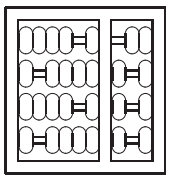

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A Survey on Information Systems Interoperability

Renato Fileto *Claudia Bauzer Medeiros*

Technical Report - IC-03-030 - Relatório Técnico

December - 2003 - Dezembro

A Survey on Information Systems Interoperability

Renato Fileto[1,2] Claudia Bauzer Medeiros[2]

[1]Embrapa Information Technology
Brazilian Agricultural Research Corporation
fileto@cnptia.embrapa.br

[2]Institute of Computing
University of Campinas
{fileto|cmbm}@ic.unicamp.br

Abstract

The interoperability of information systems has been pursued for a long time and is even more demanded in the Internet era. This paper reviews the literature in this area, from the database perspective. It covers work on interconnection of databases, classification of data integration problems, major standards and architectures, and the most recent developments in the fields of semantic Web, Web services and scientific workflows.

Keywords: Interoperability, data integration, semantic Web, Web services, ontologies, scientific workflows.

1 Introduction

The traditional paradigm for information systems development is based on the cycle modeling-design-implementation, and considers a single database framework, with one schema using one data model. The advent of heterogeneous systems and, more recently, the Web, is changing this picture. Large amounts of data are available in distinct formats and platforms. Data repositories varies from structured database management systems to unstructured files. The lack of agreement on data representation and semantics across heterogeneous systems makes the interoperability problem very complex.

Web systems are in permanent evolution, with new devices, new data sources and new requirements. The possibility of dynamic connections among systems components on the Web adds complexity to the situation. The demand for interoperability has boosted the development of standards and tools to facilitate data transformation and integration. Nevertheless, there are still many challenges to be met, especially those concerned with data semantics and behavior of cooperative systems.

This work surveys some results from the literature related with interoperability and, more specifically, data integration. Our goal is the construction of data warehouses (or materialized views) integrating several kinds of data sources, particularly for scientific applications in agriculture. Data warehouses are a suitable starting point for research and experiments on data integration. The maintenance of consolidated data at the warehouse confers greater versatility to data representation and manipulation. The unidirectional flow

of data from the sources to the warehouse, as well as the warehouse update policy which does not require on-line access to data sources, simplifies data processing. The problem can be decomposed into two steps (i) extracting data from the sources to feed the warehouse, and (ii) integrating these multiple source data into the warehouse. The emphasis of this work is on the second step. The focus is on representational and semantic issues, and the fundamental data integration problems.

Distinct data sources may be maintained independently. In fact, autonomous management of databases is frequently a prerequisite for information systems. However, valuable information may be extracted when collections of data obtained from different data sources are analyzed as a whole. The integrated analysis of data from different sources triggers a wide variety of data heterogeneity problems. Furthermore, connection of autonomous heterogeneous databases complicates classical database problems such as consistency maintenance, concurrency control, transactions and distributed query processing, and optimization. Our research is not concerned with any of these problems. Only consistency maintenance is considered in some degree. The core of our research is semantic data heterogeneity, especially when scientific data are involved.

Instead of trying to coerce all data into a single unified view in one step, we consider integration of small collections of data, in several points of distributed and cooperative processes. Integrated views of selected data sets, materialized or not, define the inputs of data processing activities of distributed processes. The outputs of such an activity, regarded as a data set or service, can be the input of another one. Thus, complex processes involving data integration can be built by composing data sets and services in an open environment like the Web.

The remainder of this paper is organized in the following way. Section 2 presents basic concepts related with information systems interoperability. Section 3 analyzes interoperability in the context of database systems. Section 4 focuses on data representation, data heterogeneity conflicts, and data integration, establishing a framework to analyze related problems and proposed solutions. Section 5 presents the most typical apparatus for data integration. Section 6 describes the the major standards and technologies of the semantic Web. Section 7 outlines the Web services technology and how it can be used to build cooperative distributed systems. Section 8 refer to applications demanding technology to support interoperability, particularly in scientific realms. Finally, Section 9 presents the conclusions.

