A Study on Self-adaptive Heterogeneous Data Integration Systems

Yan Cao, Yan Chen and Buyuan Jiang

College of Economics and Management, Dalian Maritime University, Dalian 116026, P.R. China caoy123@163.com chenyan_dlmu@163.com sophyblaze@hotmail.com

Abstract. Along with the rapid development of Internet and the extensive use of information technology in various fields, large amount of heterogeneous data has been produced. The way of integrating these heterogeneous data has been a hot issue. In this paper, from the problems which enterprise information integration faced, a framework of self-adaptive heterogeneous data integration system (AHDIS) has been given, and by using ontology, semantic similarity, web service and XML techniques, a self-adapted heterogeneous data integration platform which can be dynamically regulated has been built up successfully. With this integrated platform, the global data model can automatically or semi-automatically be adjusted while a change has been made to the local data model, so that it can realize the data sharing among heterogeneous data sources.

Keywords: *Data integration, AHDIS, Schema mapping, Semantic similarity, Web Services*

1. INTRODUCTION

In the process of information construction, the enterprise has constructed many information systems to manage the enterprise data. For differences in business, functions and phases of information system construction of every department etc, enterprise's internal data has obvious distributives, autonomy and the heterogeneity (platform, application, data format and semantic heterogeneity). However in many situations, enterprise needs sharing information among several applications more and more which is distributed in different positions in the network to improve their operating efficiency and provide the support for high-level consolidated decision. Therefore, it's imperative to establish the integration system. A good integration system should not only meet the needs for existing application, but also have good extensibility; enterprise's future application system should be able to add to the integration system conveniently, the adaptive heterogeneous data integration system is designed for the need.

Self-adaptive Heterogeneous Data Integration System--AHDIS, means to complete the adjustment of global data pattern automatically or semi-automatically when heterogeneous local data pattern changes and it can enable the system to continue steady operation. In this paper, we bring forward a self-adaptive heterogeneous data

Please use the following format when citing this chapter:

Cao, Y., Chen, Y., Jiang, B., 2007, in IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer), pp. 65-74.

integration model based on correlative technology, using XML Schema to express the data pattern of heterogeneous data source; Using correlative technology such as ontology to solve semantic heterogeneity; Using Web Service to solve mutual operation between heterogeneous systems, and realize actual operation to the data source; completing construction and modification to the heterogeneous data integration system automatically or semi-automatically.

2. HETEROGENEOUS DATA INTEGRATION METHOD

Schema mapping is a key technology in realizing the heterogeneous data source integration, it usually takes more than half the efforts to produce schema mapping during the process of integration, and may cause mapping redefinition when the data source pattern changes. Because of the increasing complexity of the data source's local schema or the integration system's global schema, manual and detailed definition of schema mapping becomes the biggest bottleneck in realizing integration systems. Therefore, reducing manual participation as far as possible and intensifying automation of Schema mapping become the universally- pursued goal.

Schema mapping mainly uses form definitions, such as GAV (Global As View), LAV (Local As View), GLAV etc. Global schema in GAV is established basing on the data view of data source, it's made up of a series of elements; each element corresponds to an query of data source which shows data structure and operation of corresponding data source. In LAV, the global schema is firstly constructed and the data view of data source is defined on the basis of it, and obtained by the global schema according to inferring with certain rules. GLAV is the united product of GAV and LAV, made by uniting goal pattern view and the source pattern view, so it combines the above two's advantages or has the higher expression ability. Integration system with the GAV mapping description deals with inquiry through the unfolding technology, so though its efficiency is higher, its expansibility is bad and it's unsuitable for application of the data source's dynamic changing; Integration system with the LAV mapping description deals with query through the rewriting technology, its complexity is higher but expansibility is better. In this paper, we absorb the advantages of GAV, to the disadvantages of GAV, we solve semantic heterogeneity with correlative technology such as ontology etc; Use Web Service to solve mutual operation between heterogeneous systems, and then realize actual operation to the data source; complete construction and modification to the heterogeneous data integration system automatically or semi-automatically, improve expansibility of the entire heterogeneous data integration system.

