Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

Web-based Service Processor (WSP) for Embedded Systems

Tadashi Ohashi†

Abstract :
The conventional Service Processor (SVP) [1] [2] is attached to a host computer (like server)which is fabricated and tested in factories. The computer includes embedded system in storage devices such as HDD and the like. SVP takes roles of three major RAS (Reliability, Availability and Service) functions for host-computer OS support, maintenance of host computer and services of host computer operation. This conventional three functions are less capable of these services because special mini-computer is needed and enough skill is needed. To solve the problem, Web-based Service Processor (WSP) which is easily utilized by conventional PCs is proposed in this paper. As result, performance of services and cost are drastically improved.

1. Introduction

The object of the proposed method is to enhance versatility relating to Web-based Service Processor controlled by delivering and maintaining component information with less labor and lower cost. The proposed method comprises a client's storing component information on all hardware and embedded system constituting a product, connected to a drawing server, an EC (Engineering Control) sheet server, a program server, or a manual server. A Web-based Service Processor connected to the Internet as well as to the extranet, having a console function for at least the Web-based Service Processor , extracting component information through an SWP browser, and allowing the Web-based Service Processor to conduct a maintenance service, a host OS console service and the like. This paper relates to a Web-based Service Processor control system for extracting component information on all the hardware and embedded system constituting a product through the Internet and to a storage devices storing Web-based Service Processor control programs.

† Production Engineering Department of Common Technology Div., Fujitsu Limited

2. WEB-based Service Processor

2.1 Background of this research

Conventionally, at manufacturing process involved in the manufacture of products, various codes relating to embedded system and documents relating to embedded system and hardware are generated in a series of processes including a development process, a design process, a production process, an inspection process, a shipment process, a delivery and installation process and a user-side operation and maintenance process, and there take considerable labor and cost for delivery and maintenance. Thus, the conventional methods having these disadvantages are strongly desired to be improved. As described above, at manufacturers-side, various codes and documents are conventionally generated in a series of processes. The various codes relating to embedded system includes a source code, an object code, patch data, hardware state setting information on a host (or server, etc.) subjected to a maintenance service for the Service Processor and the like. The documents include manuals, specifications, design drawings, EC sheets (a new design notice, a design modification notice) and the like relating to embedded system, hardware and the like.

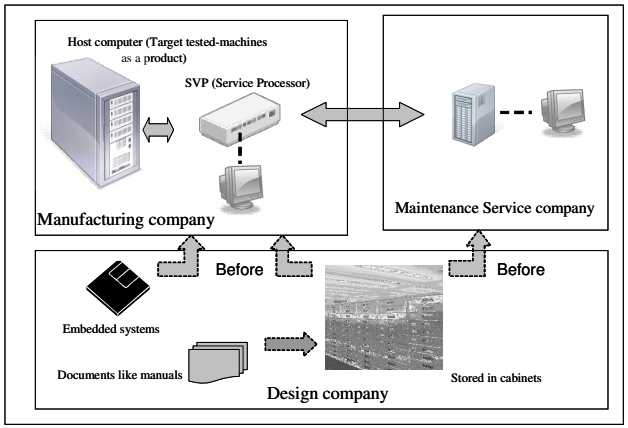

Fig. 1 Conventional document based process

It is assumed here that the documents shown in Fig. 1 include all paper mediums such as the above-described manuals, specifications, design drawings and EC sheets. As shown in Fig. 1, the documents in great volume are conventionally stored in a

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

cabinets or delivered, if necessary, to an inspection division, a manufacturing division, customers, a maintenance company and the like. Further, the parties which received the documents read the documents by visual observation. Likewise, the source codes, object codes relating to embedded system, hardware state setting information on a host (or a server, etc.) to which Service Processor maintenance service is conducted, and the like are distributed by hand to related divisions, a maintenance company and the like. Further, at the manufacture side, there are installed a host like a mainframe used in various processing and a console. Service Processor is a computer service system independent of the host. Service Processor controls the hardware of the host, provides conversational means for conversation with an operating system, monitors and diagnoses the operation state of the host and the like. The console is a specific terminal for issuing commands to Service Processor and is controlled by an operator. A remote maintenance terminal is installed at a maintenance company side and connected to the console (Service Processor) through a line. The remote maintenance terminal conducts a remote maintenance through the line. Meanwhile, it has been conventionally necessary to deliver source codes and object codes of the embedded system developed by the design company and the documents (see Fig.1) (which will be referred to as ”component information” hereinafter) one by one. Due to this, it disadvantageously takes considerable labor and cost for delivery and maintenance. In case of the documents, in particular, it disadvantageously takes considerable labor takes cost for the preparation and creation of materials, the creation of block copies, maintenance, delivery and the like. Moreover, the console has been conventionally connected to Service Processor (host) as a part of the system. For that reason, only the console can execute control over Service Processor, which makes the system disadvantageously less versatile.

