Information Integration across Heterogeneous Domains: Current Scenario, Challenges and the InfoMosaic Approach

Aditya Telang and Sharma Chakravarthy

Information Integration across Heterogeneous Domains:
Current Scenario, Challenges and The InfoMosaic Approach

Aditya Telang and Sharma Chakravarthy
IT Laboratory & Department of Computer Science & Engineering
The University of Texas at Arlington, Arlington, TX 76019.
{telang, sharma}@cse.uta.edu

Abstract

Today, information retrieval and integration has assumed a totally different, complex connotation than what it used to be. The advent of the Internet, the proliferation of information sources, the presence of structured, semi-structured, and unstructured data - all of have added new dimensions to the problem of information retrieval and integration as known earlier. From the time of distributed databases leading to heterogeneous, federated, and multi-databases, retrieval and integration of heterogeneous information has been an important problem for which a complete solution has eluded researchers for a number of decades. Techniques such as global schemas, schema integration, dealing with multiple schemas, domain specific wrappers, and global transactions have produced significant steps but never reached the stage of maturity for large scale deployment and usage. Currently, the problem is even more complicated as repositories exist in various formats (HTML, XML, Web Databases with query-interfaces, ...) and schemas, and both the content and the structure are changing autonomously. In this survey paper, we describe the general problem of information retrieval and integration and discuss the challenges that need to be addressed to deal with the general problem of information retrieval and integration as it pertains to data sources we encounter today. As the number of repositories/sources will increase in an uncontrolled manner, there is no other option but to find a solution for integrating information from different autonomous sources as needed for a search/query whose (partial) answers have to be retrieved and integrated from multiple sources. We survey the current work, identify challenges that need to be addressed, and present InfoMosaic, a framework we are currently designing to handle the challenge of multi-domain integration of data on the web.

1 Introduction

The volume of information accessible via the web is staggeringly large and growing rapidly. The coverage of information available from web sources is difficult to match by any other means. Although the quality of information provided by these sources tends to vary from one to another (sometime drastically), it is easy to search these repositories and obtain information of interest. Hence, over the last decade or so, search engines (e.g. Google, Yahoo, etc.) [1] have become extremely popular and have facilitated users to quickly and effortlessly retrieve information from individual sources. Conceptually, search engines only perform the equivalence of a simple lookup operation on one or more keywords, followed by a sophisticated ranking of the results [2]. There exist several advanced search mechanisms (or meta-search engines) that post-process the output of normal search engines to organize and classify the sources in a meaningful manner (e.g., Vivisimo [3]). Additionally, question-answering frameworks (e.g., START [4]) that are language-based retrieval mechanisms, address the issues of providing detailed information in response to queries on varied concepts (e.g., geography, science, etc.) expressed in English (or other) language. Furthermore, several commercial
web-portals (e.g. http://www.expedia.com, http://www.amazon.com, etc.) have been designed to operate on individual domains (e.g., airlines, book purchase, hotels, etc.) or a set of pre-determined sources for searching multiple repositories belonging to the same domain\(^1\) of interest and provide results based on some user criteria, such as cost, schedule, proximity, etc. In summary, efficient and effective search and meta-search engines are available that do specific tasks very well.

These retrieval mechanisms are widely used mainly because of their simplicity and absence of a learning curve as compared to other forms of information retrieval. This simplicity, on the other hand, makes it difficult to specify queries that require extraction of data from multiple repositories across diverse domains or results that are relevant to a context. Currently, there are no retrieval mechanisms that support – framing queries that retrieve results from multiple documents and sources, and combining them in meaningful ways to produce the desired result. For example, consider the query: “Retrieve all castles within 2 hours by train from London”. Although all the information for answering different parts of this query 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.

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. In this survey paper, we analyze the problem of information integration as it pertains to extracting and combining heterogeneous data from autonomous Web sources in response to user queries spanning across several domains.

The rest of the survey is organized as follows – In Section 2, we elaborate on the challenges encountered in heterogeneous information integration. Section 3 explains the dimensions along which existing information integration systems are measured and differentiated. Section 4 elaborates some of the salient approaches embraced by the research community for addressing these challenges. In Section 5, we analyze the existing frameworks, mechanisms and techniques that have been proposed to solve some of the intricate problems that are encountered in data integration. Section 6 elucidates our approach in the context of InfoMosaic, a framework proposed for web-based multi-domain information integration. Section 7 concludes the survey.