3. THE SYSTEM STRUCTURE OF AHDIS

In the heterogeneous data integration system, data schemas of each local data source are all established at the different time by different users according to different needs, so there may be kinds of differences and conflicts between them. In order to

realize users' transparent visit to multi-data sources system, it needs to establish a global layer in the integration system to shield these differences and conflicts. In the heterogeneous data integration system, the global schema constitutes a virtual database. The global schema is visited by users, but their actual data needs to be obtained from each local data source.

In order to establish mapping from the global schema to the local schema in the integration system, the following problems must be solved:

(1) To seek one kind of unified method to express each local data schema,
(2) To establishing a common data model of integration system,
(3) To transform users' inquiry of the global schema to one or multi-sub inquires.

If heterogeneous data integration system of GAV pattern changes in the local schema, its global schema can realize auto-adapted adjustment in certain degree, following functions are needed:

(1) The system has the function of monitoring local schema's changes,
(2) The system has the function of adjusting the global schema according to the local schema's changes.

Therefore, the auto-adapted heterogeneous data integration system, ought to be able to provide the matching algorithm for users, enables user to complete the mapping of synonymous elements between each data source and establish the global schema conveniently (provide the keys to the unified visit connection to outsiders). The operation to data is through invoking Web service, and completed by each data source's subsystem. The self-adaptive heterogeneous data integration system can modify data source and invoke its corresponding Web service conveniently. General structure for Heterogeneous data integration is shown in figure 1:

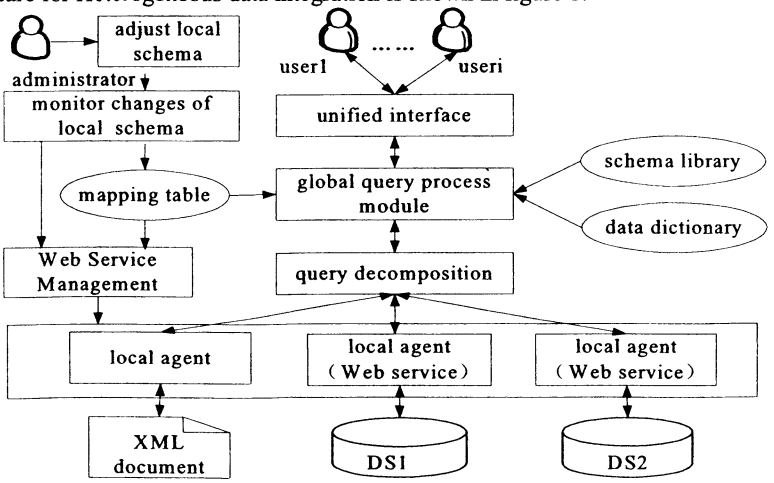

Figure 1. The Framework of Self-Adaptive Heterogeneous Data Integration System

It mainly includes the following several parts: data dictionary, common data model, decomposition of query, local pattern change monitor and Web Service management etc.

4. THE DESIGN OF AHDIS FRAMEWORK

4.1 Data Dictionary

This paper manages data source by establishing data dictionary for it, which is convenient to establish the mapping. And it divides metadata which describes the data source into three kinds, which is schema information, position (navigation) information and other corresponding information [1-2].

Using XML Schema sets modeling for metadata of the data source, establishes metadata dictionary (MDD) of the data source. Because the heterogeneous data integration system usually has many heterogeneous data sources, we let each data source firstly register in MDD before integrating, describing the position and type of the data source, and providing detailed schema structure and semantic information. Metadata descriptions of many data sources compose MDD of data source. Mapping rules between relational data model and XML Schema are as follows:

- Tables of relational data model correspond to ComplexType complex types of XML Schema;
- Each field in a table is mapped to an attribute or sub-element of ComplexType type;
- Attributes or elements which are mapped by table's primary keys are defined as key attributes, mapped by foreign keys are defined as keyref attributes;
- Create sub-elements based on relations of primary keys and foreign keys in a table. If foreign keys of a table are as primary keys or a part of primary keys in another table, then the table with the identical field as foreign key is mapped to the father element, while the other table is mapped to the sub-element [3].

