# A XML Schema Matching based on Fuzzy Similarity Measure

Chang Suk Kim* and Kwee-Bo Sim**

* Department of Computer Education, Kongju National University, Chungnam, Korea
(Tel : +82-41-850-8822; E-mail: csk@kongju.ac.kr)
**School of Electrical and Electronic Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-823-5319; E-mail: kbsim@cau.ac.kr)

**Abstract**: An equivalent schema matching among several different source schemas is very important for information integration or mining on the XML based World Wide Web. Finding most similar source schema corresponding mediated schema is a major bottleneck because of the arbitrary nesting property and hierarchical structures of XML DTD schemas. It is complex and both very labor intensive and error prune job. In this paper, we present the first complex matching of XML schema, i.e. XML DTD, inlining two dimensional DTD graph into flat feature values. The proposed method captures not only schematic information but also integrity constraints information of DTD to match different structured DTD. We show the integrity constraints based hierarchical schema matching is more semantic than the schema matching only to use schematic information and stored data.

**Keywords:** XML, schema matching, fuzzy similarity, DTD, integrity constraints

## 1. INTRODUCTION

As Extensible Markup Language (XML) is fast emerging as the dominant standard for representing data in the World Wide Web, numerous researches have been spurred to facilitate research on the integration of information on the Web.

To integrate information or to query in an information integration system, the system has to select the relevant information of several diverse Web sites. In Fig.1, each site has different XML schema such as Source schema_*1*, Source schema_*2* and Source schema_*n* in Realty.com, Homeseekers.com and City.co.kr respectively. Though we can usually find a similar XML schema in a same type of business Web sites, finding most similar source schema corresponding mediated schema manually is very labor intensive and error prune job.

One major bottleneck in information integration on the XML based World Wide Web is an equivalent schema matching between *mediated schema* and several different *source schemas*. In this paper, we present the first complex matching of XML schema, i.e. XML DTD. The proposed method captures not only schematic information but also integrity constraints information of DTD to match different structured DTD. We show the integrity constraints based hierarchical schema matching is more semantic than the schema matching only to use schematic information and stored data.

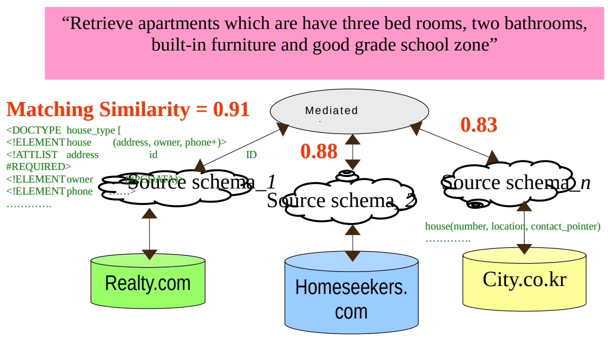


Fig. 1. Schema matching in an information integration system

## 2. PROBLEMS AND RELATED WORKS

As XML DTD has nested tag structures, matching hierarchical XML DTD to other XML DTD is not a trivial task (Fig. 2). There are several difficulties including non 1-to-1 mapping, set values and recursion issues. Recently, several matching method for XML DTD to other XML DTD, but most of all match 1-to-1 matching of each leaf node [1, 2, 3].

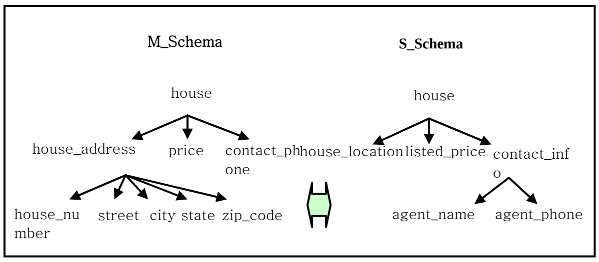


Fig. 2. Matching problem of mediated schema and source schema

Extracting integrity constraints as well as structural information play an important role in schema matching. Doan [3] proposed an automatic schema mapping using machine learning approach. But he used similarity of attribute name and semantic of stored data. Doan did not consider operator node ?, * or + of XML DTD. Milo [2] performed schema matching based on the name and structure of schema elements. The SEMINT system uses neural network learner to compare both schema. SEMINT consider only relational schema, so it could not be applied in XML schema matching [1]. SEMINT did not consider hierarchical structures of schema.

