2008－EMB－10　(6)
2008／11／27

Web Distribution Management System (WDMS) for Embedded System's Modules and its Documents (ESM&D)

Tadashi Ohashi [†]

Abstract : Embedded system's modules implemented in hardware system are distributed by WDMS (Web Distribution Management System) to users such as factories or vendors. This distribution is controlled by standards or procedures as formal knowledge [1], which is produced by experiences of engineers or vendors as well as manufactures. These standards or procedures are automatically arranged and processed by WDMS whose method is proposed in order to adjust the needs of a recipient of the distributed ESM&D (Embedded System's Modules and Documents). Standards or procedures are controlled by the decomposition of Schema (RDFs[2],) as registration, reference, or receipt process. As a result of this proposed method, This WDMS brings forth versatility and rapid accommodation to the recipients' needs.

1. Introduction

A current technology to adjust recipient needs, An engineers must intervene between the recipients and PDMs [3][4]. When a designer registers the designed embedded system's modules and documents , one must set destination like company name, department name, section etc. a document and data have a security level to let document and data be distributed depending on security level as recipients' needs. Under these circumstances, the proposed WDMS is preferable to be used for a managing control of embedded system's modules, a design drawing and specification form of a hardware or the like (hereinafter, collectively means a component). WDMS is able to accommodate users' needs as the component managing control system, which are capable of registering, receiving and referring a component easily, accurately and at a lower cost.

2. WDMS

2.1 Background

WDMS regards that hardware and embedded system's modules are as same level as component *organizing units. WDMS is a total management* system applied to components (i.e., hardware drawings, embedded system's modules drawings, Other documents, and source codes, and specifications are inevitable to manufacture units) and software programs.

Enterprise stands in need for a series of development process, design process, manufacturing process, inspection process, and shipment process. It also stands in need for many embedded system's modules (software products) in addition to design drawings, specifications and contracts. At this stage, hardware and embedded system's modules are separately managed. However, his conventional process should be revised. the both components are separately managed. However, because that both hardware and embedded system's modules are mounted on the same products' unit at the same time. Which makes lot of nascence trouble. For examples, the conventional process makes management mistakes, the degradation of management efficiency and needed lot of time-consuming in the process from development of products to the shipment. Therefore, in order to solve the problems, new approach should be applied to this conventional process.

2.2 Conventional PDM environment

Fig. 1 shows conventional development and manufacturing environment-using PDM.

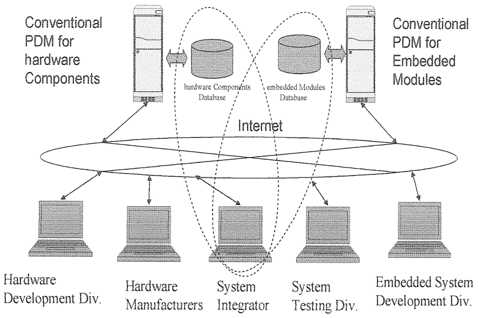

Conventional PDM for hardware Components
hardware Components Database
embedded Modules Database
Conventional PDM for Embedded Modules
Internet
Hardware Development Div.
Hardware Manufacturers
System Integrator
System Testing Div.
Embedded System Development Div.

Fig. 1 Conventional PDM environment

Development division registers embedded system's modules, drawing and document. Embedded system's modules and hardware drawings are stored in database respectively. Vendors refers catalog to sell products. manufactures and factories use embedded system's modules, drawing and document to produce products separately.

2. 3 conventional rule process

[†] Equipments Development Division, Fujitsu Limited

Fig. 2 is an explanatory view illustrating a conventional rule applicable method with respect to designing and manufacturing of the commodities. The method for studying and improving the operation process and analyzing a workflow is illustrated. According to this method, all persons in charge of designing and manufacturing of the products interchange and supply the operation information to study and improve the operation process and analyze a workflow. In Fig.2, a purchasing department M_1 applies control information C_1 (ISO standard [5]) with respect to a product development plan I_1 to calculate the cost as an activity A_1 and make the cost account into an output O_1. Next, a logical design department M_2 receives the output O_1 on the basis of a control information C_2 (logical standard) to create a logical design as the activity A_1 and make a logical design drawing into an output O_2. Then, a mounting design department M_3 receives the output O_2 and thermal design information I_2 to perform a mounting design on the basis of a control information C_3 (ISO standard) and make the mounting designing form into an output O_3. Hereinafter, the source code and the object code or the like of the embedded system's modules are outputted.

