PRE

Journal of Robotics and Mechatronics
Vol.18 No.2, 2006 pp.186-194

An Agent System for Operating Web-based Sensor Nodes via the Internet

Tokihiro FUKATSU, Masayuki HIRAFUJI and Takuji KIURA

National Agricultural Research Center
3-1-1 Kannondai, Tsukuba, Ibaraki 305-8666, Japan
E-mail: fserver@zoushoku.narc.affrc.go.jp

Abstract: Web-based sensor nodes, which have a Web server for monitoring data and operating devices, could be used for their easy installation, monitoring, and management with small effort on the part of users. To realize a sensor network system with a high scalability, fewer legacy problems, and general versatility, we propose an agent system for operating Web-based sensor nodes via the Internet. In this agent system, we can handle many kinds of sensor nodes flexibly and uniformly with the agent program and the configuration file. By constructing all objects of the agent system based on the Internet, it is possible to enhance the expansion of operation, control, and scale. By making the agent architecture, algorithm, and implementation, we demonstrate the capabilities and reliability of this system as a useful sensor network.

Key Words: Agent System, Sensor-Node, Remote Monitoring, Internet, Field Server

1. Introduction

In the agricultural and environmental sciences, it is important to be able to easily monitor field and environmental information over long periods of time, but such monitoring is difficult and requires much effort because it must be performed in a wide geographic area under harsh conditions [1]. One approach to solving this problem would be to employ a sensor network that would enable users to monitor in these conditions using many sensor nodes made up of small sensor units with radio data links. Research on sensor networks, which are typically known as “Mote [2]” or “Sensor Web [3]”, has focused on the networks for power saving [4, 5] and on the Artificial Intelligence (AI) algorithm [6] for autonomous operation, and these traditional sensor networks use exclusive radio communication and expensive hardware. The issues examined are mainly routing [7, 8], data transmission [9, 10], discussing position sensing [11], synchronization [12], and optimal arrangement [13]. Some researches show using of sensor networks as habitat monitoring applications [14, 15] and environment observation applications [16, 17], but these systems only consider situations in which many identical specific units are arranged and monitor with simple similar sensors.

In the case of agriculture, it is important to handle various kinds of sensors in real time. Various types of information are necessary for monitoring crop fields in order to simulate crop growth and to predict the occurrence of harmful organism [18, 19]. We require not only measurement but also the control and operation of some devices based on measured data at a remote site [20]. The system must also be able to treat large blocks of data such as image sources for crop or livestock conditions to respond to various situations with flexibility [21]. Moreover, it is necessary to change sensor nodes’ locations, configurations, and operation according to the situation, because the specific required information differs continually for every growth stage or crop. A useful sensor network in the agricultural field must deal with various nodes for general-purpose usage.

In our previous work, we proposed a Web-based monitoring system to realize long-term field monitoring that would consider these requirements [22], and we developed general-purpose sensor nodes called “Field Servers” (**Fig.1**). Field Servers are equipped with Web servers for managing various sensors including cameras, and a wireless LAN to provide a high-speed transmission network at a low price. In this type of monitoring, Web-based monitoring nodes can easily treat a variety of data from any place at any time and can reduce the amount of effort required of a user. Along with this Web-based system, the use of high-quality, low-cost Web camera is also gaining popularity according to advancement of Information Technology (IT).

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

In this circumstance, we propose an agent system for operating Web-based sensor nodes collectively and uniformly via the Internet as a new approach to building a sensor network (**Fig.2**). By constructing the system using Web-based sensor nodes effectively, it can achieve a high scalability, fewer legacy problems, and general versatility that will reduce the user’s effort in using the system. In the present paper, we propose the concept and impact of the agent system for a Web-based sensor network, and we describe the agent architecture, algorithm, and implementation for operating the system effectively.

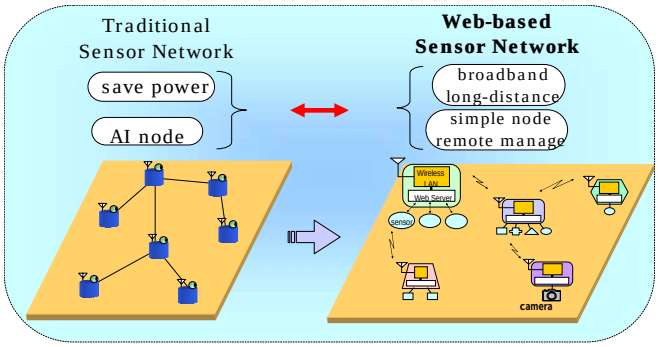

Fig.2. Traditional Sensor Network vs. Web-based Sensor Network.

2. Web-based Sensor Network

2.1. Web-based Sensor Node

A Web server is software that can present Web information to a client based on TCP/IP. By building a Web server into sensor nodes, it can realize universal, expansive, and complicated management in comparison with ordinary nodes. These features of the nodes have several positive effects.

