Information Integration Across Heterogeneous Sources: Where Do We Stand and How to Proceed? *

Aditya Telang and Sharma Chakravarthy and Yan Huang†

IT Laboratory & Department of Computer Science & Engineering
The University of Texas at Arlington, Arlington, TX 76019.
{aditya.telang, sharmac}@uta.edu, huangyan@unt.edu

Abstract

Today, information integration has assumed a completely different, complex connotation than what it used to be. The advent of the Internet, the proliferation of information sources on the surface Web as well as the deep Web, the presence of structured, semi-structured, and unstructured data - all have added new dimensions to the problem of information integration as known earlier. From the time of distributed databases leading to heterogeneous, federated, and multi-databases, retrieval and integration of information from heterogeneous sources has been an important and complex problem. Currently, the problem is even more complicated as repositories exist in various formats (HTML, XML, spatial data sources to name a few) and schemas, and both the content and the structure of the data within them are changing autonomously. As the number of repositories/sources will continue to increase in an uncontrolled manner, there is no other option but to find extensible techniques for answering a complex search/query whose (partial) answers have to be retrieved and integrated from multiple sources. In this ***survey*** paper, we identify the set of challenges that need to be addressed for this form of heterogeneous information integration, and compare the current state-of-the-art as to how they fare. We then propose a framework with functional components – termed *InfoMosaic*, that aims to address some of these important challenges, and briefly elaborate on the data and control flow involved in answering a complex query/search.

*The work is supported, in part, by NSF grants IIS - 0534611, IIS - 0326505, and EIA - 0216500.

†This author is from University of North Texas, Denton, TX - 76203, USA

International Conference on Management of Data
COMAD 2008, Mumbai, India, December 17–19, 2008
©*Computer Society of India, 2008*

1 Introduction

The immense scale and elaborate spread of the Web has rendered it as an ultimate information repository of data-rich pages on the *surface Web* of static URLs and the *deep Web* of database-backed contents. Hence, over the last decade or so, a multitude of retrieval techniques such as *search engines* (e.g., Google [5]), *meta-search engines* (e.g., Vivisimo [54]), *faceted search engines* (e.g., Faceted DBLP [35]), and *question-answering* frameworks (e.g., START [29]) have been developed to facilitate inexperienced users to quickly and effortlessly retrieve information from the *surface Web*. Additionally, several *domain-specific* retrieval systems[1] (e.g., commercial portals such as Amazon, Yahoo Autos, etc.) have aided users in formulating queries across specific domains on the *deep Web*.

However, the simplicity associated with the usage of these mechanisms makes it difficult to specify queries that require data to be extracted from multiple repositories across diverse domains, followed by its meaningful integration to produce the desired results. For instance, consider some sample query intents:

Query 1: *Retrieve castles near London that are reachable by train in less than 2 hours*

Query 2: *Obtain a list of 3-bedroom houses in Houston within 2 miles of a school and within 5 miles of a highway and priced under 250,000$*

Query 3: *Retrieve French restaurants within 1 mile of IMAX Theater in Dallas, Texas*

Query 4: *Find a place to buy kitchen furniture within walking distance of a metro stop in the Washington DC area [33]*

Although all the information for answering different parts of each of the above queries (e.g., "castles

[1]We realize that the notion of a *domain* is subjective. In the context of this paper, a *domain* indicates a collection of sources providing information of similar interest such as travel, books, literature, shopping etc.

near London", "train schedules in London", "French restaurants in Dallas", etc.) is available on the Web, it is currently **not possible** to frame it as a single query and get a comprehensive set of relevant answers. The above example underlines the "*Tower of Babel*" problem for integrating information to answer a query that requires data from multiple independent sources to be combined intelligently. The islands of information that we are experiencing now is not very different from the islands of automation seen earlier. This gap needs to be bridged to move from current search and meta-search approaches to true *information integration*.

In this paper, we address the problem of information integration as it pertains to extracting and combining data from heterogeneous autonomous Web sources in response to user queries spanning across several domains. In Section 2, we survey the existing state-of-the-art in terms of the challenges addressed, the frameworks designed and the approaches adopted for addressing the problem. In Section 3, we identify and elaborate on the salient challenges encountered in heterogeneous information integration. We analyze each challenge with respect to the current work towards addressing it, and provide our viewpoint for the same. Section 4 elucidates our approach in the context of *InfoMosaic*, a framework proposed for web-based multi-domain information integration. Section 5 concludes the paper.

2 Current Frameworks and Approaches

It is important to understand that information integration is not a new problem. It existed in the form of querying distributed, heterogeneous, multiple and federated databases. What has really changed in the last decade is the complexity of the problem (types/models of data sources, number of data sources, infeasibility of schema integration) and the kind of solution that is being sought.

2.1 Integration Frameworks

Currently, there exist a number of frameworks that address several challenges encountered in heterogeneous data integration. For instance, *Havasu*, a multi-objective query processing framework comprising of multiple functional modules, addresses the challenges of *imprecise-query specification* [41], *query optimization* [42], and *source-statistics collection* [43] for single-domain Web integration. The *MetaQuerier* framework [6] addresses the challenges in exploration and integration of deep-web sources using two functional modules – i) *MetaExplorer* [6], which is responsible for *dynamic source discovery* [27] and *on-the-fly integration* [23] for the discovery, modeling, and structuring of web databases to build a search-able source repository, and ii) *MetaIntegrator* [21], that focuses on the issues of on-line source integration such as *source selection*, *query mediation*, and *schema integration*. The *Ariadne* framework (an extension of the SIMS mediator architecture [3]) facilitates integration of data across *semi-structured* and *unstructured* data sources (e.g., web data) in addition to structured databases by using specially designed *wrappers* [40]. It also constructs an independent *domain model* [51] (using the Loom representation system [37]) for each application that integrates the information from the sources and provides a single terminology for querying.