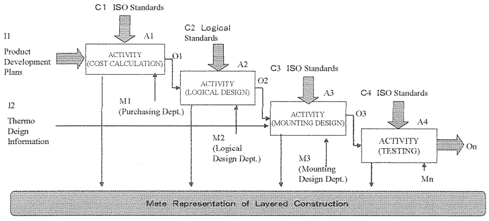

C1 ISO Standards
A1
I1
Product Development Plans
ACTIVITY (COST CALCULATION)
O1
C2 Logical Standards
A2
ACTIVITY (LOGICAL DESIGN)
O2
M1 (Purchasing Dept.)
I2
Thermo Deign Information
C3 ISO Standards
A3
ACTIVITY (MOUNTING DESIGN)
O3
C4 ISO Standards
A4
M2 (Logical Design Dept.)
ACTIVITY (TESTING)
On
M3 (Mounting Design Dept.)
Mn
Mete Representation of Layered Construction

Fig. 2 Conventional activities

2.4. Feature of WDMS

WDMS consists of component name server, predominant server and manufacturer server, those of them are described as follows. A component name server stores a name of a component (hereinafter, referred to as a component name information), attributes of component information and a storage address of component information or the like in a storage device. The component name server is connected to the Internet. A Predominant server stores meta information showing a layered construction between the component information (a cross relationship). The Predominant server is connected to the internet. This meta information is obtained by representing the layered construction of the component by the schema (RDFs) in the RDF or OWL. This schema (RDFs) is created broadly by three categories, namely, a category for registration, a category for making reference and a category for receiving. Further, for every category, the schema (RDFs) is created by respective departments, namely, a designing department, a manufacturing department, a purchasing department, a cost managing department, a maintenance department and a quality managing department.

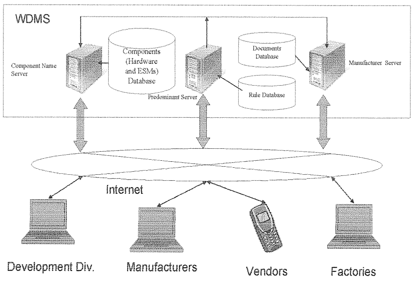


Fig. 3 Proposed WDMS environment

A manufacturer server stored the component information itself in a storage device. In fact, a plurality of manufacturer servers is provided to distributed to respective departments and the enterprises exploiting the outsourcing or the like. In other words, component information is distributed to the manufacturer servers in the locations, where the information is created, and is stored therein. Then, the operation of the above-described embodiment is explained with reference to flow charts shown in Figs. 8 to 12. On the respective displays of the registration system client, the receipt client system and the reference system client, a menu for selecting events (registration, reference, receipt and written deliberation) is displayed.

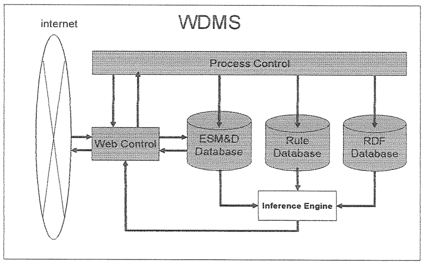


Fig. 4 Embodiment of WDMS

(1) Meta information

WDMS provides the component managing control system according to any one of the first aspect or the second aspect, which comprises a meta information storage server, which is connected to the network, for storing a meta information having a layered construction of at least component information. Said registration client registers the Meta information in the meta information storage server and registers the component information in the component information storage server. According to this WDMS, at least a meta information having a layered construction of the component information is made to be stored in the meta information storage server so that the user is capable of easily and accurately registering the component information without considering the complicated layered construction.

