A Distributed Agent System for Managing a Web-based Sensor Network with Field Servers

Tokihiro FUKATSU[1], Masayuki HIRAFUJI[1], Takuji KIURA[1]

1 National Agricultural Research Center, National Agriculture and Food Research Organization,
3-1-1 Kannondai, Tsukuba, Ibaraki, 305-8666 Japan, fukatsu@affrc.go.jp

Abstract. This paper presents the framework and function of an agent system developed for Field Server to manage Web-based sensor network. This system consisting of an agent program, program configuration files, and a Web interface, can easily and autonomously manage Web-based sensor nodes via the Internet. It can perform simple function such as monitoring to more complex and intelligent operations to adapt to various conditions as specified by the user in the program configuration files. The users can easily access and change the agent management via the Web interface in the agent system. It is also designed to handle Web-based nodes, Web applications and Web information as nodes, so its functions can be easily expanded by simply interacting with these nodes without changing the agent program. Moreover, the agent system is designed to treat itself as just one of the nodes resulting to a highly scalable and robust system. A meta-agent system, which can treat another agent system, coordinates the function of managing many nodes in the whole system so that it can achieve scalability system. The nested structure of the agent system in which several agent systems handle several nodes multiply, serves as a data backup mechanism as well as a way to avoid bottlenecks. Results of the trial experiments show the agent system for Field Servers is a robust and scalable system for long-term field monitoring, effective data handling, and management based on the situation.

Keywords. Field Server, Sensor Network, Agent System, Web monitoring, Scalability, Robustness.

Introduction

The study of agent systems and its applications to several areas has been attracting attention due to the increasing use of complex and large-scale information systems. Many believe that its characteristics such as autonomy, goal-orientedness, collaboration and mobility provide important means to address the development of distributed and complex information systems (Russell et al., 1995, Ferber, 1999). Field Servers (Fig.1), which were developed in our previous work (Hirafuji et al., 2005), are sensor nodes equipped with Web servers and enabled with various sensors and peripherals used to monitor environmental conditions and agricultural processes. This device can be easily accessed remotely using a Web browser such as Internet Explorer to get sensing data and to control its peripherals. By developing an agent system that will execute these operations autonomously, a Field Server system will be able to perform long-term monitoring, effective data handling, and management based on the particular situation, and provide high scalability and robust system. Also, with Field Servers' ability to enable ubiquitous network environment, the scope of the systems operation can be easily extended. In this paper, we discuss the agent system developed for Field Servers and the results of the initial trials.

The agent system serves as a data logging program for Field Servers, but it also works as a unit which makes an intelligent sensor network of Web-based sensor nodes that includes Field Servers which generally do not function by themselves to provide complicated judgments or an autonomous operation (Fukatsu et al., 2004b). The agent system can also handles not only Field Servers but also other Web-based equipments, Web applications and Web information as Web-based sensor nodes in the same way. Moreover, the agent itself can be treated as an independent node in our system, so it provides a scalable multilayered system and an upper tier sensor network system consists of agents cooperating with each other as nodes.

The Agent Program

The agent system has design accessible via the Internet and is constructed by an agent program, program configuration files, and a Web interface (Fig.2). The agent program written in Java, which has less machine dependence and provides easy network handling, can be easily operated on a java-installed computer. The agent program regularly confirms program configuration files (Profiles) which are described in a XML format and operates based on the contents of these files. Users can utilize the agent program for their Field Servers by simply changing the IP address, the file storage location, or sensor parameters in the Profile form. Even where each Field Server is equipped with different sensors, cameras, or other peripherals, it can easily be managed by only having to prepare unique Profiles for each Field Servers.