2 Information Systems Interoperability

Interoperability is the ability of two systems to exchange information, and correct interpret and process this information [131, 105, 92, 9]. It requires some degree of compatibility between systems, to enable data exchange and correct interpretation. Ideally, cooperative systems should be compliant with computational and application domain standards. However, this level of standardization may be impossible to attain in practice, due to the rate of technological changes, the lack of universally accepted standards, the existence of legacy systems, or just for reasons of autonomy of each information system. Thus, in many cases,

the only way to reach interoperability is by publishing the interfaces, schemas and formats used for information exchange, making their semantics as explicit as possible, so that they can be properly handled by the cooperative systems.

2.1 Viewpoints of Systems Interoperability

Hasselbring [107] shows that information systems' interoperability must be considered from three viewpoints: application domain, conceptual design and software systems technology. Figure 1 illustrates the structure of a set of information systems and their interoperability in each one of these viewpoints.

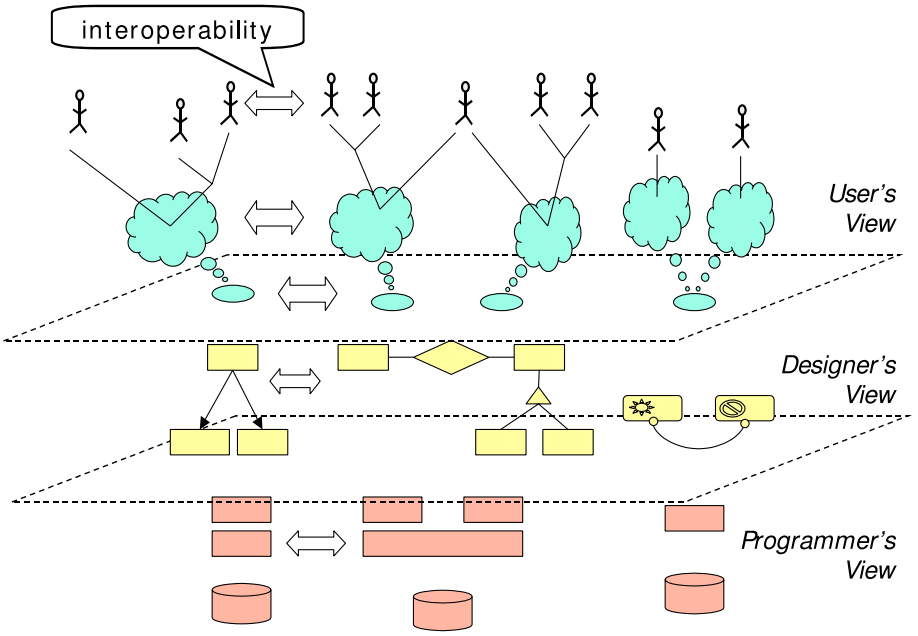

Figure 1: The viewpoints of information systems interoperability

The *user's viewpoint* concerns the distinct views and specializations of domain experts. The *designer's viewpoint* refers to requirements modeling and systems design. The *programmer's viewpoint* refers to the systems implementation.

Conflicts may appear in each of those three viewpoints. On the other hand, interoperability must be achieved in all these viewpoints, i.e., users of a system must understand information coming from another system, the system design must accommodate the "foreign" data, and the computer programs must automate information exchange (i.e., the data transfers and transformations). The hardest problems of data interoperability occur at the application and conceptual viewpoints [2].

Furthermore, each viewpoint has the instance level (solutions, projects, application programs), the meta-level (with approaches and models used to describe the characteristics of the instances), and, maybe, the meta-meta level, where the models are defined. Hence, heterogeneity can also be considered at successive levels of abstraction.

2.2 Technologies addressing Interoperability

The growth of computer networks has pushed the development of systems communication technologies beyond protocols for message passing. Several paradigms related with distributed heterogeneous systems interoperability can be singled out in the literature. Some of the most prominent of these paradigms in the Internet era are described in the following.