2.2 Summary of the proposed method

It is, therefore, an object of the proposed method to provide a Web-based Service Processor control system and a storage devices installing a Web-based Service Processor control program capable of delivering and maintaining component information with less labor and lower cost and capable of realizing enhanced versatility relating to Web-based Service Processor control. To obtain the above object, the proposed method according to a first aspect comprises, a drawing server, an EC sheet server , a program server and a manual server in one embodiment to be described later) storing component information on all of hardware and embedded system constituting a product, control information for controlling at least a hardware state of a host and setting information for setting the hardware state, and connected to the Internet; and a client (corresponding to a client and a client in one embodiment to be described later) connected to a Web-based Service Processor connected to the host and the Internet, having at least a maintenance service function, as a console function for the Web-based Service Processor, based on the control information and the setting information, and drawing the component information, the control information and the setting information through a browser. According to this The proposed method, the client having the console function for the Web-based Service Processor and connected to the Internet is provided, and component information, control information and setting information are extracted through the browser. Thus, compared with a conventional case where component information is delivered and maintained by paper mediums and the like, delivery and maintenance can be conducted with less labor and lower cost and latest component information can be easily obtained due to the interactive characteristics of the Internet. Further, according to this The proposed method, it is not necessary to store conventionally-used paper mediums in great volume in a warehouse and it is, therefore, unnecessary to dispose of bad resources of paper mediums. The proposed method according to a second aspect is relied on the Web-based Service Processor control system according to the first aspect, wherein the client executes control relating to the Web-based Service Processor through the browser, thereby setting and controlling the hardware state of the host based on the control information and the setting information. According to this The proposed method, the client provided with the browser instead of the conventional console sets and controls the hardware state of the host. This can enhance versatility with respect to Web-based Service Processor control. The proposed method according to a third aspect is based on the Web-based Service Processor control system according to the first or the second aspect, wherein the component information, the control information and the setting information are described in XML[3]and the browser is made to correspond to the XML. According to this proposed method, component information is described in XML. Due to the characteristics of XML, therefore, various types of component information can be easily structuralized. The proposed method according to a fourth aspect is based on the Web-based Service Processor control system according to any one of the first to the third aspects, wherein another client connected to the Internet is also provided with the browser. According to this proposed method, by providing another client connected to the Internet with a browser, control relating to the Web-based Service Processor can be executed by