4.2 Common Data Model

In order to solve heterogeneity of each member's system data model in the integration system, the system which integrates many data models has to provide a mapping for concepts in a model to another model, the most common method is to provide a Common Data Model(CDM), each member model is mapped on CDM. Choosing common data model and common data language generally follows the two principles [4]:

- The common data model and common data language should be easily converted with the member database system data model and data language. This requests that common data model should be as simple as possible.
- The common data model and common data language should be able to conveniently express the data and treating process in integration system, and support dealings with the structural and semi- structural data.

In the realization process, global schema in the integration system is expressed with the global ontology, here global ontology can be understood as a sharing glossary storehouse, and uses XML Schema as the description language of common

data model (CDM), is used to describe the inner structure of data source, determine mapping from the integrated pattern to each data source's local schema, and transform queries which are based on the integrated pattern to sub-queries of each data source.

The naming of heterogeneity is the main reason for the semantic conflicts between patterns, we can solve synonym problem through assigning a unified name for field information of each data source in CDM, and renaming in CDM. Take the vehicle license plate number for example, use vehicle_no to express in one data source but vehicle_num in another one, use vehicle_no to express uniformly in CDM, <Field name= "vehicle_num" type= "String" > vehicle_no </Field>.

4.3 Query Decomposition

In the heterogeneous data integration system, users can operate global schema directly, therefore system's query processing function need automatically realize transformation from global query to sub-query of each local data source transformation. System's query processing generally includes several following stages: standardization of query, query decomposition, query transformation and result synthesis. This paper uses XQuery language to query the global schema, query processing module transforms query of global schema to one or multi-sub-query. If local data source is XML documents, it can directly return to the child result, if the local data source is the relational database, it firstly transforms XQuery into SQL sentence.

4.3.1 Algorithm of Query Decomposition

The purpose of query decomposition is decomposing global query which involves many data sources to a group of local sub-query operated on single data source. Global query decomposition should follow the following principles [5-7]):

(1) Decomposition of global query take data source as a unit, and decompose query of one local data source in one sentence, each local sub-query can only involve the object of one local data source.

(2) Nesting query in global query should be decomposed and then be distributed to each sub-query before executing.

(3) Query conditions should be decomposed to sub-query according to the mapping.

According to query decomposition principles, here we give steps of query decomposition algorithm as follows:

- Take out the correlative mapping MapList from the mapping table according to the 'for' sentence.
- Traverse MapList according to conditional expression in the where sentence, establish a where sub-sentence for each mapping, and add to sub-query subsets Qs.
- Confirm the 'for' sentence of sub-query according to the where sub-sentence of each sub-query.
- Search the mapping in MapList for each return element in the return sentence. To each mapping glossary, if its data source section and data table section match with the 'for' sentence of sub-query in Qs, then establish the return sub-sentence and add it to the sub-query sentence.
- Transform XQuery to sub-query to SQL which is for local data source.

For example, one company uses two kinds of GPS vehicles monitoring system during different periods, the vehicle marked with "SB3327" have used different system successively, the vehicle's location information is stored in different systems, XQuery sentences for querying the vehicle's whole localization information through the integration system are as follows:

for $i in document("GobalDB.xml")//vehicle
where $i//vehicleno="SB3327"
return $i//mobileno, $i//timegps, $i//vehiclelatitude, $i//vehiclelongitude

Fragments of XML pattern structure for two data sources and its ontology structure are shown in figure 2.

