

New Challenges : Large Scale Automatic Semantic Integration

Khalid Saleem, Zohra Bellahsene

▶ **To cite this version:**

Khalid Saleem, Zohra Bellahsene. New Challenges : Large Scale Automatic Semantic Integration. 2007. lirmm-00170765v1

HAL Id: lirmm-00170765

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00170765v1

Submitted on 10 Sep 2007 (v1), last revised 16 Apr 2008 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Challenges : Large Scale Automatic Semantic Integration

Khalid Saleem[1] and Zohra Bellahsene[1]

LIRMM - UMR 5506 CNRS University Montpellier 2,
161 Rue Ada, F-34392 Montpellier
{saleem, bella}@lirmm.fr

Abstract. *Today schema matching is a basic problem 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 summerize 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. Today data is viewed as a semantic entity, motivating new algorithms and strategies. The evolving large scale distributed information systems are further pushing the schema matching research to utilize the processing power not available in the past. Thus directly increasing the industry investment proportion in the matching domain. This article reviews the latest application domains in which schema matching is being utilized. It discusses the shift from manual to automatic schema matching, along with new definition of match covering the automation objectivity. Another panorama which is covered by this survey is the schema matching classification in the context of latest strategies and algorithms in the field of schema based information retrieval and management.*

Keywords: XML schema tree; schema matching; schema mapping; schema integration; schema evolution; large scale

1 Introduction

Over the years technology has made this world a web of digital information, where digital systems are appearing at an exponential rate. At individual level, personal or professional, or organisational level, there exists an unending list of digital devices cooperating together to solve problems. Every day a new gadget hits the market, creating a ripple-effect in its surrounding operating environment. Thus giving rise to new innovations in the field around it. For us, the database people, it is like 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 [59]. Thus the domain of data interoperability have also evolved with emergence of new devices and systems.

Every device has to know the meaning encoded in the input data, referred to as structure of data, giving rise to the use of term *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 , at conceptual level we have the entity relationship diagram and physical database schema having tables and fields at physical level. Thus for a computer application, schema gives the best way to understand the semantics of the underlying data instances.

Matching schemas for data interoperability 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. The matching activity found the new dimension, with the separation of metadata information of the data from real data instance, 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.

By definition, schema matching is the task of discovering correspondences between semantically similar elements of two schemas or ontologies [16, 47, 53]. In this paper our emphasis is on schema matching, if other wise explicitly mentioned e.g., ontology alignment. Basic syntax based match definition has been discussed in the survey by Rahm and Bernstien [60], extended by Shvaiko and Euzenat in [63] with respect to semantic aspect. In this article, we provide a new version of schema match definition which encompasses the problem of automatic large scale mapping, also incorporating the previous ideas of matching. In our text, we try to 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 [14][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.

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

Contributions: Our contributions in this paper are related to large scale match and integrate scenario, with the need for automation.

- Provide a new definition for schema matching supporting schema semantics acquisition at structural level.
- Give the user, the latest trends of application development in the field of large scale schema matching and integration e.g. data mashups, schema based P2P database systems, web services and query answering.
- Outline the aspects missing till to date in schema matching research; like visualization of mappings, use of clustering and mining techniques in the large scale schema matching scenario.
- Present a taxonomy of schema matching strategies with respect to large scale scenario, from the input, output and execution perspective.
- Survey of 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. In section 3, we give the new definitions for the domain of schema matching and integration. Sections 4 outlines with brief explanations about the current application research domains with respect to large scale data interoperability. Section 5 enumerates the basic schema matching algorithms at element and structure level, along with explanation of strategies for large scale schema matching. Section 6 outlines a comparison of some current tools and prototypes in schema matching domain. Section 7 concludes the paper, giving an outline of the future research perspectives in schema matching.

2 Motivation

2.1 The Three Dimensions of Schema Matching

Schema Matching has been researched from various perspectives by researchers belonging to different domains. While reviewing the plethora of approaches and techniques, we came to understand that schema matching is related to three different interlinked dimensions; Application Domain, Research Domain and Basic Match Research. 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.

Basic match techniques (Table 1)revolve around the granularity aspect of schema. It can be seen as the match algorithm development process for a certain entity in the schema, which can be the most basic constituent of schema e.g., field in a relational database schema table [6, 11], or the whole schema structure itself exploited, using some graph techniques [17, 47, 51]. A combination of basic match techniques are utilized to resolve problems indicated in Table 2.

In research (data interoperability) domain (Table 2), Schema Integration [4] can follow an incremental (binary/clustering) [62, 43] or a holistic approach [36].

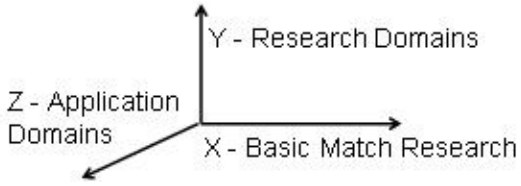


Fig. 1. Schema Matching Dimensions

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

Generating schema mapping expressions and their updating with changes in schemas is a highly demanding research domain [1, 67]. The cardinality of mapping demonstrate the numeric relationship of element correspondences [60] i.e., 1:1, 1:n, n:1 and n:m element map cardinality. Ontologies are used for knowledge representation and are similar to schemas to a certain extent; as both describe a domain with the help of terms with constrained meanings [63].

