [3]. WoT approach offers an abstraction of IoT device and enables heterogeneous IoT devices to be utilized by a web service. The abstraction makes it easier for a user to manually construct service by interconnecting the abstracted functions of IoT devices.

### B. Agent-based Approach for Autonomous Service Provisioning in IoT systems

The conventional approaches require constant re-configurations whenever a system faces change of user requirement, partial failure, change of situation, etc. The prior survey [4] points out that the characteristics of intelligent agents are well suited for the utilization of IoT devices. In order to sustain the service, there are investigations on introducing agent technologies into Smart Object (SO) [5]. Agent-based smart object provides a context-awareness, and adaptability based on the context to perform and adjust device’s action using the given knowledge to the agent. By introducing the agent technology, and knowledge on the context of user and situated environment, agent-based SO provides a capability to sustain service by adjusting its actions based on contexts.

As the application of agent-based SO, there is a centralized approach to device discovery and orchestration

---

Takumi Kato<sup>1,2</sup>, Hideyuki Takahashi<sup>1,2</sup>, Tetsuo Kinoshita<sup>1,2</sup>

<sup>1</sup>Graduate School of Information Sciences, Tohoku University, Japan.

<sup>2</sup>Research Institute of Electrical Communication, Tohoku University, Japan.

in home and building automation [6][7]. They utilize a single knowledge base shared among SOs. The cooperation among devices is organized manually by a user based on the web page of user interface. The agents in the system monitor the state of each appliance, and execute actions based on the user requirements.

Agent-based decentralized approach of IoT devices is proposed as various middleware in the existing works. UBIWARE [8], UBIROAD [9] and S-APL [10] provide semantic knowledge representation and reasoning capability to the utilize functions of IoT devices in situated environment. Impala [11], Smart Messages [12], ActorNet [13], Agilla [14] are introducing intelligent agents and mobile agents to manage sensor network to sustain sensing service by adapting the system towards the partial failure or the status of system component. These middleware demonstrates the IoT device's capability to adapt toward other component's failure, and situational changes in the IoT systems.

### C. Underlying Problem and Our Approach

Despite of the number of investigations explained above, it is still a remaining problem to autonomously organize the functions of IoT devices to realize service as IoT system, as well as to re-configure the organization of heterogeneous IoT devices towards partial failure and situational change.

One of the reasons of difficulties on the autonomous organization and re-organization of IoT device was the burden in implementing the negotiation process to form an organization. Another reason is that heterogeneity of IoT devices makes it hard for each device to negotiate and share task with one another. The followings are a summary of the underlying problems:

- (P1) Organization of IoT application by taking into account the user's requirement and diverse IoT device functionalities without manual monitoring and setting procedures
- (P2) Tuning and replacing system components to deal with systematical and environmental change without manual monitoring and setting procedures

In order to deal with these problems mentioned above, we propose agent-based IoT application architecture and the middleware to systematically help users to build IoT application based on the proposal.

## III. Autonomic Service Organization Scheme of Agent-based Internet of Things Devices

We employ agent-based approach to resolve the underlying problem mentioned in the previous section, by taking advantage of the nature of intelligent software agents which are autonomous, flexible and social. Following subsections describe the details of proposal.

### A. Agent-based IoT Architecture for IoT Systems

Figure 1. illustrates the overview of agent-based IoT

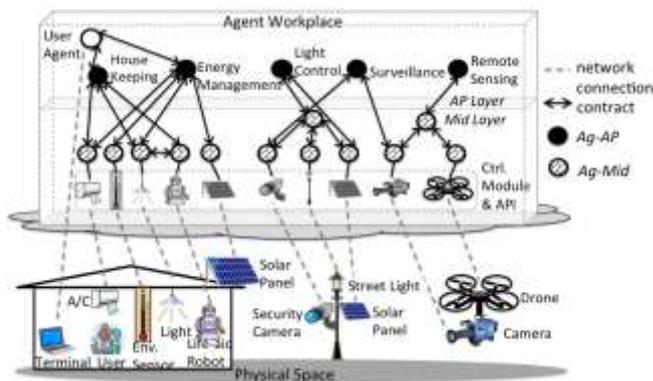


Figure 1. Overview of AIoT application architecture based on Agent-based IoT (AIoT) devices and service organization scheme.