TSIMMIS [8] uses a schema-less approach for retrieving information from dynamic sources. Frameworks such as *InfoMaster* [12], *Information Manifold* [34], *Whirl* [9] and others focus on integrating structured and semi-structured data extracted from multiple pre-defined databases by adopting a taxonomy for mapping domain concepts to database attributes. Additionally, *spatial* and *temporal* data integration has been addressed in mediated systems (e.g., *Hermes* [49], *TerraWorld* [38], etc.) as well as ontology-driven geographic information systems (e.g., *eMerges* system [50]).

2.2 Approaches for Data Integration

In the pursuit of achieving an ideal information integration system, a number of approaches have been proposed over the past two decades. Some of the notable ones include:

Mediator: One of the most prominent approaches adopted by many integration frameworks (Ariadne, TSIMMIS, Havasu, ...), it proposes the use of a *mediator*, a system responsible for reformulating user queries formed on a single schema into queries on the local schema of the underlying data sources. The sources contain the actual data, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources. The mapping can be achieved by adopting – i) *Global-as-view* (GAV) approach (that requires the global schema to be represented in terms of the underlying data sources), or ii) *Local-as-view* (LAV) approach (that requires the global schema to be defined independently from the sources, and the relationships between them are established by defining every source as a view over the global schema). The former is preferred when the sources being integrated is known and stable, whereas the latter is considered suitable for large-scale ad-hoc integration.

Ontology-based: In the last decade, *semantics* (that are an important component for data integration) gained popularity leading to the inception of the *ontology-based* approach. The *Semantic Web* research community [44], [11] has focused exten-

sively on the problem of semantic integration and the use of *ontologies* for blending heterogeneous schema across multiple concepts. Their pioneering efforts have provided a new dimension for researchers to investigate the challenges in information integration. However, the literature associated with these systems (elaborated in [25]) makes it clear that there is an obvious lack of *real* methodology for ontology-based development.

Navigation-based: Also known as *link-based* approach [17], it is based on the fact that an increasing number of sources on the web require users to manually browse through several webpages in order to obtain the desired information. Pure navigational integration eliminates relational modeling of the data, and applies a model where sources are defined as sets of pages with their interconnections and specific entry-points, as well as additional information such as content, path constraints, and mandatory input parameters. This approach is considered to be vital in *deep web*-based information integration [22] that requires extracting data hidden behind web query-interfaces. However, maintaining the relationship between web sources that change at a rapid rate is a difficult task that renders this approach infeasible in the current context of the Web.

Federated: It [47] is developed on the premise that, information needed to answer a query is gathered directly from the data sources. Hence, the results are up-to-date with respect to the contents of the data sources at the time the query is posted. More importantly, the database federation approach lends itself to be more readily adapted to applications that require users to be able to impose their own ontologies on data from distributed autonomous information sources. The federated approach is preferred in scenarios when the data sources are autonomous, and support for multiple ontologies is needed. However, this approach fails in situations where the querying frequency is much higher than the frequency of changes to the underlying sources.

Warehouse-based: This approach [16] derives its basis from traditional data warehousing techniques. Data from heterogeneous distributed information sources is gathered, mapped to a common structure and stored in a centralized location. In order to ensure that the information in the warehouse reflects the current contents of the individual sources, it is necessary to periodically update the warehouse. In the case of large information repositories, this is not feasible unless the individual information sources support mechanisms for detecting and retrieving changes in their contents. This is an inordinate expectation in the case of autonomous information sources, spread across a number of heterogeneous domains, whose internal data changes at a frequent and aperiodic rate.

3 Integration Challenges and The InfoMosaic Approach

Although the gist of information integration has not changed and this topic has been investigated over two decades, the problem at hand is quite different and far more complex than the one attempted earlier. Although techniques developed earlier – global schemas, schema integration, dealing with multiple schemas, domain specific wrappers, and global transactions – have produced significant steps, they never reached the stage of maturity for deployment and usage. The multi-domain problem is more complicated as repositories exist in various formats (HTML, XML, Web Databases with query-interfaces, etc.), with and without schemas, and both the content and the structure are changing autonomously. As the enumeration of challenges below indicate, existing techniques from multiple areas need to be eclectically combined as well as new solutions developed in order to address this problem. We describe the InfoMosaic approach to address these challenges.

3.1 Capturing User Intent

One of the primary challenges is to provide a mechanism for the user to express his/her intent in an intuitive, easy-to-describe form. As elaborated by *Query-1*, user queries are complex, and are difficult to express using existing query specification formats. In an ideal scenario, the user should be able to express the query in a natural language. This is one of the primary reasons for the popularity of search engines since, there is no query language to learn. However, unlike a search engine operation (that involves a simple lookup of a word, phrase or expression in existing document repositories), information integration is a complex process of retrieving and combining data from different sources for different sub-queries embedded in the given user query. An alternate option is the use of DBMS-style query languages (e.g., SQL) that allow users to specify a query in a pre-defined format. However, in sharp contrast to Database models (that assume the user knows what to access and from where to access), the anonymity of the sources and the complexity of the query involved in a data integration scenario makes it difficult to express the intent using the hard semantics of these data models.

Existing frameworks (e.g., Ariadne [30], TSIMMIS [8], and Whirl [9]) extend the database querying models using combinations of *templates* or *menu-based*

forms to incorporate queries that are restricted to a single domain (or a set of domains). Other frameworks (such as Havasu [28]) employ an interface similar to search engines, that take relevant keywords (associated with a concept) from the user and retrieve information for this particular concept from a range of sources. However, as the domains for querying established by these systems are fixed (although the sources within the domain might change), the problem of designing a querying mechanism is simplified to a great extent. When a more involved query needs to be posed, users may not know how to unambiguously express their needs and may formulate queries that lead to unsatisfactory results. Moreover, providing a rigid specification format may restrict the user from providing complete information about his/her intent.

Additionally, most of these frameworks fail to capture queries that involve a combination of *spatial*, *temporal*, and *spatio-temporal* conditions. A few systems (e.g., Hermes [49], TerraWorld [38], etc.) allow a limited set of spatial operations (such as *close to*, *travel time*) through its *push-button listing-based* interface or a *form-based* interface. Currently, centralized *web-based mapping* interfaces (e.g. Google Maps and Virtual Earth) allow searching and overlaying spatial layers (e.g., all hotels and metro stations in current window or a given geo-region) to examine the relationships among them visually. However, these user interfaces are not expressive enough and restrict users from specifying their intent in a flexible manner.

