A COMPOSITE APPROACH TO AUTOMATING XML SCHEMA MATCHING FOR SCHEMA INTEGRATION

MOHAMMED HASAN MOHAMMAD AL-GHANIM

THESIS SUBMITTED IN FULFILMENT OF THE DEGREE OF
DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2013

PENDEKATAN KOMPOSIT DALAM PENGAUTOMASIAN PADANAN SKEMA XML UNTUK PENGINTEGRASIAN SKEMA

MOHAMMED HASAN MOHAMMAD AL-GHANIM

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH IJAZAH
DOKTOR FALSAFAH

FAKULTI TEKNOLOGI SIANS DAN MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2013

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

5 July 2013 MOHAMMED HASAN MOHAMMAD AL-GHANIM

P31598

ACKNOWLEDGMENTS

First and foremost praise be to Almighty Allah for all his blessings for giving me patience and guidance throughout the duration of this PhD research.

I would like to thank Professor Dr. Shahrul Azman Moh'd Noah for his efforts and sharing his experience as a research supervisor.

Also, I would like to thank all people and colleagues, how helped me throughout my research.

To my beloved father, Hajj Hasan; I seek Allah's obedience by dedicating this thesis to him. I hope Allah rewards hem for this work, as he sacrificed to afford us a better life. I am sending my attributes to him while he is laid down to rest, may Allah cast continuous mercy upon him. I ask Allah to gather us altogether in paradise.

To my grate mother, Ezzyyah; who enclosed me with her continuous care and fond. Her never-ending prayers to Allah were opening the closed ways in front of me. I am asking Allah to length her life time in healthiness and happiness.

To my dearest wife, Fatemah, for her boundless patience and support along the years of this PhD. I ask Allah to reward her best for her time spent, caring after us; me and our children.

Last but not least, I would not forget to thank, from my deepest heart, my brothers and sisters in my family for their unlimited support. Surely, this research would not be completed without their generous sustain. May Allah bless them all.

ABSTRACT

One of the long-standing and challenging problems in extensible markup language (XML) data integration is schema matching. Schema matching is the task of identifying semantic similarities between two different schemas. Automated schema matching is required to minimize the labor-intensive work of data integration. Many schema-matching algorithms, such as the QMatch and Cupid algorithms, have been proposed to tackle this problem. However, most of these do not tackle indirect schema mappings, and when they do, they discard special cases of XML schemas by treating them as graph structures. As a result, the problem on accurate automatic matching remains unresolved. A composite approach called the automatic XML matcher (AXM) is hereby proposed to solve this problem. AXM includes structural and element mapping and accounts for a domain ontology snippet, which is primarily extracted from data extraction activities, and describes the relationships among a collection of objects in a specified domain. Enhanced domain ontology snippets are developed to obtain a sophisticated domain ontology that fits an XML schema structure. The approach automatically detects direct and indirect mappings for XML schemas and more kinds of mappings mentioned above. The proposed approach to automate XML schema matching combines semantic similarity techniques and data value characteristics with XML element property mapping for element-level matching. The algorithm depends on two primary axes of XML schemas, namely, label and properties. The label matcher includes a name matcher, a value characteristic matcher, and a data value matcher. The property matcher involves the structural aspects of XML. The XML structural and element mappings are applied within a well-defined XML match taxonomy. The proposed approach is then tested and evaluated to assess its effectiveness using a set of test cases. Results revealed that precision and recall measures were enhanced compared with those of five other algorithms, including the QMatch and Cupid algorithms.