This Profile consists of not only detailed target node information but also agent program instructions combined with some action commands which emulate user operations on a Web browser, such as Post, Get, Write and so on (Fig.3). By changing the combination of action commands, the agent program operates flexibly in response to the user's requests. For performing more intelligent instructions in adapting to various situations, IF-THEN rules and Operator functions, which are derived whether or not the agent action is executed from a monitoring environment, can also be described in the Profile. By providing these specifications in the node Profiles with these functions, the agent program can act as an artificial intelligence (AI) function of the Web-based sensor nodes in order to construct the Web-based sensor network. This also enables an easy way to dynamically change various algorithms in the sensor network from a remote site (Fukatsu et al., 2004a).

The agent program operates by accessing Web nodes and extracting target information with character masking as described in their Profiles. By only having to change these Profiles, it can be modified to treat not only Field Servers but in the same way also various Web-based targets with the http protocol such as other Web-based equipments, Web applications and Web information accessible on the Internet. By treating Web applications and Web information that function as nodes in the agent system, the agent system is more versatile and van easily expand its function without changing itself as a whole. This makes it possible to distribute calculation tasks on the Internet (Fig.4).

An Agent Web Interface

The agent system consists of an agent program, node Profiles, and a Web interface that facilitates access to node Profiles, the agent information, and monitor data. When constructing the Web interface of an agent system in the most simple way, it is only necessary to put the Profiles into a directly accessible folder so that one can create, edit, and delete them in order to change their action. Linking with some Web pages which support making Profiles and transferring created files, one can easily configure them with a GUI-based function. For a more secure system and to exchange more information, a Web interface is constructed using a Java servlet in the standard agent system (Fig.5). In this system, we can monitor and operate the elements of the agent system via an http protocol simply by accessing it. To respond to this access, the agent program is modified to operate itself based on the agent Profiles which are described as handling nodes, operational parameters, and a management algorithm via the Web interface. Consequently, the single agent system itself can be treated as a node in our system so that another agent system can control it via the Web interface in order to change its operation.

A Multi-Agent System

Designing the agent system to treat itself as just one of the nodes bring about a highly scalable and robust system without changing the agent program. In this system, an agent system, which is called a meta-agent system, can treat another agent system with some nodes. To manage many nodes with a whole system, it is only necessary to handle a meta-agent system by providing node tasks. The meta-agent system manages how many tasks should be allocated to each agent system. The number of tasks handled by each agent system is allocated based on its capacity and the nodes' situations. The meta-agent system accesses each agent system in order to consign some tasks, and each agent system manages some nodes based on its tasks. It can perform effective and automatic management so as to decrease users' efforts (Fig.6). When the number of nodes increases, the system can adapt with high scalability to be implemented in layered meta-agent architecture in which a meta-agent handles other meta-agents according to their scale.

A nested structure with several agent systems can be also realized so as to provide a high guarantee for various factors. In this system, several agent systems can handle several nodes that multiply in the nested structure. For the practical demand to reduce the accessing load of nodes, the system is constructed so that a secondary agent system doesn't access the target nodes while the primary agent system works normally. The secondary agent system periodically accesses the primary agent system in order to check its condition, and it operates on the target nodes only when the primary agent system is down. By cooperating with the multiple agent system, it performs as an effective data backup system making the system robust (Fig.7). In addition, for the exchange of task node conditions and circumstance information among agent systems, we can construct a sensor network for the agent system which will autonomously adjust routing or task nodes.

Trial Management

We installed dozens of Field Servers in different places and managed them using the agent system for more than four years via the Internet (Fukatsu et al., 2005). A single agent program independently handles many nodes via multi-thread processing based on machine power. In some cases where managing large numbers of nodes exceeded machine power, the system can work with a distributed agent system which is easily expandable in order to manage them with several agent programs and computers (PC clusters). The multi-agent system was constructed with a cluster of 7 computers for the agent system and 1 computer for the meta-agent system (Fukatsu et al., 2006). The meta-agent system receives the requests of node management from users via its Web interface and updates a node list. Other agent systems regularly receive their tasks from the meta-agent system and automatically manage them (Fig.8). In this experiment, the agent program doesn't have a built-in algorithm to optimally divide the tasks. But the meta-agent system in collaboration with a separate Web application which designed to divide the tasks from a node list and the condition of each agent system performs this function.