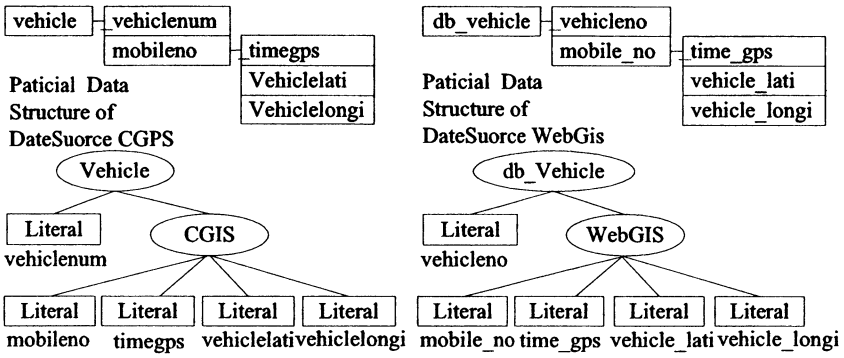

Figure 2. XML Schema Structure for Part of Data Source and their Ontology Structure

The steps of query decomposition are as follows: according to the mapping, we establish sub-query: the "where sentence", then according to sub-query: the where sentence, we establish sub-query: the "for sentence", according to the return sentence and the maplist, we get the sub-query after reorganizing:

for $i in document("WebGIS.xml")//db_vehicle where $i//vehicleno="SB3327"
return $i//mobile_no,$i//time_gps,$i//vehicle_lati,$i//vehicle_longi;
for $i in document("CGPS.xml")//vehicle where $i//vehiclenum="SB3327"
return $i//mobileno,$i//timegps,$i//vehiclelati,$i//vehiclelongi;

According to data dictionary, we transform it to the SQL sentence:

Select mobile_no,time_gps,vehicle_lati,vehicle_longi
from db_gps_normal where mobile_no in select mobile_no
from db_vehicle where vehicleno="SB3327";
Select mobileno,timegps,vehiclelati,vehiclelongi
from gpsinform where mobileno in select mobileno
from vehicle where vehiclenum="SB3327"

4.4 Inspection of Local Schema's Change

The adaptability of heterogeneous data integration system means: in the process of integration system construction, when we use GAV to pattern integration, once the

local schema changes, global schema can realize auto-adapted adjustment in certain degree. When the data schema of Data Source changes, the manager submits the changes of local schema to the AHDIS, it makes some adjustments according to the changes. The flow is shown in figure 3. It mainly completes the pattern mapping.

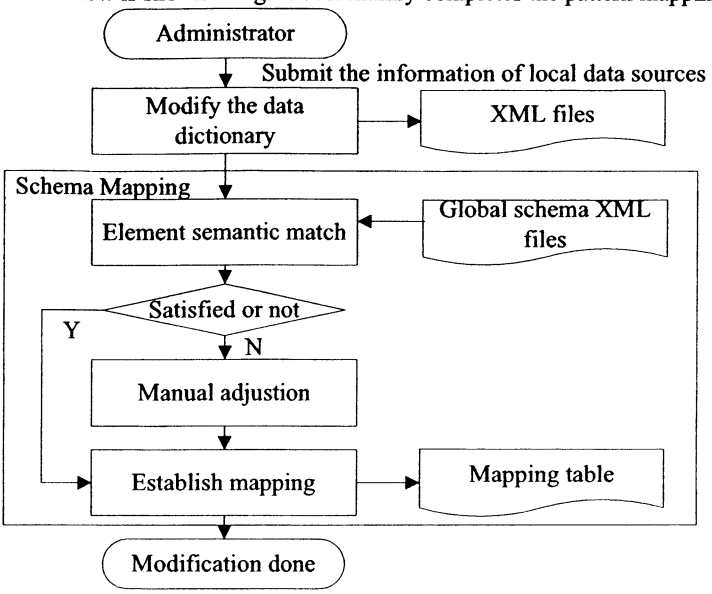

Figure 3. Schema Modify Flowchart

4.4.1 Schema Mapping

The schema mapping can be divided into two steps: schema matching and mapping generation. The target of schema matching is to discover the relations among the schema elements; the goal of the latter is to create logic expressions of equality and containing which are in accordance with schema semantic restriction among element gatherings.