2 Challenges in Heterogeneous Information Integration

The challenge posed by information integration is further exemplified by the following query example posed by Lesk [5]: “Find kitchen furniture and a place to buy it which is within walking distance of a metro stop in the Washington DC area”. Again all the individual pieces of information for answering this request are available on the web. However, it takes considerable effort to correlate this diverse data (including spatial data) to arrive at the answers for the above query. Below, we list additional illustrative queries that indicate the gravity of the problem:

**Query 1:** Retrieve all castles within 2 hours by train from London.

**Query 2:** Retrieve flights from Dallas for VLDB 2007 Conference that have fares under $1000

\(^1\)We realize that the notion of a domain is subjective. In the context of this survey, a domain indicates a collection of sources providing information of similar interest such as travel, books, literature, entertainment etc.
Query 3: List 3-bedroom houses in Austin within 2 miles of school and within 5 miles of highway and priced under 250000$.

Query 4: Retrieve Movie Theaters near Parks Mall in Arlington, Texas showing Simpsons movie.

Query 5: Retrieve Attorney details in Colorado specializing in Immigrations and speaking Spanish along with their experience in years.

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. On the other hand, current meta-search engines (that integrate information from multiple sources of a single domain) had more success. 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 number of repositories/sources will increase steadily, there is no other option but to find a solution for integrating information from different autonomous sources as needed for a search/query whose (partial) answers have to be retrieved and integrated from multiple domains and sources. As the enumeration of challenges below indicate, existing techniques from multiple domains need to be eclectically combined as well as new solutions developed in order to address this problem.

2.1 Capturing Imprecise 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 [6], TSIMMIS [7], and Whirl [8]) 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 [9]) 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 [10], TerraWorld [11], 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.

2.2 Mapping Imprecise Intent into Precise 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 [12], Whirl [8], etc.), semi-structured data (SemQL [13], CARIN [14], StruQL [15], etc.) and vague (or unstructured) data (VAGUE [16]) 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 [17] maps imprecise user queries to a more generic query using a combination of data-mining techniques. Similarly, Ariadne [6] interprets the user-specified conditions as a sequence of LOOM statements that are combined to generate a single query. MetaQuerier’s form assistant [18] 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.

2.3 Discovery of Domain and Source Semantics

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-valued hierarchies [9] maintain a classification of the attributes of the data sources over which the user queries are formed. Ariadne uses an independent domain model [6] 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 [19]. TSIMMIS adopts an Object Exchange Model (OEM) [7], a self-describing (tagged) object model, in which objects are identified by labels, types, values, and an optional identifier. Information Manifold’s CARIN [14] 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 [20] 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 [21], 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 [22], SIMS [23], OntoBroker [24], 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 [25]. 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 [26] is enormous. The MetaQuerier project [27] 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 sourcespecific 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.

2.4 Query Planning 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) [9] provides a guarantee on the cost and coverage of the results generated on a query by approximating appropriate source statistics. Ariadne’s Theseus [6] 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 [14] 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 [9] 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.

2.5 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 [6] 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 [28], MetaQuerier [27], TSIMMIS [29], InfoMaster [30], and Tukwila [21]), 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* [31], *Stalker* [28], etc.. In the case of spatial data integration, (e.g., eMerges system [32]), ontologies and semantic web-services are defined for integrating spatial objects, in addition to wrappers and mediators. *Heracles* [33] (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 [34] to organize data and services for handling distributed datasets.

Information Manifold [14] 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.

### 2.6 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.

### 2.7 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.

**Ranking Integrated Results:** 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 [35].

However, unlike the domains of information retrieval [36] or even databases [37], 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.

**Decentralized Data Sharing:** Current data integration systems employ a centralized mediation approach [38] 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 companies. To expect these data sources to be under the control of a single mediator is an unrealistic assumption.

**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.

**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 [39]. 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 Dimensions for Integration

Existing information integration systems (elaborated in Section 5) tend to be designed along several dimensions such as:

**Goal of Integration:** It indicates the overall goal of the integration framework. Some systems are portal-oriented (e.g., Whirl [8], Ariadne [40] [41], InfoMaster [42]) in that they aim to support an integrated browsing experience for the user. Others are more ambitious in that they take user queries and return results of running those queries on the diverse yet appropriate sources (e.g., Havasu [9] [43]).