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

another client through the Internet. The proposed method according to a fifth aspect is based on the Web-based Service Processor control system according to any one of the first to the fourth aspects wherein the client executes control over information on the Web-based Service Processor using the XML including a tag for defining a type of information on the hardware of the host by xmlschema[4] . According to the proposed method, Web-based Service Processor information control is executed using the XML defining an information type (e.g., binary, hexadecimal, decimal) by xmlschema. Thus, the use of XML can be extended to an information processing. The proposed method according to a sixth aspect is based on the Web-based Service Processor control system according to any one of the first to the fifth aspects wherein the client displays the hardware state by a predetermined type of information by using the xmlschema. According to this proposed method, the hardware state of the host is displayed by a predetermined type of information by using xmlschema. Thus, the use of XML can be extended to display control relating to the Web-based Service Processor. The proposed method according to a seventh aspect is based on the Web-based Service Processor control system according to the sixth aspect wherein the client writes the setting information of a predetermined type into the hardware of the host by using the xmlschema. According to this The proposed method, the setting information of a predetermined type is written into the hardware of the host by using xmlschema. Thus, the use of XML can be extended to control for writing information into the hardware. The proposed method according to an eighth aspect is based on the Web-based Service Processor control system according to the sixth aspect wherein the client displays a message from the host by scrolling up or down the message by using the xmlschema. According to this The proposed method, the message from the host is displayed while being scrolled up or down by using xmlschema. Thus, the use of XML can be extended to host message display control. The proposed method according to a ninth aspect is based on the Web-based Service Processor control system according to the first to the eighth aspects wherein the Web-based Service Processor control system comprises a program server connected to the Internet, storing a program, a loading module for loading the program and control information for controlling execution of the program; and the client extracts the program, the loading module and the control information by way of the browser through the Internet and then executes the program. According to this proposed method, a program, a loading module and control information are extracted from the program server by way of the browser corresponding to XML through the Internet. Thus, compared with a conventional case, the execution of the program can be conducted at low cost, instantly and easily. The proposed method according to a tenth aspect is a storage devices recording a Web-based Service Processor control program, connected to a Web-based Service Processor and adapted to a client connected as a console for at least the Web-based Service Processor , the Web-based Service Processor connected to a component information server storing component information on all hardware and embedded system constituting a product, control information for controlling at least a hardware state of a host and setting information for setting the hardware state through the Internet and connected to the host, wherein storage device allows a computer to execute an extraction step of extracting the component information, the control information and the setting information through a browser. According to this proposed method, component information, control information and setting information are extracted through the browser. Thus, compared with a conventional case where component information is delivered and maintained by paper mediums and the like, delivery and maintenance can be conducted with less labor and lower cost and latest component information can be easily obtained due to the interactive characteristics of the Internet. Further, according to this proposed method, it is not necessary to store conventionally-used paper mediums in great volume in a warehouse and it is, therefore, unnecessary to dispose of bad resources of paper mediums. The proposed method according to an eleventh aspect is based on storage device recording a Web-based Service Processor control program according to the tenth aspect, wherein storage device comprises a control execution step of setting and controlling the hardware state of the host based on the control information and the setting information by executing control relating to the Web-based Service Processor through the browser. According to this The proposed method, control relating to the Web-based Service Processor is conducted through the browser instead of the conventional console. Thus, versatility relating to Web-based Service Processor control can be enhanced. The proposed method according to a twelfth aspect based on storage device recording a Web-based Service Processor control program according to the tenth or the eleventh aspect comprises an information control execution step of executing control over information on the Web-based Service Processor by using an XML including a tag defining a type of hardware information on the host by xmlschema. According to this proposed method, Web-based Service Processor information control is executed using the XML defining an information type (e.g., binary, hexadecimal, and decimal) by xmlschema. Thus, the use of XML can be extended to an information processing. Other objects

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

and features of this proposed method will become understood from the following description with reference to the accompanying drawings.

2.3 What this paper intends to solve

(1) The Web-based Service Processor control system comprising: a component information storage server storing component information on all of hardware and embedded system constituting a product, control information for controlling at least a hardware state of a host and setting information for setting the hardware state, and connected to the Internet; and a client connected to a Web-based Service Processor connected to host and Internet, having at least a maintenance service function, as a console function for Web-based Service Processor, based on control information and setting information, and drawing component information, said control information and said setting information through a browser.

(2)The Web-based Service Processor control system according to claim, wherein client executes control relating to Web-based Service Processor through browser, thereby setting and controlling the hardware state of said host based on said control information and said setting information.

(3)The Web-based Service Processor control system, wherein component information, control information and setting information are described in XML and browser is made to correspond to XML.

(4)The Web-based Service Processor control system, wherein another client connected to Internet is provided with browser.

(5)The Web-based Service Processor control system, wherein client executes control over information on Web-based Service Processor using the XML including a tag for defining a type of information on the hardware of host by xmlschema.

(6)The Web-based Service Processor control system, wherein client displays hardware state by a predetermined type of information by using said xmlschema.

(7)The Web-based Service Processor control system, wherein client writes setting information of a predetermined type into hardware of said host by using xmlschema.

(8)The Web-based Service Processor control system, wherein client displays a message from host by scrolling up or down the message by using schema.

(9)The Web-based Service Processor control system to claim, wherein the Web-based Service Processor control system comprises a program server connected to Internet, storing a program, a loading module for loading program and control information for controlling execution of program; and client extracts program, loading module and said control information by way of browser through the Internet and then executes said program.

(10)The storage device recording a Web-based Service Processor control program, connected to a Web-based Service Processor and adapted to a client connected as a console for at least Web-based Service Processor, Web-based Service Processor connected to a component information server storing component information on all hardware and embedded system constituting a product, control information for controlling at least a hardware state of a host and setting information for setting the hardware state through the Internet and connected to host, wherein storage device allows a computer to execute: an extraction step of extracting said component information, control information and setting information through a browser.

(11)Storage device of Web-based Service Processor control program, comprising an information control execution step of executing control over information on Web-based Service Processor by using an XML including a tag defining a type of hardware information on host by xmlschema .

