New Challenges in Data Integration:

Large Scale Automatic Schema Matching

Category of submission: Survey Paper

Khalid Saleem, Zohra Bellahsene

LIRMM - UMR 5506 CNRS University Montpellier 2,

161 Rue Ada, F-34392 Montpellier

Tel. +33467418585

Fax. +33467418500

email:{saleem, bella}@lirmm.fr

New Challenges in Data Integration: Large Scale Automatic Schema Matching

Abstract. Today schema matching is a basic task in almost every data intensive distributed application, namely enterprise information integration, collaborating web services, ontology based agents communication, web catalogue integration and schema based P2P database systems. There has been a plethora of algorithms and techniques researched in schema matching and integration for data interoperability. Numerous surveys have been presented in the past to summarize this research. The requirement for extending the previous surveys has been created because of the mushrooming of the dynamic nature of these data intensive applications. Indeed, evolving large scale distributed information systems are further pushing the schema matching research to utilize the processing power not available in the past and directly increasing the industry investment proportion in the matching domain. This article reviews the latest application domains in which schema matching is being utilized. The paper gives a detailed insight about the desiderata for schema matching and integration in the large scale scenarios. Another panorama which is covered by this survey is the shift from manual to automatic schema matching. Finally the paper presents the state of the art in large scale schema matching, classifying the tools and prototypes according to their input, output and execution strategies and algorithms.

Keywords: Data interoperability; schema matching; schema mapping; schema integration; schema evolution; large scale

1 Introduction

There exists an unending list of digital devices cooperating together to solve problems at individual level, personal or professional, and organisational level. The collaboration between these devices eventuates in better performance and results. Every day a new gadget hits the market, creating a ripple-effect in its surrounding operating environment. For the database community, it is an emergence of new form of data or information, which has to be utilised in the most efficient and effective manner. The ability to exchange and use of data/information between different devices (physical or logical), is the basic activity in any type of system, usually referred to as *data interoperability* (Parent & Spaccapietra, 2000).

Every device has to know the meaning encoded in the input data, which can be learned primarily from the structure of data, called *Schema*. The word schema has its origin in Greek, meaning "shape" or "plan". From computer science perspective it is defined as description of the relationship of data/ information in some structured way or a set of rules defining the relationship. For inception of a system there are different levels, and each level can have its own description. For example, in relation database systems, one has at conceptual level, the entity relationship diagram and at physical level, physical database schema design having tables and fields . Thus for an application, schema gives the best way to understand the semantics of the underlying data instances.

Matching schemas for data interoperability purpose has its roots in information retrieval methods researched since early 80s. Over the period of time, the information retrieval process has gone through a number of changes. Mainly, its evolution has been governed by the introduction of new types of distributed database systems. From text similarity search to ontology alignment applications, the matching process has always been there to be

researched. With the separation of metadata information of the data from real data instance, the matching activity found the new dimension, known as schema matching. Any application involving more then one systems, requires some sort of matching. Thus making study of schema matching a problem applicable to any such scenario, with a difference in the use of matching.

Previous work on schema matching was developed in the context of schema translation and integration (Bernstein, Melnik, Petropoulos, & Quix, 2004; Do & Rahm, 2007; A. Halevy, Ives, Suciu, & Tatarinov, 2003), knowledge representation (Giunchiglia, Shvaiko, & Yatskevich, 2004; Shvaiko & Euzenat, 2005), machine learning, and information retrieval (Doan, Madhavan, Dhamankar, Domingos, & Halevy, 2003). All these approaches aimed to provide a *good* quality matching but require significant human intervention (Bernstein et al., 2004; Doan et al., 2003; Do & Rahm, 2007; Giunchiglia et al., 2004; A. Halevy et al., 2003; Lu, Wang, & Wang, 2005; Madhavan, Bernstein, & Rahm, 2001). However, they missed to consider the performance aspect, which is equally important in large scale scenario (large schema or a large number of schema to be matched).

By definition, schema matching is the task of discovering correspondences between semantically similar elements of two schemas or ontologies (Do, Melnik, & Rahm, 2002; Madhavan et al., 2001; Milo & Zohar, 1998). Basic syntax based match definition has been discussed in the survey by Rahm and Bernstein (Rahm & Bernstein, 2001), extended by Shvaiko and Euzenat in (Shvaiko & Euzenat, 2005) with respect to semantic aspect. In this article, we discuss a new dimension of schema match, which focus on the requirements of automatic large scale schema matching and integration, also incorporating the previous ideas of mappings. We highlight the structural aspect of schema and its credibility for extraction of data semantics.

The requirement for enhancing the previous works of matching definition has been created because of the evolving large scale distributed information integration applications, which are also directly increasing the industry investment proportion (Davis, 2006)[1] in the matching domain. The schema matching task of these applications which need to be automated are also discussed in length in this paper. Another aspect of this survey is the presentation of the schema matching classification from the perspective of latest strategies and algorithms in the field of schema based information retrieval and management.

Contributions: Our contributions in this paper are related to large scale schema matching with the need for automation. We

- evince the relationship between the basic schema matching techniques, research domains utilizing these techniques and the application domains which benifit and propel this research;
- present the latest trends of application development in the field of large scale schema matching and integration;
- propose a taxonomy of schema matching strategies with respect to large scale scenario, from the input, output and execution perspective; and
- discuss the latest tools/prototypes in large scale schema matching and integration.

The rest of the paper is organized as follows. In section 2 we explain the motivation for this article with respect to large scale schema matching. Sections 3 outlines the emerging application domains requiring large scale data interoperability. Section 4 describes basic schema matching algorithms at element and structure level. In section 5 we discuss different strategies suitable

[1] Markets for semantic technology products and services will grow 10-fold from 2006 to 2010 to more than 50 billion dollars worldwide

for large scale schema matching scenarios. Section 6 presents a comparison of some current tools and prototypes applied in schema matching domain. Section 7 concludes the paper, giving an outline of the future research perspectives in schema matching.

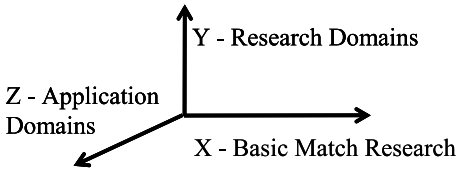

Fig. 1. Schema Matching Dimensions

2 Revisiting the Schema Matching Problem

In this section, we present a classification of schema matching problem along three dimensions; basic match research, related research domains in schema matching and application domains dependent upon data interoperability. We focus on each dimension and show how these dimensions are interlinked, with the help of an example. Further, we give an overview of the research domain of large scale schema matching.

2.1 The Three Dimensions of Schema Matching

Schema matching has been researched from various perspectives by researchers belonging to different research and application domains. While reviewing the numerous approaches and techniques, we came to understand that schema matching is related to three different interlinked dimensions: (i) *basic match research*, related (ii) *research domains* and (iii) *application domains*. The three dimensions (Figure 1) are related as: algorithms are developed for *Basic Match*

Techniques for exploiting some *Research Domain*, which is in turn responsible to carry out the objective of a certain *Application Domain*.

Fig. 2. Query Interfaces Over the Web for Travel

Example: Let us consider an example of a traveler searching for good deals for traveling for his next holidays. There are hundreds of web interfaces available for query purposes. She can not query each interface and then compare each query result. The best answer to his problem would be a virtual mediated interface which is mapped to each of the physical interfaces over the web in the travel domain. The results from each query are in turn integrated and displayed according to her preferences. The first step in the implementation of the virtual interface is the matching of all possible/ available interfaces over the web in the specified domain. Once the mappings between the individual interfaces and the integrated interface has been done, query processing and results display processes can be initiated. The mediated schema with mappings

can be cached for future utilisation by other users with similar requirements. The web query interfaces follow a hierarchical tree like structure as shown in figure 2 for travel domain taken from TEL-8 dataset [2].●.

The example presents the three dimensions as: (i) Basic match techniques applied to query interface attributes (based on attribute label, default value, data type, list of available values etc.) and their structural aspects, (ii) for query interface forms schema integration and mediation (iii) in the application domain of querying the web based travel resources.
Knowledge extraction for schema matching is done by exploiting two entities, (i) data and (ii) schema; structure describing the data. The availability of the two entities is governed by the application specific constraints. For example *data security*, where direct data access is restricted and only controlled access through schema is granted. There are large number of techniques researched for schema matching with respect to data instances and schemas (section 4).

Table 1. Dimensions of Schema Matching - Basic Match Research

X - Basic Match Research	**Level**
Linguistic based	Element
Constraints based	Element
Graph based	Structure
Data Instance/Machine Learning based	Element/Structure
Use of External Oracle	Element/Structure

Basic match techniques (table 1)exploit the granularity aspect of a schema. It can be seen as the match algorithm development process for a certain entity in the schema. The entity can be the most basic constituent of schema e.g., field in a relational database schema table (Benkley, Fandozzi, Housman, &

[2] http://metaquerier.cs.uiuc.edu/repository

Woodhouse, 1995; Bilke & Naumann, 2005), or the whole schema structure itself exploited, using some graph match techniques (Do & Rahm, 2007; Madhavan et al., 2001; Melnik, Rahm, & Bernstein, 2003). A combination of basic match techniques are utilized to resolve problems indicated in Table 2. Detail discussion on these techniques is given in section 4.