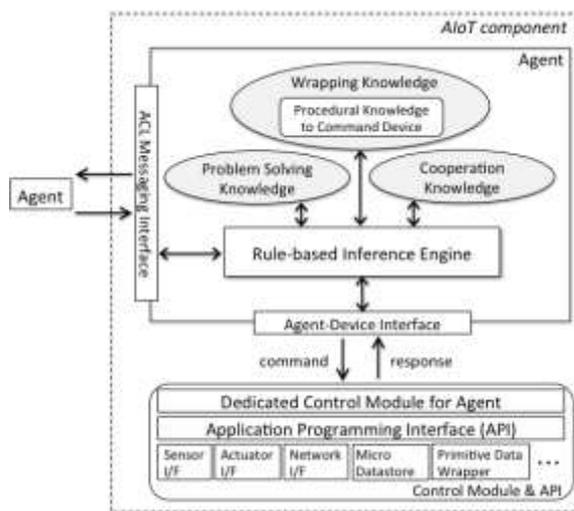


Figure 2. Architecture of agent-based IoT device.

(AIoT) devices and service organization scheme. We propose to manage IoT devices using software agents. The agents that control IoT devices are Middleware agents (Ag-Mid). In addition, the abstracted, agent-based IoT devices are organized by Application agents (Ag-AP), which have task descriptions to realize services. These agents compose Agent-based IoT Application. There are 2 layers that compose AIoT application, namely, application (AP) layer, and middleware (Mid) layer. Each layer is composed by application agents and middleware agents, respectively.

AIoT application layer provides functionalities to provide services to the user based on the design requirement of application agents, i.e. the task description given to the application agents. Application agents cooperate with middleware agents, or work on the task independently depending on their given problem solving knowlele. In case of cooperating with middleware agents, application agents work as a manager of the target task. In addition, the service requirement is acquired through user agents. User agents select application agent(s) to fulfill the service requirement by executing task description of application agent(s).

On the other hands, AIoT middleware layer interconnects AIoT application layer and the devices which work in the real world. Middleware agents is implemented through the process "agentification," which gives software agent's

capability to the various IoT devices. They work with the other middleware agent(s), or work independently.

## B. Architecture of Agent-based IoT Device

IoT devices gain the ability to flexibly cooperate with other devices through the design and implementation process called “agentification,” which provides devices with the software agent’s capability. The heterogeneity is resolved by the inter-agent communication based on the conventional agent communication techniques.

Figure 2. shows the architecture of software agent which controls IoT device. The design and implementation process of this architecture is the “agentification” process to implement agent-based IoT device. AIoT component is combination of controller software agent and the control software of IoT device such as API or provided middleware, as well as the dedicated interface module, which interfaces the agent command and APIs. The application agents and middleware agents are designed as AIoT components. The difference between these 2 agents is that middleware agents have IoT device’s control program while application agents have the task processing programs as the control module and API.

The knowledge given to the software agents in the AIoT component is the problem solving knowledge defined based on individual task, cooperation knowledge which defines the service organization scheme among agents, and the wrapping knowledge which represents the control logics of target IoT device. The knowledge of agent is represented as a set of if-then rules and processed by rule-based inference engine that determines agent’s actions. Based on the decisions of inference, the agent sends commands to the device, and receives responses of the executed actions.

## C. Autonomic Service Organization Scheme for IoT Systems

Through the agentification process, IoT devices acquire the ability to make decisions based on the given knowledge, and communicate with the other AIoT devices through agent communication language and techniques. As the core functionality to solve the problem of IoT device organization, we introduce the common cooperation knowledge to the AIoT component, so that the agents are able to cooperate with each other as AIoT application.

In our proposal, we apply Contract Net Protocol [15] to the IoT device organization. The flow of cooperation is shown in Figure 3. Application agent is the manager of the selected task by the user, and the middleware agent is the possible task executing agent which has the appropriate IoT device. The flow shown in Figure 3. Figure 3. is the contract conclusion process among such agents, as well as the task execution and reporting phase of the middleware agent.