(2) Rule verification

WDMS provides the component managing control system, which comprises rule verification means for verifying a deliberation result of a written rule, which is *a source of the rule information, and registering rule* information in response to the verification result in the rule information storage server.

According to this WDMS, a deliberation result of a written rule as a source of the information rules is verified and the rule information in response to this verification result is made to be registered in the rule information storage server, so that the violation of the rule and the error or the like can be prevented compared with the case to register the information manually.

(3) Decomposition

WDMS provides the component managing control system, which comprises replacing means for repeatedly replacing a schema(RDFs) of the component information, which is described by the RDF or OWL by *using a predetermined method so that the evaluation* value in the agent means becomes maximum or minimum, when the evaluation value does not satisfy a target value. According to this WDMS, even if the evaluation value is under the desired value, the decomposition means repeatedly recomposes the schema(RDFs) so that this evaluation value becomes maximum or minimum. Therefore, the registration, reference or receipt of the component information is capable of being easily and accurately performed with detection closer to a man's detection.

3. Mechanism of WDMS

An embodiment of the component managing control system according to the proposed WDMS will more fully be apparent from the following detailed description with accompanying drawings.

3.1 embodiments of components' structure

Components list includes documents e_1 to e_n The layered construction [6] [7] [8] shown in Fig. 5 corresponds to a managing information PK , managing information CK_1 to CK_n as the component lists , which are belonged to this managing information PK, respectively and documents D_{11} to D_{1n} through documents D_{n1} to D_{nn}, which link to these managing information CK_1 to CK_n, respectively.

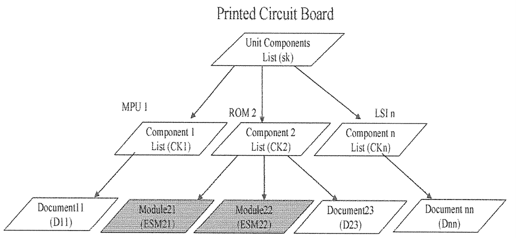
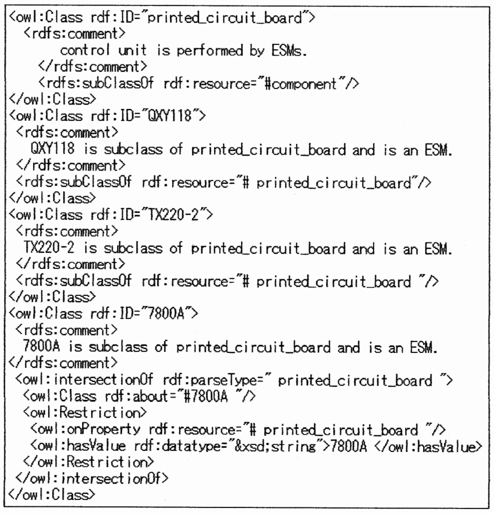


Fig.5 Components hierarchical structure

Figs. 5 and 6 are explanatory views illustrating a specific example of the layered construction of the identical embodiment.



```
<owl:Class rdf:ID="printed_circuit_board">
  <rdfs:comment>
       control unit is performed by ESMs.
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#component"/>
</owl:Class>
<owl:Class rdf:ID="QXY118">
 <rdfs:comment>
   QXY118 is subclass of printed_circuit_board and is an ESM.
 </rdfs:comment>
 <rdfs:subClassOf rdf:resource="# printed_circuit_board"/>
</owl:Class>
<owl:Class rdf:ID="TX220-2">
 <rdfs:comment>
  TX220-2 is subclass of printed_circuit_board and is an ESM.
 </rdfs:comment>
 <rdfs:subClassOf rdf:resource="# printed_circuit_board "/>
</owl:Class>
<owl:Class rdf:ID="7800A">
 <rdfs:comment>
  7800A is subclass of printed_circuit_board and is an ESM.
</rdfs:comment>
 <owl:intersectionOf rdf:parseType=" printed_circuit_board ">
  <owl:Class rdf:about="#7800A "/>
  <owl:Restriction>
   <owl:onProperty rdf:resource="# printed_circuit_board "/>
   <owl:hasValue rdf:datatype="&xsd;string">7800A </owl:hasValue>
  </owl:Restriction>
 </owl:intersectionOf>
</owl:Class>
```