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	
Match Quality Evaluation	

The most challenging research domain has been the match quality evaluation. Techniques like recall and precision [25, 27], borrowed from information retrieval domain, have been tailored to suit the needs of schema matching but still require a lot of work.

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 Ware-

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

housing, Message Translation, E-commerce [60], 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 [63], 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 4.

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:

- Creating a merged schema for data integration from *two large size schemas* (with thousands of nodes). For example Bio-genetic taxonomies [17]
- Creating a mediated schema from *a large set of schemas* (with hundreds of schemas and thousands of nodes) in the same application domain for querying purposes. For example creating a mediated web interface input form (schema) from the hundreds of web interface forms (schemas) related to travel domain [36, 68]

For the first type of problem, the tools developed to date are applicable with modifications as demonstrated in [17] and [55]. Research in [17] breaks down the bio-genatic taxonomies into fragments and apply their matching tool COMA++ [2] on pairs of these fragments to find similarities between the two taxonomies. Whereas, work in [55] uses three levels of matching; using CUPID [47] for lexical analysis of nodes using external oracles, then applying Simi-

larity Flooding [51], 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.



Fig. 2. Query Interfaces Over the Web for Travel

2.3 Motivating Example

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 [51], clustering [52, 64, 68] and mining [36, 62] 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 [36, 68].

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. (S)He 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 his/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 utilization by other users with similar requirement. The web query interfaces follow a hierarchical tree like structure as shown in figure 2 for travel domain taken from TEL-8 dataset [2].

3 Preliminaries

We follow the work in [63], molded to our perceptions. The problem presented is matching, integration and mediation of more than two schemas with no previous match reference. The schema characteristics followed by us are:

- Schema are related to same conceptual knowledge domain
- Schema provide the elements' labels and the constraints information (element level and structure level)

Structural constraints exploitation has become a major research avenue in schema/ontology matching.

3.1 Definitions

Schema matching has evolved over the period of time depending upon the applications and technologies utilizing it, as listed in Table 3. Here we present the definitions from the perspective of large scale automatic semantic matching, integration and mapping. Automatic schema matching has been explained in [60] and extended in [63]. But their focus has been a match between two schemas.

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

Semantic matching requires the comparison of concepts structured as schema elements. Labels naming the schema elements can be considered as concepts and each element's contextual placement information in the schema further elaborates the semantics of the concept. Let us consider a XML schema instance as a XML tree, the combination of the node label, the structural placement of the node and other constraints information like node data type, range of values, gives the concept at that node. Following definitions have been chalked out bounded by the above mentioned requirements.

Definition 1 (Schema): A schema $S = (V, E)$ is a rooted, labeled graph, consisting of nodes/elements $V = \{0, 1, \ldots, n\}$, and edges $E = \{(x, y) \mid \text{x,y} \in V\}$. One distinguished node $r \in V$ is called the root, and for all $x \in V$, there is a unique path from r to x. Further, lab:V $\rightarrow$ L is a labeling function mapping nodes to labels in $L = \{l_1, l_2, \ldots\}$ and ctr:V $\rightarrow$ R is a constraint function, mapping nodes to node level constraints in schema $R = \{r_1, r_2, \ldots\}$. For example constraint can be a data type of the element .

Definition 2 (Node Semantics): Node semantics of node x, Sem_x, combines the semantics of the node label C_{l_x} [31] with its contextual placement in the schema structure $Context_x$, along with a set of node specific constraints R_x. These can be viewed as structural constraints, for example class relationship with another element. Node Semantics are given via a composition function $NodeSem$, as
$Sem_x : x \rightarrow NodeSem(C(l_x), Context_x, R_x)$.

Definition 3 (Label Semantics) Label semantics corresponds to the conceptual meaning of the label (irrespective of the node it is related to). A label l is a composition of m strings, called tokens. $tok : L \rightarrow T$ is a tokenization function which maps a label to a set of tokens T=$\{t_1, t_2, \ldots, t_m\}$. Tokenization [31] helps in establishing similarity between two labels.

Thus label concept is a composition of concepts attached to the tokens making it up. Let P is a composition function, which combines the concepts of q tokens making up the label: $C(l) = P(C(t_1), C(t_2), \ldots, C(t_q))$.

Example 1: For labels 'IssuedAt' and 'IssuedOn', lemma for the token 'Issued' is the same, 'issue (verb)', but the tokens 'At' and 'On' the lemmas are different. Using some external natural language oracles, one can infer different semantics for the two labels. For label 'IssuedAt', the reference is to a place, whereas 'IssuedOn' refers to a date. If the two labels are 'IssueAt' and 'IssueOn', the semantics may be different, as lemma 'issue' is a noun and not a verb.•

Token Semantics discovery is the process of lemmatization and elimination using some linguistic oracle [63].