In InfoMosaic, we address the *query-specification* challenge in a multi-domain environment by combining and enhancing techniques from *natural language processing*, *database query specification* and *information retrieval* to incorporate the following characteristics: i) specification of *soft* semantics instead of *hard* queries, ii) ability to accept minimal specification and refine it to meet user intent and in the process collect feedback for future usage, iii) support queries that include spatial, temporal, spatio-temporal, and cost-based conditions in addition to regular query conditions, iv) accepting optional *ranking* metrics based on user-specified criteria, and v) support *query approximation* and *query relaxation* for retrieving approximate answers instead of exact answers.

3.2 Mapping Intent into Queries

The next challenge is to transform the user intent into an appropriate query format that can be represented using a variant of relational algebra (or similar established mechanisms). Since the queries in the context of information integration are complex and involve a myriad set of conditions, it is obvious that applying the existing formalisms of relational algebra may not be sufficient.

Over the past decade, several querying languages that extend the basics of relational algebra and allow access to *structured data* (SQL, OOQL [36], Whirl [9], etc.), *semi-structured data* (SemQL [32], CARIN [34], StruQL [16], etc.) and *vague (or unstructured) data* (VAGUE [39]) have been designed. These languages have, *with limited success*, incorporated imprecise user queries posed on a single-domain (or fixed set of multiple domains). Additionally, several frameworks have deployed customized models that translate the user query to a query format supported by the internal global schema (that provides an interface to the underlying sources). Briefly, Havasu's *QPIAD* [15] maps *imprecise* user queries to a more generic query using a combination of data-mining techniques. Similarly, Ariadne [30] interprets the user-specified conditions as a sequence of *LOOM statements* that are combined to generate a single query. MetaQuerier's *form assistant* [53] consists of built-in type handlers that aids the query translation process with moderate human efforts.

However, existing mechanisms will prove to be insufficient to represent complex intent spanning several domains. Hence, it becomes necessary to use domain-related taxonomies/ontologies and source-related semantics to disambiguate as well as generate multiple potential queries from the user intent. A feedback and learning mechanism may be appropriate to learn user intent from the combinations of concepts provided based on user feedback. If multiple queries are generated (which is very much possible on account of the ambiguity of natural language and the volume of concepts involved in the domains of integration), an ordering mechanism may be useful to obtain valuable feedback from the user. Once the query is finalized, a canonical representation can be used to further transform the query into its components and elaboration.

In the InfoMosaic framework, we elaborate and refine the user input by using a*knowledge-base* couple with interactive. Moreover, instead of designing a new language that supports all the query conditions, we are currently extending the capabilities of SQL to incorporate soft-semantics and conditions based on domains rather than sources. In particular, we are trying to enhance the semantics of SQL-based spatial query languages [14] for easy specification of spatial relations including metric, topological, and directional relationships pertaining to heterogeneous datasets from the web.

3.3 Domain Discovery and Source Identification

As elucidated by *Query-1*, user queries inherently consist of multiple sub-queries posed on distinct domains (or concepts). Gathering appropriate knowledge about the domains and the corresponding sources within these domains is vital to the success of heterogeneous integration of information. In order to relate various parts of a user query to appropriate domains (or

concepts), the meaning of *information* that is interchanged across the system has to be understood.

Over the past decade, several customized techniques have been adapted by different frameworks that focus on capturing such meta-data about concepts and sources that facilitate easy mapping of queries over the global schema and/or the underlying sources. Havasu's *attribute value hierarchies* [28], InfoMaster's *knowledge-base* [12], Information Manifold's *CARIN* [34], TSIMMIS's *OLE model* [8], Ariadne's *LIM* [30], and Tukwila's *data-source catalog* [26] are some of the important advances in formulation of a comprehensive source repository replete with adequate domain knowledge. Additionally, the use of *ontologies* for modeling implicit and hidden knowledge has been considered as a possible technique to overcome the problem of semantic heterogeneity by a number of frameworks such as SIMS [24], OntoBroker [10], etc..

Havasu's *attribute-valued hierarchies* [28] maintain a classification of the attributes of the data sources over which the user queries are formed. Ariadne uses an independent *domain model* [30] for each application, that integrates the information from the underlying sources and provides a single terminology for querying. This model is represented using the LOOM knowledge representation system [37]. TSIMMIS adopts an *Object Exchange Model* (OEM) [8], a self-describing (tagged) object model, in which objects are identified by labels, types, values, and an optional identifier. Information Manifold's CARIN [34] proposes a method for representing local-source completeness and an algorithm for exploiting source information in query processing. This is an important feature for integration systems, since, in most scenarios, data sources may be incomplete for the domain they are covering. Furthermore, it suggests the use of *probabilistic reasoning* for the ordering of data sources that appear relevant to answer a given query. InfoMaster's *knowledge base* [12] is responsible for the storage of all the rules and constraints required to describe heterogeneous data sources and their relationships with each other. In Tukwila, the metadata obtained from several sources is stored in a single *data source catalog* [26], and holds different type of information about the data sources such as – *semantic description* of the contents of the data sources, *overlap information* about pairs of data sources, and *key statistics* about the data, such as the cost of accessing each source, the sizes of the relations in the sources, and selectivity information. Additionally, the use of *ontologies* for modeling implicit and hidden knowledge has been considered as a possible technique to overcome the problem of semantic heterogeneity by a number of frameworks such as KRAFT [19], SIMS [24], OntoBroker [10], etc..