PENDEKATAN KOMPOSIT DALAM PENGAUTOMASIAN PADANAN SKEMA XML UNTUK PENGINTEGRASIAN SKEMA

ABSTRAK

Pemadanan skema merupakan salah satu permasalahan penting yang masih berlarutan dan perlu diatasi dalam bidang integrasi data XML (extensible markup language). Pemadanan skema ialah suatu tugasan untuk mengenalpasti persamaan semantik antara dua skema yang berbeza. Pemadanan skema secara automatik amat diperlukan bagi mengurangkan bebanan kerja bagi pelaksanaan proses pengintegrasian data. Beberapa algoritma pemadanan skema seperti QMatch dan Cupid telah pun diketengahkan bagi menyelesaikan masalah berkaitan pengintegrasian data ini. Walaubagaimanapun, kebanyakan algoritma yang diperkenalkan ini tidak mencangkupi pemadan skema secara tidak langsung. Bahkan, algoritma tersebut mengabaikan keadaan khusus skema XML dengan mengandaikan skema XML sebagai struktur graf (graph structures). Sebagai kesannya, hasrat untuk menghasilkan pemadanan skema automatik yang tepat masih tinggal tanpa penyelesaian. Sehubungan itu, suatu pendekatan bersepadu yang dinamakan sebagai Automatic XML Matcher (AXM) telah diusulkan dalam kajian ini. AXM meliputi pemadanan struktur dan elemen serta melibatkan penggunaan ontologi domain ringkas. Ontologi ringkas ini diekstrak daripada aktiviti pengekstrakan data dan menyatakan hubungan antara sekumpulan objek dalam sesuatu domain khusus. Ontologi domain yang telah melalui proses penambahbaikan dibangunkan bagi mendapatkan ontologi yang lebih canggih bersesuaian dengan struktur skema XML. Pendekatan AXM ini mengesan pemetaan langsung dan tidak langsung untuk skema XML secara automatik dan juga mampu untuk mengesan lebih banyak padanan seperti yang dinyatakan sebelum ini. Algoritma pemadanan skema secara automatik yang diperkenalkan ini menggabungkan teknik kesamaan semantik dan sifat-sifat nilai data dengan pemetaan properti elemen XML untuk pemadanan diperingkat elemen. Algoritma yang dicipta bergantung kepada dua paksi utama skema XML iaitu label dan properti. Pemadanan label meliputi pemadanan nama, sifat-sifat nilai dan nilai data. Pemadanan properti melibatkan aspek struktur XML. Pemetaan struktur dan elemen XML dilaksanakan pada taksonomi padanan XML yang ditaktrif secara jelas. Pendekatan yang dicadangkan dalam kajian ini diuji terhadap satu set ujian untuk menilai tahap keberkesanannya. Melalui perbandingan yang dibuat antara algoritma yang diperkenalkan ini dengan lima algoritma yang lain (termasuk QMatch dan Cupid), keputusan ujian membuktikan terdapat peningkatan pada ukuran kejituan dan dapatan semula.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API: Application Programming Interface

AXM: Automatic XML Matcher

DBR: Design-Based Research

ER-Model: Entity Relationship Model

GAV: Global-As-View

GLAV: Global-Local-As-View

LAV: Local-As-View

OO-Model: Object Oriented Model

OSM: Object-Oriented Systems Model

OSML: Object-Oriented Systems Model Language

OSMX: Object-Oriented Systems Model XML

RAD: Rapid Application Development

XML: eXtensible Markup Language

CHAPTER I

INTRODUCTION

1.1 RESEARCH OVERVIEW

The amount of accessible information over the web has increased in recent years, due to the emergence of hyperlinked networks. Each source has its own set of concepts, semantics, data formats and access methods, indicating a very wide differentiation in terms of kind and structure. Achieving transparency in these heterogeneous data sources remains a critical problem for many domains, and this is a problem that should be addressed immediately. Data integration refers to the problem of combining data located at different sources. Most jobs that are conducted manually via programs convert between data formats, resolve conflicts, integrate data, and interpret results to utilize such information (Bernstein et al. 2001). Many studies on data integration have been conducted. These studies aim to have a fully automated data integration system with full interoperability among these different data sources.

In cases where the extensible markup language (XML) standard is widely accepted, the number of XML documents on the web has grown in recent years, making the comparison of the web to a "database" closer to reality than ever before. Therefore, the demand for a fast and efficient querying to obtain the desired information is becoming increasingly crucial.