**Data Representation:** Refers to the design assumptions made by the integration system regards to the syntactic nature of the data being exported by the sources. Some systems assume the existence of structured data models (e.g., SIMS [44], Havasu [9]). However, since most integration frameworks perform a web-based fusion of data, they assume the co-existence of semi-structured and unstructured data (e.g. Ariadne [40], TSIMMIS [45]).

**Source Structure:** Illustrates the assumptions made on the inter-relationship between sources. Most systems assume that sources they are integrating are complementary (horizontal integration) in that they export different parts of the schema (e.g. Ariadne [41]). Others consider the possibility that sources may be overlapping (vertical integration) (e.g. Havasu [46], Tukwila [47]) in which case aggregation of information is required, as opposed to pure integration of information.
Domain and Source Dynamics: Refers to the extent to which the user has control over and/or is required to specify the particular sources that are needed to be used in answering the query. Some systems (e.g. Ariadne [33], BioKleisli [48]) require the user to select the appropriate sources to be used. Others (e.g. TSMIMMIS [7], Havasu [43]) circumvent this problem by hard-wiring specific parts of the integrated schema to specific sources.

User Expertise: Indicates the type of users that the system is directed towards. The systems that primarily support browsing need to assume very rudimentary expertise on the part of users. In contrast, systems that support user-queries need to assume some level of expertise on the users part in formulating queries. Some systems might require user queries to be formulated in specific languages, while others might provide significant interactive support for users in formulating their query.

4 Approaches for Integration

Over the past two decades, various approaches have been suggested and adopted in the pursuit of achieving an ideal information integration system. In this section, we provide a brief description of the prominent approaches that form the basis for the design of existing integration frameworks:

Mediator: It is one of the most notable approaches adopted by many integration frameworks (Ariadne [6], TSMIMMIS [7], Havasu [9], ...). A mediator (in the information integration context) is a system responsible for reformulating user queries (formed on a single mediated 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. Modeling the relation between the sources and the global schema is therefore a crucial aspect. The two distinct approaches for establishing the mapping between each source schema and the centralized global schema are: i) Global-as-view (GAV), that requires the global schema to be represented in terms of the underlying data sources, and ii) Local-as-view (LAV), 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.

Warehousing: This approach [15] 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. Warehousing emphasizes data translation, as opposed to query translation in mediator-based integration [38]. In fact, warehousing requires that all the data loaded from the sources be converted through data mapping to a standard unique format before it is stored locally. 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.

Ontological: In the last decade, semantics (which are an important component for data integration) gained popularity leading to the inception of the celebrated ontology-based integration approach [49]. The Semantic Web research community [50], [51], [52], [53] has focused extensively on the problem of semantic integration [54] and the use of ontologies for blending heterogeneous schemas across multiple domains. Their pioneering efforts have provided a new dimension for researchers to investigate the challenges in information integration. A number of frameworks designed
using ontology-based integration approaches [49] have evolved in the past few years.

**Federated:** It [55] is developed on the premise that, information needed to answer a query is gathered directly from the data sources in response to the posted query. 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 [56] 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 (e.g., Indus [55]), 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.

**Navigational:** Also known as *link-based approach* [57], is based on the fact that an increasing number of sources on the web require users to manually browse through several web-pages and data sources in order to obtain the desired information. In fact, the major premise and motive justifying this approach is that some sources provide the users with pages that would be difficult to access without point-and-click navigation (e.g., *hidden-web* [26]).

5 Current Integration Techniques and Frameworks

Currently, a number of techniques, and frameworks have tried to address several challenges in the problem of heterogeneous data integration in a delimited context. In this section, we elaborate some of the prominent frameworks.

**Havasu:** A multi-objective query processing framework comprising of multiple functional modules, Havasu [9], addresses the challenges of *imprecise-query specification* [58], *query optimization* [59], and *source-statistics collection* [60] in single-domain web integration. The AIMQ module provides query independent solution to efficiently handle *imprecise* user queries using a combination of *source schema collection*, *attribute dependency mining* [61], and *source similarity mining* [46]. Unlike traditional data integration systems that aim towards minimizing the cost of query processing, Havasu’s *StatMiner* [62] provides a guarantee on the cost as well as the coverage of the results generated on a given user query, by approximating the coverage and overlap statistics of the data sources based on *attribute-valued hierarchies*. The *Multi-R optimizer* [9] module uses these source statistics for generating a multi-objective (cost and coverage) query optimization plan. The Havasu framework has been applied for the design of BibFinder [43], a publicly available computer science literature retriever, that integrates several autonomous and partially overlapping bibliography sources.

