Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Web Service-Based System for Sharing Distributed XML Data Using Customizable Schema

Akira Hattori, Kuniaki Tabata and Haruo Hayami
Faculty of Information Technology
Kanagawa Institute of Technology
1030 Shimo-ogino, Atsugi, Kanagawa, 243-0292, Japan
{ahattori, tabata, hayami}@ic.kanagawa-it.ac.jp

***Abstract*—We describe a system for sharing distributed XML data among organizations using customizable schema. In our system, each organization can customize common DTD (Document Type Definition) within the bound of rule to create its own DTD, which is called customized DTD. And then it can describe information with tags according to the customized DTD. Basically, the upper structure of shared XML data is common. Proposed system consists of a management server, data providing servers, and application servers. They communicate with each other through web services technology. Customized DTDs are managed centrally by the management server. The server also examines whether they conform to the rule of customization. As a result of prototype system using welfare information, we found that our system was effective for sharing information among organizations of which each works freely in the same field.**

***Index Terms*—Sharing distributed XML data, Web services, DTD, Welfare information**

I. INTRODUCTION

Recently, a system for sharing distributed XML (Extensible Markup Language) data among organizations is strongly desired. This is because XML is widely accepted as a format for data exchange and representation on the web [1].

In general, an XML data explicitly or implicitly conforms to a certain schema. Thus, standardization of the schema is necessary when organizations share their XML data. However, because each of them has its own purpose, the standardization is difficult. On the other hand, if they freely make their own schemata without any constraints, it will be hard to share XML data.

In this paper, a new type of system that allows for variations based on a common schema is proposed to share XML data among organizations. We use DTD (Document Type Definition), and refer to a common schema and variation based on it as a common DTD and a customized DTD respectively. In our proposed system, organizations agree on a common DTD, and each of them makes its own customized DTD within the bound of customization rules. And then it makes the XML data according to the customized DTD available to provide the information which it has. Basically, a common DTD and customization rules make upper portion of tree structure of shared XML data unified.

II. RELATED WORK

A number of studies have focused on the integration of heterogeneous XML data sources [2][3], which are distributed over the web. They typically extend the schema integration approach in database field to handle XML data [4]. Their systems map the structures of distributed XML data to a global schema, which is an integrated view of them, or introduce a common ontology. This overcomes the problem of the structural or semantic heterogeneous of the distributed XML data. Users do not need to construct queries according to the different structures. There is also a study to search for web services in the same manner [5]. However, because each of organizations actives independently, it is difficult to prepare and maintain the global schema and the common ontology.

To solve the problem, a system which enables integrated retrieval for heterogeneous XML data sources without referring to the structure of the XML data has been proposed [6]. This kind of system has to place a certain amount of constraint on the way to define tags. However, the constraint has not been discussed in the study.

Several articles have been devoted to the studies of managing the similarity among the schemas of the distributed XML data [7][8]. To extract useful information from various XML data in response to users' requirements, a method of creating compound schema has been proposed. The method is like our system because it uses a common schema as a core one. However, it manages the correspondence between two schemas using the common schema. It is hard to manage the similarity or the correspondence.

Architecture to provide support for distributed data management in loosely coupled data sources has been suggested [9]. In the system, community schemas are introduced, which are used to wrap each of the schemas of data sources. They can be constructed based on the existing ones. The study does not target XML data, but it is interesting because it has the same awareness of the issue as our study. However, when we apply the architecture to XML data, organizations cannot add new tags to the community schemas.

III. A SYSTEM FOR SHARING DISTRIBUTED XML DATA USING CUSTOMIZABLE SCHEMA

A. Outline of the System

In our system, each organization creates customized DTD by adding and changing some element type declarations and attribute-list declarations in a common DTD. And they provide XML data conforming to their own customized DTDs to each

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

other using web services technology. To share distributed XML data, our system manages all customized DTDs and interfaces of their web services centrally.