Example 2: For labels 'DateOfBirth', tokenized as {date, of, birth}, and 'BirthDate', tokenized as {birth, date}, we proceed as follows. Since 'of' is a preposi-

tion, it is discarded. As the remaining tokens are identical, we obtain 100 percent similarity for the two labels.•

Definition 4 (Schema Concepts): Schema concepts can be formally considered as a set of semantics of nodes of schema. For schema S with set of nodes V, set of schema concepts S_C is given as : $S_C = \{Sem_{x_1}, Sem_{x_2}, \ldots, Sem_{x_n}\}$ where $1 \leq n \leq |V|$.

Definition 5 (Semantic Similarity Measure): Let V_1 be the set of elements in schema, S_1, and V_2 be the set of elements in schema, S_2. Semantic similarity of two elements $x_1 \in V_1$ and $x_2 \in V_2$ is based on the likeness of their meaning/semantics, given as a couple (d,k) i.e., degree of similarity d and similarity type k. Degree of similarity is taken as $M_S(Sem_{x_1}, Sem_{x_2})$. It is given by:
$M_S : V_1 \times V_2 \rightarrow [0, 1]$, where a zero means complete dissimilarity and 1 semantic equality. So, $d_{x_1x_2} = M_S(Sem_{x_1}, Sem_{x_2})$. Where as $k \in K$ a set of possible correspondence types e.g., inclusion, synonym, string equivalence etc.

Definition 6 (Similarity Threshold): A similarity threshold $t \in [0, 1]$ is a value used for detecting a good match. It can be set by some external factor like an expert or automatically adjusted depending upon the strategy, domain or algorithms used for matching, keeping in view the similarity type k. If degree of similarity $d_{x_1x_2} \geq t$, it is considered to be a good match.

Example 3: If *M_S(dept, department)* is calculated using inverted weighted edit distance (defined as $1 - (ed/max(|x_1|, |x_2|))$) [3], *d=0.4* and if 3-gram[4] distance is used, *d=0.125*. Thus showing that threshold value fixing is crucial. •

Definition 7 (Best Match): Matching process can detect more than one match, good or bad, for an element from schema S_1 to elements in schema S_2. Automatic Schema Matching requires some method to detect the most convincing match among them i.e., the match with the highest semantic similarity. This is formally defined as: $BestM_S(x_1, V_2) = \{x_i \in V_2 : \bigwedge_{x_j \in V_2} d_{x_1x_j} \leq d_{x_1x_i}\}$.

Given $V_{ij} \subseteq V_2$ of size n, such that $\forall\ x_{ij} \in V_1$ corresponding to element x_i, $M_S(x_i,x_{ij})$ is good match; where $1 \leq j \leq n$. Best match for element x_i of V_1 noted as $match_{ib}$ is given as following:
$match_{ib} = \max_{j=1}^{n}(d_{x_ix_{ij}})$

Definition 8 (Schema Map Expression): Assume I is a set of mappings identifiers, F_s is the composition of similarity measures, and K is a set of similarity types, as defined by the data model used for schemas S_1 and S_2. We define a mapping, *Map*, as follows:
Map: $I \times V_1 \times V_2 \times Fs \rightarrow I \times V_1 \times V_2 \times [0, 1] \times K$.

[3] ed = min no of inserts, deletes or replacements needed to transform x_1 into x_2
[4] 3-gram distance = the number of shared 3-grams for x_1 and x_2, divided by the number of 3-grams in the longer word

$Map(id, x_1, x_2, M_S) = (id, x_1, x_2, d, k)$ represents a mapping composed of tuples consisting of an id, pairs of mapped nodes annotated with their degree of similarity, and correspondence type. Schema mapping can be unidirectional, from S_1 to S_2, or bidirectional, where the correspondence holds in both directions [17] managed by the element k in the tuple.

Definition 9 (Schema Mediation) : For a set of given schemas $SSet$ with size u, schema mediation can be defined as a couple of mediated schema S_M and set of mapping identifiers I, (S_M,I). S_M can be considered as an integrated schema, which is a composition of all distinct concepts in $SSet$ and I identifies the mappings between S_M and the individual schema from $SSet$. Mediated schema concepts composition is formally given as $S_{mC} = \bigcup_{i=1}^{u}(S_{iC})$, which includes all distinct concepts in each schema S_i.

The above set of definitions formalize the schema matching and its application in large scale schema integration and mediation scenario.

4 Current Application Domains

Work in schema matching was initiated by 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 [4, 65]. 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 [60], E-commerce, B2B, B2C [63] 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 [26]. But the core problems faced in the changing world for communication and integration are the same, whether it is ontologies or schemas [33, 35] .

The latest trends in applications development requiring data interoperability can be explicitly attributed to the technologies harnessing the web. For example ontologies alignment [26], integration of XML data on the Web [52] etc. In the subsequent subsections we give the current application domains motivating our work on schema matching.

4.1 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. Once this feet was achieved, the need

arised for integration of one e-business with other e-businesses. This gave rise to the web service concept, 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 [40] for 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 [3]. 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.