To realize the automatic regulation of the global schema, firstly we need to identify the relations between the changing schema elements of local schema and the elements of global, and determine whether there is any difference on the semantics (for there is one mapping between every pair of the same semantic elements in the local and global schema) and structure, and then process accordingly. This paper can complete the schema mapping better by calculating the similarity of the semantics and description.

4.4.1.1 Schema Matching

(1) The semantic similarity: in the research of the Linguistics, the distance between words is an important relationship between them. It's a real number in [0, $+\infty$] [8]. The distance from one word to itself is 0. Generally speaking, there is a tight

relationship between the distance and the semantic similarity. The distance between two phrases is longer, their similarity is lower, and vice versa. The similarity functions to definite objects x and y are below:

- sim(x, y)∈[0..1];
- sim(x, y) =1, then x=y, the two objects are equivalent;
- sim(x, y)=0, it represents that there is no common characteristic between the two objects;
- sim(x, y) = sim(y, x), it represents that the two objects' similarity is symmetrical.
- sim(x, y) ≥λ, λ∈[0..1], if the similarity of the objects is equal to or above λ, they are similitude.

(2) Semantic similarity computation: in the foundation of paper [8], similarity computational method of the newly-added data source concepts and global pattern concepts in this article is: obtain by attribute semantic similarity and attribute description similarity weighted array. Princeton University's WordNet is a tree shape English semantics dictionary. In the tree diagram the distance between two leaves are two concepts' semantic distance, and semantic similarity can be further obtained by the semantic distance. In this foundation, two words and expressions C_1 and C_2 similarity can be recorded as $Sim(C_1, C_2)$, and their distance can be recorded as $Dis(C_1, C_2)$. Then their semantic similarity can be obtained through formula (1):

$$Sim(C_1, C_2) = \frac{\alpha(l_1 + l_2)}{(Dis(C_1, C_2) + \alpha) \times 2 \times maxl \times \max(|l_1 - l_2|)} \quad (1)$$

In the formula, l_1 and l_2 are the layers which C_1 and C_2 locate, α is the parameter which can be adjusted and ordinary α>0. $Dis(C_1, C_2)$ is the ontology tree's most short-path between concept C_1 and C_2, maxl refers to the text trees' greatest depth. Elimination here by this parameter makes it convenient to normalize computation results.

The practice indicated that it's available to obtain good similarity by comparing the regular words using WordNet. However, in the data integration system, the concept (namely field name) is often the irregular abbreviations for example: vehicle_no, carNo and so on. In view of this kind of situation, the results of the match are not very ideal. To solve this problem, we may define the sharing glossary storehouse in advance. For example: in the actual situation, the database design of the logistics information system, here is the sharing glossary listed in table 1.

Table 1. Sharing Words Table

Target Words	Sharing Words
vehicle_no	chepai , chepaiNo, carNo, vehicleno
driver_no	sijiNo, driNum , driverno
company_no	group_no
mobile_no	Sim, telno
...	...

(3) The description similarity computation: in data integration, description similarity is calculated through the description information of its attributes, for

example: data type, data length, key or not and whether it is allowed null or not. In this way, the attributes can be regarded as a vector, and each description of the attributes is a characteristic vector of the vector such as s1 (data type, data length, key or not and whether it is allowed null or not). After separate calculation of each characteristic vector's similarity of the two vectors, we can obtain the weighted average as these two vectors' similarity. It is showed as formula (2):

$$\mathrm{Sim}(s_1, s_2) = \sum_{i=1}^{sum} W_i \mathrm{Sim}_i(s_1, s_2), \qquad \sum_{i=1}^{sum} W_i = 1, \quad W_i \in [0,1] \tag{2}$$

In the formula, Simi(s_1, s_2) represents the similarity of s_1 and s_2's characteristic vector numbered i and W_i is their weight, sum represents the number of the characteristic vectors.

The similarity which is obtained through the two computation methods can be respectively recorded as LSim, DSim. Finally the semantic similarity of concept is calculated by the formula Wsim(s_1, s_2)=Wdesc Dsim+(1-Wdesc)Lsim. Wdesc shows the relative importance of semantic similarity and description similarity.

