# *Construction of Support Tools to Assist an Agent System*

*Tokihiro Fukatsu [1], Masayuki Hirafuji [1] and Takuji Kiura[1]*

1: National Agriculture and Food Research Organization, Japan,
fserver@zoushoku.narc.affrc.go.jp

***Abstract***

We previously developed an agent system with a function to access Web servers via HTTP and to operate flexibly according to users' requests with agent configuration files (Profiles) to autonomously manage Web-based sensor nodes such as Field Servers. In the Profiles, a combination of agent commands and production rules are described in order to respond to various situations. The more they are used in many situations, the more support tools for the agent system are required for easy management. In this paper, we constructed assist components, a standardized form for handling data, an application to control the agent system and a tool for making the Profiles, to assist in this through a Web interface for the agent system.
The standardized form was constructed by some default files and defined directory structure to treat collected data easily for users on the Web. The control application was designed using a Java Servlet, and displays a status list of agent operation, accepts new Profiles, and interacts with the agent system. With the control application installed, users can manage the agent system easily via the Web page. The making tool helps users describe the Profiles by selecting and combining templates (accessed via the Internet) from items such as node types, sensor parameters, and operation contents. Users can make Profiles to input suitable information from these templates. By coordination of the standardized form, the control application and the making tool, users can seamlessly send new requests to the agent system without efforts via the Web. It is expected that the system will become a new user interface that can be customized to handle various Web systems in the same way.

**Keywords**: Field Server, Agent System, Web Interface, Monitoring, Management tool

## *Introduction*

A Field Server (Fukatsu and Hirafuji, 2005) is a Web-based sensor node that performs various monitoring and other complicated operations. There currently exist many kinds of Field Servers which have different measurement modules equipped with a Web server (Table 1). In order to manage these different Field Servers, we developed an agent system (Fukatsu *et.al.* 2006a) which is able to control Web servers flexibly with configuration files (Profiles). Although this system meets users' requests when it has an adequate Profile, it requires a certain amount of effort and experience to describe a Profile. The agent system is provided free (if conditions permit) and a Field Server Agent Box (Fukatsu *et.al.* 2007) has been developed, so some users have begun to manage the system by themselves. The more extensively it is used in many applications, the more necessary support tools of the agent system become for easy management. It is also desirable to provide the collected data to users over the Web in an easily understandable format.

Table 1. List of Field Servers and their modules.

| Field Server | Main Computer | Network Camera |
|---|---|---|
| ・NARC version<br>with (a,b,c & 1,2,5,6,7)<br>・e-Lab. version<br>with (a,d,e & 1,2,3)<br>・Panasonic version<br>with (f & 2) | a. PICNIC<br>b. FS Engine/CS55<br>c. FS Engine/AD12588<br>d. e-Lab. Engine/H8<br>e. e-Lab. Engine/VAISALA<br>f. Panasonic Engine | 1. CG-NCMN(Corega)<br>2. BB-HCM580(Panasonic)<br>3. IK-WB21(Toshiba)<br>4. VB-C50i(Canon)<br>5. iCam33(UNIQUE)<br>6. RDC-i700(RICOH)<br>7. EOS-Kiss(Canon) |

In the present study, we developed a standardized form for displaying data on the Web, an interface used to observe the Field Servers and the agent system, and an assisting tool that can create Profiles easily. By using the agent system with these components, users are able to manage their Field Servers with little effort.

### *Standardized Form*

In the agent system, collected data is usually stored based on a definite hereinafter structure (Fig. 1) in the Web server's folder, which is available on the Internet. Each folder identified with the name of the Field Server (FS_Name) is created under the topmost folder; additionally, there is a hierarchical directory used to store raw data, which are identified in a year-month-day format. One day's data with the results of raw data analysis are saved as an XML file in the directory of the month and its path name is automatically registered on the List file. In order to obtain monitoring data easily on the Web, a sitemap of the data is created by the agent system in each FS folder. On the sitemap page, there is a drop down menu generated by JavaScript code. After selecting year and month through the menu, a filtered result linked to one day's data is shown by using the List file dynamically. The chosen day's data are displayed directly in tabular form on the Web with a default XSL file for the agent system. For image data, one day's data are displayed with a time-lapse function.