## 3. XML DTD MATCHING USING INLINE ALGORITHM

We tried a new trial to matching hierarchical XML DTD to other hierarchical XML DTD. But we could not directly compare them because XML DTD structures are different case by case though they have similar meaning (Fig. 2). The basic idea of this paper is to transform a XML DTD into some kinds of template that has structural information and integrity constraints extracted from the DTD. In order to directly DTD matching, hierarchical DTD structure is transformed to a *flat feature* array (FFA) structure.

Recently, several transformation algorithms that XML data into relational data, have been proposed [4, 5, 6]. We chose one particular transform algorithm, called the *hybrid inline algorithm* and add constraints properties. Lee [4] proposed semantic knowledge derivation from XML DTD for transforming XML data to relational database. These algorithms will be used to generate a *flat feature array* structure that is template to compare XML DTD.

### 3.1 Transform Hierarchical DTD Structure into Annotated DTDGraph

In this section, converting a XML DTD into annotated DTD graph is shown. Fig. 3 shows a DTD for publication that states a paper element to have four sub-element: title, contact, author and cite in that order. As common in regular expression, zero or one occurrence is represented by the symbol ?, zero or more occurrences is represented by the symbol *, and one or more occurrence is represented by symbol +. Keywords #PCDATA and CDATA are used as string types for elements and attributes, respectively.

A XML document example of the DTD of Fig, 3 is shown in Fig. 4.

```
<DOCTYPE   publication [
<!ELEMENT paper    (title, contact?, author, cite?)>
<!ATTLIST  paper    id ID     #REQUIRED>
<!ELEMENT title     (#PCDATA)>
<!ELEMENT contact   EMPTY>
<!ATTLIST  contact   aid IDREF  #REQUIRED>
<!ELEMENT author    (person+)>
<!ATTLIST  author    id ID     #REQUIRED>
<!ELEMENT person   (name, email?)>
<!ATTLIST  person   id ID     #REQUIRED>
<!ELEMENT name    EMPTY>
<!ATTLIST  name    fn CDATA #IMPILED
                ln CDATA #REQUIRED>
<!ELEMENT email    (#PCDATA)>
<!ELEMENT cite     (paper*)>
<!ATTLIST  cite     pid ID     #REQUIRED
                format (ACM|IEEE) #IMPLIED>
]>
```

Fig. 3. A DTD for the publication

```
<paper id=TR-2003-006    >
   <title>XML Schema Matching</title>
   <contact>aid=    Chulsoo      </contact>
   <author>
        <person id=Chulsoo    >
            <name fn=Chulsoo    ln=Park    />
            <email>clpark@hanmail.net</mail>
        </person>
        <person id=    Chulsoo>
            <name fn=Soonee ln=Hong    />
            <email>soon@yahoo.com</mail>
            </person>
   </author>
</paper>
```

Fig. 4. A XML document for the publication

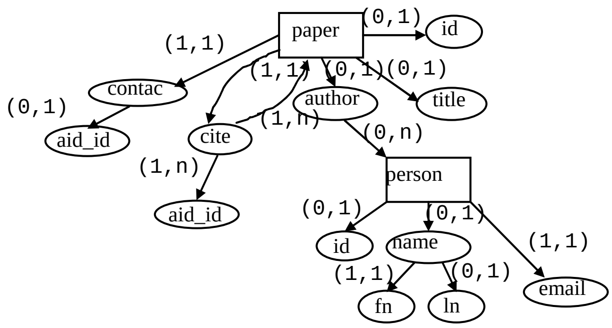


Fig. 5. Annotated DTD for publication in Fig. 3

To make a FFA, hierarchical DTD have to be created an annotated DTD graph that represents the structure of DTD and cardinality relationship type A, B, C or D. Its nodes are element attributes and operators in the DTD. Each element appears exactly once in the graph, while attributes and operators as many times as they appear in the DTD. Attributes with #IMPLIED or IDREFS type are converted to operator node? or + in a DTD graph.

### 3.2 Extracting Semantic Integrity Constraints from DTD

Some types of semantic constraints hidden in DTD are presented in this section. Several papers have been proposed about it [4, 7]. For clear presentation, we chose Lees notation since it exhibited more clear and explicit. There are five possible constraints in DTD such as *cardinality constraints, domain constraints, inclusion dependencies, equality-generating Dependencies (EGDs) and tuple-generating dependencies (TGD)* [4]. We use the cardinality constraints and the domain constraints to generate a FFA. There are four possible cardinality relationships between an elements and its sub-element as illustrated below.

```
<!ELEMENT paper (title, contact?, author+, publisher*)>
```

Let us call each cardinality relationship type A, B, C and D, respectively. We can infer four mapping information from these cardinality relationships.