4.4.1.2 The Mapping Output

The system can set the weight value Wdesc and the threshold value λ which are used in the semantic match calculation. If the similarity is bigger than λ, the two vocabularies are regarded to have the same semantics. The system establishes the direct mapping between the local schema and global schema to the concepts with same semantics. If there is no semantic match, it's necessary to add into the global schema. Then the system will establish the schema mapping. Tables are used in the system to display matching results. It also can realize automatic matching in certain degree. Because the semantic matching is an extremely complex process, the matching precision is influenced by kinds of factors; therefore, the users can realize the result revision manually.

4.5 Management of Web Service

In the AHDIS model of this article, the actual operations to the data sources are completed by different local systems distributed in various regions. When the system decomposes the inquiries, it can transform the XQueries of the sub-inquiries to SQL sentences which operate to the local data sources. Then we encapsulate these data operations to local data sources in Web Service. In this way, data processing can be realized in the system which provides Web Service through the long-distance transfer. And the maintenance of the transferred Web Service such as addition, deletion and modification will be carried on by the management of Web Service. The main work of Web Service management is to manage the Web service naming, the transfer ways and some simple semantic description information.

Usually when we need to transfer Web Service in our programs, we "insert the Web quotation", and then the VS.NET environment will generate service proxy for us, and transfer corresponding Web service. This measure can have certain limitations: when the Web service's physical location has changed, the client code

must be modified, otherwise the transfer will fail. Considering this, we need the ability to transfer Web Service dynamically. We may preserve the Web Service URL in the configuration document (a kind of Web Service information documents). We only need to modify the configuration document correspondingly when the service URL changes.

5. SUMMARY AND FUTURE WORK

The AHDIS model in this paper can simplify the foundation process of the heterogeneous data integration system to some extent. The realization of automatic or semiautomatic foundation of the system will provide much convenience to modify, add and delete the data sources dynamically in it. As the operations to data in this scheme are completed by each local data source, the consistency and real-time of data will be well ensured. Based on the semantic information acquired, the conception similarity can be further divided into two parts-semantic similarity and description similarity. By calculating the semantic similarity from various angles, the precision of the semantic matching will be improved. Although this paper has resolved some problems in the semiautomatic foundation of the heterogeneous data integration system, there are still much to be further researched in future: improve and perfect the concept similarity model to calculate the concepts' similarity better, thus enhance the accuracy rate of the output mapping relations.

REFERENCES

1. S. Zhang, H. Li, and Z. Lu, MetaData Management Model Design in WEB Data Integration System, *Computer Engineering and Applications*. Number 21, pp.189-191, (2005).
2. J. Song, W. Zhang, W. Xiao, and G. Li, Research on Metadata Based Heterogeneous Data Management in the Same Domain, *Computer Engineering and Applications*. Number 14, pp.168-171, (2005).
3. H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, *XML schema part 1: structures*, W3C Recommendation (2001). http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/
4. R. Li, *Query Processing and Optimization in Heterogeneous Information Integration*. Ph.D Thesis, Huazhong University of sience and Technology (2004).
5. N. Wang and N. Wang, Query Decomposition and Optimization in Heterogeneous Data Integration System, *Journal of Software*. Volume 11, Number 2, pp.222-228, (2000).
6. R. Li, Z. Lu, W. Wu, and W. Xiao, A Study of Algorithm on Query Decomposition in Mutidatabase Systems, *Mini-micro Systems*. Volume 22, Number 4, pp.488-491, (2001).
7. X. Chen, Z. Pan, and Q. Zhao, A Schema-Reusable Method on Heterogenous Databases Access and Integration in Grid Environment, *Journal of Software*. Volume 17, Number 11, pp.8-17, (2006).
8. Q. Liu and S. Li, Word Similarity Computing Based on Hownet, *Computational Linguistics and Chinese Language Processing*. Volume 7, Number 2, pp.59-76, (2002).