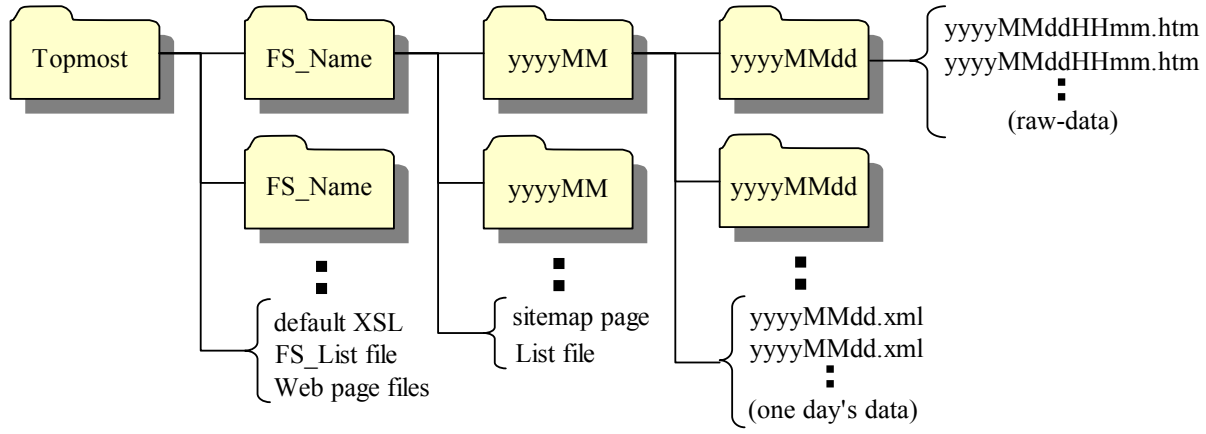


Fig. 1. Folder structure of the agent system.

In the topmost folder of the agent system, some default files are set to display all Field Server data. When the addresses of the target Profiles in the FS_List file are registered, the top page automatically displays each real-time piece of data and a link list to each sitemap file (Fig. 2). These default files also contain JavaScript code which seeks the address of a sitemap file via the relevant Profile from the FS_List file. When the agent system is executed with the FS_List address, the system automatically begins to collect data every time the address of new Profile is registered in it. Users can manage the agent system simply by inputting the default files and executing the system, describing Profiles with users' requests and registering the Profiles' addresses in the FS_List file.