1. 1-to-{1} mapping (only semantics) : NOT NULL (A Type)
2. 1-to-{0, 1} mapping (at most semantics) : NULLable (B Type)
3. 1-to-{1,…} mapping (at least semantics) : NOT NULL(C Type)
4. 1-to-{0,…} mapping (any semantics) : NULLable(D Type)

Extracting semantic constraints from DTD systematically are described in CPI(Constraints preserving Inline Algorithm) [4]. We show how to represent them in FFA. It can be find semantic constraints from the annotated DTD graph in Fig. 5. We can see not only the relational schema information, but the semantic constraints such as not null, primary key, foreign key or data type.

The *feature value* is normalized form of the structural information and constraints for an attribute. For example, if the feature of an attribute is primary key, non-foreign key, numeric data type, four bytes data type length, non-nullable, then its feature value is (1, 0, 0, 0.2, 0). The feature values element has numeric values from 0 to 1 as [0, 1]. The value range of 0-1 is depends on functions of normalization.

| Attribute | P. Key | F. Key | Data type | Length | Null |
|---|---|---|---|---|---|
| id | yes | no | numeric | 4 | no |
| title | no | no | string | 20 | no |
| person_id | yes | yes | numeric | 4 | yes |
| person_fn | no | no | string | 10 | yes |
| person_ln | no | no | string | 10 | no |
| person_email | no | no | string | 20 | yes |
| cite_aid | no | no | string | 20 | yes |
| contact_aid | no | yes | string | 20 | yes |

| Attribute | Feature value |
|---|---|
| id | (1, 0, 0, 0.2, 0) |
| title | (0, 0, 1, 0.7, 0) |
| person_id | (1, 1, 0, 0.2, 1) |
| person_루 | (0, 0, 1, 0.4, 1) |
| person_ln | (0, 0, 1, 0.4, 0) |
| person_email | (0, 0, 1, 0.7, 1) |
| cite_aid | (0, 0, 1, 0.7, 1) |
| contact_aid | (0, 0, 1, 0.7, 1) |

Fig. 6. Flat feature array for Paper

## 4. FUZZY SIMILARITY MEASURE USING FEATURE COMPARISON

The *flat feature array(FFA)* includes all the properties of DTD for matching other DTD. The process of FFA comparison is shown in Figure 7. There are two types of FFA. One is *simple type FFA* and the other is *clustering type FFA*. The simple type FFA consists of information for primary key, non-foreign key, numeric data type, four bytes data type length, non-nullable of an attribute. The clustering type FFA is combining the feature values of several clustering attributes. The similarity rate of clustering type FFA can be calculated following formula *Sim(M, S)* [10]. It represent a similarity or likeness between a attribute and clustered attribute.

We can see the extent of similarity between *mediated schema* (M_schema) and *source schema* (S_schema) is high such as 0.92, 0.90 and 1.0. If the value of Sim(M, S) is 1.0, then the both attributes have the same meaning.

In this example, Mediated_Schemas house_addr has five attributes as house_number, street, city, state and zip_code. But, on the other hand, S_Schemas house_location attribute is single. So we cant compare the similarity directly. First of all, clustering feature value of the five attributes should be obtained. Then we can find similary between the clustering attribute house_addr and the single attribute house_location.

The similarity rate between M_Schemas house_addr and S_Schemas house_location is calculated by Kims method as following [8, 9].

$$\mathrm{Sim}(M, S) = 1 - \frac{\sum |\text{feature value}(M) - \text{feature value}(S)|}{|X|}$$

Here, M and S means M_Schema and S_Schema respectively. |X| is number of feature value's element. Following expression is a similarity between M_Schema and S_Schema.

Sim (contact_phone, contact_info)

$$= 1 - \frac{|0-0| + |0-0| + |1-1| + |0.4-0.4| + |1-0.5|}{|5|}$$

$$= 0.90$$

SIM (house, HOUSE) ≈ 0.94
(similarity between M_Schema and S_Schema)

## 5. CONTRIBUTION AND FURTHER WORKS

In this paper, we have presented the first *hierarchial matching of XML DTD* that provides practical assistance in finding equivalent schema between mediated schema and several different source schemas. XML DTD is usually complex since its structure is hierarchical and nested. Thus, matching these schemas directly is laborious and error prune job. So most of all the previous works were 1-to-1 matching. The proposed method makes a hierarchical DTD structure to be flattened and then it captures not only schematic information but also integrity constraints information of DTD. These information makes effective schema matching processing. The first contribution of this work is the complex schema matching not 1-to-1 matching. Next contribution is to use constraints information for more accurate comparison.