In **research domain** (data interoperability) (table 2), *Schema Integration* (Batini, Lenzerini, & Navathe, 1986) can follow three possible approaches (i) *binary incremental* : two schemas are integrated at a time, following an upward binary tree pattern, (ii) *clustering incremental* : clusters of schemas are created based upon some similarity function, an integrated schema is generated for each cluster and the resulting schemas are further grouped into clusters and so on (Saleem, Bellahsene, & Hunt, 2008; M.-L. Lee, Yang, Hsu, & Yang, 2002), or (iii) *holistic* : all schemas are pruned together and integrated (B. He, Chang, & Han, 2004) .

Table 2. Dimensions of Schema Matching - Research Domains

Y - Research Domains	**Type**
Schema Integration	Binary/Holistic
Schema Mapping Generation	Cardinality
Schema/ Mapping Evolution	
Ontology Alignment	Binary/Holistic
Match Quality Evaluation	

While comparing schemas, there is high probability that source element can have more than one matches in the target schema. One match has to be ranked the best manually or automatically, for the mapping purpose. The cardinality of mapping (Rahm & Bernstein, 2001) demonstrate the numeric relationship of element correspondences . Semantically speaking, it is the number (combination) of elements in each of the two schemas, representing the same

concept; 1:1, 1:n, n:1 and n:m element map cardinality. More specifically it is also called *Local Cardinality*. In schema matching we also come across another type of cardinality called *Global Cardinality*. It refers to the problem when one element is involved in more than one mappings. Generation of schema mapping expressions and their updating with changes in schemas is a highly demanding research domain (An, Borgida, Miller, & Mylopoulos, 2007; Velegrakis, Miller, & Popa, 2004). For example, a concept "totalPrice" in source schema is mapped to a combination of "price" and "tax" elements in target schema, with the mapping expression calculated as
$(totalPrice)_{source} \leftrightarrow (price + (price * tax))_{target}$.

The temporal changes to a schema and its effects on the existing mappings also provide another research domain. Since the web is an evolving entity, the *schema evolution* and related *mapping evolution* require attention. Methods like domain level corpus based schema matching (Madhavan, Bernstein, Doan, & Halevy, 2005) demonstrate how to maintain a repository of schemas, concepts representations and related mappings for subsequent handling of temporal changes in the constituent schemas of the domain. In another research work (Velegrakis et al., 2004), the authors show how the changes in a schema are used to rewrite the queries representing the mappings. The research benefits from CLIO (Hernedez, Miller, & Haas, 2002) which generates queries as the mappings expressions.

The ontology concept hes been around since the early 90s. Today it is vigorously used in different applications requiring interoperability. Ontologies are used for knowledge representation and are similar to schemas to a certain extent; as both describe data domain with the help of terms with constrained meanings (Shvaiko & Euzenat, 2005). Techniques used for schema matching

have been tailored for ontology matching (ontology alignment) (Euzenat et al., 2004).

Another similar research area, which has emerged as a by product of research in agents communication, is the tracking of changes in the source ontologies of agents called *ontology evolution* (Noy, Kunnatur, Klein, & Musen, 2004). Since agents are independent entities, following their own rules, they requires different techniques for comparing and registering of changes with in their and the counter-part agent ontology.

The most challenging research domain has been the match quality evaluation. Measures like *precision* (the proportion of retrieved and relevant mappings to all the mappings retrieved) and *recall* (the proportion of relevant mappings that are retrieved, out of all relevant mappings available) (Euzenat, 2007; Gal, 2006b), have been borrowed from information retrieval domain. These metrics have been customized to quantify the quality of schema matching but still require a lot of work.

Table 3. Dimensions of Schema Matching - Application Domains

Z - Application Domains	**Type**
Data Warehousing	static
Message Translation	static
E-Commerce	static
Catalogue Integration	static/dynamic
Web Services Discovery and Integration	dynamic
Agents Communication	dynamic
Enterprise Information Integration	static/dynamic
Data Mashups	static/dynamic
Schema based P2P Database Systems	dynamic
Federated Systems	static/dynamic
Business Processes Integration	static
Query Answering (Web/Distributed Systems)	static/dynamic
Ontology Management	static/dynamic

The **application domains** for schema matching research can have a long list. Some prominent and latest fields are enumerated in table 3. The application domains can be categorized with reference to the time line and the data interoperability static or dynamic aspect. Late 80s and early 90s have been dominated by the static nature of matching. For example, in applications like Data Warehousing, Message Translation, E-commerce (Rahm & Bernstein, 2001), the source schemas have been created and their matching and integration is one time fixed process. Whereas the applications of late 90s and current era, have a much dynamic nature propelled by the internet and its changing technologies. The concepts like Web Services, P2P Databases, Large Scale Querying (Shvaiko & Euzenat, 2005), demand techniques which can support the independence and changing nature of contributing sources. A detail review of the current trends and related applications is given in section 3.

2.2 Large Scale Schema Matching

We have our motivation from the current trends of large scale dynamic aspect of schema matching. Large scale schema matching can be categorized into two types of problems depending upon the input, (i) two large size schemas (with thousands of nodes). For example bio-genetic taxonomies (Do & Rahm, 2007), (ii) a large set of schemas (with hundreds of schemas and thousands of nodes). For example hundreds of web interface forms (schemas) related to travel domain (B. He et al., 2004; Wu, Doan, & Yu, 2005).

The schema matching tools available today can be used for applications which require matching of two large schemas. (Do & Rahm, 2007) and (Mork & Bernstein, 2004) demonstrate that with some modification to the available tools/infrastructures, the required goals can be achieved. Research in (Do & Rahm, 2007) breaks down the bio-genetic taxonomies into fragments and apply their matching tool COMA++ (Aumueller, Do, Massmann, & Rahm, 2005) on

pairs of these fragments to find similarities between the two taxonomies. Whereas, work in (Mork & Bernstein, 2004) uses three levels of matching; using CUPID (Madhavan et al., 2001) for lexical analysis of nodes using external oracles, then applying Similarity Flooding (Melnik et al., 2003), fix point computation algorithm based on the idea of neighborhood affinity, and in last phase the hierarchical matching finds similar descendants. The ideas work well in case of two schemas but when the scenario has large number of schemas, the formalization, techniques and algorithms for the problem change. For us the motivating scenario lies in the integration of large number of schemas with automated matching aspect. Today this problem is specifically encountered in applications like schema based P2P database systems, query answering over the web, web services discovery/integration and data mashups in enterprise information integration. The problem has been researched using holistic matching or incremental pair-wise matching and integration algorithms, using recursive (Melnik et al., 2003), clustering (Meo, Quattrone, Terracina, & Ursino, 2006; Smiljanic, Keulen, & Jonker, 2006; Wu et al., 2005) and mining (B. He et al., 2004; Saleem et al., 2008) techniques. The automation factor is a must to solve this problem. Since large number of schema matching can not be handled semi-automatically, therefore the notion of approximate semantic matching rather than exact match, with performance has been advocated (B. He et al., 2004; Wu et al., 2005).

3 New Application Domains for Data Interoperability

Schema matching research has its roots in schema integration applications in distributed database systems. The task is to produce a global schema from independently constructed schemas. The requirements for such an integration have been presented in (Batini et al., 1986; Spaccapietra, Parent, & Dupont,

1992). The research highlights the issues in schema integration of relational schemas, the integrity of integrated schema and different possible techniques to integrate schemas (binary or n-ary). Data Warehousing, Message Translation (Rahm & Bernstein, 2001), E-commerce, B2B, B2C (Shvaiko & Euzenat, 2005) applications are examples of implementation of this research.

Today, from the artificial intelligence view point the research in this domain revolves around *ontologies*. Ontology is a way to describe data elements along with inter-element relationship rules, based upon object oriented techniques but coded in a semi-structured way. In the last couple of years domain specific ontologies have been incorporated in the data integration processes, demonstrating acceptable results (Euzenat et al., 2004). But the core problems faced in the changing world for communication and integration are the same, whether it is ontologies or schemas (Haas, 2007; A. Y. Halevy, Rajaraman, & Ordille, 2006) .

The latest trends in applications development requiring data interoperability can be explicitly attributed to the technologies harnessing the web. For example ontologies alignment (Euzenat et al., 2004), integration of XML data on the web (Meo et al., 2006) etc. In the subsequent subsections we give the current application domains motivating our work on schema matching.

3.1 Web Services Discovery and Integration

Initial concept of web was to share scientific research, followed by web sites for advertisement of products and services. Next the business community used it to do transactions with their customers, followed by secure business transactions between two e-business ventures, called B2B systems. This gave rise to the web service concept i.e., set of functions which can be invoked by other programs over the web. So, to achieve a certain goal, the user/program has to first discover the services, perform some matching to select the

appropriate services, do some planning for execution of the services to get to the subgoals and finally combine the subgoals (Huhns & Singh, 2005) to achieve the main goal. One approach to search for web services is to access a UDDI (Universal Description, Discovery, and Integration - standard for centralized service repositories) Business Registry (UBR) as the search point. Web service providers register their services with the UBRs for subsequent usage by others. Another approach is to use web search engines which restrict their search to WSDL (Web Service Description Language) files only (Bachlechner, Siorpaes, Fensel, & Toma, 2006). WDSL is an XML based language standard for describing a web service. The need for matching and merging is quite evident, as web services have to be searched against user goal requirements, compared and integrated for subgoals achievement.