However, if each organization creates customized DTD without any constraints, it will be inconvenient for sharing their XML data. There is no point in introducing a common DTD just as they freely make their own DTD. Thus, our system has rules of the way to customize the common DTD. For example, element name can be changed when its content is the semantically same as corresponding element in the common DTD, a child element can be added to an element in it but not a sibling element, and so on. We refer to the rules as customization rules.

Organizations are allowed to create customized DTDs within the limitations of the customization rules. To realize the proposed system, it is necessary to build up a mechanism which makes it possible for them to share their XML data confirming to customization rules.

B. Fundamental Technology of the System

1) Web Services: A Web service is a kind of distributed processing technology. Its interface is described by WSDL (Web Services Description Language). To exchange data encoded in XML between systems, SOAP (Simple Object Access Protocol) is used [10].

2) Entity Declaration: An XML data may consist of one or many storage units. These are called entities; they all have content. There are two types of entities: general entity which is used in the content of XML, and parameter entity which is used in DTD. To use entities, we need to declare them [11].

3) Internal Subset and External Subset: An XML data may have an optional DTD, which defines its structure. DTD is included inside the XML data, or is stored in a separate file. The former is called an internal subset and the latter is called an external subset. XML data can contain both subsets, and internal subset is processed first. Although we are not allowed to declare the type of an element more than once, we are allowed to declare an entity more than once [11]. When an entity is defined more than once, only the first declaration is used.

Thus, an entity declaration in the internal subset can be used to override the value of an entity in the external subset. That is to say, by using a parameter entity, we can redefine the element type definition which is defined in an external subset in an internal subset.

C. System Structure

Fig. 1 shows the structure of our proposed system. The system consists of a management server, data provision servers, and application servers. A data providing server provides XML data using by Web services, and an application server is an application program to use XML data provided by data providing servers. The management server will be explained later.

A method to operate proposed system is follows:

1) Organizations that take part in sharing XML data decide a common DTD, the syntax and semantics of the tags declared in it, and customization rules in collaboration with each other. And they are registered on the management server.
2) To provide XML data, each organization makes its own customized DTD, XML data conforming to the DTD, web service to provide the data, and WSDL of the service.
3) The organization checks whether the XML data follows customization rules or not, and register management information on the management server. The information consists of customized DTD, WSDL document, and so on.
4) To use provided XML data, each organization refers to the management information, and makes or modifies the application used by the organization.
5) The organization uses the application to get XML data.

On the other hand, the management server executes the following processes:

1) It creates sample XML data from the common DTD and the syntax and semantics of the tags declared in the DTD.
2) It verifies whether tags of the XML data provided by an organization conforms to the customization rules.
3) It extracts customized portion and the name of the web service, and display input form of management information.
4) It creates management information according to input by the organization.
5) It provides the management information in response to a search request from user organizations of XML data or their application servers.

In the next section, we will explain the functions of the management server.

D. Design of the Management Server

1) Sample XML Data: In our system, because parent-child relationship and sibling one among elements of an XML data is important, the system needs to manipulate its tree structure. On the other hand, DOM (Document Object Model) is one of APIs for accessing the components of an XML data [11]. It allows an application to navigate through a tree of nodes representing elements, attributes, and so on, which is called a DOM tree. Thus, our system creates an XML data from a common DTD. To do that, to begin with, the common DTD is parsed using an XML editor or a tool to parse a DTD. Then, DOM tree is created in accordance with the parsed DTD. Finally, the DOM tree is saved as an XML data, in which any text node is not included. Resulting XML data is the skeleton of some XML data conforming to the common DTD and we call it ”sample XML data”. There are two objectives of sample XML data. The first is to make it possible to execute verification and search according to tree structure. The second is to make it possible for organizations to use it as a template in making their own XML data.