The proliferation of data on the Internet has ensured that within each domain, there exist vast number of sources providing adequate yet similar information. For instance, portals such as *Expedia*, *Travelocity*, *Orbitz*, etc. provide information for the domain of *air-travel*. Similarly, sources such as *Google Scholar*, *DBLP*, *CiteSeer*, etc. generate adequate and similar results for the domain of *publications and literature*. Thus, the next logical challenge is to automate the current manual process of identifying appropriate sources associated with individual domains. Semantic discovery of sources, that involves a combination of - web crawling, interface extraction, source clustering, semantic matching and source classification, has been extensively researched by the *Semantic Web* community [48]. Currently, a significant and increasing amount of information obtained from the web is hidden behind the query interfaces of searchable databases. The potential of integrating data from such *hidden* data sources [22] is enormous. The MetaQuerier project [6] addresses the challenges for integrating these deep-web sources such as – discovering and integrating sources automatically, finding an appropriate mechanism for mapping independent user-queries to source-specific sub-queries, and developing mass collaboration techniques for the management, description and rating of such sources.

An ideal archetype would be to design a *global taxonomy* (that models all the heterogeneous domains across which user queries might be posed), and a *domain taxonomy* (that models all the sources belonging to the domain and orders them based on distinct criteria specified by the integration system). The construction of such a multi-level ontology requires extensive efforts in the areas of – *domain knowledge aggregation*, *deep-web exploration*, and *statistics collection*. However, the earlier work on databases (use of equivalences and statistics in centralized databases, use of source schemas for obtaining a global schema) and recent work on information integration (as elaborated earlier) provide adequate reasons to believe that this can be extended to multi-domain queries and computations that include spatial and temporal constraints, which is being addressed in our InfoMosaic framework.

3.4 Plan Generation and Optimization

Plan generation and optimization in an information integration environment differs from traditional database query processing in several aspects – i) volume of sources to be integrated is much larger than in a normal database environment, ii) heterogeneity between the data (legacy database systems, web-sites, web-services, hidden web-data, etc.) makes it difficult to maintain the same processing capability as found in a typical database system (e.g., the ability to perform joins), iii) the query planner and optimizer in information integration has little information about the data since it resides in remote autonomous sources, and iv) unlike relational databases, there can be several restrictions on how an autonomous source can be

accessed.

Current frameworks have devised several novel approaches for generating effective plans in the context of data integration. Havasu's *StatMiner* (in association with the *Multi-R Optimizer*) [28] provides a guarantee on the *cost* and *coverage* of the results generated on a query by approximating appropriate source statistics. Ariadne's *Theseus* [30] pre-compiles part of the integration model and uses a local search method for generating query plans across a large number of sources. Information Manifold's *query-answering approach* [34] translates user queries, posed on the mediated schema of data sources, into a format that maps to the actual relations within the data sources. This approach differs from the one adopted by Ariadne, and ensures that only the relevant set of data sources are accessed when answering a particular user query. In Tukwila, if the *query planner* concludes that it does not have enough meta-data with which to reliably compare candidate query execution plans, it chooses to send only a partial plan to the execution engine, and takes further action only after the partial plan has been completed.

However, since for these frameworks, the domains involved in the user query are pre-determined, generalizing and applying these techniques to autonomous heterogeneous sources is not possible. This is particularly true for techniques that generate their plans based on the type of modeling applied for the underlying data sources. Furthermore, current optimization strategies [28] focus on a restricted set of metrics (such as cost, coverage and overlap of sources) for optimization. Additional metrics such as – volume of data retrieved from each source, number of calls made to and amount of data sent to each source, quantity of data processed, and the number of integration queries executed – are currently not considered. It is important to understand that in this problem space, exact values of some of these measures may not be available and the information available about the ability of the sources and their characteristics may determine how these measures can be used. Thus, effective plan generation and evaluation is significantly more complex than a traditional system and requires to be investigated thoroughly.

In InfoMosaic, we view the plan generation and optimization challenge as an intelligent query optimization problem involving two stages: *logical* and *physical*. In the logical phase, we identify the *individual* domain sub-queries how they come together as a larger query by using appropriate domain knowledge. In the physical phase, various source semantics and characteristics are used to generate effective plans for each individual sub-query. In addition, we are also investigating query optimization techniques for handling spatial, temporal and spatio-temporal conditions.

3.5 Data Extraction

Typically, in schema-based systems (e.g., RDBMS), the description of data (or meta-data) is available, query-language syntax is known, and the type and format of results are well-defined, and hence they can be retrieved programmatically (e.g., ODBC/JDBC connection to a database). However, in the case of web repositories, although a page can be retrieved based on a URL (or filling forms in the case of *hidden web*), or through a standard or non-standard web-service, the output structure of data is neither pre-determined nor remains the same over extended periods of time. The extracted information needs to be parsed as HTML or XML data types (using the meta-data of the page) and interpreted.

Currently, *wrappers* [30] are typically employed by most frameworks for the extraction of heterogeneous data. However, as the number of data sources on the web and the diversity in their representation format continues to grow at a rapid rate, manual construction of wrappers proves to be an expensive task. There is a rapid need for developing automation tools that can design, develop and maintain wrappers effectively. Even though a number of integration systems have focussed on automated wrapper generation (Ariadne's Stalker [40], MetaQuerier [6], TSIMMIS [20], InfoMaster [13], and Tukwila [26]), since the domains (and the corresponding sources) embedded within these systems are known and predefined, the task of generating automated wrappers using mining and learning techniques is simplified by a large extent. There also exist several independent tools based on solid formal foundations that focus on low-level data extraction from autonomous sources such as *Lixto* [4], *Stalker* [40], etc.. In the case of spatial data integration, (e.g., eMerges system [50]), ontologies and semantic web-services are defined for integrating spatial objects, in addition to wrappers and mediators. *Heracles* [2] (part of TerraWorld and derived from the concepts in Ariadne) combines online and geo-spatial data in a single integrated framework for assisting travel arrangement and integrating world events in a common interface. A Storage Resource Broker was proposed in the LTER spatial data workbench [45] to organize data and services for handling distributed datasets.

Information Manifold [34] claimed that the problem of wrapping semi-structured sources would be irrelevant as XML will eliminate the need for wrapper construction tools. This is an optimistic yet unrealistic assumption since there are some problems in querying semi-structured data that will not disappear, for several reasons: 1) some data applications may not want to actively share their data with anyone who can access their web-page, 2) legacy web applications will continue to exist for many years to come, and 3) within individual domains, XML will greatly simplify the access to sources; however, across diverse domains, it is

highly unlikely that an agreement on the granularity for modeling the information will be established.