4.2 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 [35]. At the enterprise level, the mashup idea helps in building situational applications, for some transient need in the enterprise more quickly, 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) [41]. 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.

4.3 Schema based P2P Database Systems

One of the latest trend in databases over the web has been *P2P Databases* [34]. There have been numerous successful P2P systems delivered in the last couple of years. The dynamic nature of P2P networks make them very flexible. 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. Today, the P2P technology has transformed into sharing of any kind of data, whether it is semi structured XML data or continuous multimedia streaming [49]. 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. For querying these data sources some matching method is required to broker between their structures, giving rise to the new generation of application research of schema based P2P data sharing systems [45].

4.4 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 un-structured data, posing a big challenge for searching over it. Deep Web [36] 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 search and integrate the results.ling and lodging are inter-linked for tourist information web sites. Continuous addition of new content further complicates the problem for search and integrate the results.

Google Base is an attempt toward solving some of these problems. It provides standard annotation to the entities (anything) available on the web, done by the entity provider itself by uploading the entity information on the google provided form. It can provide a larger base for query, but still cannot guaranty the right and latest results for the query.

4.5 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 [63]. For subsequent encounters the agents may utilize the mappings discovered and stored within them. To answer the request 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.

The above set of application domains have one thing in common, the information they encounter is dynamic, changing over time and the scale of data they encounter is enormous. Something not achievable with research revolving around semi-automatic matching and integration approach.

5 Schema Matching

This section gives an overview of the basic techniques used in the schema matching and describes the strategies for large scale matching and integration research.

5.1 Basic Match Techniques

Till to date a number of algorithms have been devised and implemented for finding correspondences between schemas. These algorithms have been dependent on some basic techniques of element level string matching, linguistic similarities or constraints likeliness at element level or higher schema structure level. Graph algorithms utilized in schema matching is a special form of constraints matching [63] for managing structural similarity. In some cases, these algorithms are further supported by data instances of schemas.

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 [20]. These approaches include string prefix, suffix comparisons, soundx similarities and more sophisticated algorithms based on hamming distance technique. There is a large list of these algorithms with various variations researched over time. The mainly talked about approaches are the n-gram[5] and the edit distance[6].

Linguistic techniques are based on the tokenization, lemmatization and elimination. The idea is to extract basic sense of the word used in the string. And then find its contextual meaning [12, 20] 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 [29], domain specific ontologies or upper level ontologies [56].

[5] Google use n-gram for statistical machine translation, speech recognition, spelling correction, information extraction and other applications
[6] Listing with detail available at http://www.dcs.shef.ac.uk/ sam/stringmetrics.html

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 e.g., 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 concept of an element, relative to its surrounding elements for matching. This research domain is further elaborated in the next section of structure level schema matching.

Structure Level
Structure level matching is refered as matching a combination of elements from one schema to another schema [60]. The algorithms developed are based on graph matching research. It can also utilize external oracles like known patterns [23], ontologies [19] or Corpus of structures [46] 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) these structures are flexible enough to handle the possible input schema elements and perform mapping. Nearly all schema match research projects based on graphs, use the notion of neighborhood affinity to elevate the similarity match value for individual elements. This aspect has been presented in similarity flooding algorithm [50]. In large scale scenarios, structure level matching techniques help in enhancing the performance of the match implementations, by using neighborhood search algorithms [22]. Our notion of matching is similar, as given in section 3, semantics extraction from structure. In litrature holistic [36] or level-wise algorithms (children-parent relationships) [47, 17] have been used to determine the correspondences among structures of schemas.

Another variation of structure level matching is based on taxonomy of ontologies. For example *bounded path matching* [22] 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* [26] 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 [31] demonstrates the effectiveness of this technique but with worst time performance, when compared to other available tools.

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 as in case of semi-structured data, instance data is used to create a representation of the data [10]. Even if the schema is available, data instances can augment the schema matching by giving more insight about the schema element semantics [38]. 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. 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 [42]. In second case data instances are used in schema matching for training machine learning algorithms. In [19], 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 extended research [15], n:m mappings are predicted, involving arithmetic functions, like totalprice is equal to price+(1+taxrate). It uses an external global domain ontology to first map the elements and then by using data instances find the function.

In the dynamic environment where the load of schemas itself is quite large, data instance approach is difficult to implement because of its drawbacks.

5.2 Match Strategies

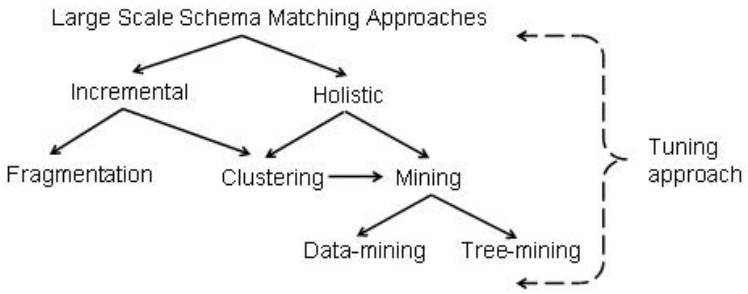


Fig. 3. Taxonomy for Large Scale Schema Matching and Integration Strategies

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

domain. Researchers have adopted different strategies depending on application domain or researcher's objectives. The strategy is basically governed by the Input, Execution and Output aspects of the match tools.