- Ethernet module
  In Web-based sensor nodes, every module is connected to others via Ethernet, so it is easy to exchange or add modules. Such a system can be used to introduce the latest high-performance commercial modules of IT, such as network cameras and wireless LAN. A wireless LAN (IEEE802.11b/g) provides high-speed transmission and long-distance communication at low cost. Use of the wireless LAN slightly increases power consumption but it provides better performance than other wireless technologies [23] from the point of view of the distance of calls and the traffic efficiency (transfer rate per power consumption) (**Table.1**). A wireless LAN can be used to handle large amounts of data and to manage these nodes from a remote site.

Table 1. Performance comparison of wireless technologies.

	Telemeter ARIB STD-T67	ZigBee IEEE802. 15.4	Bluetooth IEEE802. 15.1	Wireless LAN IEEE802. 11b/g
Frequency	429MHz	2.4G/915 MHz	2.4GHz	2.4GHz
Transfer rate	4.8kbps	250kbps	768kbps	11/54Mbps
Power consumption	130mW	60mW	110mW	1200mW
Distance of calls	～2km	～75m	～100m	～300m
Traffic efficiency	0.04kbps /mW	4.2kbps /mW	7.0kbps /mW	～45kbps /mW

- HTTP protocol
  Web-based sensor nodes present data with http protocol, which is a universally accepted on the Internet. We can easily access and manage the nodes using a Web browser such as Internet Explorer, without installing exclusive software. In this way, it is possible to treat all nodes with the Web server in the same manner. By controlling them with IP addresses, we can construct a high scalability system that is less affected by the number of nodes. In addition, every type of unit can be handled on the Internet to reduce legacy problems, in which old measurement equipment cannot be adapted to a new system.

- Simple firmware
  A Web server can be built with simple firmware and on a low-end CPU, so the sensor nodes mounted on them can be constructed cheaply and remain stable, resulting in less bug generation. These nodes themselves don't have highly programmable function such as an AI algorithm inside; they respond to remote access through the network to perform various monitoring and other complicated operations.

2.2. Agent System Concept

Using Web-based sensor nodes, we can construct a Web-based sensor network system with high scalability, fewer legacy problems, and general versatility. In this system, every sensor node presents data at any time by directly accessing it with a Web browser. However, some nodes show only real-time data because they lack a data logging function, and other nodes periodically need to be accessed in order to operate them accordingly. It is also possible to manage them by local access of a user periodically, but such a system imposes a heavy burden on a user. Thus, software for an agent to manage the sensor nodes is a necessary part of the sensor network system over long periods of time.

Only collecting node data periodically on the Web, we can apply software or Web crawlers that automatically collect Web pages and store the data in databases. However, these programs have only a logging function. To realize the agent system, it is important to include the following functions:

- handling any kinds of pages on the Web
- autonomously changing management according to the situation
- cooperating with various nodes, data, and applications
- forming an easily distributed and extensible system for scalability

Figure 3 shows the architecture of the agent system which is composed of an agent program, configuration files for the program, a database of monitoring data, and sensor nodes. This system is designed on the assumption that all objects of the system can be accessed via the Internet. By distributing objects on the Internet, we can manage and improve them separately from anywhere at any time. The agent program is designed so that it may operate according to the configuration file, which indicates a series of operation. For example, it performs to access the intended node, analyze

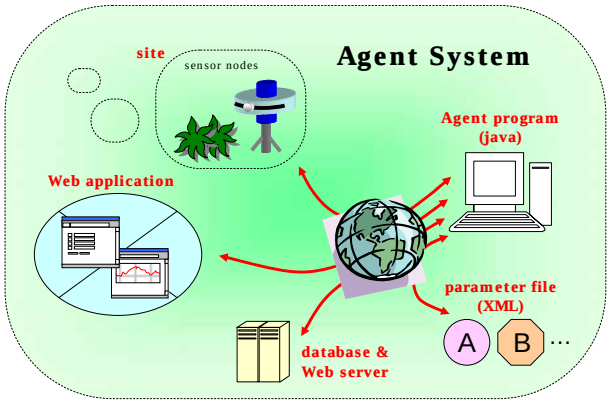

Fig.3. Architecture of the agent system.

the data, and store the results at the dictation.

By simply changing a configuration file via the Internet, a user can easily manage the agent operation from a remote site with general versatility. Thus, the program need not be located at the remote site individually. Users can utilize this system simply by installing sensor nodes and getting a connection at a remote site, which reduces the effort needed to accomplish monitoring.

Measurement data collected by the agent program is stored as a database in the extensible markup language (XML) format on the Internet, and we can easily store enormous amounts of data via distributed storage on the network. Moreover, databases are commoditized on the network, so they can be easily linked with various applications such as “MetBroker [24],” a middleware for weather data that provides agricultural models with consistent access to many different weather databases. Several MetBroker-compatible models have already been designed, and many of their applications can be used for agriculture involving on-site field information from nodes’ databases [25, 26].

To construct this agent system, we designed an agent program written in Java and the description of the configuration file in XML format. Here we discuss the core technology of the architecture, algorithm, and implementation for operating the system effectively (**Fig.4**).