3.2 Data Mashups in Enterprise Information Integration

Data Mashups is the most recent buzz word in the Enterprise Information Integration (EII) domain. Its definition can be: making new knowledge by joining available information. Web mashups are emerging at a rapid pace. *Programmable.com* provides a list of such mashups. A typical web mashup joins information from related web sites. For example a mashup website about cars can get quotes about a certain car from quotes websites, pictures and reviews from cars forums along with video footage from some social network like *youtube.com*. Thus the information resources can range from a simple database table to complex multimedia presentation i.e., the search can be on any structured or unstructured data.

Thus the core concept in mashups is to extract some new necessary knowledge from all these sources existing in different formats. This is a new challenging issue in information extraction and integration. The research aim is to provide light and fast protocols which can work through different meta models and

types of documents (A. Y. Halevy et al., 2006). At the enterprise level, the mashup idea helps in building quick situational applications, for some transient need in the enterprise, complementing the more robust and scalable integration technologies that the enterprises invest in.

An example of enterprise mashup implementation is done at IBM as Information Mashup Fabric(MAFIA) (Jhingran, 2006). In MAFIA the data input are complimented with those normally not covered by traditional EII systems, e.g., emails, presentations, multimedia data etc. In the coming years, mashups will open up a new enterprise application market, providing business users and IT departments with a quick and inexpensive approach to develop and implement applications, requiring matching and joining data in diverse formats.

3.3 Schema based P2P Database Systems

One of the latest emerging research field in databases over the web is *P2P Databases* (A. Halevy et al., 2003). There have been numerous successful P2P systems delivered in the last couple of years. Traditionally, the P2P systems have been simple file sharing systems which can self tune, depending upon the arrival and departure of contributing peers. Industrial-strength file sharing P2P systems, like Kazaa and bitTorrent, allow the peer autonomy of participation but they still restrict the design autonomy of how to describe the data. Secondly, sharing of data objects described by one P2P system are not available in another P2P setup. Today, the P2P technology has transformed into sharing of any kind of data, whether it is semi structured XML data or continuous multimedia streaming (Meddour, Mushtaq, & Ahmed, 2006). The next generation of data sources are going to be totally independent of each other, i.e., they will have the design autonomy, utilizing their own terminologies for their data structuring, with capabilities to interact with

others. For querying these data sources some matching method will be required to broker between their structures, giving rise to the new generation of application research of schema based P2P data sharing systems (Loser, Siberski, Sintek, & Nejdl, 2003).

3.4 Querying over the Web

Query processing has two intrinsic problems; understanding the query and then finding the results for it. The web contains vast heterogeneous collections of structured, semi-structured and unstructured data, posing a big challenge for searching over it. Deep Web (B. He et al., 2004) scenario highlight this aspect. Firstly, the heterogeneity problem allows the same domain to be modeled using different schemas. As we have discussed in the example for our motivation. Secondly, it is very difficult to define the boundary of a domain. For example, traveling and lodging are inter-linked for tourist information web sites. Continuous addition of new content further complicates the problem for searching and integrating the results.

3.5 Online Communities

People have been using online spaces to communicate, since the beginning of the internet. Today, with the available resources for the web, these communities have mushroomed to an unprecedented level. These virtual connections of people is also called social networks. Every community has a purpose or goal with a target audience. For example, videos or photos sharing communities or simple forums regarding a specific subject. To be more business oriented, distributed work groups within companies and between companies use online community to build their team, keep in touch and even work on projects together. Sometimes, one can find more exact answers to queries from specific online community rather than from search engine.

What ever the reason for the community, it needs a structure to support the underlying collaborative data. The data sources can be as diverse as discussed in mashups. In such virtual communities there is no central authority to monitor the structure and the performance. Users can join and leave, contribute or simply use the resources like P2P systems. In such a scenario, the matching of data resources is an extreme problem in schema matching domain. The problem complexity is further elevated if we consider an inter community communication. With the semantic web around the corner, this domain requires lots of attention.

There are very few studies in this area. In (McCann, Shen, & Doan, 2008), authors show a question answer based technique to solve the match problem in online communities. The method automatically generates questions for element names which are not possible to be compared. Choices are presented to several users and then the results are heuristically evaluated to assess the correct match.

3.6 Agents Communication

Agents Communication can be considered as a dialogue between two intelligent entities. Each agent working out its actions according to its own intelligence or ontology. When two independent agents come in contact for the first time, they need some protocol to translate the message of one agent into the ontology of the other agent (Shvaiko & Euzenat, 2005). For subsequent encounters the agents may utilize the mappings discovered and stored within them. To answer the query of its user, an agent may have to interact with number of other agents, compare and integrate their responses, just like web services. Only agents have inbuilt mechanisms to learn and counter the changes around them. P2P Ontology Integration (Besana, Robertson, & Rovatsos, 2005) proposes a framework for agents communication in a P2P network. Its main feature is

that, it efficiently tries to map dynamically only the part of ontologies, which are required for the communication.

The above set of application domains have one thing in common, they encounter dynamic information requirements, changing over time and process web scale data. It is very difficult to achieve desired performance oriented goals with research revolving around semi-automatic schema matching and integration approach. The scenarios require an automatic intelligent and self-tuning solution.

4 Schema Matching Techniques

This section gives an overview of the basic techniques used in the schema matching and integration research. Schema comprises of some basic entities called elements. The composition of elements within the schema follow rules outlined by a data model. While to date, a number of algorithms have been devised and implemented for finding correspondences between schemas. These algorithms have been dependent on techniques of string matching, linguistic similarities or constraints likeliness at element level or higher schema structure level. Graph algorithms utilised in schema matching is a special form of constraints matching (Shvaiko & Euzenat, 2005) for managing structural similarity. In some cases, these algorithms are further supported by data instances of schemas.

4.1 Element Level

Schema matching is a complex problem, which starts by discovering similarities between individual schema elements. Every element, disregarding the level of granularity, is considered alone for a match. The techniques used, basically rely on the element's name and associated description, using basic **string**

matching approaches adapted from the information retrieval domain (Duchateau, Bellahsene, & Roche, 2007). These approaches include string prefix, suffix comparisons, soundx similarities and more sophisticated algorithms based on string distance. There is a large list of these algorithms with various variations researched over time. The mainly talked about approaches are the n-gram and the edit distance[3]. For example Google use n-gram for statistical machine translation, speech recognition, spelling correction, information extraction and other applications.

Linguistic techniques are based on the tokenisation, lemmatisation and elimination. The idea is to extract basic sense of the word used in the string. And then find its contextual meaning (Bohannon, Elnahrawy, Fan, & Flaster, 2006; Duchateau et al., 2007) i.e., meaning extraction according to the elements around it. These techniques have been adopted from linguistic morphological analysis domain. The algorithms are further enriched to provide synonym, hypernym, hyponym similarities by using external oracles, dictionaries, thesauri like WordNet (Gangemi, Guarino, Masolo, & Oltramari, 2003), domain specific ontologies or upper level ontologies (Niles & Pease, 2003).

Constraints similarity is data model dependent. One of the basic constraint found in almost every model is the element type e.g. integer, string etc. Different data models have their own list of constraints. Relational model has primary key constraint to bind different attributes data in a tuple or foreign key constraint to relate to table elements. Similarly, is-a and has-a relationship constraints in object oriented model and parent-child relationship in hierarchical structure of XML data model. These relationship constraints help in extracting the relative contextual concept of an element.

[3] Listing with detail available at http://www.dcs.shef.ac.uk/∼sam/stringmetrics.html

4.2 Structure Level

Structure level matching is referred as matching a combination of elements from one schema to another schema (Rahm & Bernstein, 2001). The algorithms developed are based on graph matching research. It can also utilize external oracles like known patterns (Embley, Xu, & Ding, 2004), ontologies (Doan et al., 2003) or corpus of structures (Madhavan et al., 2005) to recognize the similarity. It also helps in solving n:m complex match problem.

Today, almost every schema matching implementation uses some form of **graph structures** for internal representation of schemas. Graph matching is a combinatorial problem with exponential complexity. Researchers use directed acyclic graphs or trees to represent schemas, ontologies or taxonomies, to reduce the complexity aspect of the problem. In generic schema matching tools (which can take as input different data model schemas) the graph structures are flexible enough to support the possible input schema elements and perform mapping. Nearly all schema match research projects based on graphs, use the notion of neighborhood affinity to compute the similarity match value for individual elements. This aspect has been presented in Similarity Flooding algorithm (Melnik, Garcia-Molina, & Rahm, 2002).

In large scale scenarios, structure level matching techniques help in enhancing the performance of the match implementations, by using neighborhood search algorithms (Ehrig & Staab, 2004). In literature holistic (B. He et al., 2004) or level-wise algorithms (children-parent relationships) (Madhavan et al., 2001; Do & Rahm, 2007) have been used to determine the correspondences among two schemas.