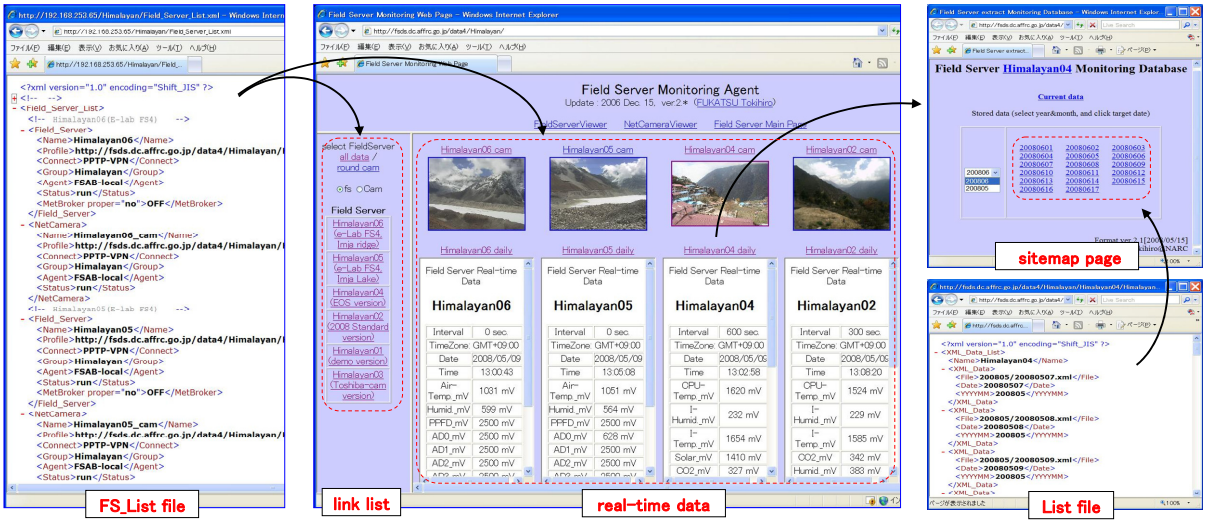


Fig. 2. Web pages of the agent system.

## *Agent Interface Program*

The agent interface program, which runs under Java Servlet, accesses the FS_List file and registered Profiles and assists with its information in managing the agent system on the Web. This program has functions, including a control panel, Profile upload and Profile editor (Fig. 3).

The control panel displays various parameters and the current situation of each Profile and allows users to change Profiles on the Web. The latest access time of collected data is displayed in the control panel in black (stable), yellow (beginning to fail) or red (constant data loss) according to the number of times missing data have been detected and periods of time

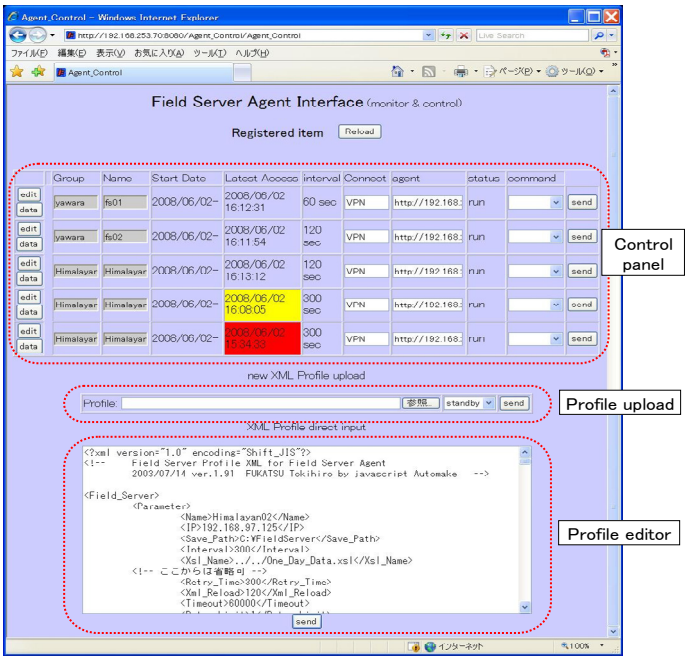


Fig. 3. Web page of the agent interface program.

during which the loss occurred. In the information on each Profile, there is a status report on the agent operating conditions, which users can easily change using drop down list which dynamically provides only valid transitions. The interface program registers the new status given on the Web in the FS_List file, and the agent system starts, stops and reboots the Profile operation depending on its status (Fig. 4).

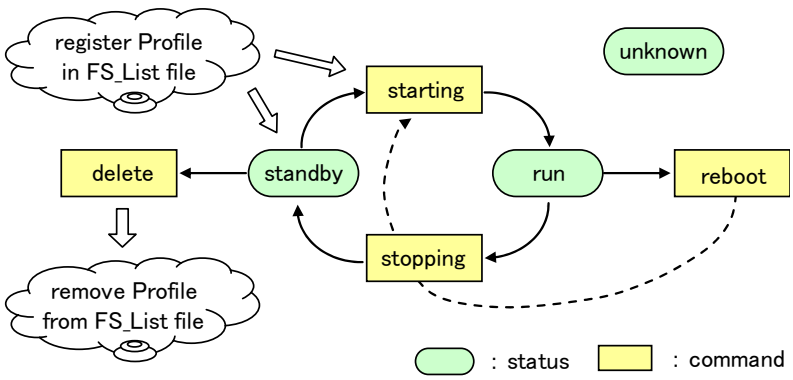


Fig. 4. Agent status and operating commands.