There are still rooms for improvement extracting semantic information from DTD and experiments to evaluate the feasibility of our approach. In the near future we would like to explore these issues.

## ACKKNOWLWDGEMENTS

This work was supported by grant (04-Basic-018) from the Basic Research Program supported by Ministry of Information and Communication and (R01-2002-000-00068-0) from the Basic Research Program of the Korea Science & Engineering Foundation in Republic of Korea.

## REFERNCES

[1] W. Li and C. Clifton, SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks, *Data & Knowlwdge Engineering* 33, pp. 49-84, 2000.

[2] T. Milo and S. Zohar, Using Schema Matching to Simplify Heterogeneous Data Translation , *Proc. on VLDB*, New York, USA, pp. 122-133, 1998.

[3] A. Doan, P. Domingos and A. Levy, Learning Source Descriptions for Data Integration, *Proc. on WebDB 2000*, pp. 81-86, 2000.

[4] D. Lee and W. Chu, Constraints-preserving Transforma -tion from XML Document Type Definition to Relational Schema, UCLA-CS-TR-200001, 2001.

[5] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. Dewitt and J. Naughton, Relational Databases for Querying XML Documents: Limitations and Opportunities, *Proc. on VLDB*, Edinburgh, Scotland, pp. 302 314, 1999.

[6] A. Deutsch, M. Fernandez and D. Suciu, Storing Semistructured Data with STORED, *Proc. on SIGMOD*, Philadelphia, USA., 1998.
[7] P. Wood, Optimizing Web Queries Using Document Type Definitions, *Proc. Intl Workshop on Web Information and Data Management*, pp. 1-5, 1999.
[8] C. S. Kim, Systematic Generation Method and Efficient Representation of Proximity Relations for Fuzzy Relational Database Systems, *Proc. of the 20th EUROMICRO Conference*, Liverpool, England, IEEE Computer Society Press, pp.549-555, 1994.
[9] C. S. Kim, A Complex Matching of XML Schema, *Proc. of the 20th International Conferenceon Internet Computing*, Las Vegas, Nevada, U. S. A., CSREA Press, pp.484-489, 2002.

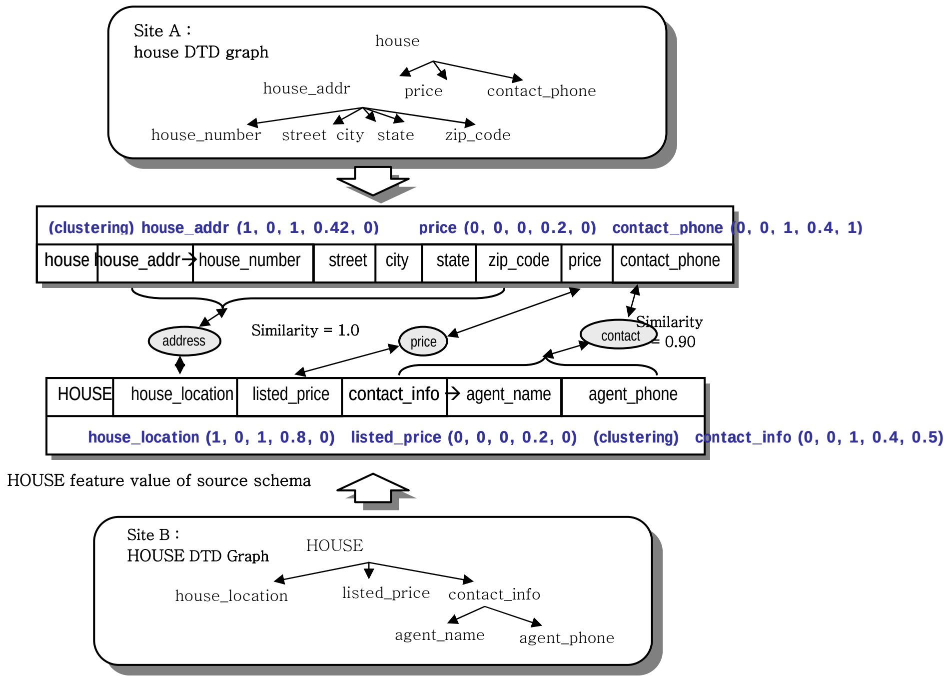


Fig. 7. The process of FFA comparison between M_schema and S_Schema