Another variation of structure level matching is based on taxonomy of ontologies. For example *bounded path matching* (Ehrig & Staab, 2004) takes two paths with links between classes, defined by the hierarchical relations,

compare terms and their positions along these paths, and identify similar terms. *Super(sub)-concepts rules* oriented match follows the idea that if super-concepts are the same, the actual concepts are similar to each other. Another related interesting measure called *upward cotopic distance* (Euzenat et al., 2004) measures the ratio of common super classes to find similarity of classes of two taxonomies.

Structure level matching also follows model-based techniques. The graph(tree) matching problem is decomposed into a set of node matching problems. Each node matching problem is translated into a propositional formula, namely pairs of nodes with possible relations between them. And finally the propositional formula is checked for validity. Research in (Giunchiglia et al., 2004) demonstrates the effectiveness of this technique but with worst time performance, when compared to other available tools.

4.3 Use of Data Instances and Machine Learning

Data instance in schema matching is used in two ways. First, if the schema information is very limited or not available, instance data is used to create a representation of the data (Bex, Neven, & Vansummeren, 2007). For example from any XML document, a basic tree hierarchy of elements can be extracted. Even, if the schema is available, data instances can augment the schema matching by giving more insight about the schema element semantics (Hernedez et al., 2002). For example city names encountered in data instances (found in a general list of city names) can infer that the field is a component of address field.

In second case data instances are used in schema matching for training machine learning algorithms. In (Doan et al., 2003), XML schema inner nodes are matched by comparing concatenated values of their corresponding leave nodes using learning techniques, e.g., address is a composition of street, zip and city.

In another research (Dhamankar, Lee, Doan, Halevy, & Domingos, 2004), n:m mapping expressions are predicted, involving arithmetic functions, like totalprice is equal to price+(price*taxrate). First, it uses an external global domain ontology to map the elements and then find the function by employing the data instances with a set of arithmetic and string concatenation rules . The drawbacks in use of data instances can be either the bulk of data to be analysed, thus down-grading the performance or the verification of the quality and granularity of data instance, which may require some cleansing technique (M.-L. Lee, Ling, Lu, & Ko, 1999). In the dynamic environment where the load of schemas itself is quite large, data instance approach is difficult to implement because of its drawbacks.

5 Match Strategies

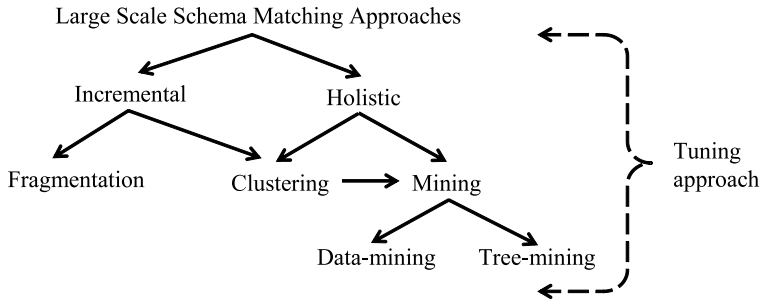

Fig. 3. Taxonomy for Large Scale Schema Matching and Integration Strategies

Different schema match research projects have shown that single match algorithm is not enough to have a quality match. It is necessary to employ a range of algorithms, applied in a sequence or parallel, optimized for the application domain. Researchers have followed different strategies depending

on application domain or researcher's objectives. The strategy is basically governed by the input and output requirements of the match tool.

Input aspect of the tool outlines the information about the entities, available for matching. For example *schema-based vs data instance-based* or the tool is for some *explicit input domain* or not. The output requirements depend upon the research domain, in which the tool is to be utilized. The output can also dictate the tool to be can be *manual, semi-automatic* or *automatic*. For example, web services require an automatic environment and comparison of two large bio-genetic ontologies can be worked out with a semi-automatic tool, where possible matches are presented to the user to select the appropriate one as the mapping.

The execution part is responsible for rest of the categorisation of schema matching tools. Namely, *internal vs external* (Rahm & Bernstein, 2001), *syntactic* vs *semantic* (Shvaiko & Euzenat, 2005) and *Hybrid* (Madhavan et al., 2001; Giunchiglia et al., 2004) vs *composite* (Do & Rahm, 2007) approaches. Some latest developments in matching approaches are being guided by the large scale scenarios like P2P data networks, semantic web, query over the web and semantic grid services. These large scale scenarios are being dealt using techniques which can retrieve good match results directly or enhance (Y. Lee, Sayyadain, Doan, & Rosenthal, 2007; Mitra, Noy, & Jaiswal, 2005) the already existing results automatically. In some work, performance with approximate mapping is being preferred over exact mapping (B. He et al., 2004; Saleem et al., 2008).

In next sub-section, we give an account of the strategies adopted by the researchers or which can be exploited for large scale schema matching. Figure 3 shows a classification of these strategies, with inter-strategy relationships.

5.1 Schema Fragmentation Approach

In the domain of semi-structured data, more and more schemas are being defined in XML, a standard language adopted by W3C. It is being widely used in E-business solutions and other data sharing applications over the web. Over time, emergence of distributed schemas and namespaces concepts have introduced more complexity to the matching problem.

Research work in (Do & Rahm, 2007) demonstrates, how these emergent problems can be tackled. The authors propose the idea of fragmentation of schemas for matching purposes. The approach, first creates a single complete schema , including the instances for the distributed elements or namespaces used in the schema. In second step the large schema instance is broken down into logical fragments which are basically manageable small tree structures. The tool COMA++ (Aumueller et al., 2005) is used to compare each fragment from source schema to each fragment of target schema for correspondences, with the help of GUI and human input. The approach decomposes a large schema matching problem into several smaller ones and reuses previous match results at the level of schema fragment. The authors have reported satisfactory results.

In (Hu, Zhao, & Qu, 2006), the authors apply the fragmentation (partitioning) approach on large class hierarchies extracted from ontologies. Each partition is called a block with an anchor class. Matches for anchor classes are pre-detected, thus elements of blocks with similar anchors are further matched in the system.

5.2 Clustering Approach

Clustering refers to the grouping of items into clusters such that items in one cluster are more similar to one another (high affinity) and those in separate clusters are less similar to one another (low affinity). The level of similarity can

vary from application or technique which is using clustering approach. Since the schema matching problem is a combinatorial problem with an exponential complexity, clustering works as an intermediate technique and improves the efficiency of the large scale schema matching. In schema matching and integration, clustering can be considered at element level or schema level.

Element Level clustering can be applied on a single schema or holistically on the given set of schemas. The authors of (Smiljanic et al., 2006) give a generic approach using the element level clustering method to detect element clusters in schema repository which are probably similar to a given personal source schema. Personal schema is then fully compared to detected list of clusters. So, rather comparing and applying all match algorithms on all schema elements in the repository, only a subset of elements are considered.

In another research work (Saleem et al., 2008), element clustering is applied at the holistic level of schemas. The work is directed toward large scale schema integration. Initially a set of clusters is created, in which each cluster have linguistically similar label elements. Intuitively, the nodes having similar labels are also clustered together. The largest size schema in the input schemas is considered as initial mediated schema. Each input schema is compared to the mediated schema. The source element is only compared to the elements found in its cluster belonging to the mediated schema.

Schema Level clustering is an extended version of element level clustering. The approach clusters together schemas which show some level of elements' similarity among them. In (M.-L. Lee et al., 2002), the authors demonstrate a recursive algorithm which finds similar elements in XML DTDs and creates their clusters. In second step, it performs the integration on each DTD cluster. The process goes on until one global DTD has been created.

A very comprehensive work on XML schema clustering techniques is given in (Dalamagasa, Chengb, Winkelc, & Sellisa, 2006).

5.3 Data Mining Approach

Data Mining is the technique for finding similar patterns in large data sets. Very recently, it has been used as schema matching method. Work in (B. He et al., 2004; Su, Wang, & Lochovsky, 2006) highlight this method for matching and integrating deep web schema interfaces. (B. He et al., 2004) uses a positive correlational algorithm based on heuristics of schema attributes. Whereas (Su et al., 2006) applies negative correlational method to match and integrate schemas.

Tree mining approach is a variation of data mining, in which data is considered to posses a hierarchical structure. It shows more affinity to XML schemas, which are intrinsically tree structures. (Saleem et al., 2008) demonstrates a method which combines the element clustering and a tree mining method. the work provides a time performance oriented solution for integrating large set of schema trees, resulting in an integrated schema along with mappings from source to the mediated schema.

5.4 Strategies for Enhancing Match Results

There have been a lot of work on schema matching but proof of exact results in the semantic world have been hard to achieve. In most of the research the quality of results has been said to be approximate (Rahm & Bernstein, 2001; Noy, Doan, & Halevy, 2005; Shvaiko & Euzenat, 2005). As a result of these observations new avenues of research opened up for finding ways to achieve the maximum correctness in schema matching. Following are the approaches under active research.