In some experiment sites, there is a specialized small computer "Field Server Agent Box (Fig.9)" which is mounted with an agent system, a storage device, a VPN function, a Web server, and a data browsing application. This agent box works as a single agent system for managing nodes in its site and at the same time works as an agent node for a data backup system that cooperates with a remote agent system. In this experiment, the agent box has a low AI algorithm which passively operates to manage nodes only when the primary remote agent system is down. There are some issues in the future that will require the upgrading of the advanced functions of effectively transferring data and synchronizing or accessing other agents in an active way.

Future Work

Our experience in using Field Servers showed that many users don't have enough specialized knowledge and time to manage the agent system by themselves. For these users, it is important that the agent system can be utilized without much effort. In the past, there were many required procedures to manage nodes, such as setting up node configuration, sensor calibration, connecting other nodes, making Profiles, maintaining storage, selecting the agent in charge, creating output Web pages, and uploading the node list. The proposed system reduces some of these initial steps and makes it more effective in upgrading certain functions such as management algorithms, making active connections, sharing Profiles, automatic searching, fault-tolerance and so on. On the other hand, there remain some important issues such as the automatic calibration of nodes as in the case of conventional sensor networks. Other issues include effectively transferring data and synchronizing or accessing other agents actively. Research on sensor networks is improving everyday. Cooperating with this work and the solutions coming from it, our agent system will progress and develop further.

Acknowledgements

This study was supported by funds from the MEXT (Ministry of Education, Culture, Sports, Science and Technology) via the "Development of the Fundamental Technology about Data Integration and Information Fusion on Earth Observation" Project.

References

J. Ferber. 1999. *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. Harlow, Addison Wesley Longman.

M. Hirafuji *et al.*. 2005. Field Server: Multi-functional Wireless Sensor Network Node for Earth Observation. *Proc. of the 3rd International Conference on Embedded Networked Sensor Systems* 304.

S. Russell, and P. Norvig. 1995. *Artificial Intelligence A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey 07458.

T. Fukatsu, and M. Hirafuji. 2004a. The Agent System for Field Monitoring Servers to Construct Smart Sensor-Network. *Proc. of 5th International Workshop on Artificial Intelligence in Agriculture* 1-5.

T. Fukatsu, M. Hirafuji, T. Kiura, A. Imada, and S. Ninomiya. 2004b. Long-Term Monitoring System using Field Monitoring Servers. *Proc. of AFITA/WCCA2004 Joint Conference* 685-691.

T. Fukatsu, and M. Hirafuji. 2005. Field Monitoring Using Sensor-Nodes with a Web Server. *Journal of Robotics and Mechatronics* 17(2): 164-172.

T. Fukatsu, M. Hirafuji, and T. Kiura. 2006. Agent System for Operating Web-Based Sensor Nodes via the Internet. *Journal of Robotics and Mechatronics* 18(2): 186-194.