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 can be stated as if the tool is *Manual vs Semi-Automatic vs Automatic*, incorporated in which application domain. For example Web services will require an automatic environment and comparison of two large bio-genetic ontologies can be worked out with a semi-automatic tool.

The execution part is responsible for rest of the categorizations of schema matching tools. Namely, *Internal vs External* [60], *Syntactic* vs *Semantic* [63] and *Hybrid* [47, 31] vs *Composite* [17] 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 [44, 54] the already existing results automatically. In some work, performance with approximate mapping is being preferred over exact mapping [36]. Some known approaches are:

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 used widely in E-business solutions and other data sharing applications over the web. With it the concepts of distributed schemas and namespaces have also emerged over time, to introduce more complexity to the matching problem. Research work in [17] 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 full 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++ [2] 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 [39], 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.

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 [64] give a generic approach using the element level clustering approach 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 [62], element clustering is applied at the holistic level of schemas. The work is directed toward large scale schema integration. An initial set of clusters, each cluster having linguistically similar label elements, is created. 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 [43], 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 own until one global DTD has been created. l one global DTD has been created.

Here we have given some examples of use of clustering in schema matching. A very comprehensive work on XML schema clustering techniques is given in [13].

Data Mining Approach
Data Mining is the technique for finding similar patterns in large data sets. It has very recently been used as schema matching method. Work in [36, 66] highlight this method for matching and integrating deep web schema interfaces. [36] uses a positive correlational algorithm based on schema heuristics. Whereas [66] 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. [62] demonstrates a method which combines the element clustering and a tree mining method. And providing 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. Details of these methods are given in section 6.

Enhancing Match Strategies and Results
There has 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

results quality has been said to be approximate [60, 57, 63]. 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 [44] provides a framework capitalizing on instance based machine learning. The authors describe, how their use of synthetic data sets can provide the matching tool the ability to perform well in a similar real senario. The tuning module executes, totally separate from the real tool.

Post-Match Strategies: There has also been some work in improving the already obtained results from a schema matching tool. OMEN [54] 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.[48] 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. Thus minimizing the human intervention in case there are more than one possible mappings for a source element to target schema.

One of the latest work for detecting the best match results, according to the user preferences, using fuzzy logic has been demonstrated in[32]. The work also enhances COMA++ results for deriving best semantic mappings. The research proposes to apply fuzzy sets theory on schemas in order to express user preferences.

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 the generic schema matching tools. Current available schema matching tools only support subject domain experts with good computer science background to utilize schema matching tools. And schema matching tools in large scale scenarios still lack the initiatives in interface development research. However with matching becoming need of today in the the ever expanding data integration domain, new user friendly interfaces are emerging for mapping environments.

These environments have augmented the match task with pre-match phase and post-match phase. As discussed in previous section, pre-match phase interfaces provide the facility to define a domain or application specific strategy, to align the different schema matching algorithms, include configuration of various parameters of the match algorithms [2], training of the machine learning-based matchers [44] and specification of auxiliary information (synonyms, abbreviations and other domain specific constraints) [2]. While the post-match phase

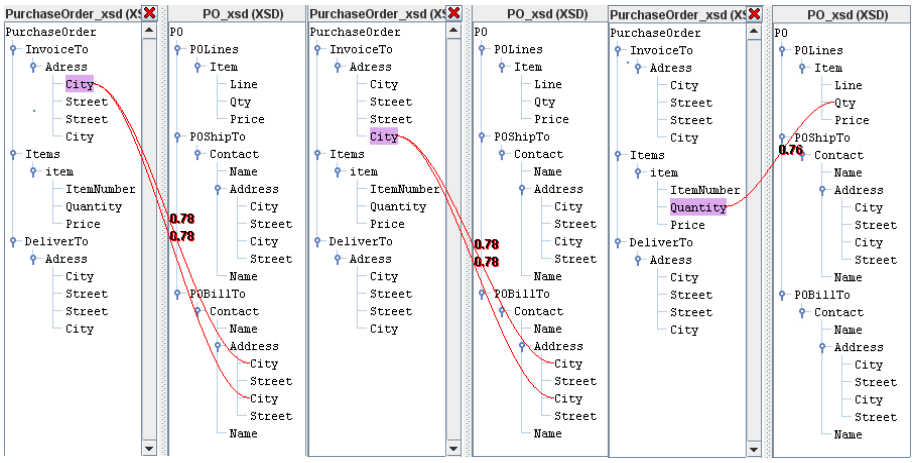


Fig. 4. Three cases for selecting the best match: COMA++

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 [2, 7, 38]. These techniques are termed as match quality measures in the literature [17]. The three tools CLIO [38], COMA++ [2] and Microsoft BizTalk Mapper [7] generate the possible mappings along with degree of match. And then graphically allow the user to select the mappings according to his/her expertise from its interpretations (Figure 4).