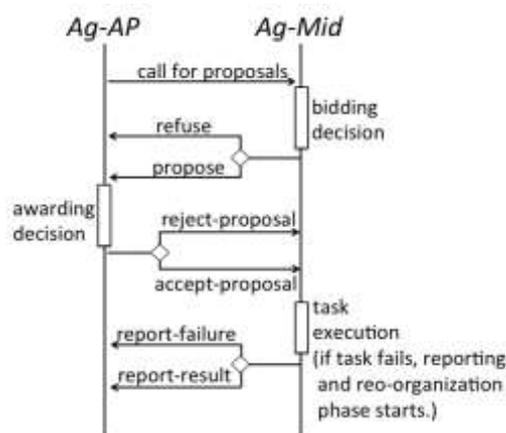


Figure 3. Contract conclusion process between application agent and middleware agent.

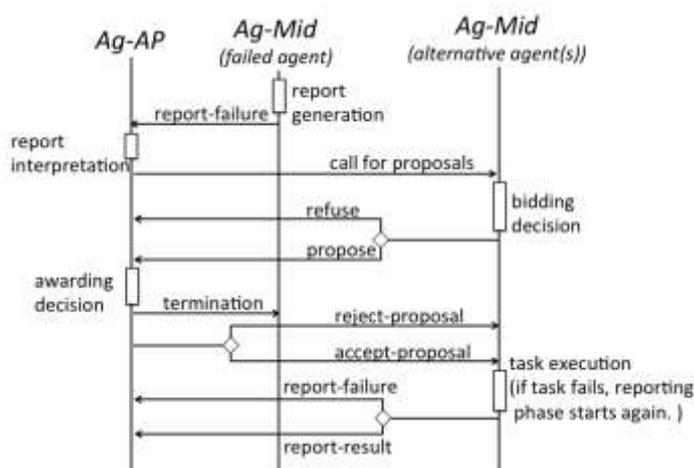


Figure 4. Re-organization process among application agent, failed and alternative middleware agent.

Application agent is called the manager of task, and the middleware agent is called the contractor. Application agent sends call for proposal to the middleware agents, which include the contract name, task abstract, and the requirement to become the contractor to process the task, as well as the deadline to bid on the sent task. The way of sending call for proposal is, point to point, multicast, and broadcast which the sending agent select one of them. The task received A contractor, the middleware agent, decides weather or not to bid on the task to process based on the condition of IoT device and software agent itself (bidding decision). If the contractor decides to bid on the task, it sends back the proposal defined in the call for proposal, otherwise it sends back a refusal message.

The manager, the application agent is likely to receive multiple bidding messages. Therefore, after the deadline of the task bidding, the manager selects one of the proposals based on the knowledge of the target task (awarding decision), and sends back the accept or reject message to the agents which sent the proposal messages. The contractor agent which received the accept proposal message execute the task by communicating with the manager. After the task

execution, it reports the result, or reports a failure report. In case of failure, or if the manager decides the replacement of contractor is necessary, the manager starts the re-organization process as described in Figure 4. The difference between the initial conclusion process and the re-organization process is the report interpretation phase. After the manager receives the failure report, the manager modifies the task description as needed, and start the contract conclusion process, and the termination of previous contractor after the new contractor is selected from the prospective alternative middleware agents.

#### IV. Implementation and Experiment of AIoT Application in Logistics Scenario

In order to confirm the effectiveness of proposed AIoT application architecture, agent architecture and organization scheme, we have implemented prototypical logistics application using small robots with attached sensors.

##### A. Implementation Configuration of Agents and IoT Devices

We used the repository-based multiagent framework of ADIPS [16]. The configuration of one of the implemented logistics robots is shown in Figure 5. The robot consists of 2 IoT devices, SunSPOT (multi-functional sensors) [17] and LEGO Mindstorms [18] NXT Intelligent Brick (simplified robot), and these devices have different controlling software, and controlled by different middleware agents, namely NXT1 agent and SunSPOT agent. Unmanned Ground Vehicle (UGV1) agent manages these 2 agents to work as a logistics robot. NXT1 agent and SunSPOT agent have the knowledge of controlling each device, and UGV1 agent has the knowledge to solve problems as a set of IoT devices.

##### B. AIoT Application in Logistics Scenario among Heterogeneous Devices

The experiment and result are shown in Figure 6. In the logistics experiment, we confirm the effectiveness of agent architecture and the autonomic organization scheme. The task is to carry a virtual packages form Point A through Point D. There are 3 areas, Place A-B, Place B-C, Place C-D. In each area, there are UGVs are logistics robots. There is also a backup logistics robot of Unmanned Aerial Vehicle (UAV) in area Place B-C. UGV and UAV both have the ability to carry virtual packages, and UAV has the advantage to carry things above obstacles. We use Parrot AR. Drone [19] in this experiment.