2) Management Method of Customization Rules: An XML data basically contains some elements and attributes. The elements lie at the heart of it, while the attributes complement

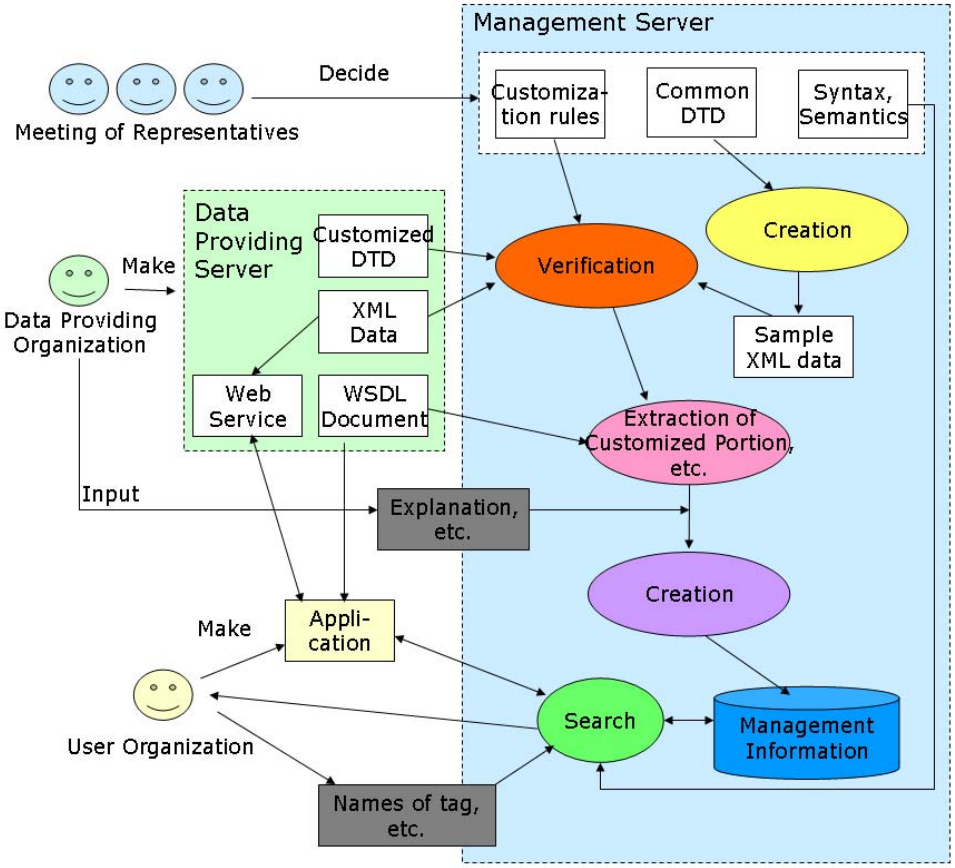


Fig. 1. System Structure

the information about elements. This is because we focused on XML elements in the management method of customization rules. Table I shows allowable customization manners in our system. The manners (1) to (4) are related to element itself, manner (5) and (6) are related to element content, and manners (7) to (9) are related to attributes. Our system manages whether each element declared in a common DTD is allowed the manners (1) to (7) or not. It also manages whether each attribute in it is allowed the manner (8) and (9). These information are customization rules and managed in the form of ”element name or attribute name: list of numbers of allowed customization manner”

When verifying whether XML data of each organization follows customization rules or not, our system traverses the entire tree from the root element of customized DTD to process each node, which is element node, as follows. Fig. 2 indicates image of the verification algorithm. In the figure, element C is a processed node.

First, processed node and its child nodes are mapped to nodes declared in a common DTD. If necessary, a person in charge of the organization may specify the mapping occasionally. The mapping results are managed by using a table.

TABLE I
Allowable Customization Manner of a Common DTD

#	Customization manner
(1)	Change element name
(2)	Abbreviate element
(3)	Add sibling element before this one
(4)	Add sibling element after this one
(5)	Change the order of child elements
(6)	Add child element
(7)	Add attribute
(8)	Change the name of attribute
(9)	Abbreviate attribute

Then, it is verified whether the child nodes of the processed node follow the rules, which are changed and addition of child nodes. And then, if there is a child node of the node mapping to the processed node which is not mapping to any child nodes of the processed node, for example node G in Fig. 2, whether abbreviating the child node is allowed is confirmed. Finally, it is verified whether the order of the child nodes of the processed node and addition of sibling nodes to them follow the rules. To execute this verification process, our system retains the