Discussion

In the preceding sections, we have discussed some recent techniques to counter the expanding context of schema matching. These techniques supplement the already existing schema matching and integration algorithms and highlight the fact that structural analysis of schemas for matching is becoming more and more important. 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. The definitions for schema matching and integration given in section 3, have been

designed to support this fact. The triplet in node semantics definition (Definition 2) providing us a way to extract semantics from the the structure.

Other then these, there are some issues which also need to be treated with some priority. These are related to the temporal changes to a schema and its effects on the existing mappings to or from to that schema. Since the web is an evolving entity, the *schema evolution* and related *mapping evolution* is here to stay. Methods like domain level CORPUS based schema matching [46] 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 [67], work benefiting from CLIO [38] which generates queries for the mappings, the authors show how the changes in a schema are used to rewrite the queries representing the mappings.

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* [58]. 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 *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 (the proportion of retrieved and relevant mappings to all the mappings retrieved) and the *recall* measure (the proportion of relevant mappings that are retrieved, out of all relevant mappings available) are most widely used to verify the quality [16]. 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 [25, 28].

6 State of the Art

The previous surveys [60, 63, 69] 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 matching found in first step. In this section we review some works which can be classified under the large scale schema matching research domain till to date, which adheres to the strategies discussed in section 5.2.

COMA++[2] 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 parent/child similarities.

Similarity of pairs of elements is calculated into a similarity matrix. It has a very comprehensive graphical user interface for matching, 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 using COMA++ described in [17], match large taxonomies by fragmenting them. The source and target schemas are broken down into fragments, subtrees of schema tree representations. 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[8] target is to provide a flexible and a customizable framework for combining different match algorithms. Currently *Cupid* [47] implementation and *Similarity Flooding* (SF) [50] algorithms are being used as the base matchers. A graphical interface for it has been proposed by the name of *BizTalk Mapper* [30] and an enhanced version demonstrated in [7]. The GUI is very heavily dependent on other microsoft technologies. Protoplasm supports various operators for computing, aggregating, and filtering similarity matrices. 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.

Cupid, first calculates linguistic similarity of pair of nodes using external oracles of synonyms and abbreviations. Then the structural matching is applied on the tree structures in post order manner. Supported by SF, this technique gives similarity possibilities for non-leaf nodes, depending upon the similarity of their leaves. For each source element, Cupid selects the target element with the highest aggregated weighted similarity exceeding a given threshold as the match candidate. In [55], the authors have utilized Protoplasm framework to perform matchings on very large medical taxonomies. To reduce match complexity, only the direct child and grandchildren structural similarities are considered.

CLIO [38] 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 transforma-

tion 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** [67], 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

MOMIS[5] 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. External dictionary WordNet is utilized to compute the synonym, hypernym or general relationship of 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 [37] 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 for which a match candidate is not found in the local interface, are added to it. Wise-Integrator employs several algorithms to compute attribute similarity; namely exact and approximate string matching, dictionary lookup for semantic name similarity, specific rules for compatibility of data types, of value scales/units, of value distribution (e.g., average), and of default values, all exploiting information from interface specifications. For each pair of attributes, 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 and one element is decided as the global attribute to be used in the integrated interface. Otherwise, the local attributes are considered distinct and added as new attributes to the integrated interface.

DCM framework (Dual Correlation Mining) [36] 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. Based on the assumptions; independence of attributes, non-overlapping semantics, uniqueness within an interface, and the same semantics for the same names, they propose a statistical approach, extracted from data mining domain. The algorithm identifies and clusters synonym attributes by analyzing the co-occurrence of attributes in different interfaces.

PSM (Parallel Schema Matching)[66], 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 attributes to identify synonyms may give high score for rare elements but PSM does not. And secondly the time complexity 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.

Next we summarize some of the implemented research works specific to ontology mapping and integration. Basic matching techniques remain the same like string based matching supported by structural matching using internal data structures based on graphs.

GLUE[19] is the extended version of LSD, which finds ontology/ taxonomy mapping using machine learning techniques. The system is input with set of 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* [18], 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* [46], 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* [15]. 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 matchings e.g., name is a concatenation of firstname and lastname. It also provides the information about the prediction criteria for a match or mismatch.

ONTOBUILDER [61] 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.

QOM [22] (Quick Ontology Matching) has been incorporated in complete tool FOAM [21] (Framework for Ontology Alignment and Mapping). It maps ontologies in RDF(S) format. It utilizes edit distance algorithm to compute the similarity between the names of two entities of same category as in *OLA* [24], e.g., properties are compared to properties. The correspondences found for different kinds of elements are aggregated using a weighted sum. To reduce the target search space in large ontologies, QOM looks for neighboring elements of the preceding matched entities.

PORSCHE (Performance Oriented Schema Matching) [62] 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 [4] 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 [64]. 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.

Future prospective of schema matching is in the large scale level, which is mainly related to schemas and ontologies in P2P networks. P2P Ontology Integration [9] 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.

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

Table 4. Schema Matching Tools and Prototypes Comparison - General

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

give a general outlook of tools. Whereas table 5 gives much deeper insight into the algorithms used by the tools.