For the challenge of executing sub-queries and extracting relevant data from the sources in InfoMosaic, we plan to use *Lixto* [4], a powerful data extraction engine for programmatically extracting portions of a HTML page (based on the need) and converting the result into a specific format. It is based on monadic query languages over trees (based on monadic second order logic), and automatically generates Elog [4] (a variant of Datalog) programs for data extraction. For handling extraction of spatial data (that is larger in size and hence difficult to extract in a short time), we are planning to use a combination of – i) building a local spatial data repository by dynamically downloading related spatial files (using data clearing houses such as Map Bureau [18], etc.) of data that is relatively static , and ii) querying spatial web-services for fetching data that tends to change on a more frequent basis.

3.6 Data Integration

The most important challenge in the entire integration process involves fusion of the data extracted from multiple repositories. Since most of the existing frameworks are designed for a single domain or a set of predetermined domains, the integration task is generalized such that the data generated by different sources only needs to be "*appended*" and represented in a homogeneous format. Frameworks, such as Havasu, support the "*one-query on multiple-sources in single-domain*" format in which, the data fetched from multiple sources is checked for overlap, appended, and displayed in a homogeneous format to the user. Others, such as Ariadne, support the "*multiple sub-queries on multiple-sources in separate-domains*" format which is an extension to the above format, such that the task of checking data overlap is done at the sub-query level. The non-overlapping results from each sub-query are then appended and displayed.

However, the problem of integration becomes more acute when the sub-queries, although belonging to distinct domains, are dependent on each other for generating a final result-set. For instance, in *Query-1*, although it is possible to extract data independently for "*castles near London*", and "*train-schedules to destinations within 2 hours from London*", the final result-set that requires generating "*castles that are near London and yet reachable in 2 hours by train*" cannot be obtained by simply appending the results of the two sub-queries. For this (and similar complex) query, it becomes necessary to perform additional processing on the extracted data based on the sub-query dependencies, before it can be integrated and displayed.

In InfoMosaic, we extract spatial and non-spatial data into Post-GIS [46] and XML repositories respectively. Next, we generate and execute queries (XQuery for XML and spatial queries for Post-GIS whose results are converted to GML for further processing) and then integrate this extracted and processed data. The generation of these queries is based on the DTD (generated from the logical query plan) of the stored sub-query results and the attributes that need to be joined/combined from different sources. The join can be an arbitrary join (not necessarily equality) on multiple attributes. Our approach involves generating XQueries for each sub-query and combine them into a larger query using *FLOWR* expressions. It might be possible that the results of some sub-queries are already integrated during the execution and extraction phase. This information, based on the physical query plan, is taken into consideration for generating the required XQuery.

3.7 Result Ranking and Representation

Users should be able to access available information; however, this information should be presented in a structured and easy-to-digest format. Returning hundreds and thousands of information snippets will not help the user to make sense of the information. An interesting option would be to apply a rank on the final integrated results and provide only a percentage (top-k) of the total answers generated. For spatial queries, the result may be presented in a visual format.

However, unlike the domains of information retrieval [31] or even databases [7], the computation of ranking in information integration is more complex due to – autonomous nature of sources, lack of information about the quality of information from a source, lack of information about the amount of information (equivalent of cardinality) for a query on the source, and lack of support for retrieving results in some order or based on some metrics. To the best of our knowledge, *ranking* has not been addressed explicitly in any of the major projects on information integration. In InfoMosaic, we are currently addressing this challenge by investigating the application of ranking at different stages in the integration process (i.e., at sub-query execution phase, before the integration phase, after the integration phase, etc.) [52].

3.8 Other Challenges

In addition to the above challenges, there exist a number of issues that will prove to be significant as integration frameworks move from prototype designs to large-scale commercial systems.

3.8.1 On-the-fly Data Integration

It refers to scenarios where data needs to be integrated from a source immediately after discovering it. This is needed for supporting the most general form of information integration where any query on any domain is allowed. The problem is compounded in situations

where data sources might be used only for limited tasks (i.e., only for a particular type of user query). Such sources need to be *discovered* as soon as the user query is posed on the system, *analyzed* to determine the data type, access mechanisms and queries supported, and *discarded*, once the results are generated and displayed to the user. The primary challenge encountered for *on-the-fly integration* is to significantly reduce the time and skill needed to integrate data sources, and to determine whether the source needs to be modeled and integrated in the mediated schema. Hence, developing techniques for automatically modeling this functionality by probing the sources with reasonable inputs is required. Once the system understands the functionality being provided it can incorporate the source into a new or existing work-flow.

3.8.2 Decentralized Data Sharing

Current data integration systems employ a centralized mediation approach for answering user queries that access multiple sources. A centralized schema accepts user queries and reformulates them over the schema of different sources. However, the design, construction and maintenance of such a mediated schema is often hard to agree upon. For instance, data sources providing *castle* information and *train schedules* are independent, belong to separate domains, and are governed by separate organizations. To expect these data sources to be under the control of a single mediator is an unrealistic assumption.

3.8.3 Naming Inconsistencies

Entities (such as places, countries, companies, ...) are always consistent within a single data source. However, across heterogeneous sources, the same entity might be referred to with different names and in different context. To make sense of the data that spans across multiple sources, an integration system must be able to recognize and resolve these differences. For instance, in a query requiring access to sources providing air-travel information, one source may list *Departure City* and *Arrival City* as the two input locations for querying. However, another source might use *From* and *To* as its querying input locations. Even though, these inputs indicate the same concept in the domain of *travel*, resolving this complexity for an integration environment is a difficult task. Although MetaQuerier [6] has addressed this issue in great detail for a predefined set of domains, its applicability to a range of autonomous domains is a complicated task.

3.8.4 Security and Privacy

Existing information integration systems extracting data from autonomous sources assume that the information in each source can be retrieved and shared without any security restrictions [1]. However, there is an increasing need for sharing information across autonomous entities in a manner that no data apart from the answer to the query is revealed. There exist several intricate challenges in specifying and implementing processes for ensuring security and privacy measures before data from diverse sources can be integrated.