(12)Storage device recording a Web-based Service Processor control program, wherein storage devices comprises a control execution step of setting and controlling the hardware state of host based on said control information and setting information by executing control relating to Web-based Service Processor through browser.

(13)Storage device recording medium recording a Web-based Service Processor control program, comprising an information control execution step of executing control over information on Web-based Service Processor by using an XML including a tag defining a type of hardware information on host by xmlschema .

3. Detail description of the proposed WSP

Description will be given hereinafter to one embodiment of a Web-based Service Processor control system and a storage device recording a Web-based Service Processor control program according to the proposed method with reference to the accompanying drawings. Fig. 2 is a schematic diagram showing the constitution of one embodiment according to the proposed method. In Fig.2, a client is a terminal for registering electronic information on drawings such as circuit diagrams and structural drawings generated in a series of processes including a development process, a design process, a manufacturing process, an inspection process and a shipment process, information on various EC sheets (a new design notice, a design modification notice and the like), program information and electronic information on various manuals. The client is connected to the Internet. In this client, a browser for perusing information described in XML (Extensive Markup Language) is started. Electronic information on diagrams, various EC sheets and various manuals, and program information as stated above will be generally referred to as “component information” hereinafter. A diagram server is connected to the Internet and is provided with a storage device. This storage device stores therein electronic

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

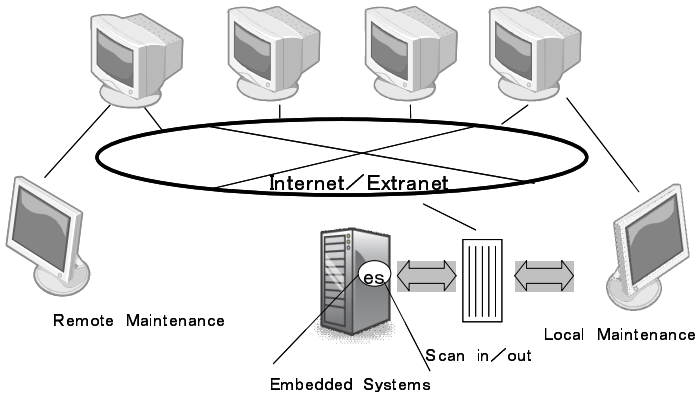

Fig.2 Schematic Diagram of proposed WSP

information, described in XML, on drawings such as circuit diagrams and structural drawings. An EC sheet server is connected to the Internet and is provided with a storage device. This storage device stores therein electronic information, described in XML, on various EC sheets. A program server is connected to the Internet and is provided with a storage device. This storage device stores therein an inspection program and the like. A manual server is connected to the Internet and is provided with a storage device. This storage device stores therein electronic information, described in XML, on various manuals. A Web-based Service Processor is connected to the Internet and a client. The Web-based Service Processor is a computer subsystem independent of a host (or server, etc.). The Web-based Service Processor controls hardware, provides conversational means for conversation with an operating system, monitors and diagnoses the operation state of the host. This Web-based Service Processor is connected to the host through a bus for scan-in and scan-out to the host. The client is connected to the Internet and the Web-based Service Processor. In this client, an SWP (Web-based Service Processor) browser is started. This client corresponds to the conventional console (see Fig.1) and conducts a maintenance service and the like to the host. Further, the client is allowed to refer to or register information on drawings, various sheets and the like through the Internet by using an SWP browser. Further, an IP address, an SMTP address and an FTP address are added to the client (SWP browser). The client is, therefore, capable of establishing communications through the Internet. Next, the operation of one embodiment stated above will be described with reference to the flow charts of Fig. 3 to7. In a state shown in Fig.3, in which the browser and the SWP browser are started, respectively, the displays of the clients and display menus for selecting events (registration, reference, reception, and loading browser. Here, if “registration” is selected while the browser of the client is started, the client outputs the determination result of a step SSP1, “Yes”. In a step SSP8, the client registers electronic information on drawings such as circuit diagrams and structural drawings, information on various EC sheets (a new design notice, a design modification notice), program information or electronic information on various manuals in the drawing server, the EC sheet server, the program server or the manual server through the Internet. In case of registering hardware information, “registration” and “hardware” shown in Fig.3 are selected. By doing so, hardware information on drawings, various EC sheets and various manuals are registered. In case of registering embedded system information, “registration” and “embedded sys” shown in Fig.3 are selected. By doing so, embedded sys information on various EC sheets, programs and various manuals are registered. Further, if “reference” shown in Fig. 3 is selected while the SWP browser of the client is started, the client outputs the determination result of a step SSP2, “Yes”. In a step SSP9, the client extracts electronic information on drawings such as circuit diagrams and structural drawings, information on various EC sheets (a new design notice, a design modification notice), program information and electronic information on various manuals from the drawing server , the EC sheet server, the program server and the manual server. These pieces of information are then displayed by the SWP browser, thereby executing a reference processing. In case of referring to hardware information, “reference” and “hardware” shown in Fig.3 are selected. By doing so, hardware information on drawings, various EC sheets and various manuals are referred to. In case of referring to embedded system information, “reference” and “embedded sys” shown in Fig. 3 are selected. By doing so, embedded system information on various EC sheets, programs and various manuals are referred to. If “reception” shown in Fig.3 is selected while the SWP browser of the client is started, the client outputs the determination result of a step SSP3, “Yes”. In a step SSP10, the client extracts electronic information on drawings such as circuit diagrams and structural drawings, information on various EC sheets (a new design notice, a design modification notice and the like), program information and electronic information on various manuals from the drawing server, the EC sheet server, the program server and the manual server through the SVP browser, receives these pieces of information and then stores the information in a storage device (not shown). In case of receiving hardware information, “reception” and “hardware” shown in Fig.3 are selected. By doing so, hardware information on drawings, various EC sheets and various manuals are received.