7 Conclusion and Perspective

In this paper we have given 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 tried to provide an insight on current emergent technologies driving the match research like data mashups and P2P database networks.

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 Cupid is hybrid in nature whereas COMA is composite thus providing much flexibility. The tools normally adopt hybrid approach for better and automatic approach. Structure level matching has been adopted by all except for web search interface schema integrators. For semantic comparison, external oracle like 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. 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 character, demonstrated in [43, 62].

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

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

It appears that the most prototypes aimed to provide a good quality of matching. While today the current application domain like the genomic or e-business deal with large schema therefore the matching tool should provide good performance and if possible automatic mapping. In the future, matching system should try to find a trade off between quality and performance.

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 applications domain. We conclude our discussion by enumerating some explicit future research concerns in the field of schema matching and integration.

- Transformation in schema mappings with schema evolution
- Visualization of mappings in multi-schema (more than 2) integration
- Development of correctness/completeness metrices and benchmark tools for evaluating schema match systems

References

1. Y. An, A. Borgida, R. J. Miller, and J. Mylopoulos. A semantic approach to discovering schema mapping expressions. In *ICDE*, 2007.
2. D. Aumueller, H. H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with coma++. In *ACM SIGMOD*, pages 906–908, 2005.

3. D. Bachlechner, K. Siorpaes, D. Fensel, and I. Toma. Web service discovery - a reality check. Technical report, Digital Enterprise Research Institute (DERI), 2006.
4. C. Batini, M. Lenzerini, and S. B. Navathe. A comparitive analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, 1986.
5. D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. The momis approach to information integration. In *ICEIS*, pages 194–198, 2001.
6. S. Benkley, J. Fandozzi, E. Housman, and G. Woodhouse. Data element tool-based analysis (delta). In *MTR*, 1995.
7. P. A. Bernstein, S. Melnik, and J. E. Churchill. Incremental schema matching. In *VLDB*, 2006.
8. P. A. Bernstein, S. Melnik, M. Petropoulos, and C. Quix. Industrial-strength schema matching. *ACM SIGMOD Record*, 33(4):38–43, 2004.
9. P. Besana, D. Robertson, and M. Rovatsos. Exploiting interaction contexts in p2p ontology mapping. In *P2PKM*, 2005.
10. G. J. Bex, F. Neven, and S. Vansummeren. Inferring xml schema definitions from xml data. In *VLDB*, 2007.
11. A. Bilke and F. Naumann. Schema matching using duplicates. In *ICDE*, 2005.
12. P. Bohannon, E. Elnahrawy, W. Fan, and M. Flaster. Putting context into schema matching. In *VLDB*, 2006.
13. T. Dalamagasa, T. Chengb, K.-J. Winkelc, and T. Sellisa. A methodology for clustering xml documents by structure. *Information Systems*, 31:187228, 2006.
14. M. Davis. Semantic wave 2006 - a guide to billion dollar markets - keynote address. In *STC*, 2006.
15. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. imap: Discovering complex semantic matches between database schemas. In *ACM SIGMOD*, 2004.
16. H. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *IWWD*, 2003.
17. H.-H. Do and E. Rahm. Matching large schemas: Approaches and evaluation. *Information Systems*, 32(6):857–885, 2007.
18. A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources - a machine learning approach. In *ACM SIGMOD*, 2001.
19. A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Y. Halevy. Learning to match ontologies on the semantic web. *VLDB J.*, 12(4):303–319, 2003.
20. F. Duchateau, Z. Bellahsene, and M. Roche. A context-based measure for discovering approximate semantic matching between schema elements. In *IEEE RCIS*, 2007.
21. M. Ehrig, J. Euzenat, and H. Stuckenschmidt. Framework for ontology alignment and mapping - results of the ontology alignment evaluation initiative. In *Integrating Ontologies*, 2005.
22. M. Ehrig and S. Staab. Qom - quick ontology mapping . In *ISWC*, 2004.
23. D. W. Embley, L. Xu, and Y. Ding. Automatic direct and indirect schema mapping: Experiences and lessons learned. *ACM SIGMOD Record*, 33(4):14–19, 2004.
24. J. Euzenat. An api for ontology alignment. In *ISWC*, pages 698–712, 2004.
25. J. Euzenat. Semantic precision and recall for ontology alignment evaluation. In *IJCAI*, 2007.
26. J. Euzenat et al. State of the art on ontology matching. Technical Report KWEB/2004/D2.2.3/v1.2, Knowledge Web, 2004.
27. A. Gal. Why is schema matching tough and what can we do about it. *SIGMOD Record*, 35(4), 2006.