3. Agent System Architecture & Implementation

3.1. Operating Instructions

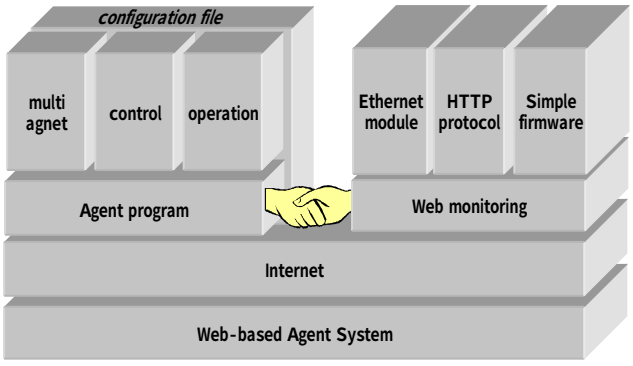

Fig.4. Technical framework of the agent system.

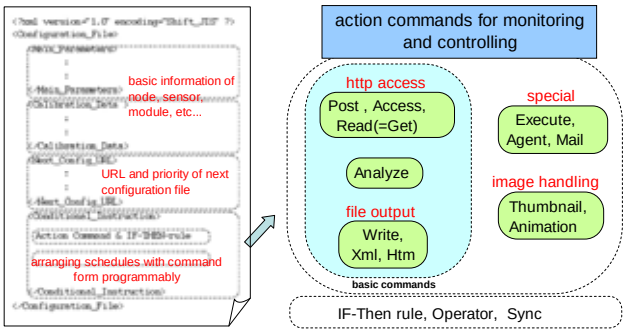

Fig.5. Configuration file and operating instructions with action commands.

In the configuration file, agent instructions are described with some combination of action commands so that the agent system may work in a variety of situations (**Fig.5**). A user can manage various kinds of agent operations such as monitoring and controlling sensor nodes by easy programming to combine these commands with the XML text data. Each action command is mainly constructed with a simple operation which emulates the user’s operations in a Web browser, including the category of accessing the uniform resource identifiers (URI) to get and send data (Post, Access, Read (=Get)), the category of analyzing and extracting information from html data (Analyze), and the category of saving the information in the desired format (Write, Xml, Htm). There are also commands such as “Animation” and “Thumbnail” for handling image data, “Execute,” “Agent,” “Mail,” and so on.

If more function is needed in the agent program, it isn’t impossible to make a new action command by re-coding the program, but it is not suitable for practical use that the agent program will be changed and rebooted each time in response to the requests. This agent system is designed so that any user can utilize it in various situations, so it is impossible to realize all requests by using this method, but it can easily access and handle Web applications via the Internet with basic action commands without changing the agent system’s program. By accessing and utilizing the Web applications that can execute the desired function, we can extend new functions to the system on demand without changing or rebooting the agent program by simply making or finding the Web application. For example, by using the image

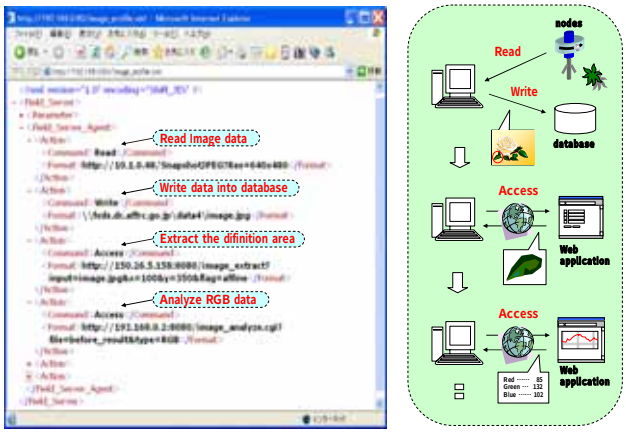

Fig.6. Example of the extensible command using Web applications.

extracting Web application and the image analyzing Web application, the agent system, which itself doesn't have these functions, can obtain the analyzed information by collecting image data from sensor nodes through the Web applications without re-coding it (**Fig.6**).

By using this method, it is possible to distribute the task for calculation by the Web application, and the agent system can be used as an expansive and general-purpose system. This promotes more effective use of the Web applications, Web information, and Web models via the Internet.

3.2. Control

Some Web-based sensor nodes can not only perform measurement but can also operate attachment devices with the GET /POST method of http protocol. The agent system has an IF-THEN rule and an Operator function so that it can control the measurement and operation according to the situation. In agent instructions, we can simply specify an IF-THEN rule to determine whether or not each command is executed according to the situation, such as time, sensor value, and some flags. **Figure 7** shows an easy example of the agent control. In this configuration file, there is a description of the conditions that cause a light to be switched on the node if the temperature falls below 4 degrees and it is later than 19:00, and the agent executes at the dictation of the configuration file.

As in the case of the extensible commands, we can also separate the function of control algorithm from the Web application by inputting parameters and receiving the results of the judgments. By using the Web application and adapting the results to an inner IF-THEN rule, we can control the sensor nodes with another algorithm. We can establish simple condition judgments and complicated rule-based AI algorithms in this system.