1.1.1 Data integration systems

The traditional approach to data integration is to create a global schema over a set of data sources without changing the data in these sources. The data integration system translates queries on the global schema into original data sources (Hull & Zhou 1996),

as shown in Figure 1.1. This system depends on two primary concepts, namely, wrappers and mediators. A wrapper is a program that translates data to a form deemed usable by the query processor in the data integration system, whereas a mediator generates and maintains mappings between sources and global schemas.

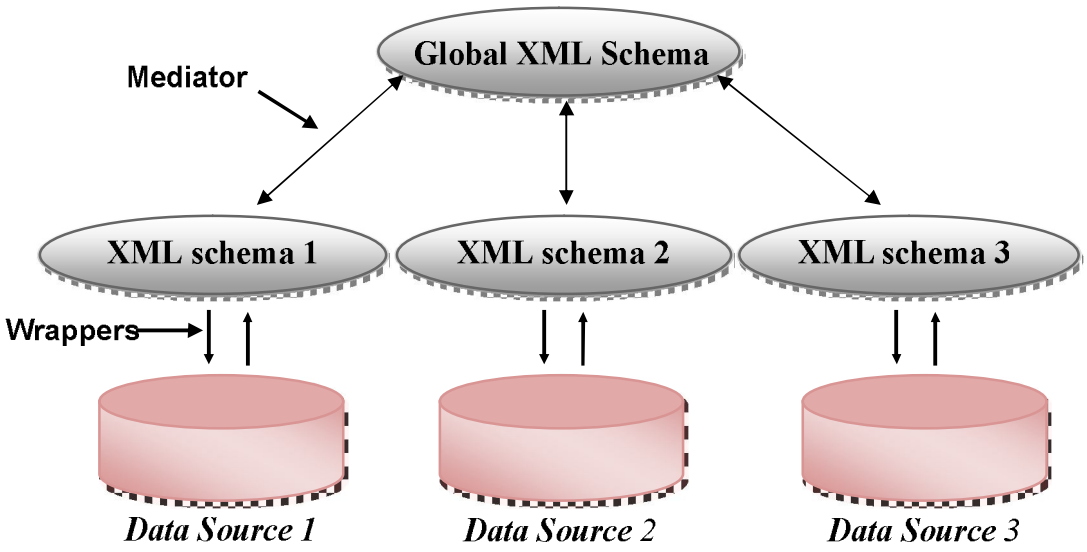

Figure 1.1 XML Data Integration Architecture

Two approaches have been proposed, namely, local-as-view (LAV) and global-as-view (GAV) (Ullman 2000). In the GAV approach, a query exists over the source relations for each relation in the global schema. However, for each data source in LAV, there exists a rule over relations in the global schema, which describes what tuples are found in this data source. Meanwhile, (Embley et al. 2004) proposed the target-based approach, in which the target schema in the global schema is matched with the schemas in the data sources in the global schema. In this sense, schema matching can be automated to obtain higher accuracy in generated mappings. Many studies (Domenig & Dittrich 2000; Och et al. 2000; Draper et al. 2001; Chukmol et al. 2005; Raghavan et al. 2005; Mork et al. 2008; Assaf et al. 2012) have attempted to solve this problem, although they generally did not fully automate the matching. Moreover, even if they did, the results still required enhancement, as mentioned in several surveys (Bernstein et al. 2011; Shvaiko & Euzenat 2012).

1.1.2 Schema matching

One of the long-standing problems in data integration systems is schema matching, which is the task of identifying semantic similarities between two different schemas. Clearly, the manual approach to schema matching is burdensome, tough, time-consuming, error-prone, and costly. As a result, there is a need to automate schema matching in order to reduce the labor-intensive work of data integration systems.

Rahm and Bernstein (2001) conducted a comprehensive survey, in which the problem of automating schema matching was studied and analyzed. This survey draws the primary lines in schema matching research, as the researchers provided a good principal classification of schema matching in general. Thus, this research is based on the definition, classifications, and conclusions of schema matching, which proposed in the mentioned survey.