Pre-Match Strategies: Pre-match methods typically deal with the matching tool's execution strategies, called *tuning match strategies*. These approaches try to enhance the performance of current schema matching tools which have the ability to rearrange the hybrid or composite execution of their match algorithms. Defining external oracles, the criteria for their use and adjustment of parametric values, like thresholds, for different algorithms is also part of pre-match. The work in (Y. Lee et al., 2007) provides a framework capitalizing on instance based machine learning. The authors describe, how the use of synthetic data sets can equip the matching tool with the ability to perform well, when applied to a similar real scenario. The tuning module execution is totally separate from the actual tool working.

Post-Match Strategies: These strategies are concerned with improving the already obtained results from a schema matching tool. OMEN (Mitra et al., 2005) Ontology Mapping Enhancer, provides a probabilistic framework to improve the existing ontology mapping tools using a bayesian network. Is uses pre-defined meta-rules which are related to the ontology structure and the meanings of relations in the ontologies. It works on the probability that if one know a mapping between two concepts from the source ontologies (i.e., they match), one can use the mapping to infer mappings between related concepts i.e., match nodes that are neighbors of already matched nodes in the two ontologies.

Manakanatas et al.(Manakanatas & Plexousakis, 2006) work is a post-match phase prototype application. It has an interface to detect the best map, from the set of mappings for a source schema element produced by COMA++. It uses WordNet as the linguistic oracle to filter the COMA++ results, in case

there are more than one possible mappings for a source element to target schema. Thus minimizing the human intervention.
One of the latest work for detecting the best match results, according to the user preferences, using fuzzy logic has been demonstrated in(Guedria, Bellahsene, & Roche, 2007). The work also enhances COMA++ results for deriving best semantic mappings. The research proposes to apply fuzzy sets theory utilizing pre-defined user preferences.

5.5 GUI aspect

User perception is getting more importance in the schema matching tools in the form of investments in the *graphical user interface* development for the generic schema matching tools. Current schema matching tools interfaces only support subject domain experts with good computer science background. And schema matching tools in large scale scenarios still lack the initiatives in user interface development. However with matching becoming need of today in the the ever expanding data integration domain, new user centric graphical environments are emerging to support the matching tasks (Wang et al., 2007). These environments have augmented the match task in *pre-match, amid-match and post-match phases*. Pre-match phase interface provide the facility to define a domain or application specific strategy, to align the different schema matching algorithms. It can include configuration of various parameters of the match algorithms selection or specification of auxiliary information like synonyms, abbreviations and other domain specific constraints (Aumueller et al., 2005).
The post-match phase uses different measures to select the best correspondence, for an element from a set of possible matches which show the semantic equivalence aspect for that element (Aumueller et al., 2005; Bernstein, Melnik, & Churchill, 2006; Hernedez et al., 2002). Tools like CLIO

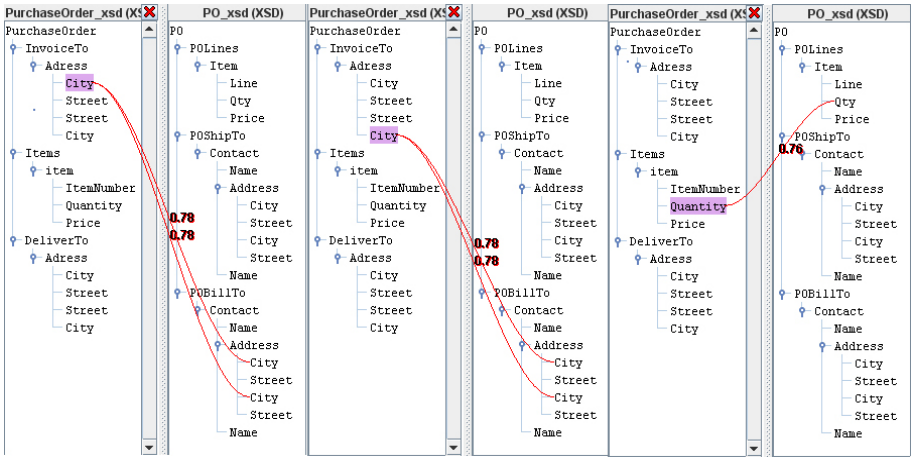

Fig. 4. Three cases for selecting the best match: COMA++

(Hernedez et al., 2002), COMA++ (Aumueller et al., 2005) and Microsoft BizTalk Mapper (Bernstein et al., 2006) generate the possible mappings along with degree of match. And then graphically allow the user to select the mappings according to her expertise (figure 4).

The amid-match phase interface interactively solve the matching problem, with the user help. Schema matching environment SCIA (Wang et al., 2007) provides a much detail interface. It focuses on minimum user intervention based on some pre-defined rules regarding the contextual matching of elements.

5.6 Top-k Methods

Top-k mappings method, semi-automatically, tries to find not the best but k best possible matches from which user can select the most appropriate. Thus intuitively increasing the recall measure of the quality of mappings. There exists two variations (i) *element level* and (ii) *schema level* top-k mappings. In former case the matches are presented for each element. Whereas in latter, top-k possible sets of mappings for whole schema are considered.

Several schema matching tools analyse the target search space in an iterative manner, which can be considered as top-K approach. Most of these tools, find element level top-k matches. For example the CLIO (Hernedez et al., 2002) tool calculates the best matching and user have to select or reject the matchings. A rejection results in re-evaluation for next best possible match for that element. Thus producing a highly user dependent iterative system. LSD (Doan, Domingos, & Halevy, 2001) also works in the similar fashion; accepting the rejection as a constraint for its learning process. In contrast, COMA++ (Do & Rahm, 2007), presents the user with the best matches with match confidence higher than the defined threshold. There is no fix value for k in this case. The user can select one of the proposed matchings or reject all. Another approach presented in QOM (Ehrig & Staab, 2004), also works in the similar fashion; reutilising the results from previous iteration and proposing a new set of k matches. It demonstrates a more robust automatic approach with good results.

In literature, there are very few works regarding schema level top-k mappings. (Gal, 2006a) presents an automatic approach which finds top-k possible sets of mappings between two schemas. Next, the similarity between the sets is analysed heuristically, resulting in a set of mappings which are most frequent in all the sets. The author gives very solid results and arguments to support the high recall value for this set.

Another variation of the technique is discussed in (Sarma, Dong, & Halevy, 2008). The authors demonstrate an automatic technique for generating top-k mediated schemas along with mappings from input schemas to the integrated schemas. The research, first generates possible clusters of similar elements among the input schemas and then produce a set of probabilistic mediated schemas with probability values i.e., top probable schemas. Further, the probabilistic mappings are computed from input schemas to the probabilistic

mediated schemas. Queries from users are applied on the probabilistic mediated schemas and data results are statistically evaluated for same query. Similarity in results, provide a way to verify the integrity of probabilistic mediated schemas and helps in constructing the deterministic mediated schema with mappings.

A somewhat similar approach as above is given in (Chiticariu, Kolaitis, & Popa, 2008). Authors demonstrate a method to generate all possible integrated schemas, without duplicates, from a set of input schemas, along with mappings. These schemas are merged, based upon user defined constraints in an interactive manner to generate the final integrated schema.

5.7 Discussion

In the preceding sub-sections, we have discussed some recent strategies to enhance the quality of results of schema matching and integration. These techniques supplement the already existing basic schema matching and integration algorithms, and also highlight the fact that structural comparison of schemas is an essential part of schema matching process. The semantic aspects or concepts hidden in the schemas can be extracted with the help of algorithms exploiting the structures of schemas or taxonomies of ontologies. These algorithms search for contextual meaning of each node with in the graph(tree) structure representing the schema/ontology.

Another aspect, quite evident in schema matching research is the use of GUI and interactive user input at pre-match, post-match and during the match process. The probabilistic, uncertainty and fuzzy logic based methods are also being exploited at data and schema level, to come up with good map results for data interoperability. Use of clustering and data mining approaches are becoming more frequent to tackle the large scale scenarios.

The *quality evaluation* of schema mapping is also an open research problem. It can be divided into two parts, *Correctness* and *Completeness*. Correctness follows the idea that the mappings discovered are correct, and completeness means every possible mapping has been discovered. The current measures utilised to evaluate the quality of a match tool, have been derived from the information retrieval domain. Specifically the *precision* measure and the *recall* measure are most widely used to verify the quality (Do et al., 2002). Some variances of recall and precision are given as *F-measure*, the weighted harmonic mean of precision and recall measures, and *Fall-out* which is the proportion of irrelevant mappings that are retrieved, out of the all irrelevant mappings available. A theoretical and empirical evaluation of schema matching measures is explained in (Euzenat, 2007; Gal, Anaby-Tevor, Trombetta, & Montesi, 2005).

6 Overview of Large Scale Schema Matching Tools

The previous surveys (Rahm & Bernstein, 2001; Shvaiko & Euzenat, 2005; Yatskevich, 2003) incorporate solutions from schema level (metadata), as well as instance level (data) research, including both database and artificial intelligence domains. Most of the methods discussed in these surveys compare two schemas and work out quality matching for the elements from source schema to target schema. Some of the tools also suggest the merging process of the schemas, based on the mappings found in match step. In this section, we review the effectiveness of schema matching tools with respect to large scale scenarios.