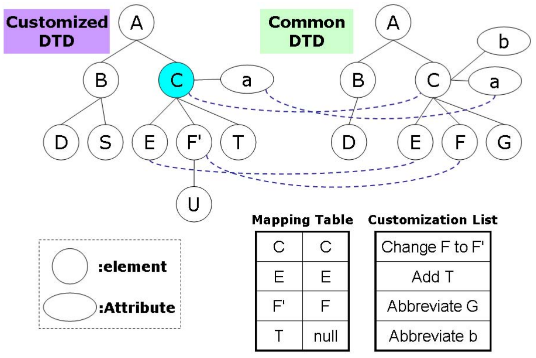


Fig. 2. Image of Validation Algorithm

information of the change of element name and addition and abbreviation of elements, which are got through preceding processes. Attributes are also verified in the same manner except for the order and sibling relationship.

The table to hold the mapping between customized DTD and common DTD and the information of the change, addition and abbreviation are produced as a result of execution of the verification algorithm. They are also used to detect customized portion, which is a next step.

3) Handling of Management Information: Management information consists of the information about elements and attributes added to a common DTD, which is their name and explanation. It also contains a child element of the root element of XML data confirming to a customized DTD, which is a part of its instance. In addition, it contains the information about the web service to provide the XML data, which is the address of its WSDL document and elements and attributes to search for XML data used by the service. Management information is created for each data providing organization and stored in an XML format. Management server enables it for users, which are persons and computer programs, to search for the information by specifying the name of an element declared in a common DTD or data providing organization. It provides computer programs with that function as a web service.

E. Examples Using Welfare Information

To explain functions of our proposed system, which are registration of and search for management information, we made a prototype using welfare information, which is about shop or public facility such as stairs in a building entrance.

1) A Common DTD and Customization Rules: We made some data providing servers and a common DTD based on existing barrier-free map systems, which provide welfare information on the Web. Its tree structure is shown in Fig. 3.

In Fig. 3, you are allowed to change FName, Address and Tel, which are element names, to another name, and to substitute an element content for text (#PCDATA) as the contents of them. However, their contents have to contain the name, address and telephone number of a facility, respectively.

On the other hand, while you are not allowed to change Toilet, Parking, Entrance and Elevator to another name, you are allowed to substitute an element content for text. In a common DTD, the contents of them are #PCDATA. Similarly, you are allowed to substitute an element content for an empty element (EMPTY) as the contents of the elements named Others, BOthers and EOthers. In a common DTD, they are defined as options. Table II summarizes the customization rules of our prototype system.

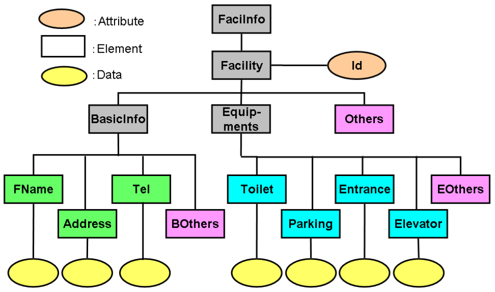


Fig. 3. Common DTD in our Prototype System

TABLE II
CUSTOMIZATION RULES OF THE PROTOTYPE SYSTEM

Element name	Allowable manner
FaciInfo, Facility, BasicInfo, Equipments	no
FName, Address, Tel	(1), (6)
Toilet, Parking, Entrance, Elevator	(5), (6)
Others, BOthers, EOthers	(2), (6)