This interface program includes an uploading function that registers new Profiles in the FS_List file on the Web. In this function, a prepared Profile and its initial status are sent to the program by the POST method of HTTP. The program automatically saves it in the appropriate folder and registers its address and status in the FS_List file. The uploading function is also available to other agent systems, so it is possible to construct a distributed processing system by sharing all Profiles fairly among several agent systems (Fukatsu *et.al.* 2006b).

In addition to the uploading function, there is also a Profile editor that can be used to change existing Profiles or to edit new ones on the Web. The editor works in conjunction with the control panel, where there is an edit button for opening Profiles. Using this editor, it is easy for users to slightly change parameters such as interval time and conversion factor. Such changes are immediately reflected in the agent operation when reboot is selected on the control panel. It is also possible to describe new Profiles from scratch or based on existing Profiles. In this program, certain parameters of new Profiles, such as save address and URL, are automatically checked and corrected.

### *Assisting Tool for Making a Profile*

It is often difficult for users to describe a Profile that responds to their requests adequately because the description differs according to module type and operation content. To solve this problem, we developed an assisting tool for making a Profile without specialized experience. The assisting tool running under Java Servlet can create a Profile only by selecting certain elements and inputting certain parameters on the Web. First, users choose to create a new Profile or change an existing one which is automatically listed by this tool, and then edit it with respect to three factors:

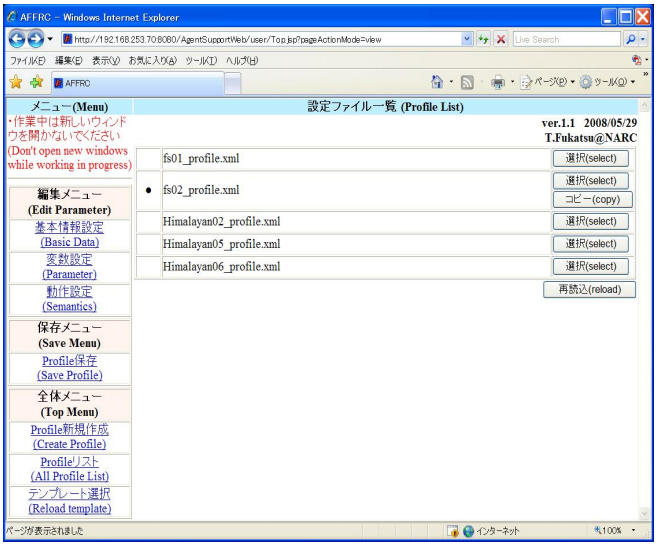


Fig. 5. Assisting tool for making a Profile.

basic parameters, variable definition and operation content (Fig. 5).

The edit window of the basic parameters helps with the entry of an initial definition and user information (Fig. 6). Required items are shown in color and optional items are shaded. Each item is already inputted with a default value and its titles are linked to a help window which gives detailed information. To make it easier to input the latitude and longitude of the deployed Field Servers, an Internet map application such as Google Map is linked.