Fig.6 Components construction

As shown in Fig. 6, in the Components lists control unit, at least parts given by the parts number QXY118, TX220-2 and 7800A (for example, a embedded system's modules) are included. A components construction list showing "a parts number", "a parts name", "a number of edition" and "a name of a manufacture" of these parts in Fig. 6. Further, in the embodiment, this parts component list is used as a parts construction list, which is described in the RDF or OWL shown in Fig. 6.In this Fig. 6, the items corresponding to the items in Fig. 7 are represented by the same reference numerals as those of Fig. 7.

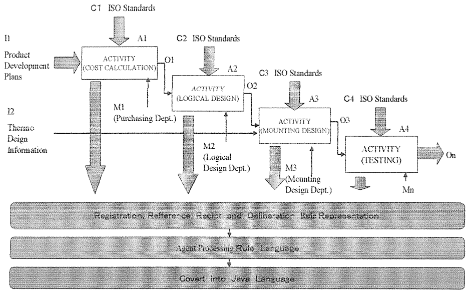

C1 ISO Standards
I1
Product Development Plans
A1
ACTIVITY (COST CALCULATION)
O1
C2 ISO Standards
A2
ACTIVITY (LOGICAL DESIGN)
O2
C3 ISO Standards
A3
ACTIVITY (MOUNTING DESIGN)
O3
C4 ISO Standards
A4
ACTIVITY (TESTING)
On
I2
Thermo Deign Information
M1 (Purchasing Dept.)
M2 (Logical Design Dept.)
M3 (Mounting Design Dept.)
Mn
Registration, Refferance, Recipt and Deliberation Rule Representation
Agent Processing Rule Language
Covert into Java Language

Fig.7 Proposed activities

In this Fig. 6, the items corresponding to the items in Fig. 7 are represented by the same reference numerals as those of Fig.2.

3.2 Registration, reference, and receipt rules

The agent rule language comprises a language for converting the registration rule information, the receipt rule information and the reference rule information into the JAVA language.

These registration rule information, the receipt rule information and the reference rule information are obtained by showing the information representing registration procedure of the component by a if/then format, respectively. Examples of the registration rule, the reference rule and the receipt rule are shown below.

(1) Registration rule

A registration rule 1 • • • • • • • • • • r11

If: a security segment of this component is in X class?

Then: this component may be registered

A registration rule 2 • • • • • • • • • • r12

If: this component has a correct title panel?

Then: this component may be registered

A registration rule n • • • • • • • • • • r1n

If: this component is related to hardware?

Then: a schema (RDFs) for the hardware may be registered end

(2) A reference rule

A reference rule 1 • • • • • • • • • • • • r21

If: this component is permitted to be made reference to?

Then: this component may be made reference to

A reference rule 2 • • • • • • • • • • • • r22

If: a purchasing department makes reference to this component

Then: the Schema (RDFs) of the purchasing department may be employed.

A reference rule n • • • • • • • • • • • • r2n

If: a manufacturing department makes reference to this component

Then: the schema (RDFs) of the manufacturing department may be employed.

(3) A receipt rule

A receipt rule 1 • • • • • • • • • • • • • • r31

If: this component is permitted to be received?

Then: this component may be received

A receipt rule 2 • • • • • • • • • • • • • • r31

If: a purchasing department receives this component?

Then: the schema (RDFs) of the purchasing department may be employed.

A receipt rule n • • • • • • • • • • • • • • r3n

If: a manufacturing department receives this component?

Then: the schema (RDFs) of the manufacturing department may be employed.