28. A. Gal, A. Anaby-Tevor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *VLDB J*, 14(1):50–67, 2005.
29. A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. Sweetening wordnet with dolce. *AI Magzine*, 24(3):13–24, 2003.
30. J. E. C. George G. Robertson, Mary P. Czerwinski. Visualization of mappings between schemas. In *CHI*, 2005.
31. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *ESWS*, 2004.
32. W. Guedria, Z. Bellahsene, and M. Roche. A flexible approach based on the user preferences for schema matching. In *IEEE RCIS*, 2007.
33. L. M. Haas. Beauty and the beast: The theory and practice of information integration. In *ICDT*, 2007.
34. A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *ICDE*, 2003.
35. A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *VLDB*, 2006.
36. B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across web query interfaces: a correlation mining approach. In *KDD*, pages 148–157, 2004.
37. H. He, W. Meng, C. T. Yu, and Z. Wu. Automatic integration of web search interfaces with wise-integrator. *VLDB J.*, 13(3):256–273, 2004.
38. M. A. Hernedez, R. J. Miller, and L. M. Haas. Clio: A semi-automatic tool for schema mapping. In *ACM SIGMOD*, 2002.
39. W. Hu, Y. Zhao, and Y. Qu. Partition-based block matching of large class hierarchies. In *ASWC*, 2006.
40. M. N. Huhns and M. P. Singh. Service-oriented computing: Key concepts and principals. *IEEE Internet Computing*, 2005.
41. A. Jhingran. Enterprise information mashups: Integrating information, simply - keynote address. In *VLDB*, 2006.
42. M.-L. Lee, T. W. Ling, H. Lu, and Y. T. Ko. Cleansing data for mining and warehousing. In *DEXA*, 1999.
43. M.-L. Lee, L. H. Yang, W. Hsu, and X. Yang. Xclust: clustering xml schemas for effective integration. In *CIKM*, pages 292–299, 2002.
44. Y. Lee, M. Sayyadain, A. Doan, and A. S. Rosenthal. etuner: tuning schema matching software using synthetic scenarios. *VLDB J.*, 16:97–122, 2007.
45. A. Loser, W. Siberski, M. Sintek, and W. Nejdl. Information integration in schema-based peer-to-peer networks. In *Caise*, 2003.
46. J. Madhavan, P. A. Bernstein, A. Doan, and A. Y. Halevy. Corpus-based schema matching. In *ICDE*, pages 57–68, 2005.
47. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, 2001.
48. D. Manakanatas and D. Plexousakis. A tool for semi-automated semantic schema mapping: Design and implementation. In *DisWEB Workshop CaiSE*, 2006.
49. D.-E. Meddour, M. Mushtaq, and T. Ahmed. Open issues in p2p multimedia streaming. In *MultiComm*, 2006.
50. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, pages 117–128, 2002.
51. S. Melnik, E. Rahm, and P. A. Bernstein. Developing metadata-intensive applications with rondo. *J. of Web Semantics*, I:47–74, 2003.
52. P. D. Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of xml schemas at various "'severity"' levels. *Information Systems J.*, pages 397–434, 2006.

53. T. Milo and S. Zohar. Using schema matching to simplify heterogeneous data translation. In *VLDB*, pages 122–133, 1998.
54. P. Mitra, N. F. Noy, and A. R. Jaiswal. Omen: A probabilistic ontology mapping tool. In *ISWC*, pages 537–547, 2005.
55. P. Mork and P. A. Bernstein. Adapting a generic match algorithm to align ontologies of human anatomy. In *ICDE*, 2004.
56. I. Niles and A. Pease. Towards a standard upper ontology. In *FOIS*, 2003.
57. N. F. Noy, A. Doan, and A. Y. Halevy. Semantic integration. *AI Magazine*, 26(1):7–10, 2005.
58. N. F. Noy, S. Kunnatur, M. Klein, and M. A. Musen. Tracking changes during ontology evolution. In *ISWC*, 2004.
59. C. Parent and S. Spaccapietra. Database integration: The key to data interoperability. In M. P. Papazoglou, S. Spaccapietra, and Z. Tari, editors, *Advances in Object Oriented Modeling*. The MIT Press, 2000.
60. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.
61. H. Roitman and A. Gal. Ontobuilder: Fully automatic extraction and consolidation of ontologies from web sources using sequence semantics. In *EDBT Workshops*, 2006.
62. K. Saleem, Z. Bellahsene, and E. Hunt. Performance oriented schema matching. In *DEXA*, 2007.
63. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *J. Data Semantics IV*, pages 146–171, 2005.
64. M. Smiljanic, M. van Keulen, and W. Jonker. Using element clustering to increase the efficiency of xml schema matching. In *Workshop ICDE*, 2006.
65. S. Spaccapietra, C. Parent, and Y. Dupont. Model independent assertions for integration of hetrogeneous schemas. *VLDB J.*, pages 81–126, 1992.
66. W. Su, J. Wang, and F. Lochovsky. Holistic query interface matching using parallel schema matching. In *ICDE*, 2006.
67. Y. Velegrakis, R. Miller, and L. Popa. On preserving mapping consistency under schema changes. *VLDB J*, 13(3):274–293, 2004.
68. W. Wu, A. Doan, and C. Yu. Merging interface schemas on the deep web via clustering aggregation. In *ICDM*, 2005.
69. M. Yatskevich. Preliminary evaluation of schema matching systems. Technical Report DIT-03-028, Informatica e Telecomunicazioni, University of Trento, 2003.