6.1 Tools: Matching Two Large Schemas

COMA++(Aumueller et al., 2005) is a generic, composite matcher with very effective match results. It can process the relational, XML, RDF schemas as well as OWL ontologies. Internally it converts the input schemas as graphs for structural matching and stores all the information in MYSQL as relational data. At present it uses 17 element/structure level matchers which can be selected and sequenced according to user's requirements. For linguistic matching it utilizes user defined synonym and abbreviation tables, along with n-gram name matchers. Structural matching is based on similar path and child/parent similarities.

Similarity of pairs of elements is calculated into a similarity matrix. It has a very comprehensive graphical user interface for candidate match selection and merging. For each source element, elements with similarity higher than the threshold are displayed to the user for final selection. The COMA++ supports a number of other features like merging, saving and aggregating match results of two schemas for reuse. An approach described in (Do & Rahm, 2007), uses COMA++ for matching large taxonomies by fragmenting them. The source and target schemas are broken down into fragments. Each source schema fragment is compared to the target schema fragments one by one for a possible match. Then best fragment matches are integrated to perform schema level matching. Thus, this approach provides a mechanism for large scale matching and merging of two schemas.

PROTOPLASM(Bernstein et al., 2004) target is to provide a flexible and a customizable infrastructure for combining different match algorithms. Currently CUPID (Madhavan et al., 2001) implementation and Similarity Flooding (SF) (Melnik et al., 2002) algorithms are being used as the base

matchers. A graphical interface for it has been proposed and demonstrated by the name of BizTalk Mapper (Bernstein et al., 2006). It is based on the HCI research presented in (George G. Robertson, 2005) and is very heavily dependent on microsoft technologies. PROTOPLASM supports numerous operators for computing, aggregating, and filtering similarity matrices. By using a script language, it provides the flexibly for defining and customizing the work flow of the match operators. SQL and XML schemas, converted into graphs internally, have been successfully matched.

Mork and Bernstein (Mork & Bernstein, 2004) present a case study of matching two large ontologies of human anatomy, using PROTOPLASM infrastructure. They use an extended version of hierarchical algorithm, which goes one step further than COMA++. The similarity of descendants is used to evaluate ancestor similarity, to child-grandparent level. The authors argue that the hierarchical approach produced disappointing results because of differences in context. They report that a lot of customization was required to get satisfactory results.

CLIO (Hernedez et al., 2002) has been developed at IBM. It is a complete schema mapping and management system. It has a comprehensive GUI and provides matching for XML and SQL schemas (Object Relational databases converted into relational with the help of a wrapper function). It uses a hybrid approach, combining approximate string matcher for element names and Naive Bayes learning algorithm for exploiting instance data. It also facilitates in producing transformation queries (SQL, XQuery, or XSLT) from source to target schemas, depending upon the computed mappings. Its interface gives the user the facility to augment the schema semantics or the data instance (to

support users expertise) in the pre-match phase and selection of best among the candidate matches in the post-match phase.

Another project, **ToMAS** (Velegrakis et al., 2004), has added a new module to CLIO. It arms CLIO with the capability to handle the temporal changes in already mapped schemas and produce the required changes in the existing mappings. The work also presents a simple and powerful model for representing schema changes

SCIA (Wang et al., 2007) is a semi-automatic schema mapping system for data integration. It creates executable mappings in the form of views between two schemas, similar to CLIO. It provides an automatic matching mechanism for simple element level mappings. In parallel, it finds points in the schemas, where user input is necessary. These points are computed where there exists ambiguous contextual information of a pair of matching elements. The authors research is based on the argument that human perception works well to select a better mapping from a given set of possible matches. But humans are not good and fast enough to identify mismatched portions of the schemas and matches missed by the tool. The system handles schemas as tree structures and tries to find context based matches. During the matching process, it interactively asks specific questions to resolve these problems. For example SCIA asks the user how to proceed, if no match is found for a non-leaf node, with a significantly large subtree rooted at that node.

QOM (Quick Ontology Matching) (Ehrig & Staab, 2004) is a semi-automatic ontology (RDF based) mapping tool. It uses heuristics to classify candidate mappings as promising or less promising. It uses multiple iterations, where in each iteration the number of possible candidate mappings is reduced. It

employs label string similarity (sorted label list) in the first iteration, and afterward, it focuses on mapping change propagation. The structural algorithm follows top down (level-wise) element similarity, which reduces time complexity. In the second iteration, depth-first search is used to select the appropriate mappings from among the candidate mappings. QOM has been incorporated into a complete schema matching tool called FOAM (Framework for Ontology Alignment and Mapping) (Ehrig, Euzenat, & Stuckenschmidt, 2005).

GLUE(Doan et al., 2003) is the extended version of *LSD* (Doan et al., 2001), which finds ontology/ taxonomy mapping using machine learning techniques. The system is input with set of data instances along with the source and target taxonomies. Glue classifies and associates the classes of instances from source to target taxonomies and vice versa. It uses a composite approach, as in LSD, but does not utilize global schema (as in LSD). LSD uses composite approach to combine different matchers (a meta-learner combines predictions of several machine learning based matchers).

LSD has been further utilized in *Corpus-based Matching* (Madhavan et al., 2005), which creates a corpus of existing schema and their matches. In this work, input schemas are first compared to schemas in the corpus before they are compared to each other. Another extension based on LSD is *IMAP* (Dhamankar et al., 2004). Here the work utilize LSD to find 1:1 and n:m mapping among relational schemas. It provides a new set of machine-learning based matchers for specific types of complex mappings expressions. For example, name is a concatenation of firstname and lastname. It also provides the information about the prediction criteria for a match or mismatch.

6.2 Tools: Matching and Integrating Large Set of Schemas

MOMIS(Beneventano, Bergamaschi, Guerra, & Vincini, 2001) is a heterogeneous database mediator. One of its components ARTEMIS is the schema integration tool which employs schema matching to integrate multiple source schemas into a virtual global schema for mediation purposes. The tool operates on hybrid relational-OO model. It first calculates elements similarity based on name and data type, thus acquiring all possible target elements. Further, external dictionary WordNet is utilized to compute the synonym, hypernym relationship between elements. In next step, structural similarity of elements is computed as the fraction of the neighbor elements showing name similarity exceeding a threshold over all neighbor elements. For each pair of elements, the name and structural similarity are aggregated to a global similarity using a weighted sum. According to the global similarities, similar elements are clustered using a hierarchical clustering algorithm for supporting complex match determination.

Wise-Integrator (H. He, Meng, Yu, & Wu, 2004) is a schema integration tool. It uses schema matching to find correspondences among web search forms so that they can be unified under an integrated interface. First a local interface is selected and then incrementally each input form is compared against it. The attributes without a match candidate in the local interface, are added to it. Wise-Integrator employs several algorithms to compute attribute similarity. Namely exact and approximate string matching, along with dictionary lookup for semantic name similarity. It also utilises specific rules for compatibility of data types supported by value scales/units and default values. For each pair of elements, the similarities predicted by the single criteria are simply summed to obtain a global weight. Elements showing the highest global weight exceeding a

threshold are considered matching. One of the element from each matchings pair, is selected as the global attribute to be used in the integrated interface.

DCM framework (Dual Correlation Mining) (B. He et al., 2004) objective is similar to Wise-Integrator. It focus on the problem of obtaining an integrated interface for a set of web search forms holistically. The authors observe that the aggregate vocabulary of schemas in a (restricted) domain, such as book, tends to converge at a small number of unique concepts, like author, subject, title, and ISBN; although different interfaces may use different names for the same concept. The research proposes a statistical approach, extracted from data mining domain, based on the assumptions: independence of elements, non-overlapping semantics, uniqueness within an interface, and the same semantics for the same names. The algorithm identifies and clusters synonym elements by analyzing the co-occurrence of elements in different interfaces.

PSM (Parallel Schema Matching)(Su et al., 2006), is another implementation of holistic schema matching, for a given set of web query interface schemas. The objectives are similar to DCM algorithm, but PSM improves on DCM on two things; first DCM negative correlation computation between two elements to identify synonyms may give high score for rare elements but PSM does not. And secondly the time complexity of DCM is exponential with respect to the number of elements whereas for PSM it is polynomial. PSM, first holistically detects all the distinct elements in the input schemas, assuming synonym elements do not coexist in the same schema. In second phase, it generates pairs of candidate synonym elements. This pair generation is dependent on a threshold calculated by the number of cross-occurrences (if element1 is in schema1 and element2 is in schema2 or vice versa) in different pairs of

schemas. The results of the experiments in this work show that it has the ability to find 1:1 and n:m matches quite efficiently.

ONTOBUILDER (Roitman & Gal, 2006) is a generic multipurpose ontology tool, which can be used for authoring, and matching RDF based ontologies. Its interface also supports the process of matching web search forms for generating an integrated form. OntoBuilder generates dictionary of terms by extracting labels and field names from web forms, and then it recognizes unique relationships among terms, and utilize them in its matching algorithms. The tool uses spacial attribute precedence based algorithm to calculate the semantics of each attribute in the form i.e., sequencing of concepts with in the form.