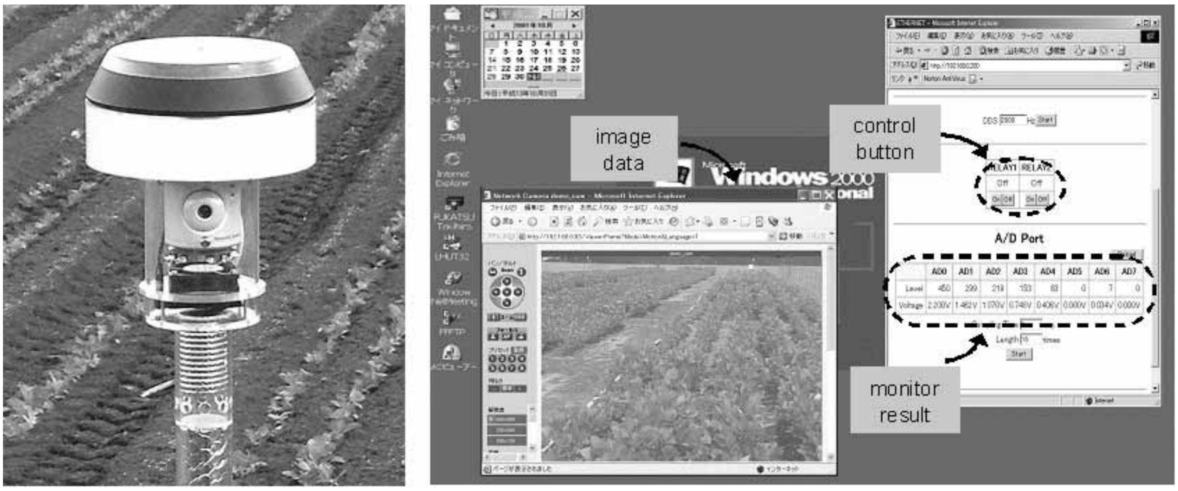

Figure 1. Field Server as a Web-based sensor node.

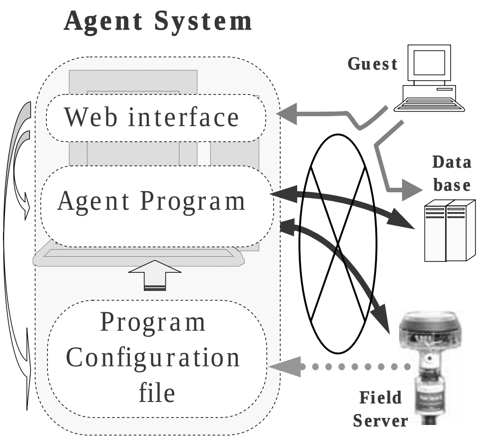

Figure 2. Structure of agent system constructed by agent program, program configuration file, and Web interface.

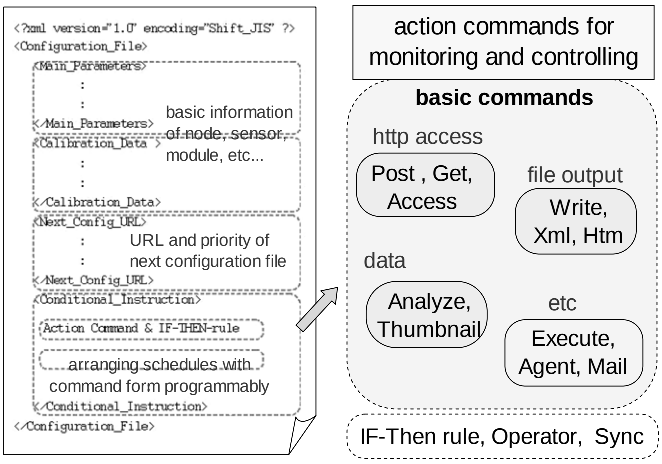

Figure 3. Program configuration file which contains basic information and operating instructions with action commands.