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

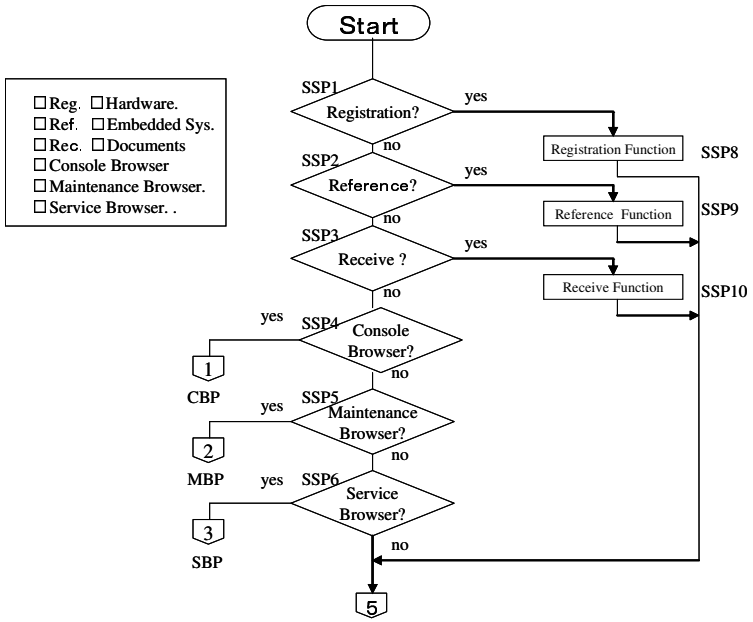

Fig.3 Selection service process

In case of receiving embedded system information, “reception” and “embedded sys” shown in Fig. 3 are selected. By doing so, embedded system information on various EC sheets, programs and various manuals are received. Further, if “console browser” is selected, the client outputs the determination result of a step SSP4 shown in Fig.3, “Yes”, and executes a console browser processing shown in Fig. 4. In this console browser processing, the client (SWP browser) functions as a console for the Web-based Service Processor . In a step CBP1, the client receives an OS message from the OS (operating system) of the host through the Web-based Service Processor . OS messages include “HOST OS MESSAGE”, “RUN01 JOB START”, “RUN02 JOB RESTART” and the like. In a step CBP2, the client embeds the received OS message into an XML tag. If “HOST OS MESSAGE”, for example, is received, the message is embedded like <PD>HOST OS MESSAGE</PD>. <PD> is a tag which means scrolling down the OS message from up to down direction (D direction) on one-line basis. If scrolling up an OS message from down to up direction (U direction) on one-line basis,