PORSCHE (Performance Oriented Schema Matching) (Saleem et al., 2008) presents a robust mapping method which creates a mediated schema tree from a large set of input XML schemas (converted to trees) and defines mappings from the contributing schema to the mediated schema. The result is an almost automatic technique giving good performance with approximate semantic match quality. The method uses node ranks calculated by pre-order traversal. It combines tree mining with semantic label clustering which minimizes the target search space and improves performance, thus making the algorithm suitable for large scale data sharing. The technique adopts a holistic approach for similar elements clustering in the given set of schemas and then applies a binary ladder incremental (Batini et al., 1986) schema match and integrate technique to produce the mediated schema, along with mappings from source schemas to mediated schema.

Bellflower is a prototype implementation for work described in (Smiljanic et al., 2006). It shows how personal schema for querying, can be efficiently matched and mapped to a large repository of related XML schemas. The method identifies fragments with in each schema of the repository, which will best match to the input personal schema, thus minimizing the target search space. Bellflower uses k-means data mining algorithm as the clustering algorithm. The authors also demonstrate that this work can be implemented as an intermediate phase with in the framework of existing matching systems. The technique does produce time efficient system but with some reduction in quality effectiveness.

6.3 Summarizing the Tools

In tables 4 and 5 we give a quick comparison of the above discussed schema matching tools and prototypes. The comparison in table 4 has been devised to give a general outlook of tools, highlighting the use of GUI, match cardinality supported by the tool, use of external oracles and related application domains. Whereas table 5 gives much deeper insight into the algorithms used by the tools with respect to input, output and execution aspects.

The analysis of the prototype tools for schema matching or ontology alignment domains shows that most of the techniques used are the same among them. For example, two most cited schema matching tools PROTOPLASM and COMA++ follow similar match characteristics and architecture, the only difference is that PROTOPLASM framework is hybrid in nature whereas COMA++ is composite, thus providing more flexibility. The tools adopt hybrid approach for better and automatic approach. Structure level matching has been adopted by all, except for some web search interface schema integrators, since query form field atributes follow more of a sequence than hierarchical structure. For semantic comparison of element labels, external oracle like

Table 4. Schema Matching Tools and Prototypes Comparison - General

Tool	GUI	Approach	Card.	Ext Orc	Internal Rep	Research Domain
BELLFLOWER	No	Hybrid	1:1	-	Directed Graph	Schema Matching
CLIO	Yes	Hybrid	1:1	-	Rel. Model, Directed Graph	Schema Matching, Mapping Evolution
COMA++	Yes	Composite	1:1	Dom Syn, Abr Thesuri	Directed Graph	Schema Matching and Merging
DCM	No	Hybrid	n:m	-	-	Schema Integration
GLUE	No	Composite	n:m	-	Attribute based	Data Integration
MOMIS	Yes	Hybrid	n:m	Thesuri	Directed Graph	Schema Integration
ONTO BUILDER	Yes	Hybrid	1:1, 1:n	-	Graph	Create/Match Ontologies
PORSCHE	No	Hybrid	1:1,1:n	Dom Syn, Abr Thesuri	Tree	Schema Integration and Mediation
PROTOPLASM	Yes	Hybrid	1:1	Wordnet	Graph	Schema Matching
PSM	No	Hybrid	n:m	-	-	Schema Integration
QOM	No	Hybrid	1:1	Dom. Thesuri	Tree	Ontology Alignment
SCIA	Yes	Hybrid	n:m	Thesuri	Tree, Graph	Data Integration
WISE INTEGRATOR	Yes	Hybrid	1:1	General Thesuri	Attribute based	Web Search form Integration

WordNet dictionary or reference domain ontology is quite frequent. The notion of neighborhood likelihood for next possible match is followed by most of the major matching tools e.g., PROTOPLASM, MOMIS, QOM, SCIA and GLUE. This feature is also intuitively used for search space optimization. Another characteristic for search space optimization in large scale scenario is clustering of elements/schemas, showing some similarity at the pre-processing level e.g., element name similarity based on edit distance or synonymous meaning, demonstrated in XClust (M.-L. Lee et al., 2002), PORSCHE and QOM.

It appears that the most prototypes aim to provide good quality matchings, with lack in time performance. Today, the application domains like the genomic or e-business, deal with large schema. Therefore the matching tool should also provide good performance and if possible automatic mapping generation. In future, matching systems should try to find a trade off between quality and performance. A recent work in this domain has been proposed in

Table 5. Schema Matching Tools and Prototypes Comparison - Strategy based

Tool	Input	Output	Match Algorithms (Level wise)			
			Element			Stucture/(Data Ins.)
			Str.	*Ling.*	*Const.*	
BELLFLOWER	XSD	Schema Matches	Yes	-	-	K-means data mining
CLIO	SQL,XSD	Mappings (Query)	Yes	-	Yes	(Naive Byes Learner)
COMA++	XSD,XDR, RDF,OWL	Mappings, Merged Schema	Yes	Yes	Yes	Path: biased to leaf nodes
DCM	Web Query Interface	Mappings between all input schemas	Yes	-	Yes	Correlational Mining
GLUE	DTD,SQL, Taxonomy	Mappings, IMap functions	Yes	-	Yes	(Whirl/Bayesian Learners)
MOMIS	Rel,OO data model	Global View	Yes	Yes	Yes	Schema Clustering, Neighborhood Affinity
ONTO BUILDER	RDF	Mediated Ontology	Yes	Yes	-	Elements Sequencing
PORSCHE	XSD Instance	Mediated Schema	-	Yes	-	Elements Clust, Tree Mining
PROTOPLASM	XDR, SQL,RDF	Mappings	Yes	Yes	Yes	Path (Parent,Child,Grand Child), Iterative Fix Point Computation
PSM	Web Query Interface	Mappings between all input schemas	Yes	-	Yes	Correlational Mining
QOM	RDF(S)	Mappings	Yes	-	Yes	Neighborhood Affinity, Taxonomic Structures
SCIA	Rel,DTD, XSD,OWL	Mappings (Query)	Yes	Yes	Yes	Iterative Fix Point Computation, Path
WISE INTEGRATOR	Web Query Interface	Integrated Schema	Yes	Yes	Yes	Clustering

(Duchateau, Bellahsene, & Colleta, 2008), which uses decision tree concept based on machine learning algorithm.

7 Conclusion and Perspective

In this paper we provide a broad overview of the current state of the art of schema matching, in the large scale schema integration and mediation for data interoperability. The paper also tries to provide an insight on current emergent technologies driving the match research, like data mashups and P2P database networks.

We have seen in this study that although schema matching has passed its teen ages, there are issues that still require to be investigated. These are harnessed

by the dynamic nature of today's application domains. Future prospective of schema matching is in the large scale level, which is mainly related to schemas and ontologies in P2P, data grids, agents and web services based networks. We conclude our discussion by enumerating some explicit future research concerns in the field of schema matching and integration.

- Maintenance of mappings with schema evolution.
- Visualization of mappings in multi-schema (more than 2) integration.
- Development of correctness/completeness metrics and benchmark tools for evaluating schema matching systems.
- Self-tunning of the matching tools, providing a balance between the quality and the performance aspects.

References

An, Y., Borgida, A., Miller, R. J., & Mylopoulos, J. (2007). A semantic approach to discovering schema mapping expressions. In *Intl. conf. on data engineering.*

Aumueller, D., Do, H. H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with coma++. In *Acm sigmod* (p. 906-908).

Bachlechner, D., Siorpaes, K., Fensel, D., & Toma, I. (2006). *Web service discovery - a reality check.* (Tech. Rep.). Digital Enterprise Research Institute (DERI).

Batini, C., Lenzerini, M., & Navathe, S. B. (1986). A comparitive analysis of methodologies for database schema integration. *ACM Computing Surveys*, *18*(4), 323-364.

Beneventano, D., Bergamaschi, S., Guerra, F., & Vincini, M. (2001). The momis approach to information integration. In *Iceis* (p. 194-198).

Benkley, S., Fandozzi, J., Housman, E., & Woodhouse, G. (1995). Data element tool-based analysis (delta). In *Mtr.*

Bernstein, P. A., Melnik, S., & Churchill, J. E. (2006). Incremental schema matching. In *Vldb.*

Bernstein, P. A., Melnik, S., Petropoulos, M., & Quix, C. (2004). Industrial-strength schema matching. *ACM SIGMOD Record*, *33*(4), 38-43.

Besana, P., Robertson, D., & Rovatsos, M. (2005). Exploiting interaction contexts in p2p ontology mapping. In *P2pkm.*

Bex, G. J., Neven, F., & Vansummeren, S. (2007). Inferring xml schema definitions from xml data. In *Vldb.*

Bilke, A., & Naumann, F. (2005). Schema matching using duplicates. In *Intl. conf. on data engineering.*

Bohannon, P., Elnahrawy, E., Fan, W., & Flaster, M. (2006). Putting context into schema matching. In *Vldb.*

Chiticariu, L., Kolaitis, P. G., & Popa, L. (2008). Interactive generation of integrated schemas. In *Acm sigmod.*

Dalamagasa, T., Chengb, T., Winkelc, K.-J., & Sellisa, T. (2006). A methodology for clustering xml documents by structure. *Information Systems (Elseveir)*, *31*, 187228.