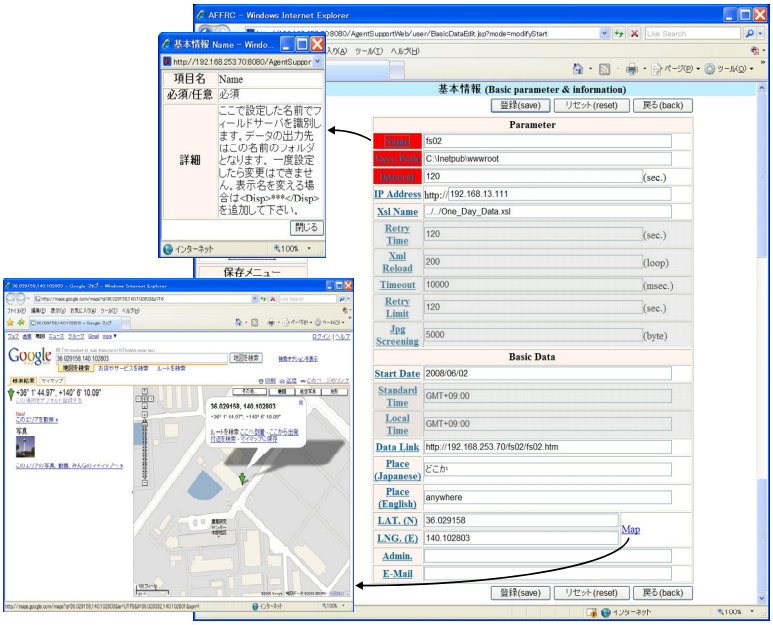


Fig. 6. Edit window of the basic parameter.

Variable definition usually depends on the Web modules embedded in the Field Servers. In this tool, the user selects some templates which individually form the definition of their modules (Fig. 7). Each template contains the variables necessary to extract monitoring data from an HTML page and to convert raw data into useful information based on each module. In order to create the desired definition, the user need only change a certain parameter such as the coefficient of sensors, and add or delete variable definitions as needed after selecting templates. These templates are also able to download via the Internet so that they can be updated for new types of modules and Field Servers.

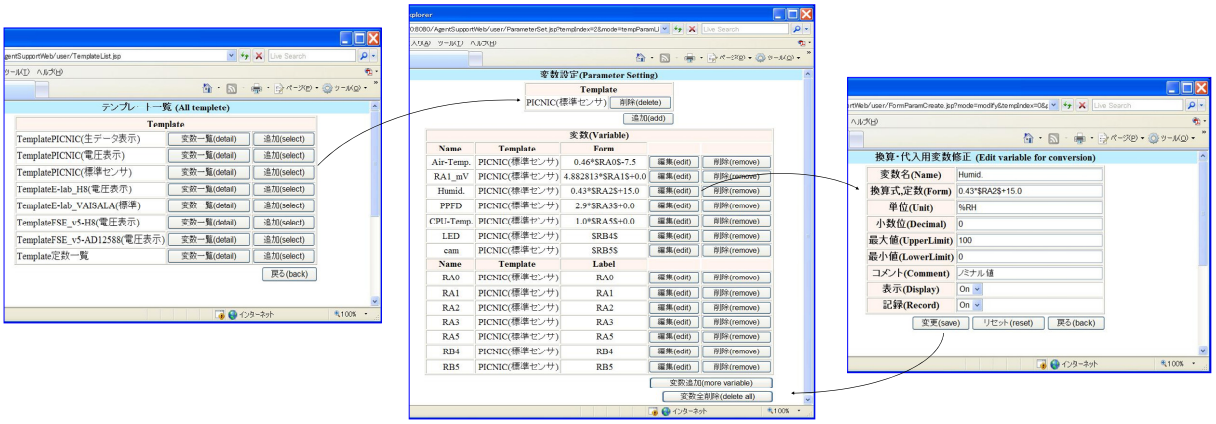


Fig. 7. Edit window of variable definition.

The agent system operates flexibly based on a combination of original commands and IF-THEN rules according to users' requests (Fukatsu *et.al.* 2006c). In addition to variable definition, operation contents are constructed by selecting templates which are described by a series of actions, such as collecting data periodically and analyzing image data (Fig. 8). Users can handle these templates as macro commands by following the easy steps described on the