Distributed objects is the paradigm on the core of technologies like CORBA and DCOM [179]. Each object has an object id, the code to implement its behavior, and a state determined by the value associated with a number of internal variables. An object encapsulates it internal state and code and provides an interface based on methods to externally access and modify its state. Distributed objects communicate with each other through remote method invocation. *CORBA* (Common Object Request Broker Architecture) [48, 179] is the architecture of OMG (Object Management Group) for distributed objects. CORBA objects can be anywhere in a network and are accessed by remote clients, via method invocations, without having to know where each server object resides, what operating system it executes on and how the object is implemented. The language and the compiler used to create CORBA server objects are transparent to clients.

Infopipes [191] are building blocks to implement stream data processing. An infopipe is a language and platform independent abstraction for a data flow from a producer to a consumer. It includes data processing, buffering and filtering. The infopipe model includes facilities for managing quality of service properties (e.g., performance, availability, security), composing and restructuring data flows during execution. This model has inherent parallelism and embraces content semantics and user requirements, allowing information flow control and resource use optimization.

Peer-to-Peer [164] refer to a class of systems that employ resources distributed across a network to perform some function in a decentralized fashion. The resources encompass processing power, data, storage means and network bandwidth. The function can be distributed computing, contents sharing, communication or collaboration. The key characteristic of a peer-to-peer system is that, in opposition to the client-server architecture, each peer can provide some service to other peers, at the same time that it benefits from the services provided by other peers of its community. Peer-to-peer systems, such as Napster, and Kazaa, became popular for allowing people to share audio and video files on the Web.

Composite Web Services [216, 231] use Web services – i.e., self-describing and independent software modules accessible through the Internet – as the building blocks to construct inter-institutional cooperative processes. Web services communicate via messages, using standard Web protocols. These services encapsulate autonomous systems components with Web-based interfaces, taking advantage of the ubiquity of the Web to provide wide access to those components. The fundamental problems of this paradigm are the discovery of the services available on the Web to fulfill a particular need; and the coordination of services in distributed processes to achieve specific

goals. Web services technology has been developed and applied in areas like electronic commerce and finance. Our research combines Web services, workflows, and semantic Web technology, to solve problems of scientific applications involving data integration and cooperative work on the Web.

XML and Java are also expected to play an important role in the implementation of interoperable distributed information systems [41, 178]: the former as a syntactic standard for data representation (Section 6.1), and the latter as a portable language, allowing the transference of source coded objects' behavior from one platform to another.

3 Database Systems Interoperability

Information systems are characterized by the flow consisting of "data input, processing and output". The uncoordinated creation of heterogeneous files to store data of autonomous systems leads to problems when different applications have to access shared data. Database systems were proposed to solve these problems in centralized environments [137].

3.1 Centralized Database Systems

Database and database management systems (DBMS) [66, 67, 5] are among the most common means of managing data. A *centralized database system* accommodates all the data of an organization in a unique internal schema. *Views* [23, 224, 81, 207], or external schemas, are distinct logical database images, allowing (groups of) users to access a central database according to their specific needs. A view is usually built by using a database query language to write a query defining an image of a limited amount of data.

Database views are assigned to particular applications according to users' requirements and privacy concerns. A view can be materialized or non-materialized. *Materialized views* are copies of data to support different database images. *Non-materialized views*, on the other hand, are just abstractions, produced by translating requests to the abstract views into requests to actual database or lower level views.

The user of a database (or view) must know the data model employed and the (external) schema, in order to access the database directly through the DBMS. An alternative approach is the construction of application programs atop the DBMS to help users in their daily activities. The development of systems integrating different databases demands considerable coordination of the teams responsible for the distinct databases, views and application programs. This coordination is very difficulty to be achieved, even when the integration involves only a few departments within the same organization.