4. Embodiment of component Information process

4.1 Registration process

In step RG1 shown in Fig. 8, the registration system client derives the schema (RDFs) information for the designing department from the predominant server. In step RG2, the registration system client derives the definition/constitution/format information relating to the component from the Predominant server. In step RG3, the registration system client derives the component name information from the component name server. In step RG4, the registration system client derives the registration rule information from the predominant server.

In the steps RG5 to RG7, the registration system client makes a deduction of the registration rule information by using a deduction method of the if/then production rule or the like. For example, in he steps RG5 to RG7, the registration system client performs the deduction processing on the basis of the previous registration rules 1 to n in 3.3 (1).

Then, with reference to Fig.9, a deduction processing will be explained. In step RD1 shown in Fig. 9, the registration system client derives a front part of the registration rule (if the steps RG5 to RG7, the registration system client performs the deduction processing on the basis of the previous registration rules

1 to n in 3.3 (1)..Then, with reference to Fig. 9, a deduction processing will be explained. In step RD1 shown in Fig.9, the registration system client derives a front part of the registration rule (if part). In step RD2, the registration system client derives a back part of the registration rule (then part). In step RD3, the registration system client detects whether if part satisfies a first condition or not. If the detection result is "Yes", the registration system client determines a crisp value C (A) as 1. On the other hand, if the detection result of the step RD3 is "No", in step RD4, the registration system client detects whether a membership function μ (A) satisfies a second condition or not. If the detection result is "Yes", in step RD5, the registration system client determines the membership function μ (A) as a value at a midpoint in between 0 and 1.

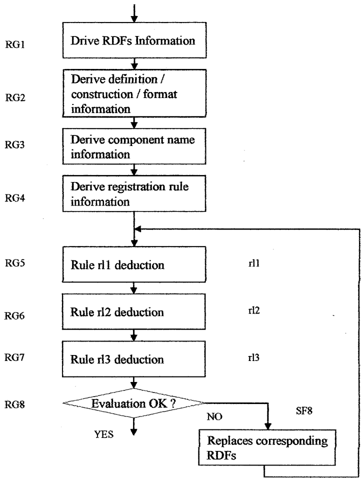

RG1
Drive RDFs Information
RG2
Derive definition / construction / format information
RG3
Derive component name information
RG4
Derive registration rule information
RG5
Rule rl1 deduction
rl1
RG6
Rule rl2 deduction
rl2
RG7
Rule rl3 deduction
rl3
RG8
Evaluation OK ?
NO
SF8
YES
Replaces corresponding RDFs

Fig.8 Registration process

The registration system detects whether an evaluation result is ok or not on the basis of an evaluation processing shown in Fig.9.

The registration system client detects whether the crisp value C (A) is 1 or not. If this detection result is "No", the registration system client detects whether the membership function μ (A) is 0 or not. If this detection result is "No", the registration system client detects whether the membership function μ (A) is 1 or not. If this detection result is "No", in step RD4, the registration system client recomposes the schema (RDFs) In Fig.8, in step RD8, the registration system client detects whether the evaluation result is a ok target value is equal to the evaluation result or not, this detection result is "Yes", the current component is registered in the storage device of the manufacturer server. On the other hand, the detection result in the step RD8 is "No", after recomposing the schema(RDFs) so that the evaluation result takes the minimum value or the maximum value, the registration system client repeats the processing after the step RD5.

4.2 Reference process

If "reference" is selected in the reference system client, the reference system client executes an electronic information relating to a drawing such as a circuit diagram and a structural drawing or the like, an information relating to respective EC forms (new design notification, design revision notification), an information relating to a program or a reference processing (agent processing) shown in electronic information relating to a drawing such as a circuit diagram and a structural drawing or the like, an information relating to respective EC forms (new design notification, design revision notification), an information relating to a program or a reference processing (agent processing) shown in Fig. 10 for making reference to the component with the layered construction such as an electronic information relating to respective manuals. In other words, in step RF1 shown in Fig.10, the reference system client derives the schema (RDFs) information for the designing department from the Predominant server. In step RF2, the reference system client derives the definition/constitution/format information relating to the component from the Predominant server. In step