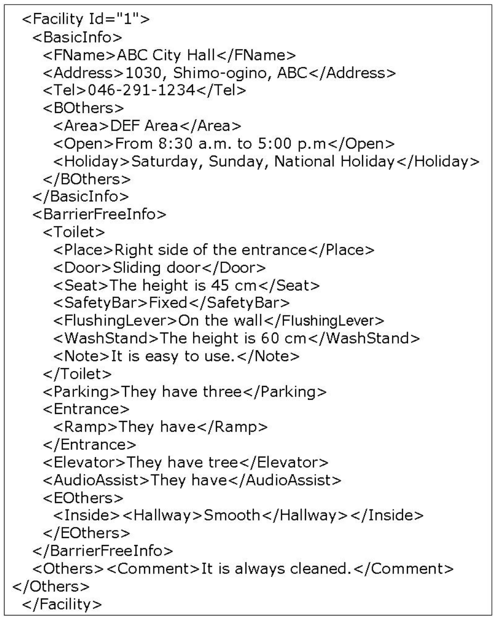
For example, DTD of toilet element is as in the following:
<!ENTITY % pcd.toilet ”#PCDATA”>
<!ELEMENT toilet (%pcd.toilet)>

2) Registration of Management Information: Fig. 4 is an example of welfare information in XML format and Fig. 5 shows the result of verification of the XML data.

In the result of verification, parts of customization which follows the customization rules shown in Table II are indicated in green. On the other hand, ones which do not follow the rules are indicated in red. When the customization does not follow the rules, the way to correct it is also suggested if possible. In Fig. 5, the change of Equipments to BarrierFreeInfo and addition of child element, which is audio assist, to it do not follow the rules. You can see that the mistakes are properly detected.

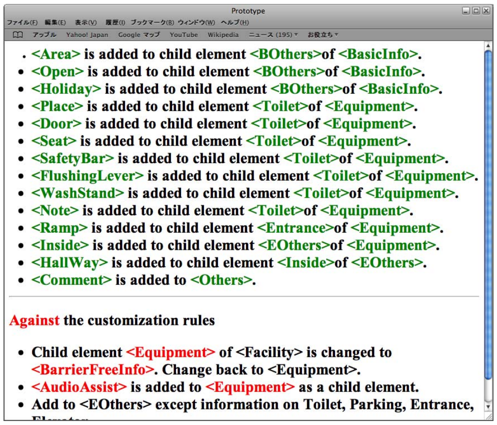
On the other hand, Fig. 6 shows the result of verification of the XML data after the mistakes are corrected according to the suggestion. Because all the customizations follow the rules, added tags and the method name of the web service are detected. In this form, you can fill in the explanations of the tags and ones used to search for welfare information. Management server creates management information from the information filled in.

3) Search for Management Information: Fig. 7 shows the search result of management information using an element name, which is BasicInfo, defined in the common DTD. The



```
  <Facility Id="1">
    <BasicInfo>
      <FName>ABC City Hall</FName>
      <Address>1030, Shimo-ogino, ABC</Address>
      <Tel>046-291-1234</Tel>
      <BOthers>
        <Area>DEF Area</Area>
        <Open>From 8:30 a.m. to 5:00 p.m</Open>
        <Holiday>Saturday, Sunday, National Holiday</Holiday>
      </BOthers>
    </BasicInfo>
    <BarrierFreeInfo>
      <Toilet>
        <Place>Right side of the entrance</Place>
        <Door>Sliding door</Door>
        <Seat>The height is 45 cm</Seat>
        <SafetyBar>Fixed</SafetyBar>
        <FlushingLever>On the wall</FlushingLever>
        <WashStand>The height is 60 cm</WashStand>
        <Note>It is easy to use.</Note>
      </Toilet>
      <Parking>They have three</Parking>
      <Entrance>
        <Ramp>They have</Ramp>
      </Entrance>
      <Elevator>They have tree</Elevator>
      <AudioAssist>They have</AudioAssist>
      <EOthers>
        <Inside><Hallway>Smooth</Hallway></Inside>
      </EOthers>
    </BarrierFreeInfo>
    <Others><Comment>It is always cleaned.</Comment>
</Others>
  </Facility>
```

Fig. 4. An Example of Welfare Information


Prototype
ファイル(F) 編集(E) 表示(V) 履歴(I) ブックマーク(B) ウィンドウ(W) ヘルプ(H)
アップル Yahoo! Japan Google マップ YouTube Wikipedia ニュース (195) お役立ち