3.2 Heterogeneous Database Systems

Heterogeneous database systems (HDBS) [66, 201, 137, 113, 5] are software packages that integrate various preexisting database systems (DBSs) or HDBSs called components. The same component can participate in various HDBSs. Components can be developed independently and without any concern about subsequent integration.

Sheth and Larson [201] characterize HDBSs using three orthogonal axes: heterogeneity, distribution, and autonomy. The *heterogeneity* of a HDBS depends on the number and severity of discrepancies among its constituent DBSs, with respect to their schemas, data models, query languages, transaction management capabilities, DBMS, hardware, operating systems and communication protocols. Discrepancies can appear at any abstraction level (data instances, schema, data model). The heterogeneity can be reflected in the data representation or be just a matter of interpretation. *Distribution* refers to the location of the HDBS' components. In principle, distribution is orthogonal to heterogeneity. A distributed system can involve different hardware, software and communication platforms. *Autonomy* refers to the freedom of the HDBS' components to define and manage their databases. The need for maintaining autonomy and the demand for sharing data are often conflicting requirements. The integration of different databases cannot completely block the capacity of each component DBS to manage its data without interference of the HDBS general manager [5]. Autonomy can be classified in four categories [201, 5]:

1. *Design autonomy* refers to the independence of each component DBS to design its database.

2. *Communication autonomy* refers to the ability of a component DBS to decide whether to communicate with other component DBSs. A component DBS with communication autonomy is able to decide when and how it responds to a request from another component DBS.

3. *Execution autonomy* means that a component DBS is independent to execute operations (requested both locally and externally), with full control of transaction processing.

4. *Association autonomy* asserts that component DBSs can independently decide what information they want to share with the HDBS, to which requests they reply, when to start and when to finish their participation in the HDBS.

3.3 Integrated Access to Multiple Databases

The approaches to enable integrated access to multiple physical databases can be roughly classified in two categories: schema integration [18, 66] and the federated approach [136, 201, 137]. The former consists in providing some unified schema through which the users access the integrated data. The latter, on the other hand, can just supply some means for accessing exported views of the heterogeneous databases, leaving much of the data integration onus to the users. Figure 2 illustrates the differences between these approaches. In the distributed approach (on the left), the schema of each distributed database is a view of the unified schema. In the federated approach, on the other hand, the export/import schemas of the federated databases are externally handled. The schema integration approach makes data heterogeneity transparent to the users, while the federated approach concede more autonomy to the component databases.

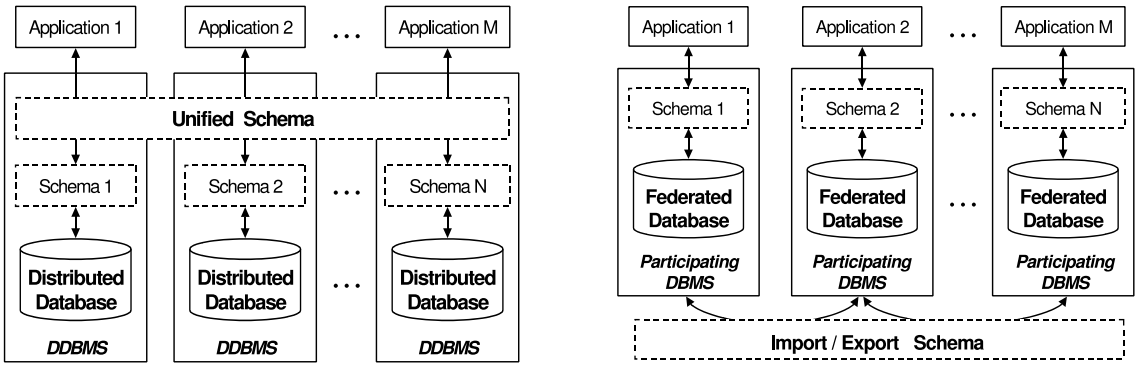

Figure 2: Distributed and federated database systems

There are several options for implementing HDBSs, with varying coupling degrees among the component DBSs, and offering different trade-offs between cooperation and autonomy. Elmagarmid and Pu [66] give an introduction to such systems, classifying them as follows.