3.8.5 Web-service and Information Integration

The development of web services has been fast paced in the past few years. Web services are Web based application components that use open protocols, XML-based standards and transport protocols to exchange data with clients. Web services can offer application components such as unit conversion and weather reports. They provide application components (such as unit conversion, weather reports, etc.) and tailored datasets in XML format, amenable to information integration with other traditional HTML and XML based data sources. Service composition techniques are being investigated heavily. However, the implications of web services to information integration in the areas of query language design, domain discovery and description, query planning and optimization, and result ranking need to be thoroughly investigated.

4 Putting It Together: The Architecture and Novelty of The InfoMosaic Approach

The architecture of our InfoMosaic framework is shown in Figure 1. The user query is accepted in an intuitive manner using domain names and keywords of interest, and elaborated/refined using the *domain knowledge* in the form of taxonomies and dictionaries (synonyms etc.). Requests are refined and presented to the user for feedback (*Query Refinement module*). User feedback is accumulated and used for elaborating/disambiguating future queries. Once the query is finalized, it is represented in a canonical form (e.g., query graphs) and transformed into a query plan using a two-phase process: i) generation of logical plans using domain characteristics, and ii) generation of physical plans using *source semantics*. The plan is further optimized by applying several metrics (*Query planner and Optimizer module*). The *Query Execution and Data Extraction module* generates the actual source queries that are used by the extractor to retrieve the requisite data. It also determines whether a previously retrieved answer can be reused by checking the data repositories (XML and PostGIS) for cached data.

We have an *XML Repository* that stores extracted results from each source in a system-determined format. A separate *PostGIS Repository* is maintained for storing spatial data extracted from sources. The *Data Integrator* formulates XQueries (with external func-

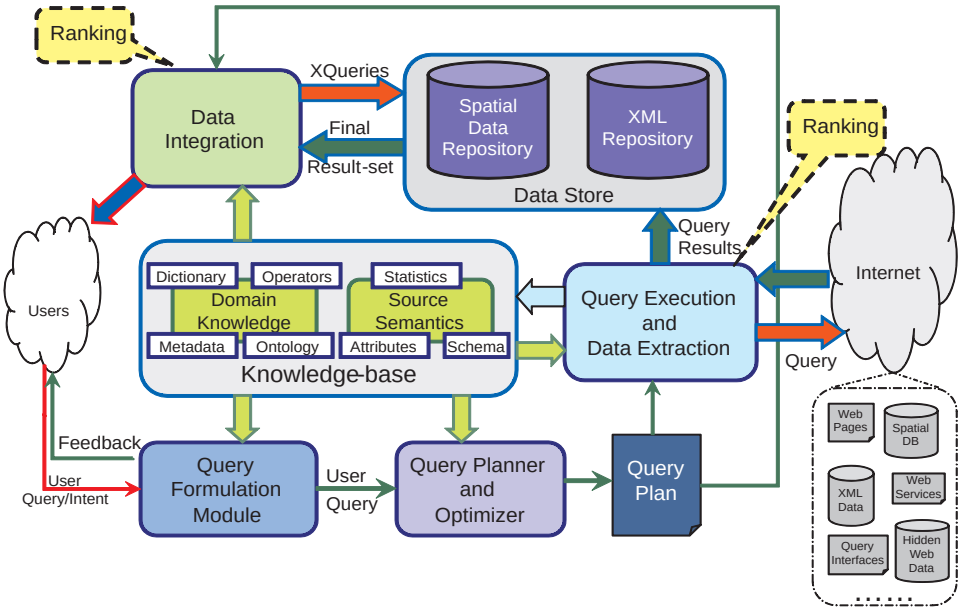

Figure 1: InfoMosaic Architecture

tions for handling spatial component) on these repositories to compute the final answers and format them for the user. *Ranking* is applied at different stages (sub-query execution phase, extraction phase, or integration phase) depending on the user-ranking metrics, the selected sources and the corresponding query plan. The *Knowledge-base* (broadly consisting of *domain knowledge* and *source semantics*) blends all the pieces together in terms of the information used by various modules. The adaptive capability of the system is based on the ability of the InfoMosaic components to update these knowledge-bases at runtime.

4.1 Novelty of Our Approach

Although several challenges in information integration problem has been addressed in a delimited context by a number of projects (as elaborated earlier), a large number of challenges still need to be tackled in the context of heterogeneous data integration on the Web. To the best of our knowledge, we are the first ones to address *multi-domain* information extraction and integration of results in conjunction with spatial and temporal data which is intended to push the state-of-the-art in functionality. We believe that it is important to establish the feasibility of the functionality before addressing performance and scalability issues. Some of the novel aspects our approach are:

1. We are formulating the problem of multi-domain information integration as an intelligent query processing and optimization problem with some fundamental differences from conventional ones. InfoMosaic considers many additional statistics, semantics, domain & source knowledge, equivalences and inferencing for plan generation and optimization. We plan to extend conventional optimization techniques to do this by building upon techniques from databases, deductive databases, taxonomies, semantic information, and inferencing where appropriate. The thrust is to develop new techniques as well as to identify and use the existing knowledge.

2. We believe that in order for this system to be acceptable, user input should be intuitive (if not in natural language). We intend to develop a feedback-centric user input which can compete with the simplicity of a keyword based search requests.

3. Incorporating spatial, temporal and spatio-temporal query conditions in addition to generic ones is a unique aspect of the InfoMosaic framework, and has not been addressed in the existing frameworks. Additionally, the issue of *ranking* integrated results at different stages in the integration process has not been considered, and is an important component of our framework.

4. We plan to choose a few communities (e.g., tourists, real-estate agents, museum visitors, etc.) each needing information from several domains and will address the problem in a real-world context. The crux of the problem here is to identify clearly the information needed (from sources, ontologies, statistics, QoS, etc.) along with the techniques and algorithms for their usage. We are addressing the problem using actual domains and web sources rather than making assumptions on data sources or using artificial sources or using small number of pre-determined sources.