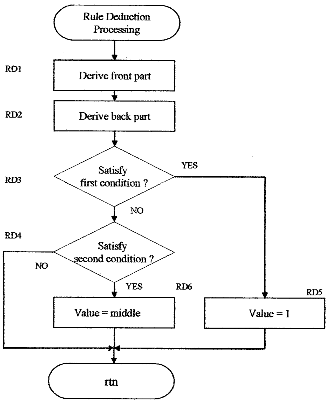


Fig.9 Deduction process

RF3, the reference system client derives the component name information from the component name server. In step RF4, the reference system client derives the reference rule information from the predominant.

In the steps RF5 to RF7, the reference system client makes a deduction of the reference rule information by using a deduction method of the if/then production rule or the like. For example, in the steps RF5 to RF7, the reference system client performs the deduction processing on the basis of the previous reference rules 1 to n in 3.3 (2).

Then, the reference system client effects the deduction processing on the basis of the reference rules 1 to n with reference to above described Fig.10.

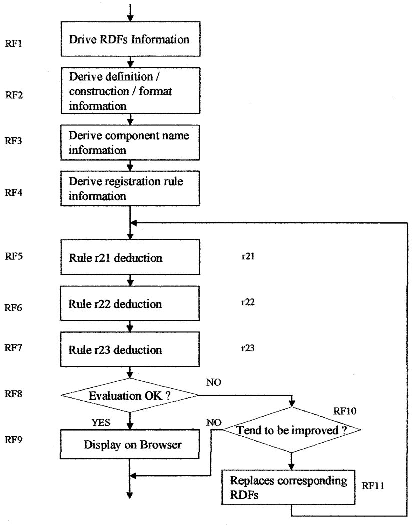


Fig.10 Reference process

In step RF8, the reference system client detects whether the evaluation result is ok or not on the basis of the evaluation result shown in Fig. 10 as it as in the RF8. If this detection result is "Yes", in step RF9, the reference system client makes reference to the component by displaying the current component registered in the manufacturer server on the RDF or OWL browser. On the other hand, if the detection result of the step RF8 is "No", in step RF10, the reference system client

Then, the reference system client effects the deduction processing on the basis of the reference rules 1 to n with reference to above described Fig.10. In step RF8, the reference system client detects whether the evaluation result is ok or not on the basis of the evaluation result shown in Fig. 8 as it as in the RF8. If this detection result is "Yes", in step RF9, the reference system client makes reference to the component by displaying the current component registered in the manufacturer server on the RDF or OWL browser. On the other hand, if the detection result of the step RF8 is "No", in step RF10, the reference system client detects whether the evaluation result tends to be improved compared with the former evaluation result or not. If this detection result is "Yes", in step RF11, after changing the schema (RDFs) information, the reference system client repeats the processing after the step RF5.

4.3 Receipt Process

If the receipt system client is selected "receipt", the receipt system client determines the receipt process. On this account, the receipt system client executes an electronic information relating to a drawing such as a circuit diagram and a structural drawing or the like, an information relating to a drawing such as a circuit diagram and a structural drawing or the like, an information relating to respective EC forms (new design notification, design revision notification), an information relating to a program or a receipt processing (agent processing) shown in Fig.11 for receiving the component with the layered construction such as an electronic information relating to respective manuals. In other words, in step RC1 shown in Fig.11, the receipt system client derives from the Predominant server any one schema (RDFs) information from respective schema (RDFs) information for the designing department, the manufacturing department, the purchasing department, the cost managing department, the maintenance department, the quality assurance department or the like and further derives the definition/constitution/format information from this schema (RDFs) information.

In step RC2, the receipt system client derives the parts list information from the component name server. In step RC3, the receipt system client derives the receipt rule information from the predominant. In step RC4, the receipt system client performs the deduction processing shown in Fig. 11 by using the deduction method or the like of the if/then production rule to make deduction of the receipt rule information. For example, the receipt system client performs the deduction processing on the basis of the previous receipt rules 1 to n in 3.3 (3).