Besides controlling the individual action command, we can control the agent parameters such as the execution interval time and the save destination by directly accessing and changing the configuration file. Moreover, the agent system can shift suitably the whole configuration file which

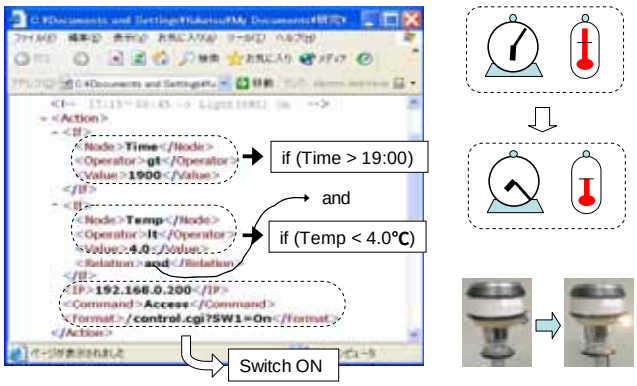

Fig.7. Example of the agent control with IF-THEN rule.

is selected from the configuration file list according to the situation (**Fig.8**). In the configuration file, the configuration file list is described with several pairs of the priority and the

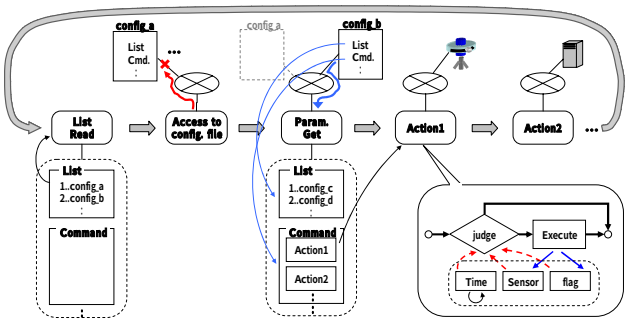

Fig.8. Flow of the agent program using the configuration file list.

URI of the configuration file. The agent system is managed by the configuration file with higher priority, which can be changed in this system according to the situation. If the program can't execute with the primary configuration file, next priority file will be accessed.

This agent system is designed to utilize cascade control with each action command and the whole configuration file.

3.3. Multi-Agent

A single agent program can independently handle many tasks of the configuration file by deriving many threads. If the computer has high-performance machine power, it can respond to manage more tasks on a single agent. But centralized management of the agent system is undesirable from the point of view of risk management and scalability. The agent program can be controlled with the configuration file, which can be accessed via the agent program. By using the meta-agent program which is the same agent program and manages some agent programs with their configuration files, we can construct the distributed agent system (**Fig.9**). A single agent program manages several tasks, and a meta-agent program manages several agent programs. By treating the meta-agent program as a single agent program, we can establish one of the multi-agent systems with layered structure and achieve scalability.

The meta-agent program manages how many tasks should be allocated to each agent program. The number of tasks that can be handled by a single agent program differs based on machine power. To allocate many tasks effectively to each agent program running on the computer, we introduce the evaluation index of the agent task. In this paper, we simply

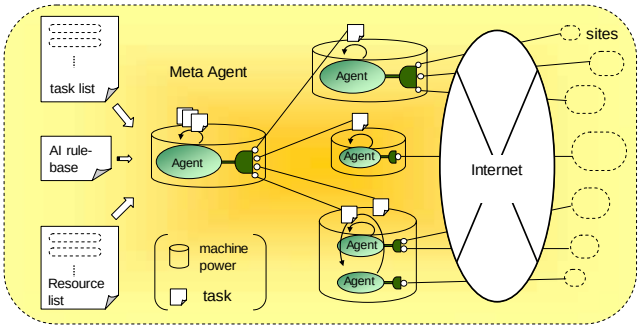

Fig.9. Meta-agent system which distributes agent tasks to other agents according to machine power.

Table 2. Evaluation index of the agent task.

$$\begin{cases} E_{agent\ task} = \Sigma (T_{task} / t_{interval}) \\ T_{task} = \Sigma (T_{command}) \\ T_{command} = f_t(P_{computer}, \text{command type, object, } P_{network}) \\ P_{computer} = f_{p1}(\text{CPU clock, memory size}) \\ P_{network} = f_{p2}(\text{throughput, response time}) \end{cases}$$

define the index (E_{agent_task}) as the summation of the proportion of the execution time (T_{task}) to the interval time ($t_{interval}$) for each task (**Table.2**).

In this definition, the execution time of each action command ($T_{command}$) mainly depends on the performance of the running computer ($P_{computer}$), which is represented by CPU clock and memory size, the type of the action command, the file size of the object, and the network environment of the site ($P_{network}$), which is represented by throughput and response time. The execution time of each task is defined by the summation of the execution time of each action command.

By applying the index to the control algorithm of the meta-agent program, the agent system can manage many tasks to divide based on each computer's machine power. In this system, it is possible to utilize legacy computers. The agent system can be constructed with many computers (PC clusters), which don't have to be high performance machines, by distributed the agent tasks according to the machine specification.