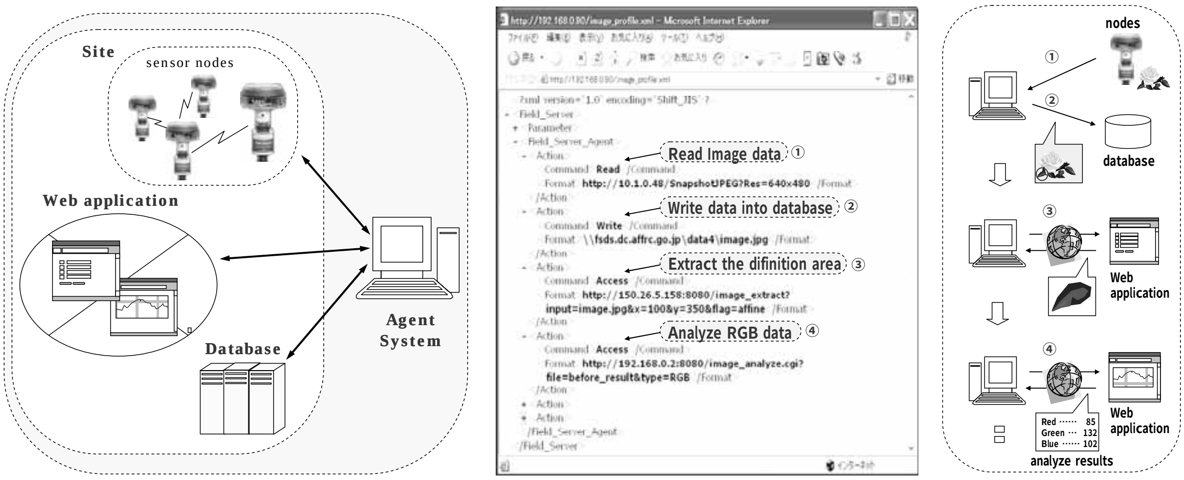

Figure 4. Example of expandable command using Web applications. By accessing Web applications which extract and analyze images, the system can obtain analyzed information from monitoring image data,

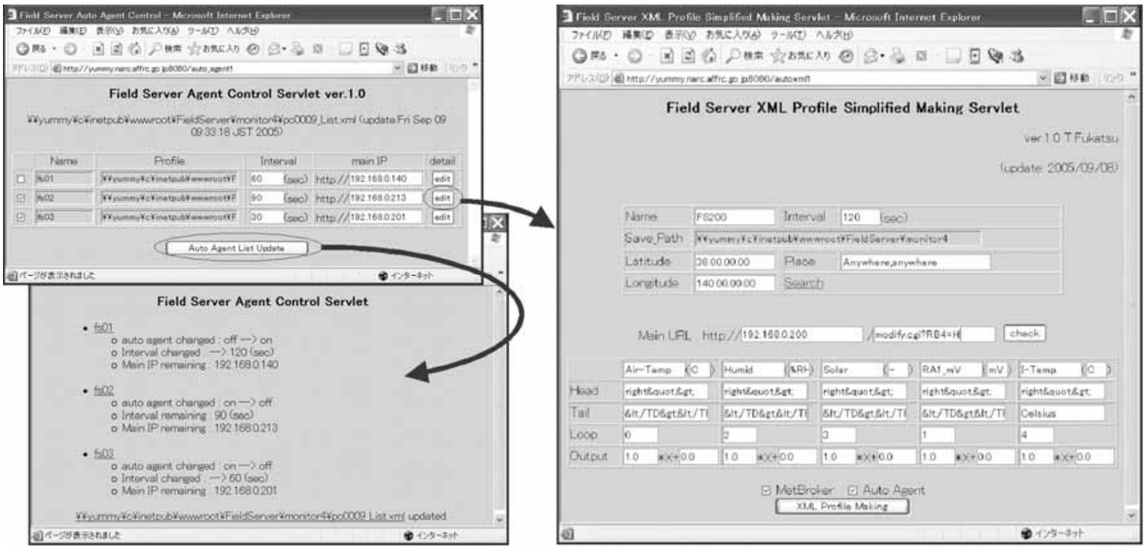

Figure 5. Web interface of agent system constructed by Java servlet. In this page, we can add or remove nodes in charge, and edit node Profile in detail with GUI interface.