- <Area> is added to child element <BOthers>of <BasicInfo>.
- <Open> is added to child element <BOthers>of <BasicInfo>.
- <Holiday> is added to child element <BOthers>of <BasicInfo>.
- <Place> is added to child element <Toilet>of <Equipment>.
- <Door> is added to child element <Toilet>of <Equipment>.
- <Seat> is added to child element <Toilet>of <Equipment>.
- <SafetyBar> is added to child element <Toilet>of <Equipment>.
- <FlushingLever> is added to child element <Toilet>of <Equipment>.
- <WashStand> is added to child element <Toilet>of <Equipment>.
- <Note> is added to child element <Toilet>of <Equipment>.
- <Ramp> is added to child element <Entrance>of <Equipment>.
- <Inside> is added to child element <EOthers>of <Equipment>.
- <HallWay> is added to child element <Inside>of <EOthers>.
- <Comment> is added to <Others>.

Against the customization rules

- Child element <Equipment> of <Facility> is changed to <BarrierFreeInfo>. Change back to <Equipment>.
- <AudioAssist> is added to <Equipment> as a child element.
- Add to <EOthers> except information on Toilet, Parking, Entrance, Elevator

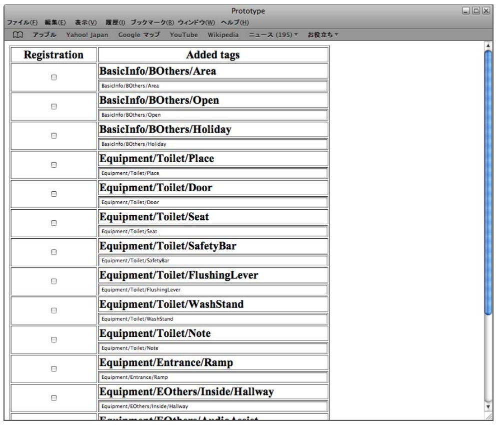
Fig. 5. An Validation Result of XML Data

result contains how each organization customized the common DTD. In the figure, tags added through customization are indicated in red, and ones of which element names are changed are indicated in green. Fig. 8 shows the search result of it by selecting data providing organization. As Fig. 7, how selected organization customized it is indicated in red and green. In this manner, you can understand what tags are used by other organizations.

IV. Discussion

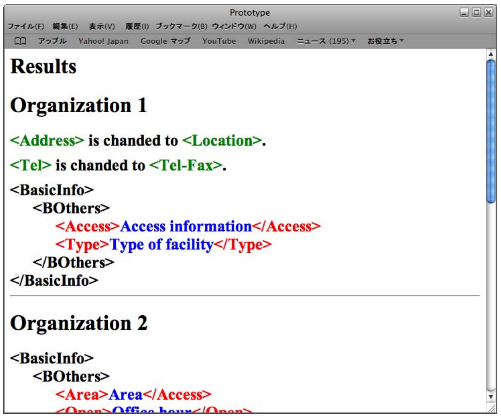
A. Advantage of our System

In our system, each organization can create its own customized DTD based on a common DTD. Management server of the system verifies whether the customization conforms


Prototype
ファイル(F) 編集(E) 表示(V) 履歴(I) ブックマーク(B) ウィンドウ(W) ヘルプ(H)
アップル Yahoo! Japan Google マップ YouTube Wikipedia ニュース (195) お役立ち

Registration	Added tags
□	BasicInfo/BOthers/Area BasicInfo/BOthers/Area
□	BasicInfo/BOthers/Open BasicInfo/BOthers/Open
□	BasicInfo/BOthers/Holiday BasicInfo/BOthers/Holiday
□	Equipment/Toilet/Place Equipment/Toilet/Place
□	Equipment/Toilet/Door Equipment/Toilet/Door
□	Equipment/Toilet/Seat Equipment/Toilet/Seat
□	Equipment/Toilet/SafetyBar Equipment/Toilet/SafetyBar
□	Equipment/Toilet/FlushingLever Equipment/Toilet/FlushingLever
□	Equipment/Toilet/WashStand Equipment/Toilet/WashStand
□	Equipment/Toilet/Note Equipment/Toilet/Note
□	Equipment/Entrance/Ramp Equipment/Entrance/Ramp
□	Equipment/EOthers/Inside/Hallway Equipment/EOthers/Inside/Hallway
	Equipment/EOthers/AudioAssist