- *Distributed database system (DDBS)* [66, 5, 67, 182] consists of a single logical database that is physically distributed. Despite the physical fragmentation of data, a DDBS supports a single data model and query language, with one schema integrating all its contents.
- *Federated database system (FDBS)* [201] (also called *heterogeneous database system – HDBS*) is a distributed database system allowing heterogeneous components with different data models, query languages or schemas.
- *Multidatabase system (MDBS)* [136, 137] is a collection of loosely coupled databases. The key properties of a MDBS are the autonomy of the participant databases and the absence of a globally integrated schema. MDBSs are employed when users want to preserve their autonomy, even to the point of refusing to participate in a globally integrated schema.

All these database systems architectures rely on some integrated or export/export schema. However, they do not address the resolution of data heterogeneity conflicts to build such an schema. They either consider that this problem has been solved or leave it to the user.

3.4 Web Databases

Web Databases [54, 221, 93, 172, 32] make data stored in local databases accessible through the Web, enabling applications like on-line stores and digital libraries. The most common interfaces for querying Web databases are forms and navigation menus on Web browsers. The query specification resulting from a user interaction with such an interface is encoded and sent to a Web Server, which submits the query to the DBMS. The result is converted into HTML format to be returned via the Internet and showed in the browser. Options

for implementing the interaction between the Web Server and the DBMS are described in [130, 65].

The challenge of the querying Web databases research is the construction of a unified and simple interface. The most common approach to solve this problem is the generation of wrappers and mediators to integrate data from Web pages provided by Web databases [221, 32, 31, 143]. These solutions tend to be complex, inefficient and unsuitable in many cases, due to the dynamics of the sources interface and availability. Other solutions available in the literature include [172, 93, 54]. Neiling *et al.* [172] present automated means to recover and integrate the contents of related Web databases (e.g., movie databases). Gravano *et al.* [93] describe a system to organize Web databases in hierarchies of classes, according their contents. Silva *et al.* [54] use keywords specified by the user to derive structured queries to be submitted to one or more DBMSs.

4 Data Integration

Heterogeneous data are those data presenting differences in their representation or interpretation, although referring to the same reality [136]. *Data heterogeneity conflicts* are the incompatibilities that may occur among distinct data sets. The interoperability problem considered in this section is *data integration* [63, 185], i.e., providing a single view for a set of heterogeneous data, with unified syntax, structure and semantics. *Data integration* involves the resolution of heterogeneity conflicts and transformations of source data to accommodate them in the integrated view.

In order to make data integration possible, it is necessary, at first, to categorize the kinds of data to be integrated and the heterogeneity conflicts. Then, conflicts can be solved in a sequence determined by their categories. The rest of this section discusses the proposals available in the literature and defines a framework to analyze and handle data integration problems.

4.1 Data Structuring

Structured Data

Conventional database systems take advantage of rather strict data structuring, expressed via a database schema using a data model, to provide data management facilities, with efficient data access and consistency maintenance. That is the case of the classical relational database management systems and even the object-oriented systems.

Data structuring presents virtues and drawbacks with respect to data integration. On the one hand, structure grants uniformity for data processing and helps maintaining consistency. On the other hand, an structured integrated view from two or more heterogeneous data sets is sometimes very difficult to obtain.

Semantic data models [18], such as the entity-relationship data model, allow data to be described in an abstract and intelligible manner, at the conceptual level. Thus, these models can facilitate data integration. However, semantic data models are not versatile enough and information can be lost on converting data among heterogeneous database schemas using

these data models. The automation of the data conversion process is also difficult, because of the gap between the implementation and the conceptual viewpoints.

Semi-structured Data

Semi-structured data [2, 1, 29, 104, 184] are those data whose structure is irregular and partially known. In order to allow the identification of the data elements in the irregular structure, semi-structured data have to be self-describing. Thus, the data and basic descriptions of their structure and meaning (metadata) are assembled together. Differently from structured data, where structure (type and schema) are defined prior to the creation of data instances, semi-structured data instances can be created at the same time their structure is defined.