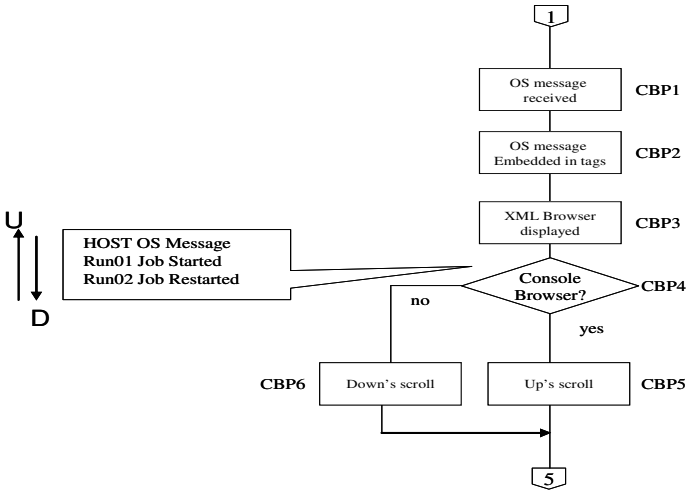

Fig.4 Console browser process

a tag <PU> is used. In a step CBP3, the OS message is displayed on the SWP browser. In a step CBP4, the tag is checked to thereby judge whether or not scroll is scroll-up. In that case, a determination result is set “No” to provide <PD> (scroll-down) tag. In a step CBP6, the OS message is scrolled down in D direction on the SWP browser. If the determination result of the step CBP4 is “Yes”, the OS message is scrolled up in U direction in a step CBP5. Further, if “maintenance browser” is selected, the client outputs the determination result of the step SSP5 shown in Fig.3, “Yes”, and executes a maintenance browser processing shown in Fig.5. In this maintenance browser, the client (SWP browser) functions to maintain the Web-based Service Processor . In that case, a select picture plane for selecting a maintenance item (“MPU START”, “MPU STOP”, “MEMORY DUMP” or the like) is displayed on the SWP browser. In a step MBP1 shown in Fig.5, it is judged whether or not DUMP (“MEMORY DUMP” stated above) has been selected. If this determination result is “Yes”, the scan-out of a hardware register in the Web-based Service Processor is executed in a step MBP2 and binary data is acquired. In a step MBP3, the client embeds the acquired binary data into an XML tag. In this case, the tag means that the data is binary data. In a step MBP4, it is judged whether or not scan-out has been completed. If the determination result is “No”, processing after the step MBP2 are repeated. If the determination result of a step MBP4 is “Yes”, the client describes scan or log information in XML in a step

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

MBP5. In a step MBP6, the client embeds the information embedded into the tag in the step MBP3 and the XML information in the step MBP5 into an applet. In a step MBP7, the client reads xmlschema for maintenance display from a storage device (not shown.