Fig. 6. An Input Form of Management Information


Prototype
ファイル(F) 編集(E) 表示(V) 履歴(I) ブックマーク(B) ウィンドウ(W) ヘルプ(H)
アップル Yahoo! Japan Google マップ YouTube Wikipedia ニュース (195) お役立ち

Results

Organization 1

<Address> is chanded to <Location>.

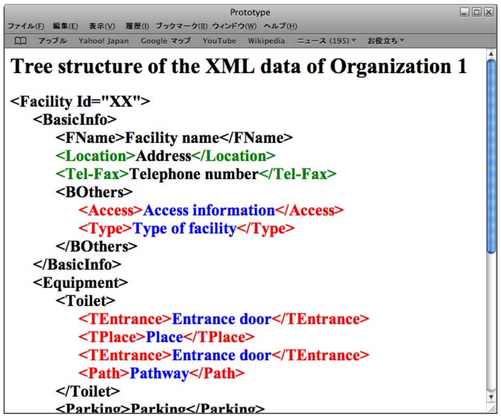
<Tel> is chanded to <Tel-Fax>.

```
<BasicInfo>
    <BOthers>
        <Access>Access information</Access>
        <Type>Type of facility</Type>
    </BOthers>
</BasicInfo>
```

Organization 2

```
<BasicInfo>
    <BOthers>
        <Area>Area</Access>
        <Open>Office hour</Open>
```

Fig. 7. Search Result by Element Name


Prototype
ファイル(F) 編集(E) 表示(V) 履歴(I) ブックマーク(B) ウィンドウ(W) ヘルプ(H)
アップル Yahoo! Japan Google マップ YouTube Wikipedia ニュース (195) お役立ち

Tree structure of the XML data of Organization 1

```
<Facility Id="XX">
    <BasicInfo>
        <FName>Facility name</FName>
        <Location>Address</Location>
        <Tel-Fax>Telephone number</Tel-Fax>
        <BOthers>
            <Access>Access information</Access>
            <Type>Type of facility</Type>
        </BOthers>
    </BasicInfo>
    <Equipment>
        <Toilet>
            <TEntrance>Entrance door</TEntrance>
            <TPlace>Place</TPlace>
            <TEntrance>Entrance door</TEntrance>
            <Path>Pathway</Path>
        </Toilet>
        <Parking>Parking</Parking>
```

Fig. 8. Search Result by Data Provider

to the rule when an organization registers the management information of its web service. These enable organizations to add their tags under some degree of constraint and to share their XML data among each other, preserving the uniformity of distributed XML data. We used welfare information in prototype system. Each organization conducts various activities and collects information it wants. That is why it is not practical to standardize the information even if they are engaged in similar activity. However, it is possible to find commonalities among their information, for example, the name and address of facility. Therefore we think that our system is effective in such situation. Previous researches use a global schema which is unique one among organizations. Thus, it is difficult to share the information which can not be described using the tags conforming to the global schema.

In our system, management information is managed centrally, and organizations can understand what tags each of them use easily. They are able to easily find organizations that might have data they want. And this will make them to know activities of each other and to take advantage of ones to plan their own activities. Managing management information centrally would also enable our system to get periodically whether the customized DTD have been update or not to keep it up to date, even after registration of management information.

In addition, each organization participating in information sharing makes a web service to provide its XML data. When using web service, a service provider decides the interface of the service and publishes WSDL document which describes it. Using the WSDL document, other system can communicate with the service to exchange their data among each other. The characteristics of web service are suitable for information sharing among organizations which are active independently.