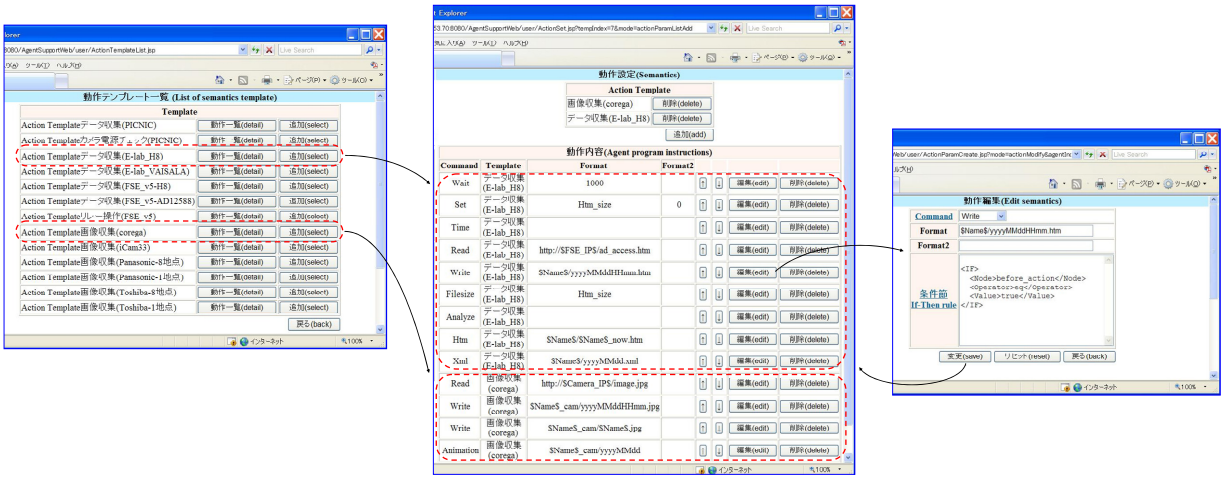


Fig. 8. Edit window of operation contents.

Web. In these templates, some required parameters, such as password and IP address, are defined in the variable definition, so there is less alteration of selected template needs for users in this tool.

The Profiles made by the tool can be sent directly to the agent interface. By using support components such as the assisting tool, the agent interface and the standardized form in a coordinated manner, users can manage the agent system simply and easily with only a few operations on the Web.

## *Future Work*

With respect to applying the support components to various situations, certain problems remain to be solved so that the Field Servers can be monitored with little effort. The Virtual Private Network (VPN) connection method is usually used to monitor several sites with installed Field Servers from a remote location. This method has to be set manually for each site, so it is difficult to use the support components in combination with it. Some Profiles are described to handle sensor data and image data in a single Profile in order to synchronize these modules. To display these data automatically using our components, users have to prepare a dummy Profile separately because the current system makes each Profile responsive to a single data display.

Templates in the assisting tool are currently updated only by administrators. By allowing users to update them, the system will reduce the administrators' workload and it will obtain a lot of templates. On the other hand, it will become more difficult for users to identify an appropriate template in a large pool of templates that includes some incomplete ones and some duplicated ones. An intelligent application able to manage templates and provide the best ones for users is required.

From the viewpoint of safety, the support components are convenient, but carry the risk of security holes and may be used illegally for fraud. Because it is an automatic and seamless system managed by users, it is vulnerable to careless treatment. We are working to develop an advanced agent system with a fail-safe design that ensures Profiles consistency and good communication for users.

***References***

Fukatsu, T. and M.Hirafuji (2005) Field monitoring using sensor-nodes with a Web server, Journal of Robotics and Mechatronics 17(2), 164-172.

Fukatsu, T., M.Hirafuji and T.Kiura (2006a) An agent system for operating Web-based sensor nodes via the Internet, Journal of Robotics and Mechatronics 18(2), 186-194.

Fukatsu, T., M.Hirafuji and T.Kiura (2006b) A distributed agent system for managing a Web-based sensor network with Field Servers, Proceeding of 4th World Congress on Computers in Agriculture, 223-228.

Fukatsu, T., M.Hirafuji and T.Kiura (2006c) Agent program for providing seamless management to Web-based monitoring system, Proceeding of AFITA2006, 581-589.

Fukatsu, T., T.Kiura, K.Tanaka and M.Hirafuji (2007) Hierarchical agent system for Web-based sensor network, Proceeding of SAINT2007, CD-ROM.