Semi-structured schemas and data models are usually formalized as graphs, whose nodes represent data elements and whose edges represent nesting and reference relationships between data elements [2, 184]. This data structuring is suitable for data integration and Web systems. Current research in databases includes how to model, query, restructure, store and manage semi-structured data [2, 60, 1]. Other research themes include extracting some structure from data in formats such as those prevalent in the Web [2, 78, 32, 31, 143, 174], text documents [4] and spreadsheets [132], in order to integrate these data.

4.2 Characterizing Data Heterogeneity

The most widespread way to characterize data heterogeneity is to separate representation from interpretation concerns [201]. *Representational conflicts* refer to syntactic or structural discrepancies in the portrayal of heterogeneous data. *Semantic conflicts* refer to disagreement about the meaning, interpretation or intended use of the same or related data.

The solution of representational conflicts usually requires the analysis of their semantic counterpart, i.e., establishing correspondences (perfect or not) between the meanings of data items from heterogeneous sources. Semantic matches are often achieved only for specific domains.

Both representational and semantic conflicts may occur in any level of abstraction: instance, schema, data model. Thus data heterogeneity conflicts can also be classified according to the following categories [105, 163, 124, 123]:

- *Data conflicts* are discrepancies in the representation or interpretation of instantiated data values, which can differ in their measurement unit, precision and spelling.
- *Schema conflicts* are differences in schemas due to alternatives to depict the same reality, such as using distinct names for the same entities or modeling attributes as entities and vice-versa.
- *Data versus schema conflicts* are disagreements about what is data and metadata; e.g., a data value under one schema can be the label of an entity or attribute in another schema.
- *Data model conflicts* result from the use of different data models.

4.3 Solving Syntactic and Structural Conflicts

The earlier solutions for representational heterogeneity [131, 163, 129] are restricted to the relational data model. They extend SQL to allow the conversion of table and attribute labels into data values and vice-versa. Other works explore languages with logical foundations, aggregation and restructuring capabilities [86, 87].

Proposals for integrating semi-structured and other diverse data sources are surveyed in [194, 78]. Several proposals concern the establishment of a standard syntax and data model. Some of them are centered in object models [105, 193], while others use semi-structured data to represent heterogeneous data at a more abstract level [43, 44, 184, 104]. The use of semi-structured data confers versatility to data representation, enabling data transformations and mappings among irregular structures. On the other hand, as data modeling constructs from typical data models often carry semantics, information can be lost on converting data from such a data model into semi-structured data. The information loss problem can be handled by maintaining proper metadata associated with the transformed semi-structured data.

4.4 Reconciling Semantics

The solution of semantic conflicts relies on the standardization of the meaning of the concepts, terminology, and structuring constructs found in source data [200, 180]. It involves metadata enrichment to support the investigation of semantic matching among data items from distinct data sets.

The first step is to semantically describe data, by associating consensual descriptions to published and exchanged data [121]. At this stage, the establishment of an accord is usually possible only for small communities [92]. Common semantics can be expanded to wider communities, as information is better understood and appropriate levels of abstraction are devised to make possible data exchange with minimal loss of meaning.

4.5 The Data Integration Steps

Data integration can be regarded as a sequence of steps, involving transformations and investigation of correspondences among data elements, in order to produce a unified view of heterogeneous data. Figure 3, adapted from [185], illustrates the information flow along the data integration steps.

Heterogeneous data are first converted to a homogeneous format (e.g. XML), using transformation rules that explain how to transform data from the source data model to the target data model. The translated data and schemas are semantically poor for integration purposes. Thus they must be enriched with semantic information (e.g., measurement units, meaning of the terms appearing in tags and data values). Then, the correspondences between elements from heterogeneous sources are investigated, using the semantic descriptions and similarity rules, to produce a collection of correspondence assertions. Finally, the correspondence assertions and integration rules are used to produce an integration specification, which describes how data elements from heterogeneous sources must be transformed and mixed to produce a unified view.