B. Problems

In our system, organizations can add and change elements and attributes freely within the limitations of customization rules. If each of them creates a customized DTD without regard of the tags used by others, a number of different DTDs will be created. This could make it difficult to process distributed XML data to use it. To solve this problem, when an organization creates an XML data, it is necessary to get management information and to present elements and attributes which were added to and changed from a common DTD. Such functionality assists it in using as same tags with other organizations as possible. The way of changing the different tags of distributed XML data, which are describing the same information, for single tag and providing the changed XML data to an application server is also effective. In this case, application servers do not get XML data from data providing servers directly, but get them through management server.

On the other hand, verifying an XML data created by a data providing organization follows customization rules is also troublesome. Because when doing that, an organization has to specify all added/changed elements and attributes. In order to solve this problem, it is necessary to automatically relate them to the tags declared in a common DTD to some degree. We believe that it is effective to manage synonyms of the names of elements and attributes in the common DTD. They could be used when relating elements and attributes in a customized DTD to ones in the common DTD. Using the existing management information is also effective for solving this problem. That is to say, they have the potential to serve as synonyms. In addition, it will be worth extracting elements sequence which you are not allowed to customize and reflecting it to verification algorithm.

V. Conclusions

This paper presented a system to share distributed XML data among organizations, in which each of them is allowed to create original schema based on a common schema to describe its own information. They customize a common DTD within customization rules, and make XML data conforming to their customized DTDs. We developed a prototype of our proposed system and discussed the effectiveness of the system.

Future work includes inventing and implementing a mechanism to solve the problem of a huge number of customized DTDs and efficient verification algorithm. We need to do that based on a quantitative evaluation. In this paper, we have not discussed update and retrieval of XML data itself called instance. Effective and efficient update and retrieval are also left for future works. In addition, we have to take into consideration issues related to information sharing in the real organizational activities. It is necessary to study system functions including operation side such as an access control to customized DTDs.

References

[1] A. D. Jhingran, N. Mattos, and H. Pirahesh, "Information integration: a research agenda, IBM Systems Journal," Vol.41, No.4, 2002, pp.555–562.

[2] Y. K. Nam, J. Goguen, and G. Wang, "A metadata integration assistant generator for heterogeneous distributed databases," Proceedings of Confederated International Conferences DOA, CoopIS and ODBASE 2002, 2002, pp.1332–1344.

[3] H. Xiao and I. Cruz, "Integrating and exchanging XML data using ontologies," Journal on Data Semantics, Vol.6, 2006, pp.67–89.

[4] L. Lakshmanan and F. Sadri, "Interoperability on XML Data," Lecture Notes in Computer Science Vol.2870, 2003, pp.146–163.

[5] T. Syeda-Mahmood, G. Shah, R. Akkiraju, A.-A. Ivan, and R. Goodwin, "Search service repositories by combining semantic and ontology matching", Proceedings of 2005 IEEE International Conference on Web Services, 2005, pp.13–20.

[6] G. Suzuki, K. Konishi, T. Hayashi, N. Kobayashi, and T. Honishi, "Mediation system for heterogeneous information sources based on XML : MediPresto/XM," IPSJ SIG Notes, Vol.2001, No.70, 2001, pp.461–467.

[7] I. Kojima and M. Hanasaka, "Design and implementation of integrated metadata management system based on XQuery/OAI and its application for OGSI/Web services," Proceedings of DEWS2003, 2003. http://www.ieice.org/ de/DEWS/proc/2003/papers/3-C/3-C-02.pdf

[8] R. Akimoto and W. Kameyama, "Implementation and evaluation of metadata schema integration mechanism based on museum information," IPSJ SIG Notes, Vol.2007, No.77, 2007, pp.1–6.

[9] E. Spyropoulou and T. Dalamagas, "SDQNET: semantic distributed querying in loosely coupled data sources," Proceedings of the 10th East European Conference, Advances in Databases and Information Systems, 2006, pp.55–70.

[10] T. Erl, Service-Oriented Architecture: a field guide to integrating XML and Web services, Prentice Hall Ptr, 2004.

[11] N. Bradley, The XML companion, ADDISON-WESLEY, 2002.