**MetaQuerier:** It is made up of two distinct components that address the challenges in *exploration* and *integration* of *deep-web* sources [26]. In contrast to the traditional approaches (e.g., Wise-Integrator [63]), that aim towards integrating web data sources based on the assumption that query-interfaces can be extracted perfectly, MetaQuerier tries to perform source integration by extracting query interfaces from raw HTML pages with subsequent schema matching. Hence, in essence, MetaQuerier strives to achieve data mining for information integration [64] i.e., it mines the observable information to discover the underlying semantics from web data sources. The *Meta-Explorer* [27] is responsible for *dynamic source discovery* [65] and *on-the-fly integration* [66] for the discovery, modeling, and structuring of web databases, to build a search-able source repository. On
the other hand, the *MetaIntegrator* [67] focuses on the issues of on-line source integration such as *source selection, query mediation, and schema integration*. In contrast to traditional integration systems, MetaIntegrator is dynamic i.e., new sources may be added as and when they are discovered.

**Ariadne:** It extended the information integration approach adopted by the SIMS mediator architecture [68] to include *semi-structured and unstructured* data sources (e.g., web data) instead of simple databases, by using specially designed *wrappers* [28]. These wrappers, built around individual web sources, allowed querying for data in a database-like manner (for example, using SQL). These wrappers were generated semi-automatically using a machine learning approach [69]. Ariadne also constructed an independent *domain model* [70] for each application, that integrates the information from the sources and provides a single terminology for querying. This model was represented using the LOOM knowledge representation system [19]. The SIMS query planner did not scale effectively when the number of sources increased beyond a certain threshold. Ariadne solved this problem by embracing an approach [71] that is capable of efficiently constructing large query plans, by pre-compiling part of the integration model and using a local search method for query planning.

Since its inception, Ariadne has been divided into a number of individual projects such as – *Apollo* [72], *Prometheus* [73], and *Mercury* [74] for addressing separate issues and challenges in heterogeneous information integration. The early applications of Ariadne included a *Country Information Agent* [40], that integrated data related to countries from a variety of information sources. Ariadne was also used to build *TheaterLoc* [41], a system that integrated data from restaurants and movie theaters, and placed the information on a map for providing efficient access to naive users. After its atomization, three individual and diverse application projects have been developed under the Ariadne project – *Heracles* [33] (an interactive data-driven constraint-based integration system), *TerraWorld* [11] (a geospatial data integration system), and *Poseidon* [75] (that focusses on the composition, optimization, and execution of query plans for bioinformatics web-services).

**Information Manifold:** It focused on efficient *query processing* [14] by accessing sources that are capable of providing an appropriate answer to the queries posed by users. In order to facilitate efficient user query formulation, as well as to represent background knowledge of the mediated schema relations (designed over several autonomous sources), Information Manifold proposed an expressive language, *CARIN* [76] modeled using a combination of *Datalog* (database query language) [77] and *Description Logics* (knowledge representation language) [78]. Information Manifold also proposed algorithms [79] for translating user queries, posed on the mediated schema of data sources, into a format that maps to the actual relations within the data sources. The query-answering approach adopted by Manifold ensured that only the relevant set of data sources are accessed when answering a particular user query. This approach differed from the one adopted by Ariadne, that uses a general purpose planner for query translation. In addition, Information Manifold proposed a method [80] for representing local-source completeness and an algorithm [81] 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, Information Manifold suggested the use of *probabilistic reasoning* [82] for the ordering of data sources that appear relevant to answer a given query. Such an ordering is dependent on the overlap between the sources and the query, and on the coverage of the sources.

**TSIMMIS:** It addressed the challenges in integration of *heterogeneous data* extracted from *structured* as well as *unstructured* sources [7]. TSIMMIS adopted a schema-less approach (i.e., no single global database or schema contained all information needed for integration) for retrieving information from dynamic sources (i.e., when source contents changed frequently). Each information
source was covered by a wrapper that logically converted the underlying data objects to a common information model. This logical translation was done by converting queries into requests based on the information in the model that the source could execute, and converting the data returned by the source into the common model. TSIMMIS adopted an Object Exchange Model (OEM) [83], a self-describing (tagged) object model, in which objects were identified by labels, types, values, and an optional identifier. These objects were requested with the aid of OEM-QL [84], a query language specifically designed for OEM. The mediators [83] were software modules that refined information from one or more sources [85], and accepted OEM-QL queries as inputs to generate OEM objects. This approach allowed the mediators to access new sources transparently in order to process and refine relevant information efficiently. End users accessed information either by writing applications that requested OEM objects, or by using generic browsing tools supported by TSIMMIS, such as MOBIE (MOsaic Based Information Explorer) [7]. The query was expressed as an interactive world wide web page or in a menu-selected format. The answer was received as a hypertext document.