Figure 6. (a) shows the agent organization of logistics robots. Logistics agent is an application agent that manages the task of carrying packages from Place A to D periodically. Through the autonomic service organization scheme described earlier, the UGVs are selected in each area to perform the task to carry packages. After the agent organization process, the logistics application starts to work.

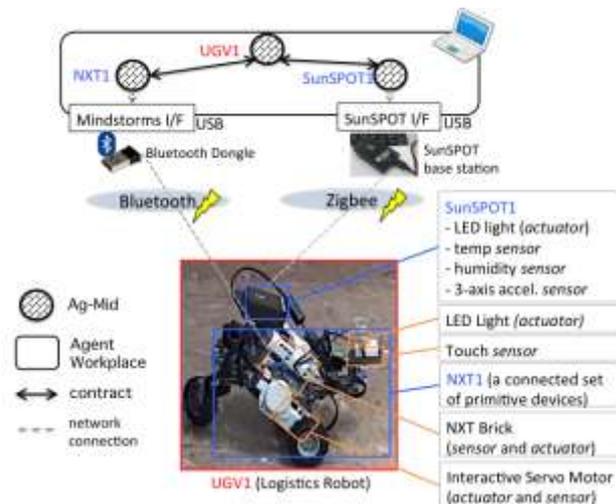


Figure 5. Agent and device configuration of UGV and sensor controlled by middleware agents.

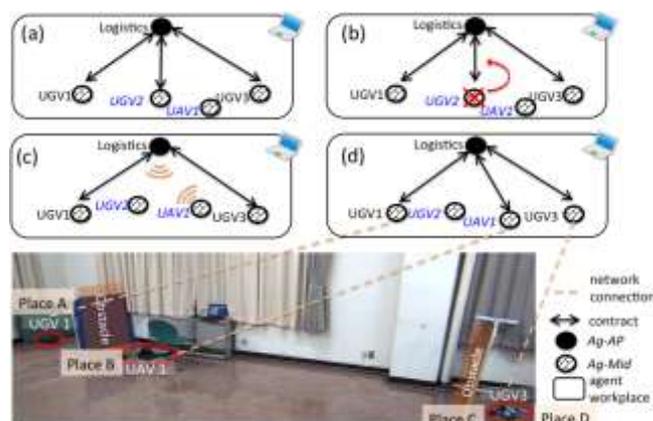


Figure 6. Experimental result of autonomic organization and re-organization to sustain logistics service among UGV and UAV controlled by middleware agents.

However, after a while, we place an obstacle in Place B-C area, so that UGV2 is unable to continue the assigned task. The obstacle is detected by the sensor attached on the UGV, and the existence of obstacle is reported to the manager. Then as in Figure 6. (b), according to the proposed re-organization process, Logistics agent organize the agents again, and select and assign task as shown in Figure 6. (c) (d). Because of the autonomic replacement in the area Place B-C, the logistics service continues.

#### v. Conclusion

In this paper, we proposed a multiagent-based autonomic service organization scheme for IoT devices, including the architecture of IoT device controlling agents. In the experiment, it is confirmed that our agent architecture and service organization scheme enable the capability to autonomously organize and re-organize services based on the requirements and situations.

#### Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 16K00118, 15J06341, and 16K00292.