By changing the agent topology, we can construct various kinds of multi-agent systems. In a Web-based system, the connection is not structurally guaranteed by various factors, so the "1: n" topology in which one node is managed by many agent programs is useful as a data backup system. And it is also effective for a data backup system to construct cooperation between the main agent and the backup agent. The backup agent program, which is usually installed and sleeps at a remote site, starts in the case the main agent program doesn't work for some troubles such as a network. This type of agent system is good for the situation of decreasing node tasks, as it saves node power.

4. Performance Demonstration

4.1. Evaluation of the System

To demonstrate the proposed agent system, we first examined the agent program and evaluation index with a single agent system. We then evaluated the multi-agent system with a large number of sensor nodes in field experiments.

The computer (CPU: Pentium 4 1.5 GHz, Memory: 128 Mbytes, OS: Windows 2000 Professional, Java: JDK 1.4.2) for the agent program and the Field Server (2005 model with 4 sensors and a 0.3 M pixels image camera) for the sensor node were used as the execution environment of this system. In this environment, we examined whether the agent system worked well and evaluated some fundamental data concerning the index.

Table 3. Execution time of each action command in a standard environment

Action Command	Read, Post, Access	Thumbnail	Animation	Xml	Analyze, Write, Htm
Execution Time	~2000ms (~1Mbyte)	~600ms (~1Mbyte)	~270ms (~720files)	~550ms (~720files)	~10ms
Variation Factor	Network	CPU	CPU & numbers	CPU & numbers	———
Fixed Factor	(file size)	file size & ratio	———	———	———

Table 3 shows the results of the execution time of each action command in this standard environment (the network environment: 10 Mbps). There are some factors which fluctuate the time during operations. One of the categories is the "Xml" and "Animation" commands which continually change the time that is proportional to the amount of collected data stored by each folder in one day to make a list of the data. This category and "Thumbnail" command are also affected by computer performance. The execution time almost corresponds with CPU road time. In the situation of many threads (N threads), the time is simply estimated at N times with a margin because of the effect of intersectional efficiency. **Figure 10** shows the experimental results of the integrated CPU time and the estimated time in N threads.

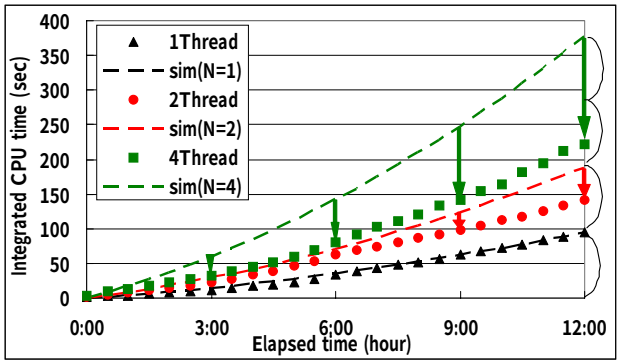

Fig.10. Experimental results of integrated CPU time in N threads

The other category is the "Read" command affected by network performance. In this category, the access time (T_{access}) is approximated by file size, throughput, and response time ($T_{response}$) in large part (**Table 4**). In http protocol with packet transfer communication, the effective transfer rate (TR) becomes slower with small file size because of the effect of response time. In the situation of many threads (N threads), the time is also simply estimated at N times. **Figure 11** shows the experimental results of access time in N threads. In some cases of the sensor nodes with a low-performance Web server, response time is greater for outputting an html file.

Table 4. Approximation of the access time

$$\begin{cases} T_{access} = K \cdot x + T_{response} \quad (s) \\ TR = x / T_{access} \quad (Mbps) \\ \left(\begin{array}{ll} x & : \text{file size} \quad (Mbyte) \\ 1/K & : \text{throughput} \quad (Mbps) \end{array} \right) \end{cases}$$

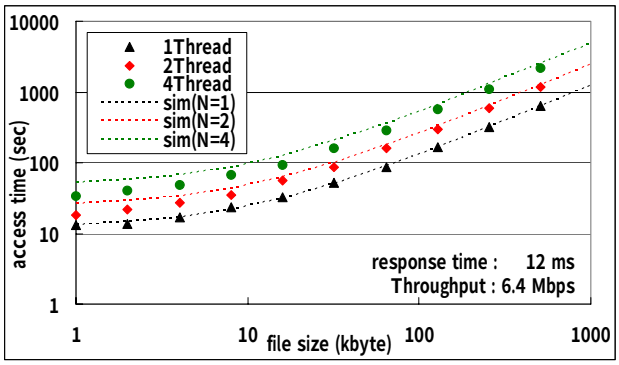

Fig.11. Experimental results of access time in N threads.

By calculating the evaluation index based on these data, we simply determine whether the number of threads is better for a single agent program or not in this system. The evaluation index shows the proportion of the execution time, so we must keep the setting of the targets at less than 1.0 (overwork). In this experiment, we set the interval time of all threads at 120 seconds, and the evaluation index in the case of 24 threads was 0.9. The situations of these threads were varied (**Table 5**), and the memory size was changed in this experiment (from 128 to 640 Mbyte) because the memory usage increases with the number of threads (**Fig.12**).