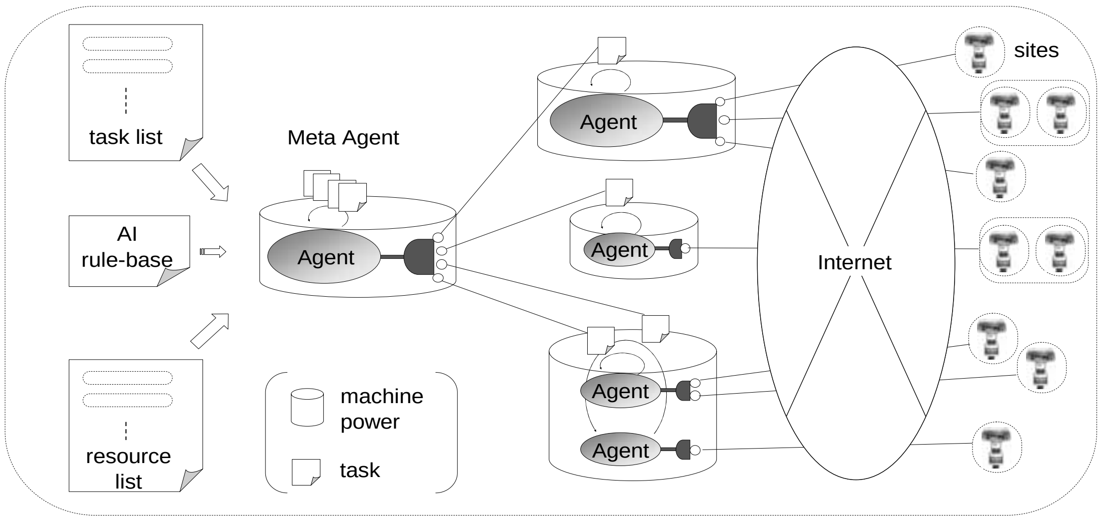

Figure 6. Meta-agent system distributing agent tasks to other agent systems based on machine power.

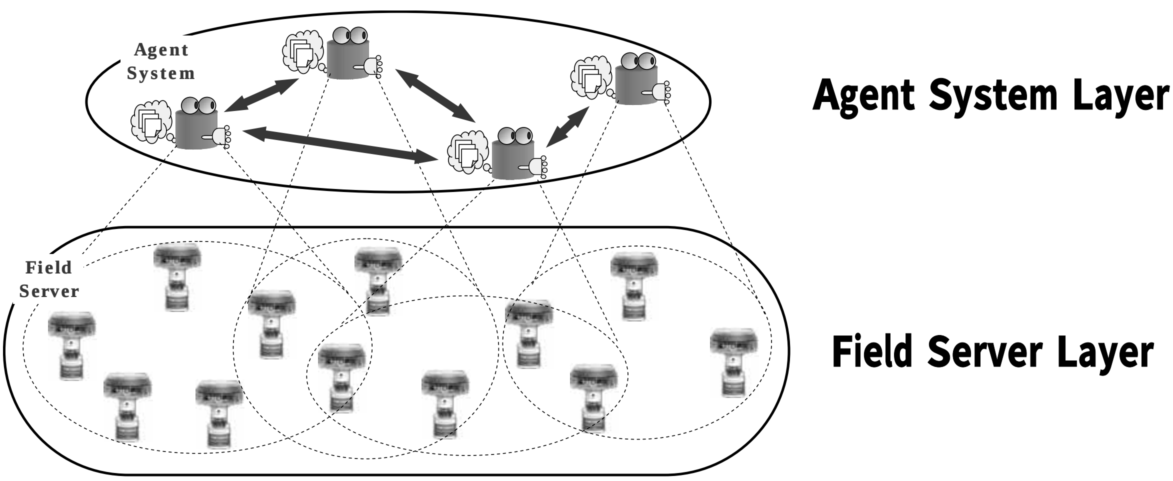

Figure 7. Multiple agent system constructed by nested structure in which several agent systems handle several nodes.