Rahm and Bernstein (2011) also investigated the emergent challenges in schema matching research since their original survey (Rahm & Bernstein 2001), as well as the latest techniques in automating schema matching. Finally, the survey described new applications for schema matching.

A match is a function that takes two schemas as inputs and provides the mappings (similarities between two elements) between them. This match is expressed by mapping expressions that describe the ways by which the similarities between the two schemas are related. More than one matching algorithm can be used in the schema matching process depending on the domain of the application. Thus, matchers are classified into single criterion matchers and a combination of these single matchers as follows: a hybrid matcher, integrated single matchers, or a composite matcher (i.e., these combine multiple match results). Single matchers can primarily be classified into the following: (1) schema level versus instance level (schema-level metadata versus data contents) and (2) element level versus structure level (individual elements versus structural similarities).

Many algorithms have addressed the automatic schema matching problem. Most of these investigated one-to-one mapping cardinality (denoted by 1:1), i.e., the

mappings between two individual elements in different schemas. (Doan et al. 2001; Madhavan et al. 2001), and other studies are examples of works that detected 1:1 mappings, as mentioned in surveys (Rahm & Bernstein 2001; Bernstein et al. 2011; Shvaiko & Euzenat 2012). The 1:1 mapping cardinality is also called direct mapping (Embley et al. 2004; Xu et al. 2006). However, techniques that detect only direct mappings do not tackle the whole problem of schema matching, because different schemas have varied element semantics and schematic structures. For example, the address attribute in one schema may be mapped to the other attributes (street, city, and country) in another schema. This mapping is called one-to-many mapping cardinality (denoted by 1:m).

Generally, in real-life schemas, a number of attributes in one schema (n) relate to a number of attributes in another schema (m); this cardinality is called many-to-many mapping cardinality (denoted by n:m). Several researchers refer to many-to-many cardinality as indirect schema matching (Embley et al. 2004; Xu & Embley 2006). Some works have studied n:m mappings (Do & Rahm 2002; Dhamankar et al. 2004; Embley et al. 2004; Engmann & Massmann 2007; Thang & Nam 2010); however, the results of the automatic schema matching process still require enhancement (Bernstein et al. 2011; Shvaiko & Euzenat 2012).

1.1.3 XML and interoperability

XML documents are widely spread as semi-structured data (i.e., between structured data, such as databases, and unstructured data, such as organizational documents).

Being in this position, XML remains ideal for interoperability and data exchange between different data sources. However, tools can easily be developed to convert structured data into XML documents or vice versa (Popa et al. 2002; Haw & Lee 2011; Chen et al. 2012). Furthermore, semi-structured information can be extracted from unstructured documents and then stored in XML documents. Although more complicated than converting structured data into XML documents, few approaches have been proposed for converting unstructured documents into XML through matured techniques in natural language processing (NLP) (Embley et al. 1999; Lee 2003; Gottlob & Koch 2004; Indumathi & Uma 2008).

For the question "Why are XML documents considered an area of interest despite the availability of many emergent semi-structured representations, such as RDF documents?" the answer is that XML databases and documents throughout the world are widely used. XML can still optimally exhibit interoperability, because it has expressive semantics and structure. Many investments have also been made on XML databases; thus, companies normally maximize the use of such investments. We may reap the benefits from the huge tools and techniques available to deal with these documents.

1.2 PROBLEM STATEMENT

XML documents are widespread and should be integrated to eliminate querying them using a data integration system. Many earlier studies on schema matching do not tackle indirect schema mappings, and the few ones that do fail to consider XML schemas as a special case but regard all schemas (e.g., XML, ER, and OO) as the same. By contrast, the studies conducted specially for XML schemas do not delve into the problem of indirect mappings or into the values of the entities.