Garlic [86], a sister project of TSIMMIS, was developed at IBM to enable large-scale integration of multimedia information. It was applied in the late 1990s for multimedia data integration in the fields of Medicine, Home Applications, and Business Agencies.

InfoMaster: It [42] was a framework that provided integrated access to structured information sources. InfoMaster’s architecture consisted of a query facilitator, knowledge base, and translation rules. The facilitator analyzed the sources containing relevant information needed to answer user queries, generated query plans to access these sources, and mapped these query plans to the sources using a self-generated wrapper [87]. The knowledge base was responsible for the storage of all the rules and constraints [20] required to describe heterogeneous data sources and their relationships with each other. The translation rules described how each distinct source could relate to the reference schema. InfoMaster was developed and deployed at Stanford University for searching housing rentals in Bay Area, San Francisco and for scheduling rooms at Stanford’s Housing Department [88]. It was also the basis for Stanford Information Network (SIN) project [89] that integrated numerous structured information sources on the Stanford campus. InfoMaster was commercialized by Epistemics in 1997.

Tukwila: An adaptive integration framework, Tukwila [47], tackles the challenges encountered during the integration of autonomous sources on the web, namely source statistics, data arrival, and source overlap and redundancy. The major contributions of the Tukwila framework are – interleaved query planning and execution, adaptive operators, and a data source catalog. Query processing in Tukwila does not require the creation of a complete query execution plan before the query evaluation step [21]. If the query optimizer concludes that it does not have enough metadata 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. Tukwila incorporates operators that are well suited for adaptive execution, and minimizing the time required to obtain the first answer to a query. In addition, the Tukwila execution engine includes a collector operator [21] that efficiently integrates data from a large set of possibly overlapping or redundant sources. The metadata obtained from several sources is stored in a single data source catalog. The metadata 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.

Whirl: The primary contribution of Whirl has been the Whirl Query Language [8]. This language, based on the concepts extracted from Database Systems (the syntax is based on soft se-
mantics [90] and very similar to SQL), Information Retrieval (uses an inverted-indexing technique for generating top-k answers), and Artificial Intelligence (using a combination of A* search [91]), has been used in incorporating single domain queries across structured and semi-structured static sources. The Whirl framework was applied for design of two applications: i) Children’s Computer Games, that integrated information from 16 similar web-sites, and ii) North American Birds that contained a set of integrated databases with large number of tuples, the majority of which pointed to external web-pages.

**Ontology-based Integration Systems:** Semantics play an important role during integration of heterogeneous data. In order to achieve semantic interoperability, the meaning of information that is interchanged across the system has to be understood. The use of ontologies, for the extraction of 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 [22], SIMS [44], OntoBroker [24], and InfoSleuth [92]. Other ontology-based systems such as PICSEL [93], OBSERVER [94], and BUSTER [25] did not propose any methods for creation of ontologies. The literature associated with these systems makes it clear that there is an obvious lack of real methodology for ontology-based development.

**GeoSpatial Integration Systems:** Vast amount of data available from the web contain spatial information explicitly or implicitly. Implicit spatial information includes the location of an event. For example, many news reader today use GeoRSS format, which is an enhancement to the W3C standard RSS XML file format to include location tags. Explicit spatial objects such as road networks and disease distributions over different parts of the earth are commonly provided by different organizations in different data formats through different interfaces. To this end, several systems have been proposed to address the issues of geospatial integration. Heracles [33] has tried to combine online and geospatial data in a single integrated framework for assisting travel arrangement and integrating world events into a common space. The spatial mediated approaches (systems derived using the concepts from Ariadne) [95] combines spatial information using wrappers and mediators. A storage Resource Broker was used in LTER spatial data workbench [34] to organize data and services to manage distributed datasets as a single collection. Query evaluation plan generation is addressed within a spatial mediation context in [96]. As generic models, XML/GML and web services are also widely used in other spatial integration systems [97], [98], [99], [100], [101]. The eMerges system [32] further defines ontologies for spatial objects and uses semantic web services for integration. The other line of work on spatial data integration focused on studying the conflation of raster images with vector data [102] and satellite image with vector maps [103].