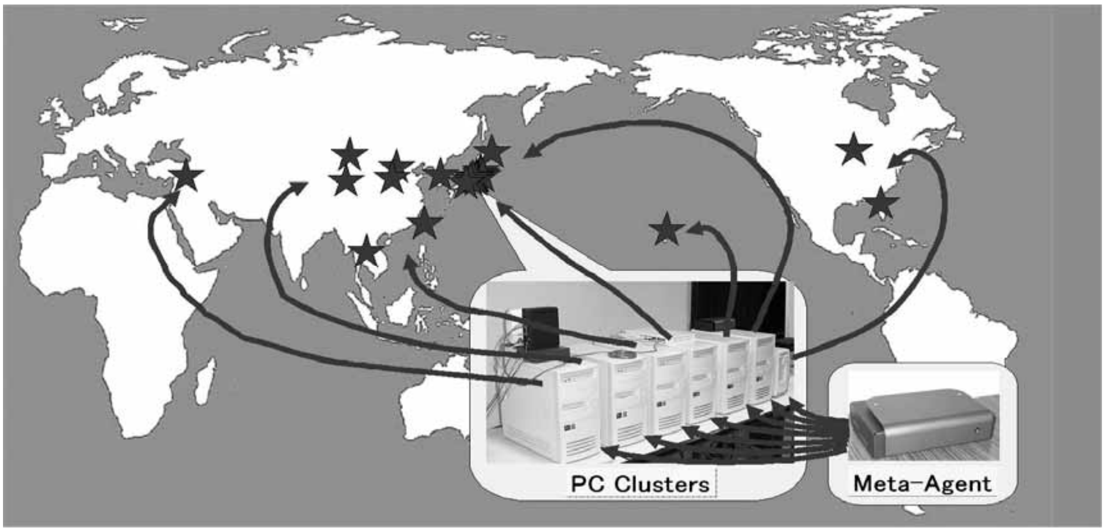

Figure 8. Trial management with PC Clusters and Meta Agent system which manage several dozens of Field Servers in deferent places since 2002.

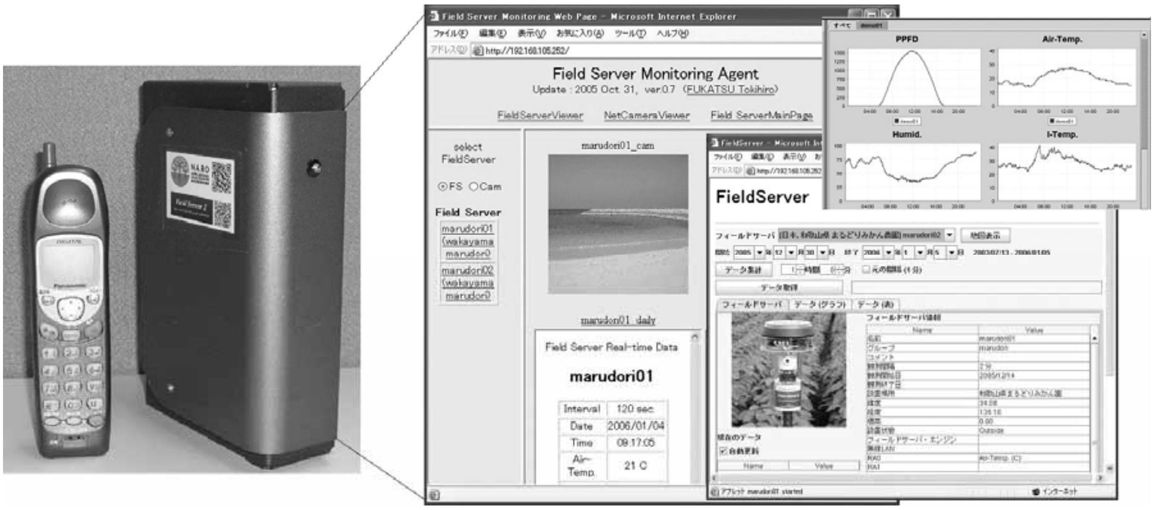

Figure 9. Field Server Agent Box which is mounted with agent system, storage device, VPN function, Web server, and data browsing application.