Many schema matching algorithms discard the special cases of XML schema by treating XML schemas as a graph structure, a node with children, and a structure with leaf and non-leaf nodes (Embley et al. 2004; Giunchiglia et al. 2004; Xu & Embley 2006; Cruz et al. 2009; Seligman et al. 2010). However, by doing this, they often limit the discovery of matches at different levels of the tree. For example, some approaches do not represent XML documents as directed graphs (Cruz et al. 2009). However, the proposed algorithms can still match XML documents, even though the techniques used in this matching process are not built specially for the metadata of XML schemas and properties of the structure.

Meanwhile, the QMatch algorithm (Claypool et al. 2005; Tansalarak & Claypool 2007) depends on the label and properties of each element to determine the weights used to calculate the QoM for each element. However, this algorithm just depends on linguistic matching, which does not provide accurate results for many indirect matches, because these need to combine structural- and element-level mappings with instance-level mappings (Embley et al. 2004).

1.3 RESEARCH OBJECTIVES

One key contribution of the proposed approach are the combination between XML taxonomies and the hybrid algorithm (Claypool et al. 2005; Tansalarak & Claypool 2007) with the composite approach to automating schema matching in (Embley et al. 2004; Xu & Embley 2006). Another contribution of the current work is the enhancement of these approaches to yield accurate results, as illustrated in the following sections. First, we illustrate below how XML schemas are represented. Afterwards, we present the resources implemented to obtain the mappings.

The final objective is to develop a new composite approach, which detects direct and indirect mappings for XML schemas through the following procedures:

- to develop domain ontology snippets and data frames to obtain sophisticated domain ontology that fits XML schema matching;

- to propose and develop composite algorithms to automate XML schema matching; and

- to test and evaluate the effectiveness of the proposed composite approach to automating XML schema integration

1.4 OVERVIEW OF THE RESEARCH METHOD

Certainly, identifying the appropriate research method is one of the most crucial issues in the success of any research project. An adequate research method assures the establishment of significant objectives, enabling the research to achieve the final goals and contributions successfully.

At the beginning of this research, the primary aim was to identify the research problems and discover the research gaps. These required solutions that were investigated carefully. After proposing the solutions, the plan to develop them was established and then evaluated and compared with other related studies. These steps were the default thoughts of the research method.

In this sense, the generic system development research method design-based research (DBR) (Barab & Squire 2004; Wang & Hannafin 2005) was extended to fit the needs of the current research. This method was the most applicable to this research, which aims to solve the research problems through the proposed approach, then developing the algorithms as a prototype.

The research method consists of four primary phases, as elaborated in Chapter 3. An overview of these phases is given below.

- **Practical Problem Identification:** In this phase, past and current works were reviewed to analyze and observe current data integration approaches and components aimed at identifying the problems and initiating the definition of research questions and objectives.

- **Proposed Solutions:** In this phase, issues from the previous phases were classified, and solutions to the research problems identified were proposed. In this way, the components, processes, algorithms, and tools for the new proposed approach were determined. The new approach with its components and processes was proposed. Each part of the approach (i.e., algorithms and resources) was verified and validated from the literature referenced in the previous phase. Pre-development methods were also identified in this phase.

- **Development:** The proposed approach as a prototype was implemented, which enabled us to validate and evaluate the proposed solutions. First, the pre-development methods (e.g., WordNet) were executed. Given that the basis was ready, the prototype AXM system was developed using RAD methodology. The divide and conquer strategy was also implemented in the development phase, which meant that every dedicated algorithm was programmed separately and tested alone. Afterwards, these algorithms were joined to form a composite prototype.

- **Evaluation:** At this stage, the evaluation measures and benchmarks were defined based on the standards found in the literature. The data set was also identified. The testing process was conducted, with the results continuously analyzed as they were obtained. Finally, the results of the AXM approach were mapped with the most related approaches in the literature.