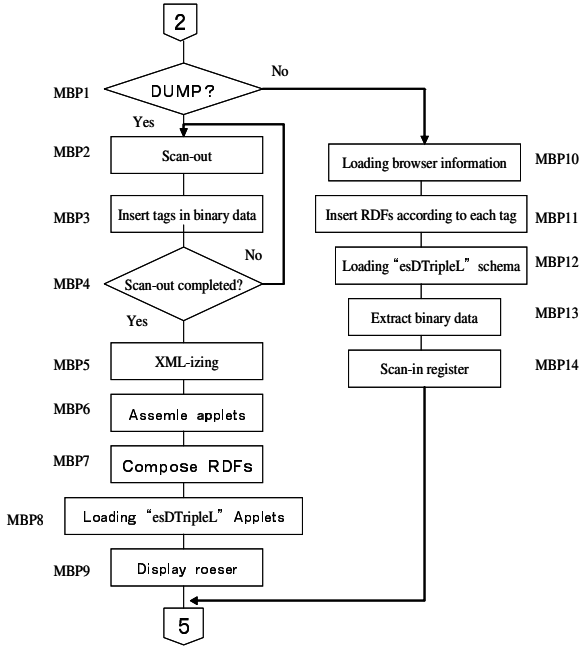

Fig.5 Maintenance browser process

This xmlschema defines a tag described in XML. The xmlschema is stored in the storage device in advance. In a step MBP8, the client reads esDTripleL (Embedded System Data Style Semantics and Specification Language) from the storage device (not shown). The schema defines the document style correspond to XML. In a step MBP9, the scanned-out binary data, log information (XML information) and the like are displayed on the SWP browser according to the esDtripleSL. On the other hand, if the determination result of the step MBP1 is "No", scan-in to a register in the Web-based Service Processor is executed in steps MBP10 to MBP14. Namely, in the step MBP10, to-be-scanned-in browser input information (e.g., binary data) described through SWP browser in XML is read (inputted) into the register in the Web-based Service Processor . Accordingly, the above-stated information is embedded into the XML tag. The browser information include hexadecimal description data, decimal data and the like besides binary data (binary description data). In case of the binary data, the XML tag is <B>binary description</B>. In case of the hexadecimal data, the XML tag is <HEX>/hexadecimal data</HEX>. In case of decimal data, the XML tag is <DEC>/decimal description</DEC>. In the step MBP11, the client reads the xmlschema according to each tag of the browser input information from the storage device (not shown). This xmlschema defines the tag of browser input information. "TAG BUS=X' 44BF FFFF' (HEX)" is an example of this xmlschema. This means that "TAG BUS" is associated with "44BF FFFF" in hexadecimal description. In the step MBP12, the client reads esDtripleSL corresponding to the browser input information from the storage device (not shown). In the step MBP13, binary bit information is extracted from the xmlschema. To be specific, "0100(4) 0100(4) 1011(B) 1111(F) 1111(F) 1111(F) 1111(F) 1111(F)" (binary bit information) is extracted from "44BF FFFF" (hexadecimal). In the step MBP14, the client scans in the binary bit information (binary data) to a target register (e.g., TAG BUS) of the Web-based Service Processor. Further, if "service browser" is selected, the client outputs the judgment result of the step SSP6 shown in Fig. 2, "Yes", and executes a service browser processing shown in Fig.6.

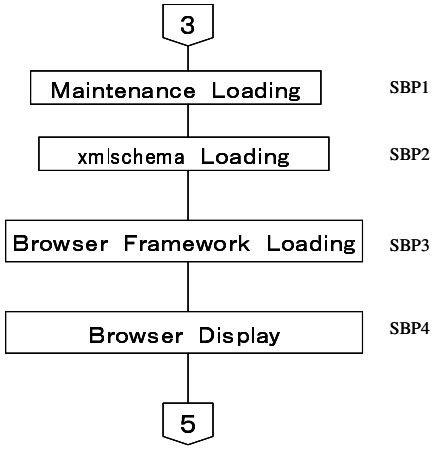

Fig. 6 Service browser process

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

In this service browser processing, the client (SWP browser) functions as a console for the Web-based Service Processor and displays update information and patch information on the Web-based Service Processor as well as information for maintaining the Web-based Service Processor and the like (all of which information are described in XML). These pieces of service information are stored in the storage device (not shown). In a step SBP1 shown in Fig.6, the client loads display service information (described in XML) from the storage device. In a step SBP2, the client loads xmlschema relating to the service information from the storage device. This xmlschema defines a tag for the service information (in XML) and is stored in the storage device in advance. In a step SBP3, the client reads esTripleSL for service information display from the storage device (not shown). This esTripleSL defines a message style corresponding to the service information. In a step SBP4, the service information loaded in the step SBP1 is displayed on the SWP browser according to esDtripleSL.