Then, in step RC5 shown in Fig. 11, the receipt system client detects whether the evaluation result is ok or not on the basis of the evaluation processing shown in Fig.11 as same as in the step RC8. If this detection result is "Yes", the receipt system client receives the current component from the manufacturer server. On the other hand, if the detection result of the step RC5 is "No", in step RC6, the receipt system client detects whether the evaluation result tends to be improved compared with the former evaluation result or not. If this detection result is "Yes", in step RC7, after recomposing the schema (RDFs) information, the receipt system client repeats the processing after the step RC4.

Further, the above described registration rules, the reference rules and the receipt rules (hereinafter, simply referred as to rules) are deliberated in writing by the persons in charge of respective department before being stored in the predominant. The rules reflect the result from the written deliberation and they are stored in the predominant. A processing of the written deliberation to make reflect the result from the written deliberation on the registration rules, the reference rules and the receipt rules will be explained below.

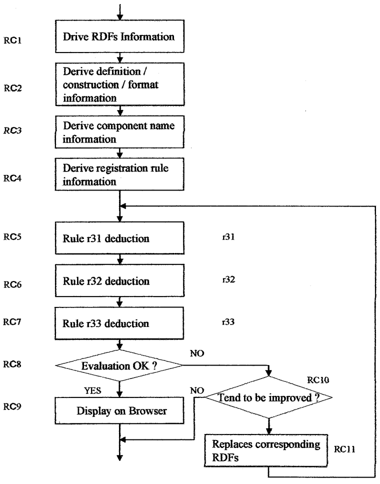


Fig. 11 Receipt process

4.4 Deliberate Process

When "written deliberation" is selected in the registration system client, the registration system client determines the detection result as "deliberate process". Thus, the registration system client executes the processing of the written deliberation shown in Fig. 12. In other words, in step DL1 shown in Fig. 12, the registration system client decomposes the original texts of the rules into some paragraphs.

In step DL2, the registration system client gives names to the decomposed paragraphs, respectively. In step DL3, the registration system client homologizes respective written deliberation results from respective departments (for example, A department, B department, C department) to the discomposed paragraph (of the original texts). In step DL4, the registration system client compares respective paragraphs with respective paragraphs of the written deliberation results by homologizing them to the schema (RDFs) information of the original texts.

In step DL5, the registration system client detects whether as a result of comparison in the step DL4, respective paragraphs agree with the written deliberation results or not. If the detection result is "Yes", in step DL6, the registration system client detects whether all paragraphs are compared with the paragraphs of the written deliberation results or not. If this detection result is "No", the registration system client repeats the processing in the step DL5. If the detection result in the step DL5 is "No", in step DL10, the registration system client detects whether this paragraph is included in the paragraphs, which do not agree with any paragraphs yet.

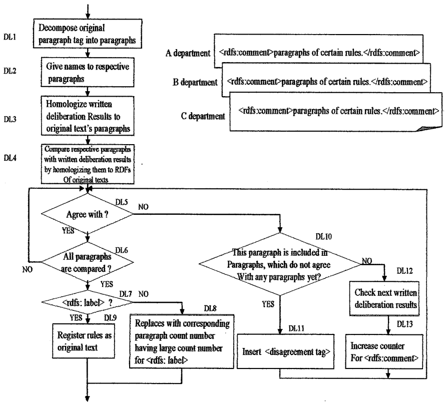

DL1
Decompose original paragraph tag into paragraphs
DL2
Give names to respective paragraphs
DL3
Homologize written deliberation Results to original text's paragraphs
DL4
Compare respective paragraphs with written deliberation results by homologizing them to RDFs Of original texts
A department
<rdfs:comment>paragraphs of certain rules.</rdfs:comment>
B department
<rdfs:comment>paragraphs of certain rules.</rdfs:comment>
C department
<rdfs:comment>paragraphs of certain rules.</rdfs:comment>
DL5
NO
Agree with ?
YES
DL6
All paragraphs are compared ?
NO
YES
DL7
NO
<rdfs: label> ?
YES
DL9
Register rules as original text
DL8
Replaces with corresponding paragraph count number having large count number for <rdfs: label>
DL10
This paragraph is included in Paragraphs, which do not agree With any paragraphs yet?
NO
YES
DL12
Check next written deliberation results
DL11
Insert <disagreement tag>
DL13
Increase counter For <rdfs:comment>