Table 5. Experimental situation of remote site and nodes.

Target place: 10 sites (0.5 ~10 Mbps, 10~200 ms)	
Data:	Field Server (Elab-exp.) * 15
	Field Server (National)* 2
	Other nodes (Web page)* 5
Camera:	3Mpixel digital camera (Ricoh) * 1
	0.4Mpixel pan-tilt-zoom camera (Panasonic)*3
	0.3Mpixel network camera (Corega)*8

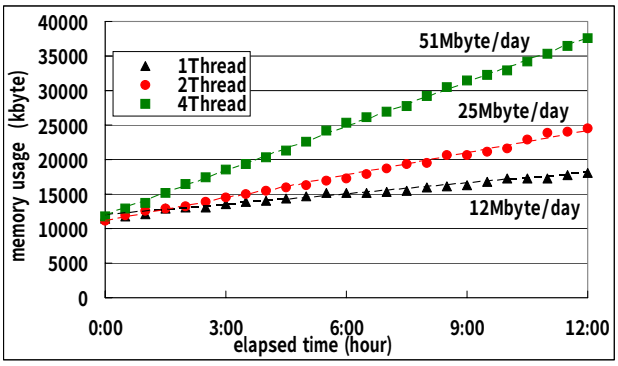

Fig.12. Experimental results of memory usage in N Threads.

Figure 13 shows the experimental results for the operational state in a single agent with 24 threads. In this result, though each execution time is longer than one in a single thread because of multi-tasking, we confirmed that the system works acceptably within the interval time.

4.2. Field Experiments

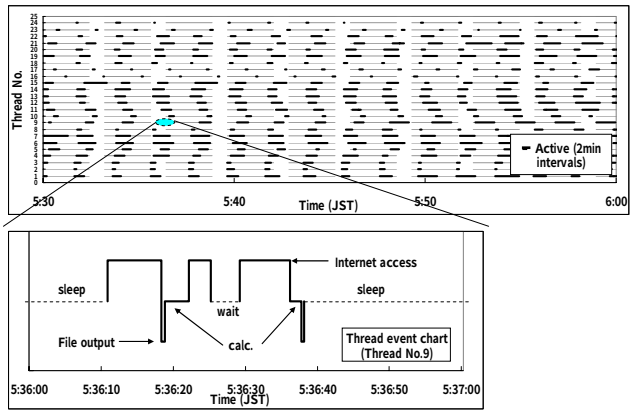

Fig.13. Experimental results in a single agent with 24 threads.

To evaluate the performance of the agent system for a Web-based sensor network, we installed many sensor nodes in various places and managed them with this system. Each task for the sensor nodes is described in the configuration file, which differs in the interval time, network environment, management, node type, and so on. In this experiment, the meta-agent system was constructed with a cluster of 7 computers in kind (section 4.1), and tasks were distributed based on the evaluation index. For this agent system to work effectively, it is also important to construct a network that is satisfactory at remote site. In this experiment, we developed the connection with the Virtual Private Network (VPN), which is a useful technology that can securely connect each remote site on the same network via the Internet [27] (**Fig.14**). In order to perform the VPN connection of each sites effectively, we improved the division algorithm of the meta-agent in which the meta-agent brings the same site task to the same agent based on the production rules and the

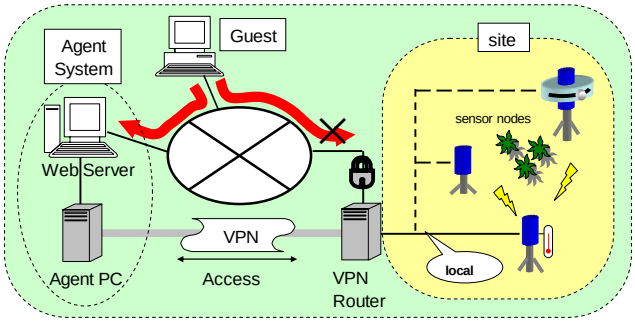

Fig.14. Network construction with VPN connection.

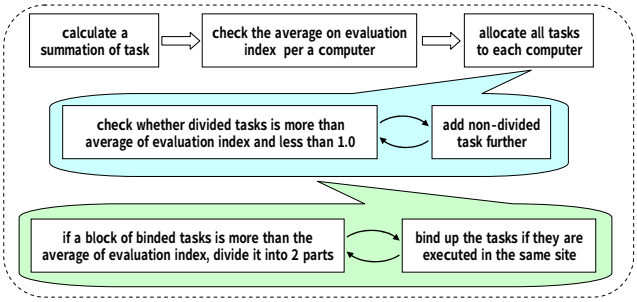

Fig.15. Division algorithm of the meta-agent.