Davis, M. (2006). Semantic wave 2006 - a guide to billion dollar markets - keynote address. In *Stc.*

Dhamankar, R., Lee, Y., Doan, A., Halevy, A., & Domingos, P. (2004). imap: Discovering complex semantic matches between database schemas. In *Acm sigmod.*

Do, H. H., Melnik, S., & Rahm, E. (2002). Comparison of schema matching evaluations. In *Web, web-services, and database systems workshop.*

Do, H.-H., & Rahm, E. (2007). Matching large schemas: Approaches and evaluation. *Information Systems*, *32*(6), 857-885.

Doan, A., Domingos, P., & Halevy, A. Y. (2001). Reconciling schemas of disparate data sources - a machine learning approach. In *Acm sigmod.*

Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., & Halevy, A. Y. (2003). Learning to match ontologies on the semantic web. *VLDB J.*, *12*(4), 303-319.

Duchateau, F., Bellahsene, Z., & Colleta, R. (2008). *Matchplanner.* (Tech. Rep.). Hal-LIRMM.

Duchateau, F., Bellahsene, Z., & Roche, M. (2007). A context-based measure for discovering approximate semantic matching between schema elements. In *Ieee rcis.*

Ehrig, M., Euzenat, J., & Stuckenschmidt, H. (2005). Framework for ontology alignment and mapping - results of the ontology alignment evaluation initiative. In *Workshop on integrating ontologies.*

Ehrig, M., & Staab, S. (2004). Qom - quick ontology mapping . In *Iswc.*

Embley, D. W., Xu, L., & Ding, Y. (2004). Automatic direct and indirect schema mapping: Experiences and lessons learned. *ACM SIGMOD Record*, *33*(4), 14-19.

Euzenat, J. (2007). Semantic precision and recall for ontology alignment evaluation. In *Intl. joint conf. on artificial intelligence.*

Euzenat, J., et al. (2004). *State of the art on ontology matching* (Tech. Rep. No. KWEB/2004/D2.2.3/v1.2). Knowledge Web.

Gal, A. (2006a). Managing uncertainty in schema matching with top-k schema mappings. *JoDS*, 90-114.

Gal, A. (2006b). Why is schema matching tough and what can we do about it. *SIGMOD Record*, *35*(4).

Gal, A., Anaby-Tevor, A., Trombetta, A., & Montesi, D. (2005). A framework for modeling and evaluating automatic semantic reconciliation. *VLDB J*, *14*(1), 50-67.

Gangemi, A., Guarino, N., Masolo, C., & Oltramari, A. (2003). Sweetening wordnet with dolce. *AI Magzine*, *24*(3), 13-24.

George G. Robertson, J. E. C., Mary P. Czerwinski. (2005). Visualization of mappings between schemas. In *Acm chi.*

Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-match: an algorithm and an implementation of semantic matching. In *European semantic web symposium.*

Guedria, W., Bellahsene, Z., & Roche, M. (2007). A flexible approach based on the user preferences for schema matching. In *Ieee rcis.*

Haas, L. M. (2007). Beauty and the beast: The theory and practice of information integration. In *Intl. conf. on database theory.*

Halevy, A., Ives, Z., Suciu, D., & Tatarinov, I. (2003). Schema mediation in peer data management systems. In *Intl. conf. on data engineering.*

Halevy, A. Y., Rajaraman, A., & Ordille, J. J. (2006). Data integration: The teenage years. In *Vldb.*

He, B., Chang, K. C.-C., & Han, J. (2004). Discovering complex matchings across web query interfaces: a correlation mining approach. In *Acm kdd* (p. 148-157).

He, H., Meng, W., Yu, C. T., & Wu, Z. (2004). Automatic integration of web search interfaces with wise-integrator. *VLDB J.*, *13*(3), 256-273.

Hernedez, M. A., Miller, R. J., & Haas, L. M. (2002). Clio: A semi-automatic tool for schema mapping. In *Acm sigmod.*

Hu, W., Zhao, Y., & Qu, Y. (2006). Partition-based block matching of large class hierarchies. In *Asian semantic web conf.*

Huhns, M. N., & Singh, M. P. (2005). Service-oriented computing: Key concepts and principals. *IEEE Internet Computing*.

Jhingran, A. (2006). Enterprise information mashups: Integrating information, simply - keynote address. In *Vldb.*

Lee, M.-L., Ling, T. W., Lu, H., & Ko, Y. T. (1999). Cleansing data for mining and warehousing. In *Intl. conf. on database and expert systems applications.*

Lee, M.-L., Yang, L. H., Hsu, W., & Yang, X. (2002). Xclust: clustering xml schemas for effective integration. In *Acm cikm* (p. 292-299).

Lee, Y., Sayyadain, M., Doan, A., & Rosenthal, A. S. (2007). etuner: tuning schema matching software using synthetic scenarios. *VLDB J.*, *16*, 97-122.

Loser, A., Siberski, W., Sintek, M., & Nejdl, W. (2003). Information integration in schema-based peer-to-peer networks. In *Caise.*

Lu, J., Wang, S., & Wang, J. (2005). An experiment on the matching and reuse of xml schemas. In *Intl. conf. on web engineering.*

Madhavan, J., Bernstein, P. A., Doan, A., & Halevy, A. Y. (2005). Corpus-based schema matching. In *Intl. conf. on data engineering* (p. 57-68).

Madhavan, J., Bernstein, P. A., & Rahm, E. (2001). Generic schema matching with cupid. In *Vldb* (p. 49-58).

Manakanatas, D., & Plexousakis, D. (2006). A tool for semi-automated semantic schema mapping: Design and implementation. In *Disweb workshop caise.*

McCann, R., Shen, W., & Doan, A. (2008). Matching schemas in online communities: A web 2.0 approach. In *Intl. conf. on data engineering.*

Meddour, D.-E., Mushtaq, M., & Ahmed, T. (2006). Open issues in p2p multimedia streaming. In *Multicomm.*

Melnik, S., Garcia-Molina, H., & Rahm, E. (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Intl. conf. on data engineering* (p. 117-128).

Melnik, S., Rahm, E., & Bernstein, P. A. (2003). Developing metadata-intensive applications with rondo. *J. of Web Semantics*, *I*, 47-74.

Meo, P. D., Quattrone, G., Terracina, G., & Ursino, D. (2006). Integration of xml schemas at various "'severity"' levels. *Information Systems J.*, 397-434.

Milo, T., & Zohar, S. (1998). Using schema matching to simplify heterogeneous data translation. In *Vldb* (p. 122-133).

Mitra, P., Noy, N. F., & Jaiswal, A. R. (2005). Omen: A probabilistic ontology mapping tool. In *Intl semantic web conf.* (p. 537-547).

Mork, P., & Bernstein, P. A. (2004). Adapting a generic match algorithm to align ontologies of human anatomy. In *Intl. conf. on data engineering.*

Niles, I., & Pease, A. (2003). Towards a standard upper ontology. In *Fois.*

Noy, N. F., Doan, A., & Halevy, A. Y. (2005). Semantic integration. *AI Magazine*, *26*(1), 7-10.

Noy, N. F., Kunnatur, S., Klein, M., & Musen, M. A. (2004). Tracking changes during ontology evolution. In *Intl. semantic web conf.*

Parent, C., & Spaccapietra, S. (2000). Database integration: The key to data interoperability. In M. P. Papazoglou, S. Spaccapietra, & Z. Tari (Eds.), *Advances in object oriented modeling.* The MIT Press.

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *VLDB J.*, *10*(4), 334-350.

Roitman, H., & Gal, A. (2006). Ontobuilder: Fully automatic extraction and consolidation of ontologies from web sources using sequence semantics. In *Edbt workshops.*

Saleem, K., Bellahsene, Z., & Hunt, E. (2008). Porsche: Performance oriented schema mediation. *to appear in Information Systems (Elseveir).*

Sarma, A. D., Dong, X., & Halevy, A. (2008). Bootstrapping pay-as-you-go data integration systems. In *Acm sigmod.*

Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. *J. Data Semantics IV*, 146-171.

Smiljanic, M., Keulen, M. van, & Jonker, W. (2006). Using element clustering to increase the efficiency of xml schema matching. In *Workshop intl. conf. on data engineering.*

Spaccapietra, S., Parent, C., & Dupont, Y. (1992). Model independent assertions for integration of hetrogeneous schemas. *VLDB J.*, 81-126.

Su, W., Wang, J., & Lochovsky, F. (2006). Holistic query interface matching using parallel schema matching. In *Intl. conf. on data engineering.*

Velegrakis, Y., Miller, R., & Popa, L. (2004). On preserving mapping consistency under schema changes. *VLDB J*, *13*(3), 274-293.

Wang, G., Zavesov, V., Rifaieh, R., Rajasekar, A., Goguen, J., & Miller, M. (2007). Towards user centric schema mapping platform. In *Vldb workshop semantic data and semantic integration.*

Wu, W., Doan, A., & Yu, C. (2005). Merging interface schemas on the deep web via clustering aggregation. In *Intl. conf. on data mining.*

Yatskevich, M. (2003). *Preliminary evaluation of schema matching systems* (Tech. Rep. Nos. DIT-03-028, Informatica e Telecomunicazioni). University of Trento.