## References

- [1] J. A. Stankovic, "Research Directions for the Internet of Things," IEEE Internet of Things Journal, vol.1, no.1, pp.3-9, Feb. 2014.
- [2] T. Tomita, K. Ushiki, Y. Kawakatsu, N. Fujino, H. Mineno, "Task-Driven Device Ensemble System Supporting Seamless Execution of User Tasks Despite Multiplexed Interruptions," International Journal of Informatics Society, vol. 5, pp.49–58, 2013.
- [3] L. Mainetti, V. Mighali, L. Patrono, "A Software Architecture Enabling the Web of Things," IEEE Internet of Things Journal, vol.2, no.6, pp.445-454, Dec. 2015.
- [4] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, S. Clarke, "Middleware for Internet of Things: A Survey," IEEE Internet of Things Journal, vol.3, no.1, pp.70-95, Feb. 2016.
- [5] G. Fortino, A. Guerrieri, M. Lacopo, M. Lucia, W. Russo, "An Agent-Based Middleware for Cooperating Smart Objects," Highlights on Practical Applications of Agents and Multi-Agent Systems, Springer Berlin Heidelberg, vol.365, Communications in Computer and Information Science, pp.387-398, 2013.
- [6] M. Ruta, F. Scioscia, E.D. Sciascio, G. Loseto, "Semantic-Based Enhancement of ISO/IEC 14543-3 EIB/KNX Standard for Building Automation," IEEE Trans. on Industrial Informatics, vol.7, pp.731-739, 2011.
- [7] M. Ruta, F. Scioscia, G. Loseto, E.D. Sciascio, "Semantic-Based Resource Discovery and Orchestration in Home and Building Automation: A Multi-Agent Approach," IEEE Trans. on Industrial Informatics, vol.10, pp.730-741, 2014.
- [8] N. Michal, K. Artem, K. Oleksiy, N. Sergiy, S. Michal, and T. Vagan, "Challenges of middleware for the Internet of Things," in Automation Control Theory and Practice. InTech, pp.247-270, 2009.
- [9] V. Terziyan, O. Kaykova, D. Zhovtobryukh, "Semantic Middleware for Context-aware Smart Road Environments," in Proc. 5th Int. Conf. Internet Web Appl. Serv. (ICIW), pp. 295–302, 2010.
- [10] A. Katasonov, V. Terziyan, "Smart Resource Platform and Semantic Agent Programming Language (S-APL)," in Proc. of 5th German Conference on Multiagent System Technologies, LNAI 4687, pp.25-36, 2007.
- [11] T. Liu, M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," ACM SIGPLAN Notices, vol.38, no.10, pp.107–118, 2003.
- [12] P. Kang, C. Borcea, G. Xu, A. Saxena, U. Kremer, L. Iftode, "Smart Messages: A Distributed Computing Platform for Networks of Embedded Systems," Comput. J. Spec. Focus Mobile Pervasive Comput., vol.47, pp.475–494, 2004.
- [13] Y. Kwon, S. Sundresh, K. Mechitov, G. Agha, "Actornet: An Actor Platform for Wireless Sensor Networks," in the Proc. of Fifth International Joint conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), pp.1297-1300, May, 2006.
- [14] C.-L. Fok, G.-C. Roman, C. Lu, "Agilla: A Mobile Agent Middleware for Self-adaptive Wireless Sensor Networks," ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol.4, no.3, pp.16:1-16:26, 2009.
- [15] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Trans. on Computers, vol.C-29, pp.1104–1113, 1980.
- [16] IDEA: Interactive Design Environment for Agent System <http://www.k.riec.tohoku.ac.jp/s/idea/>
- [17] SunSPOT <http://www.sunspotdev.org/>
- [18] LEGO Mindstorms <http://www.lego.com/en-us/mindstorms/>
- [19] Parrot AR. Drone <http://www.parrot.com/jp/products/ardrone-2/>



**Takumi Kato** received his B.S. in Engineering from Sendai National College of Technology in 2011, and the M.S. in Information Sciences from Tohoku University in 2013. He is currently a Ph.D. candidate at the Graduate School of Information Sciences (GSIS), Tohoku University. His research interests include Internet of Things (IoT), knowledge-based system and agent-based computing. He is a member of IPSJ and IEEE.



**Hideyuki Takahashi** received the doctoral degree in information sciences from Tohoku University, Japan, in 2008. He is currently an assistant professor at the Research Institute of Electrical Communication, Tohoku University. His research interests include ubiquitous computing, green computing, and agent-based computing. He is a member of IEICE and IPSJ.



**Tetsuo Kinoshita** received the BE degree in electronic engineering from Ibaraki University, Japan, in 1977, and the ME and DrEng degrees in information engineering from Tohoku University, Japan, in 1979 and 1993, respectively. He is currently a professor at the Research Institute of Electrical Communication, Tohoku University. His research interests include agent engineering, knowledge engineering, knowledge-based systems, and agent based systems. He received the IPSJ Research Award in 1989, the IPSJ Best Paper Award in 1979, and the IEICE Achievement Award in 2001. He is a senior member of the IEEE.