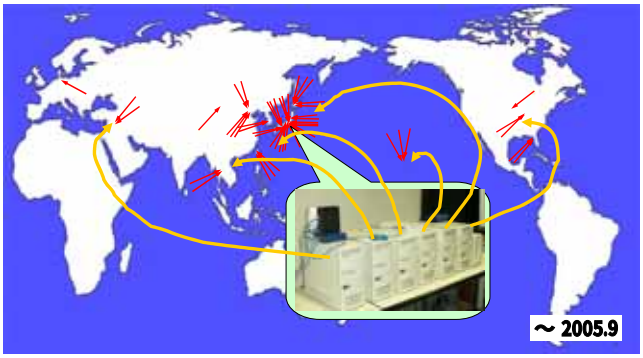

Fig.16. Site map of the agent system.

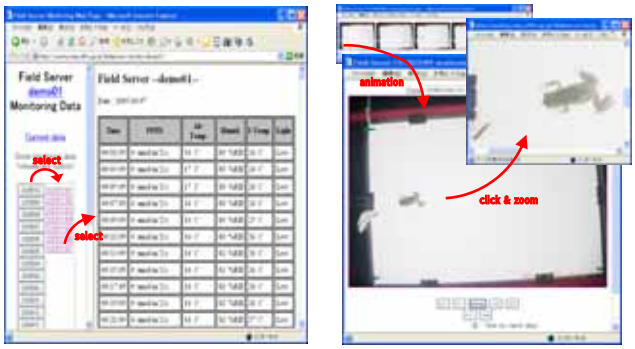

Fig.17. Web pages created by agent system.

evaluation index (**Fig.15**).

Approximately 50 sensor nodes (as of September 2005) are currently installed in various parts of the world, and they are in operation on this agent system (**Fig.16**). This system has performed reliably for more than 3 years since 2002. **Table 6** shows the condition of the nodes and sites in this target. In this system, large amounts of data are stored in our database. In a standard task, 60 kilobytes of monitoring data from one Field Server node is collected at 120-second intervals and 15 gigabytes of data is stored in a year. The total collected data in this system equals about 1 terabyte in a year.

Table 6. Condition of the experimental site.

Data size:	1kbyte (data) ~ 1Mbyte (image)
Interval time:	30sec ~ 1800sec
Country (Site):	Japan (15), China (4), U.S.A. (3), Thailand, Taiwan, Korea, Syria
Connection speed:	0.05 ~ 10Mbps
Connection method:	Direct, Port Forward, PPTP VPN, SSL VPN

In this system, the agent program stores the collected data and simultaneously creates Web pages in which a user can easily check data. The Web pages in which the stored data are displayed can be accessed with either a Web browser or a Personal Digital Assistant (PDA) or cellular phone that can use the Internet. As regards image data, the system also creates Web pages in which a user can easily search out useful image data from our vast database. This Web page is displayed with serial image data, which is thumbnailed to reduce capacity, while the original image data can be easily accessed by clicking thumbnailed image data that is linked to it. Therefore, a user can inspect a moving image by performing few network loads and view important images at high resolution (**Fig.17**).

5. Conclusion and Discussion

We proposed an agent system for operating Web-based sensor nodes collectively and uniformly via the Internet to realize a high scalability and extensibility system. We have contemplated the concept and architecture, implemented the system, and demonstrated its capabilities and its reliability.

In this paper, we don't discuss in depth the algorithm for controlling the sensor nodes or the optimization for multi-agent use including the evaluation index; however, the structure of the agent system makes it expandable, so that part can be considered and improved separately from operation of the main agent program. By improving the implementation of the program, we can manage more sensor units effectively with small resources. These are important factors to develop the agent system.

When developing these kinds of Web-based sensor nodes in the future, it is important to consider their management, as with an agent system. If the number of sensor nodes increased, the agent system itself need not consider the problem, but the management of tasks such as initialization, troubleshooting, and calibration must be studied to reduce the amount of effort that will be required of users in the future.

In a Web-based system, the network connection is not structurally guaranteed; problems of packet loss, communication delays, and limited speed response can always occur. In this agricultural usage, these problems are not so serious during field monitoring and greenhouse control as affairs now stand. But it is also necessary and important to solve these problems, as a fail safe and robust system for packet loss and a multi-session method for quick response will be needed in the future.

With the ongoing developments in IT, Web-based technology will continue gaining popularity. Nowadays, this technology is utilized via the control of the Internet connected home appliance, and in the future, it is estimated that robot units and other controlled units will be equipped with Web servers. The agent system we describe here can be applied to situations in which the agent system controls many actuators of robots embedded with Web servers via the Internet. By cooperation among people in various fields, the agent system can be further developed, and the proposed system will be more effective as one of the new methods for utilizing sensor networks.