5. Extensibility of the system and the ability to incrementally add functionality is a key aspect of our approach. That is, if we identify the information and techniques for representative communities, it should be possible to add other communities and domains without major modifications to the framework and modules. This is similar to the approach taken for DBMS extensibility (by adding blades, cartridges, and extenders).

6. Adaptability and learning from feedback and actions taken by the system is central to the entire framework. The entire knowledge base of various types of information will be updated to improve the system (in terms of accuracy, coverage, information content, etc.) on a continuous basis.

We believe that the task of identifying various types of semantic information needed for query/input refinement, plan generation, optimization, handling ranking and QoS constraints, data extraction from sources is central to the InfoMosaic framework. Our approach permits us to clearly separate *what* semantic information is needed from *how* to obtain this information.

5 Conclusion

Figure 2 provides a comparative analysis of the existing integration frameworks with respect to the challenges they aim to address.

As more and more data becomes available on the web, it is even more important to be able to search for complex queries instead of humans performing the task of information integration using *basic search* capabilities. Indeed, the use of Web needs to move towards more specialized content-based retrieval mechanisms (such as information integration) that do more than simply return documents. Towards this end, extensive work is needed on the higher-levels of the system, including managing semantic heterogeneity in a more scalable fashion, the use of domain knowledge in various parts of the system, transforming these systems from query-only tools to more active data sharing scenarios, and easy management of data integration systems. Extensibility of the system and the framework is extremely important as the coverage of the system should increase as we add more domain knowledge and source semantics.

The objective of InfoMosaic is to allow users to specify *what* information is to be retrieved without having to provide detailed instructions on *how* or from *where* to obtain this information. Our approach draws upon techniques from database systems, artificial intelligence, information retrieval, and the use of extended ontologies. We plan on combining techniques from these areas synergistically and extend/adapt them as needs to solve this interesting problem.

References

[1] R. Agrawal, A. V. Evfimievski, and R. Srikant. Information Sharing Across Private Databases. In *SIGMOD Conference*, pages 86–97, 2003.

[2] J. L. Ambite, C. A. Knoblock, S. Minton, and M. Muslea. Heracles II: Conditional constraint networks for interleaved planning and information gathering. *IEEE Intelligent Systems*, 20(2):25–33, 2005.

[3] Y. Arens, C. Y. Chee, C.-N. Hsu, and C. A. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *Int. J. Cooperative Inf. Syst.*, 2(2):127–158, 1993.

[4] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *VLDB*, pages 119–128, 2001.

[5] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30(1-7):107–117, 1998.

[6] K. C.-C. Chang, B. He, and Z. Zhang. Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web. In *CIDR*, pages 44–55, 2005.

[7] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic Ranking of Database Query Results. In *VLDB*, pages 888–899, 2004.

[8] S. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *IPSJ*, pages 7–18, 1994.

[9] W. W. Cohen. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. In *SIGMOD Conference*, pages 201–212, 1998.

[10] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In *DS-8*, pages 351–369, 1999.

[11] M. Doerr, J. Hunter, and C. Lagoze. Towards a Core Ontology for Information Integration. *Journal of Digital Information*, 4(1), 2003.

[12] O. M. Duschka and M. R. Genesereth. Infomaster: An Information Integration Tool. In *Intelligent Information Integration*, 1997.

[13] O. M. Duschka and M. R. Genesereth. Query Planning in Infomaster. In *Selected Areas in Cryptography*, pages 109–111, 1997.

Challenge / Framework	Query Specification Format	Query Elaboration	Query Planning & Optimization	Domain & Source Knowledge	Data Extraction	Data Integration	Ranking	Spatial & Multiple Domain Integration
Havasu	Search Based	Mining Techniques	Multi R Module	StatMiner & Indra	Source Call Programs	Append & Display	NO	NO
Ariadne	Form Based	LOOM Semantics	Theseus Module	Apollo & Mercury Modules	Phoebus Module	Prome theus Module	NO	Heracles + TerraWorld Modules
Meta Querier	Form Based	Form Assistant	NO	Meta Integrator + Explorer	Mining Techniques	Deep web Integrator	NO	NO
Tsimmis	MOBIE	OEM Query Language	Constraint Mgmt. Strategy	OEM Module	Classifiers + Extractors	Mediator Based	NO	NO
Information Manifold	CARIN	Translation Algorithms	Probabilistic Reasoning	Source Description	Source Interface Programs	Using Outer Joins	NO	NO
Infomaster	Form Based	Translation Rules	Query Facilitator	Knowledge Base	Database Wrappers	Append & Display	NO	NO
Whirl	Template Based	WHIRL Language	Modified A* Search	STIR Model	Views on RDBMS	Append & Display	NO	NO
Tukwila	XML Based Templates	Query Reformulation Module	Interleaved planning & Execution	Data Source Catalog	Location independent Wrappers	Append & Display	NO	NO
Infomosaic	*Search + Form Based*	*Domain Knowledge + Feedback*	*Logical & Physical Planning*	*Knowledge Base*	*Lixto + (XML & PostGIS) Repositories*	*XML + GML-Based*	*YES*	*YES*

Figure 2: Integration Frameworks and Challenges Addressed

[14] M. J. Egenhofer and A. U. Frank. Towards a Spatial Query Language: User Interface Considerations. In *VLDB*, pages 124–133, 1988.

[15] J. Fan, H. Khatri, Y. Chen, and S. Kambhampati. QPIAD: Query processing over Incomplete Autonomous Databases. Technical report, Arizona State University, 2006.

[16] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the world-wide web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.

[17] M. Friedman, A. Y. Levy, and T. D. Millstein. Navigational Plans For Data Integration. In *AAAI/IAAI*, pages 67–73, 1999.

[18] C. Goad and D. Speight. Map Bureau. http://www.mapbureau.com.