1.5 RESEARCH SCOPE

The data integration system consists of many parts that work together to achieve interoperability between different data sources. Each part is a separate field of research; for example, the wrapper described in Section 1.1.1 is a vital process in the data integration system. However, this process stands alone as a different research topic compared with the other parts of the data integration system such as mediators. Wrappers depend on crawling and data extraction techniques specific to the data source type that they are accessing, whereas mediators deal with the matching and mappings between the schemas delivered from the wrappers. The current research focuses on one part of the data integration system to provide significant research contributions.

Specifically, this work focuses on the schema matching process. Schema matching primarily influences the whole output of the data integration system, as it conforms to the virtual mediated schema. This schema provides the answers for the queries, i.e., the final resulting information.

Generic schema matching faces many challenges (Shvaiko & Euzenat 2008). Thus, XML schemas served as the inputs to the matching process, enabling this research to narrow down the problems of generic schema matching. The huge amount of various data sources available on the web poses a huge challenge. However, XML documents are semi-structured data (i.e., they are mediator data structure representations).

Moreover, the proposed techniques, such as linguistic and structural matching techniques on XML schemas, can be applied to the schema matching field. This

opportunity guarantees concentrated efforts on developing a robust approach to solve outstanding schema matching problems, such as inaccuracy in the schema matching automation outcome.

1.6 SUMMARY

An overview of this research is introduced in this chapter. The data integration system was briefly discussed as a general background topic. Studies on schema matching were also mentioned to discuss the role of XML in the interoperability between data sources.

The research gaps and problem statement were discussed, and the research objectives to solve the mentioned problems and cover the research gaps were also described.

In addition, the research method phases were described in brief. A well-known research method in this field, DBR, was adapted to address such research needs.

Finally, the research scope was discussed (i.e., the schema matching problem implemented on XML schemas), the aim of which is to more efficiently enhance the data integration system and obtain better results.

CHAPTER II

LITERATURE REVIEW

The data integration automation research area is surveyed thoroughly in this chapter, and the proposed approaches in this area classified and illustrated. The parts and tools needed to establish a working data integration system is discussed. The schema matching research area is then described, followed by a presentation of the XML schema and the matching approaches.

2.1 DATA INTEGRATION SYSTEM

The traditional approach to data integration is to create a global schema over a set of data sources without changing the data in these sources. The data integration system translates queries on the global schema into the original data sources (Hull & Zhou 1996), as shown in Figure 1.1.

During the 1990s, researchers proposed the architecture of the data integration system as a whole, wishing to solve the problem of heterogeneous data integration. However, these researchers faced many challenges, such as heterogeneity and scalability (Section 2.1.1). These challenges forced them to divide this problem into sub-problems and then collaborate to overcome these challenges (Hull & Zhou 1996; Ullman 2000; Ziegler & Dittrich 2007). Despite their best efforts, a fully automated integration system with highly accurate results has yet to be obtained. Therefore, further studies on the data integration system should be conducted.

The data integration system depends on two primary parts: wrappers and mediators. A wrapper is a program, which translates data from the source to a form that is usable to the query processor in the data integration system. These wrappers

differ depending on the data sources they are dedicated to. Meanwhile, a mediator generates and maintains mappings between sources and the global schema. The need to automate this process is a research area that has gained popularity in recent years, as elaborated in the subsequent sections.

2.1.1 Issues and challenges faced by data integration system

Several issues and challenges related to data integration can be viewed from various dimensions discussed below.

a) Heterogeneity

Heterogeneity refers to different data sources with different schemas and structures. However, the web obtains data from many sources, such as local systems for organizations and companies, significantly increasing the number and kinds of data sources to be processed. These sources are autonomous and unique in terms of schemas and structure. Several causes of semantic heterogeneity within different data sources are discussed below.

i. Semantic heterogeneity

Semantic heterogeneity occurs when the information has different shapes and kinds. However, several issues must be dealt with, as listed below.

- Scaling conflicts occur when different reference systems are used to measure the same value, such as the use of different currencies by various countries.

- Naming conflicts occur when the naming schemes of information differ significantly. A frequent phenomenon is the presence of homonyms and synonyms.