The other line of work on spatial data integration [102] focused on studying conflation of raster images and vector data. The process involves the discovery of control point pairs, which are a set of conjugate point pairs between two datasets. Extension on integrating satellite image with vector maps was investigated in [103]. A framework to combine spatial information based on a mediator that takes metadata on the information needs of the user was proposed in [95]. This infrastructure uses the view agent approach as the method of communication and is validated in mobile data settings. Grid based architecture that utilizes service-oriented view to unify various geographic resources was briefly introduce in [97].

**Biological Integration Systems:** In recent years, integration of heterogeneous biological and genomic data sources [104] has gained immense popularity. A number of prominent systems such as Indus [55], SRS [105], K2/BioKleisli [48], TAMBIS [106], DiscoveryLink [107], BioHavasu [104], Entrez [108], and BioSift Radia [104] have been designed for resolving some of the intricate
6 The InfoMosaic Approach

6.1 InfoMosaic Architecture and Dataflow

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 functions 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 challenges in integrating assorted biological data.
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.

6.2 Challenges Addressed by InfoMosaic

Capturing and Mapping Imprecise Intent into Precise Query: 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.

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 for easy specification of spatial relations including metric, topological, and directional relationships pertaining to heterogeneous datasets from the web.

Plan Generation and Optimization: We view this 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.

Data Extraction: We use Lixto [31], 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 [31] (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, 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.

Data Integration: Extraction of spatial and non-spatial is done with respect to separate Post-GIS and XML repositories respectively. We then generate and execute queries (XQuery for XML and spatial queries for Post-GIS whose results are converted to GML for further processing) to 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.

**Result Ranking:** We are currently addressing this challenge [35] 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.).

**Representation of Domain Knowledge and Source Semantics:** We address this challenge by adopting 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 is done by combining and enhancing the extensive work carried out in the areas of – knowledge representation, domain knowledge aggregation, deep-web exploration, and statistics collection. 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 adopted in our InfoMosaic framework.

### 6.3 Novelty of Our Approach

As seen in Figure 2, although several challenges in information integration problem has been addressed in a delimited context by a number of projects, 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:

- 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 will 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.

- We plan on choosing 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 rules and inferencing techniques to develop algorithms and techniques for their usage. We will address the problem using actual domains and web sources rather than making assumptions on data sources or using artificial sources (as in Havasu [9]) or using small number of pre-determined sources (as in Infomaster [42] or Ariadne [6]); however, our techniques will build upon and extend current approaches.

- Incorporating spatial and temporal data is unique to our proposal as, to the best of our knowledge, this has not been addressed in the literature on information integration2.

---

2Retrieval of images and location data has been attempted. Currently *mashup*ing locations on a map in Web 2.0
## Figure 2: Integration Frameworks and Challenges Addressed

- Extensibility of the system and the ability to incrementally add functionality will be 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).

- 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 request.

- Adaptability and learning from feedback and actions taken by the system will be central to the whole project. 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.

### 7 Conclusion

As this survey elicits, the research community has witnessed significant progress on many aspects of data integration over the past two decades. As elaborated in Sections 3, 4 and 5, flexible architectures for data integration, powerful methods for mediating between disparate data sources, tools for rapid wrapping of data sources, methods for optimizing queries across multiple data sources are a few of the important advances achieved towards solving the problem of information integration.
However, 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.

Furthermore, the emergence of web-databases and related technologies have completely changed the landscape of the problem of information integration. First, Web-databases provides access to many valuable structured data sources at a scale not seen before, and the standards underlying web services greatly facilitate sharing of data among corporations. Instead of becoming an option, data integration has become a necessity. Second, business practices are changing to rely on information integration – in order to stay competitive, corporations must employ tools for business intelligence and those, in turn, must glean data from multiple sources. Third, recent events have underscored the need for data sharing among government agencies, and life sciences have reached the point where data sharing is crucial in order to make sustained progress. Fourth, personal information management (PIM) is starting to receive significant attention from both the research community and the commercial world. A significant key to effective PIM is the ability to integrate data from multiple sources.

Finally, we believe that information integration is an inherently hard problem that cannot be solved by just a few years of research. In terms of research style, the development of benchmarks, theoretical research on foundations of information integration as well as system and toolkit building is needed. Further progress in this area will be significantly accelerated by combining expertise from the areas of Database Systems, Artificial Intelligence, and Information Retrieval.
Bibliography


[34] 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/.