4. Conclusions of proposed method

As described above, according to one embodiment of the present The proposed method, the client having the console function for the Web-based Service Processor and connected to the Internet is provided, and component information (drawing information, various EC sheet information and the like), control information for controlling at least the hardware state of the host and setting information for setting the hardware state are extracted through the SWP browser. Thus, compared with a conventional case where component information is delivered and maintained by paper mediums and the like, delivery and maintenance can be conducted with less labor and lower cost and latest component information can be easily obtained due to the interactive characteristics of the Internet. Further, according to one embodiment of the present The proposed method, it is not necessary to store conventionally-used paper mediums in great volume in a warehouse and it is not, therefore, necessary to dispose of bad resources of paper mediums. Furthermore, according to one embodiment of the present this proposed method, the client provided with the SWP browser instead of the conventional console controls the Web-based Service Processor. This can enhance versatility with respect to control over the Web-based Service Processor. Moreover, according to one embodiment of the present The proposed method, component information is described in XML. Due to the characteristics of XML, therefore, various types of component information can be easily structuralized. Additionally, according to one embodiment of the present this proposed method, by providing another client (not shown) connected to the Internet with a browser having the same function as that of the SWP browser, control relating to the Web-based Service Processor can be executed by another client through the Internet. Furthermore, according to one embodiment of the present The proposed method, information on the Web-based Service Processor is controlled using the XML defining a data type (e.g., binary, hexadecimal, decimal) by a tag. Thus, the use of XML can be extended to a data processing. Further, according to one embodiment of the present The proposed method, a program, a loading module and control information are extracted from the program server by way of the SWP browser corresponding to XML through the Internet. Thus, compared with the conventional case, the execution of the program can be conducted at low cost, instantly and easily. One embodiment of the present The proposed method has been described so far with reference to the accompanying drawings. It should be noted that concrete examples of the constitution of the proposed method are not limited to one embodiment. Even if there are design changes and the like within the range in which the scope and spirit of the present this proposed method are not deviated, these changes are included in the present this proposed method. For example, in one embodiment described above, the Web-based Service Processor control program for realizing the function of the client (WSP browser) may be recorded on a storage device including the Web-based Service Processor control programs medium may be read and executed by a computer. As described so far, according to the proposed method of the first aspect, the client having the console function for the Web-based Service Processor and connected to the Internet is provided, and component information, control information and setting information are extracted through the browser. Thus, compared with a conventional case where component information is delivered and maintained by paper mediums and the like, delivery and maintenance can be advantageously conducted with less labor and lower cost and latest component information can be advantageously, easily obtained due to the interactive characteristics of the Internet. Further, according to the proposed method of the first aspect, it is not necessary to store conventionally-used paper mediums in great volume in a warehouse and it is, therefore, advantageously unnecessary to dispose of bad resources of paper mediums. Furthermore, according to the proposed method of the second aspect, the client provided with the browser instead of the conventional console sets and controls the hardware state of the host. This can advantageously enhance versatility with respect to Web-based Service Processor control. Moreover, according to the proposed method of the third aspect, component information is described in XML. Due to the characteristics of XML, therefore, various types of component information can be advantageously, easily structuralized. Additionally, according to the proposed method of the fourth aspect, by providing another client connected to the Internet with a browser, control relating to the Web-based Service Processor can be advantageously executed by another client through the Internet. Furthermore, according to the proposed method of the fifth aspect, Web-based Service Processor information control is executed using the XML defining an information type (e.g., binary, hexadecimal, decimal) by xmlschema. Thus, the use of XML can be advantageously extended to an information processing. According to the proposed method of the sixth aspect, the hardware state of the host is displayed by a predetermined type of information by using xmlschema. Thus, the use of XML can be advantageously extended to display control relating to the Web-based Service Processor. According to the proposed method of the seventh aspect, the setting information of a predetermined type is written into the hardware of the host by using xmlschema and esDtripleSL. Thus, the use of XML can be advantageously extended to control for writing information into the hardware. According to the proposed method of the

Vol.2010-ARC-187 No.17
Vol.2010-EMB-15 No.17
2010/1/29

eighth aspect, the message from the host is displayed while being scrolled up or down by using xmlschema and esDtripleSL. Thus, the use of XML can be advantageously extended to host message display control. Further, according to the proposed method of the ninth aspect, a program, a loading module and control information are extracted from the program server by way of the browser corresponding to XML through the Internet. Thus, compared with a conventional case, the execution of the program can be conducted at low cost, instantly and easily. According to the proposed method of the tenth aspect, component information, control information and setting information are extracted through the browser. Thus, compared with a conventional case where component information is delivered and maintained by paper mediums and the like, delivery and maintenance can be advantageously conducted with less labor and lower cost and latest component information can be advantageously, easily obtained due to the interactive characteristics of the Internet. Further, according to the proposed method of the ninth aspect, it is not necessary to store conventionally-used paper mediums in great volume in a warehouse and it is, therefore, advantageously unnecessary to dispose of bad resources of paper mediums. Further, according to the proposed method of the eleventh aspect, control relating to the Web-based Service Processor is conducted through the browser instead of the conventional console. Thus, versatility relating to Web-based Service Processor control can be advantageously enhanced. In addition, according to the proposed method of the twelfth aspect, Web-based Service Processor information control is executed using the XML defining an information type (e.g., binary, hexadecimal, decimal) by xmlschema. Thus, the use of XML can be advantageously extended to an information processing. Although the proposed method has been described with respect to a specific embodiment for a complete and clear disclosure, the appended claims are not to be thus limited but are to be construed as embodying all modifications and alternative constructions that may occur to one skilled in the art which fairly fall within the basic teaching herein set forth.

References

[1] T.Ohashi "*Consideration of Embedded system Management Control System on a web*" (in Japanese). Spring Convention No.60 organized by Information Processing Society of Japan (IPSJ), Vol.3 pp.417-418, 2000.

[2] T.Ohashi "*Web Distribution Management System (WDMS) for Embedded System's Modules and its Documents (ESM&D)*" (in Japanese). IPSJ(Information Processing Society of Japan) SIG Technical Report, 2008-EMB-10 (6) pp.37-44 Nov.27,2008.

[3] http://www.w3.org/XML/

[4] http://www.w3.org/TR/xmlschema-2/