- The same names do not necessarily indicate the same semantics, whereas different names may be used to represent the same real-world concept. For example, the word "Jaguar" can be related in real life to either an animal or a

well-known car brand. A normal language dictionary cannot detect that confusion.

- Element names may be encrypted or abbreviated so that they are only comprehensible to their creators. For example, the label "Address" can be abbreviated in many ways depending on the developer, such as "Addr" or "Adrs." Unfortunately, no standards exist for abbreviations and acronyms; moreover, even if some proposals standardize naming, developers sometimes do not follow them.

- Integrity constraints may be hardwired in programs accessing data and not declaratively specified at the schema level. Thus, databases using database constraints (e.g., database triggers) are much better, more reliable, and closer to matching correctly than those depending on the integrity constraints of the programs.

- Elements may be modeled at different levels of detail. For example, address information is divided into street, zip and city in one schema, and then captured using one single field in another.

ii. Structural heterogeneity

Heterogeneity can also exist among schema structure, because analysts and database developers differ in their thinking. This is because, upon development, the information system can be especially suited to a certain organization's requirements. For example, a hospital system in London has different requirements from one in Malaysia or even from one in the UK (i.e., the same country).

iii. Syntactic heterogeneity

- Language: Language syntax differs from one application language to another. For example, Sybase database's syntax is much different from that of Oracle. Language syntax is more difficult to integrate.

- Data model: For example, using XML schemas differs from using UML and entity-relationship models.

Both metadata (i.e., describing information bundled with data) and instance-level conflicts can mislead schema matching, because no similarity or incorrect similarity between schema elements may result from such cases. For the user who is manually performing the matching task or verifying a match result, the conflicts lead to additional time and effort required to correctly understand the semantics of the schema elements. Furthermore, for automatic match approaches, the conflicts typically reduce the result quality if not properly resolved using the corresponding schema transformation or data cleaning techniques.

b) Scalability

Scalability refers to the effect of adding more data sources into the data integration system and its mediated schema. Given that the web data source is available, the number of sources to be accessed and integrated must be open, and the new sources continually become available and become part of the system. Sources within the system may change frequently.

c) Query processing

Query processing deals with the issue of how the data integration system processes and executes queries. However, sets frequently pose queries that can be very complex with respect to the collection of information sources.

d) Evolution

Evaluation stands for the ability to change the global schema as the applications evolve. Database administrators may wish to change the global schema to include some new items of interest.

Most existing approaches to data integration have been presented in previous works (Chawathe et al. 1994; Draper et al. 2001; Do & Rahm 2002; Lee et al. 2002; Melnik et al. 2002; Bernstein et al. 2004; Dhamankar et al. 2004).

2.1.2 Approaches to data integration system

Multiple approaches are used to achieve interoperability between different data sources. Studies that tackle the issues of data integration within the context of data schema can be viewed from the following three perspectives: data warehousing vs. virtual approach, GAV vs. LAV, and target-based approach. These are described below.

a) Data warehousing vs. virtual approach

Two approaches are important in building a data integration system, namely, warehousing and a virtual approach (Hull & Zhou 1996). In the warehousing approach (Zhuge et al. 1995), data from multiple information sources are loaded into a warehouse, and user queries are applied to the data warehouse.

The warehouse approach requires updates to the warehouse when the source data change and when the updates are typically conducted in batches, not on demand. Such an approach guarantees adequate query performance, because queries are read-only and operate on a single repository. In the virtual approach, the data remain in the information sources, and user queries are decomposed at run time into queries on the information sources. The virtual approach is appropriate with many information sources and frequently changing individual information sources. However, this approach requires more sophisticated query optimization and execution methods to guarantee adequate performance.

a) Traditional approaches: GAV vs. LAV

Two approaches are used to mediate schema, which are LAV and GAV (Hull & Zhou 1996; Ullman 2000).

In the GAV approach, for each relation in the global schema exists a query over the source relations. In other words, to every element of the global schema, a view over the data sources is associated such that its meaning is specified in terms of the data residing at the sources.