Fig. 12 Deliberation process

If the detection result in the step DL10 is "Yes", in step DL11, the registration system client inserts a tag <comment> ... </comment>, which represents a new disagreed paragraph in the original texts. In this case, "disagreement" is inserted in the tag <comment> ...

</comment>. On the other hand, if the detection result of the step DL10 is "No", in step DL12, the registration system client checks the next written deliberation result (paragraph). In step DL13, the registration system client increases a counter for calculating the tag <Pn>, which is given to the paragraph. Then, the registration system client repeats the processing on and after the step DL5.

Then, if the detection result in the step DL6 is "Yes", in step DL7, the registration system client detects whether the counter of <Pn> is 1 or not. If this detection result is "Yes", in step DL9, the registration system client registered the rule as the original text in the predominant. On the other hand, in the case that the detection result of the step DL7 is "No", in step DL8, the registration system client replaces the current paragraph of the original text with the corresponding paragraph of the written deliberation result and registers the rule after replacement in the predominant.

As explained above, according to the embodiment, the registration system client, the receipt system client or the reference system client derives the rule information from the predominant and derives the meta information from the predominant server to register, receive or make reference to the component information on the basis of these rule information and meta information. Compared with the conventional case that the registrant, the receptor or the person who makes reference registers, receives or makes reference to the registration rules, the receipt rules or the reference rules, which is described on the document while making reference to the registration rules, the receipt rules or the reference rules, it is possible to register, receive or make reference to more easily and accurately.

An embodiment of the present WDMS has been described in detail with reference to the accompanying drawings but it is to be understood that the practical arrangement is not limited to this specific embodiment, and that various design changes may be made without departing from the spirit and scope of the present WDMS. For example, in the above-described embodiment may record the component managing control program to realize the above described functions and the component managing control program recorded in this recording medium may be read in the computer to execute the program.

5. Conclusion of WDMS

(1) WDMS is proposed to distribute embedded system's modules and its relating documents in conjunction with hardware drawings and document. integrated distributions are very effective to for both hardware and embedded system's modules. Double web management systems are rationalized. Human powers are also rationalized. Client systems are automatically able to receive drawings and their document. As a result, recipient's works are also reduced in cost by WDMS.

(2) WDMS is able to accommodate user's needs as the component managing control system, which are capable of registering, receiving and referring a component easily, accurately and at a lower cost.

(3) Even if the evaluation value is under the desired value, the decomposition performs repeatedly for the schema (RDFs) information so that this evaluation value becomes maximum or minimum. Therefore, the registration, reference or receipt of the component information is capable of being easily and accurately performed with detection closer to a man's detection.

References

[1] I Nonaka" A Dynamic Theory of Organizational Knowledge Creation" *Organization Science*, Vol. 5, No. 1 (Feb., 1994), pp. 14-37

[2]http://www.w3.org/TR/RDFschema/

[3]T.Ohashi "*Consideration of Firmware 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.

[4] Y. Fukuda "*Enterprise Model*" (in Japanese) Journal of Japanese Society for Artificial Intelligence Vol.13 No.6 pp.3-9. 1998

[5] ISO/IEC "ISO12207"

[6] R. Mizoguchi "*Ontology* " (in Japanese) JSAI Vol.13 No.1 pp.9-10. 1998

[7] Toshihiro Moriya, Takayasu Ito "*Web Ontology Language Based on Fuzzy Knowledge Representation System*" IPSJ National Convention No.65, pp.5-151～5-154

[8] Nick Frangiadakis in the Computer Science Department at the University of Maryland. http://www.cs.umd.edu/~ntg/828Y/LessonLearned2.