[19] P. M. D. Gray, A. D. Preece, N. J. Fiddian, W. A. Gray, T. J. M. Bench-Capon, M. J. R. Shave, N. Azarmi, M. Wiegand, M. Ashwell, M. D. Beer, Z. Cui, B. M. Diaz, S. M. Embury, K. ying Hui, A. C. Jones, D. M. Jones, G. J. L. Kemp, E. W. Lawson, K. Lunn, P. Marti, J. Shao, and P. R. S. Visser. KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases. In *DEXA Workshop*, pages 682–691, 1997.

[20] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vassalos. Template-based wrappers in the TSIMMIS system. In *SIGMOD Conference*, pages 532–535, 1997.

[21] B. He and K. C.-C. Chang. A Holistic Paradigm for Large Scale Schema Matching. *SIGMOD Conference*, 33(4):20–25, 2004.

[22] B. He, M. Patel, C.-C. Chang, and Z. Zhang. Accessing the Deep Web: A Survey. Technical report, University of Illinois, Urbana-Champaign, 2004.

[23] B. He, Z. Zhang, and K. C.-C. Chang. MetaQuerier: Querying Structured Web Sources On-the-fly. In *SIGMOD Conference*, pages 927–929, 2005.

[24] C.-N. Hsu and C. A. Knoblock. Reformulating Query Plans for Multidatabase Systems. In *CIKM*, pages 423–432, 1993.

[25] H.Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner. Ontology-Based Information Integration: A Survey. *IJCAI*, 2001.

[26] Z. G. Ives, D. Florescu, M. Friedman, A. Levy, and D. S. Weld. Adaptive Query Processing for Internet Applications. In *IEEE Computer Society Technical Committee on Data Engineering*, pages 19–26, 1999.

[27] G. Kabra, C. Li, and K. C.-C. Chang. Query Routing: Finding Ways in the Maze of the Deep Web. In *International Workshop on Challenges in Web Information Retrieval and Integration*, pages 64–73, 2005.

[28] S. Kambhampati, U. Nambiar, Z. Nie, and S. Vaddi. Havasu: A Multi-Objective, Adaptive Query Processing Framework for Web Data Integration. Technical report, Arizona State University, 2002.

[29] B. Katz, J. J. Lin, and D. Quan. Natural Language Annotations for the Semantic Web. In *CoopIS/DOA/ODBASE*, pages 1317–1331, 2002.

[30] C. A. Knoblock. Planning, Executing, Sensing, and Replanning for Information Gathering. In *IJCAI*, pages 1686–1693, 1995.

[31] D. L. Lee, H. Chuang, and K. E. Seamons. Document Ranking and the Vector-Space Model. *IEEE Software*, 14(2):67–75, 1997.

[32] J.-O. Lee and D.-K. Baik. SemQL: A Semantic Query Language for Multidatabase Systems. In *CIKM*, pages 259–266, 1999.

[33] M. Lesk, D. R. Cutting, J. O. Pedersen, T. Noreault, and M. B. Koll. Real Life Information Retrieval: Commercial Search Engines (Panel). In *SIGIR*, page 333, 1997.

[34] A. Y. Levy. Information Manifold Approach to Data Integration. *IEEE Intelligent Systems*, pages 1312–1316, 1998.

[35] M. Ley. Faceted DBLP. dblp.l3s.de/, 2006.

[36] L. Liu, C. Pu, and Y. Lee. An Adaptive Approach to Query Mediation Across Heterogeneous Information Sources. In *CoopIS*, pages 144–156, 1996.

[37] R. M. MacGregor. Inside the LOOM Description Classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[38] M. Michalowski and C. A. Knoblock. A Constraint Satisfaction Approach to Geospatial Reasoning. In *AAAI*, pages 423–429, 2005.

[39] A. Motro. VAGUE: A User Interface to Relational Databases that Permits Vague Queries. *ACM Trans. Information Systems*, 6(3):187–214, 1988.

[40] I. Muslea, S. Minton, and C. A. Knoblock. Hierarchical Wrapper Induction for Semistructured Information sources. In *Autonomous Agents and Multi-Agent Systems*, 2001.

[41] U. Nambiar and S. Kambhampati. Mining Approximate Functional Dependencies and Concept Similarities to Answer Imprecise Queries. In *WebDB*, pages 73–78, 2004.

[42] Z. Nie and S. Kambhampati. Joint optimization of cost and coverage of query plans in data integration. In *CIKM*, pages 223–230, 2001.

[43] Z. Nie, S. Kambhampati, U. Nambiar, and S. Vaddi. Mining source coverage statistics for data integration. In *WIDM*, pages 1–8, 2001.

[44] N. F. Noy. Semantic Integration: A Survey Of Ontology-Based Approaches. *SIGMOD Record*, 33(4):65–70, 2004.

[45] L. N. Office, the San Diego Supercomputer Center, the Northwest Alliance for Computation Science, and Engineering. The Spatial Data Workbench. http://www.lternet.edu/technology/sdw/.

[46] R. Research. POSTGIS. http://postgis.refractions.net/.

[47] A. P. Sheth and J. A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Survey*, 22(3):183–236, 1990.

[48] H. Stuckenschmidt and H. Wache. Context Modelling and Transformation for Semantic Interoperability. In *KRDB*, 2000.

[49] V. Subrahmanian. A heterougeneous reasoning and mediator system. http://www.cs.umd.edu/projects/hermes/.

[50] L. Tanasescu, A. Gugliotta, J. Domingue, L. G. Villaras, R. Davies, M. Rowlatt, M. Richardson, and S. Stincic. Spatial Integration of Semantic Web Services: the e-Merges Approach. In *International Semantic Web Conference*, 2006.

[51] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *ACM SIGKDD*, pages 350–359, 2002.

[52] A. Telang, R. Mishra, and S. Chakravarthy. Ranking Issues for Information Integration. In *First International Workshop on Ranking in Databases in Conjunction with ICDE 2007*, 2007.

[53] Z. Zhang, B. He, and K. C.-C. Chang. Lightweight Domain-based Form Assistant: Querying Web Databases On the Fly. In *VLDB*, pages 197–208, 2005.

[54] S. M. zu Eissen and B. Stein. Analysis of Clustering Algorithms for Web-Based Search. In *PAKM*, pages 168–178, 2002.