References:

[1] T.Yamanaka, and I.Kaihotsu, "TDR Measurement of Soil Moisture in a Cold-Arid Region: Effects of Temperature Variation and Soil Freezing/Melting phenomena," Electronic

Monograph, No.1, pp.1-7, 2003 (in Japanese).
[2] J.M.Kahn, R.H.Katz, and K.S.J.Pister, "Next Century Challenges: Mobile Networking for “Smart Dust”," ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom99), pp.271-278, 1999.
[3] K.A.Delin, and S.P.Jackson, "Sensor Web for In Situ Exploration of Gaseous Biosignatures," In Proc. IEEE Aerospace Conference, Vol.7, pp.465-472, 2000.
[4] L.Doherty, B.A.Warneke, B.E.Boser, and K.S.J.Pister, "Energy and Performance Considerations for Smart Dust," International Journal of Parallel and Distributed Systems and Networks, Vol.4, No.3, 2001.
[5] V.Shnayder, M.Hempstead, B.Chen, G.W.Allen, and M.Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," Proc. of the Second ACM Conference on Embedded Networked System, 2004.
[6] S.Sahni, and X.Xu, "Algorithms for Wireless Sensor Networks," International Journal of Distributed Sensor Networks, Vol.1, No.1, pp.35-56, 2005.
[7] F.Ye, H.Luo, J.Cheng, S.Lu, and L.Zhang, "A Two-Tier Data Dissemination Model for Largescale Wireless Sensor Networks," MOBICOM02, pp.148-159, 2002.
[8] SD.Servetto, and G.Barrenechea, "Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks," WSNA02, pp.12-21, 2002.
[9] R.Govindan, W.Hong, S.Madden, M.Franklin, and S.Shenker, "The Sensor Network as a Database," Computer Science Department, TR02-02-771, 2002.
[10] S.Ratnasamy, D.Estrin, R.Govindan, B.Karp, S.Shenker, L.Yin, and F.Yu, "Data-Centric Storage in Sensornets," SIGCOMM02, 2002
[11] A.Savvides, CC.Han, and MB.Strivastava, "Dynamic Fine-grained Localization in Ad-hoc Networks of Sensors," MOBICOM01, pp.166-179, 2001.
[12] J.Elson, and K.Romer, "Wireless Sensor Networks: A New Regime for Time Synchronization," HotNets-I, 2002.
[13] S.Tilak, NB.Abu-Ghazaleh, and W.Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," WSNA02, pp.49-58, 2002.
[14] A.Mainwaring, J.Polastre, R.Szewczyk, D.Culler, and J.Anderson, "Wireless Sensor Networks for Habitat Monitoring," ACM Workshop on Sensor Networks and Applications, 2002.
[15] E.S.Biagioni, and K.W.Bridges, "The application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species," International Journal of High Performance Computing Applications, Vol.16, No.3, pp.315-324, 2002.
[16] G.W.Allen, J.Johnson, M.Ruiz, J.Lees, and M.Welsh, "Monitoring Volcanic Eruptions with a Wireless Sensor Network," Proc. of the European Workshop on Wireless Sensor Networks, 2005.
[17] R.Cardell-Oliver, K.Smettem, M.Kranz, and K.Mayer, "A Reactive Soil Moisture Sensor Network: Design and Field Evaluation," International Journal of Distributed Sensor Networks, Vol.1, No.2, pp.149-162, 2005.
[18] R.Sameshima, "A new method for predicting flowering stage in soybean," JARQ, 25, pp.149-153, 1991.
[19] T.Ohtani, K. Sugahara, K. Tanaka, M. Laurenson, T. Watanabe, and S. Umemoto, "Web based IPM system for Japanese pear diseases in Japan. III. Weather data acquisition system to estimate leaf wetness duration and scab infection severity " The 2001 KSPP International Conference and Annual Meeting. Plant Disease Forecast: Information Technology in Plant Pathology, Program and Abstracts pp.63, 2001.
[20] T.Hoshi, Y.Hayashi, S.Yokoi, T.Hasegawa, and T.Kozai, "An Environmental Control System Based on the Decentralized Autonomous Control,” Journal of Society of High Technology in Agriculture, Vol.14, No.3, pp.157-164,2002.
[21] T.Kida, M. Kamo, A. Sawamura, and D. Matsumoto, "Study on the Development of Dairy Facilities and a Cow Herd Monitoring System," Agricultural Information Research, 12(4), pp.299-306, 2003 (in Japanese).
[22] T.Fukatsu, and M.Hirafuji, "Field Monitoring Using Sensor-Nodes with a Web Server," Journal of Robotics and Mechatronics, Vol.17 No.2, 2005.
[23] S.Junnila, and J.Niittylahti, "Wireless Technologies for Data Acquisition Systems," Proc. of the 1st International Symposium on Information and Communication Technologies, pp.132-137, 2003.
[24] M.R.Laurenson, T.Kiura, and S.Ninomiya, "Providing Agricultural Models with Mediated Access to Heterogeneous Weather Databases," Applied Engineering in Agriculture, Vol.18, No.5, pp.617-625, 2002.
[25] K.Tanaka, T.Watanabe, and M.Hirafuji, "Model Design Using the Interface and Use of Models by the Modelbase System," Proc. of the 2nd Asian Conference for Information Technology in Agriculture, pp.267-272, 2000.
[26] M.R.Laurenson, T.Kiura, and S.Ninomiya, "A Tool for estimating the risk of extreme climatic events," Agricultural Information Research, Vol.10, No.2, pp.79-90, 2002.
[27] T.Kiura, T.Fukatsu, and M.Hirafuji, "Field Server Gateway: Gateway Box for Field Monitoring Servers," Proc. of the 3rd Asian Conference for Information Technology in Agriculture, pp.410-413, 2002.