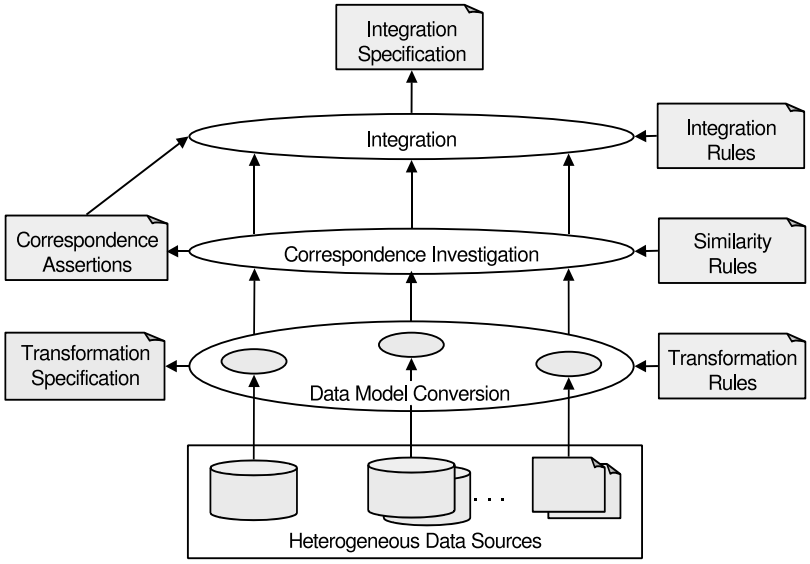

Figure 3: The data integration steps

Even though data integration ultimately requires human intervention, it is crucial to automate or at least assist some laborious tasks, in order to make data integration practicable. The goal of automated facilities is to make data integration easier and repeatable, while allowing users to make decisions along the integration process.

5 Building Blocks to Integrate Data in Cooperative Systems

This section describes some categories of software apparatus that have been proposed to support integrated data views. Such apparatus allow the interconnection of heterogeneous data repositories, programs, materialized and non-materialized views, in such a way that the output of one software module can supply the input to another module.

5.1 Gateways

A *Gateway* is a software component that allows a DBMS and/or an application program directly connected to this DBMS to access data maintained by another DBMS, using the data model and data manipulation language of the former. It is necessary to develop one specific gateway for each DBMS pair. Gateways do not provide transparency for heterogeneous database schema and instances. Hence, gateways do not offer support to establish a unifying view of heterogeneous data. Figure 4 presents a gateway providing access to database "Y" for an application program and its directly connected database "X".

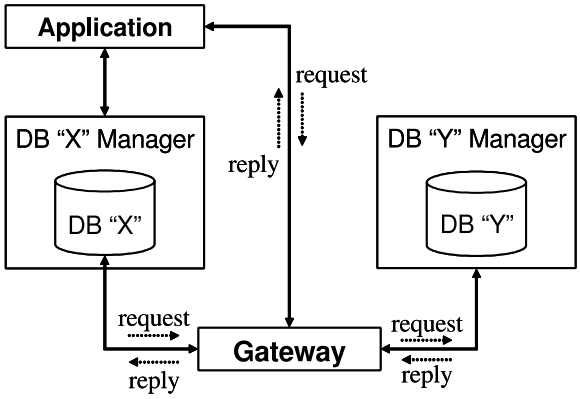

Figure 4: A database gateway

5.2 Wrappers and Mediators

Wrappers and mediators [225, 84] provide data manipulation services over a reconciled view of heterogeneous data. Wrappers encapsulate details of each data source, allowing data access under a homogeneous data representation and manipulation style (common data model and, sometimes, standardized schema). Mediators offer an integrated view of the data sets of several data sources that can include wrappers and other mediators. Some systems adopt multiple levels of mediators in order to modularize the data transformation and integration along successive levels of abstraction.

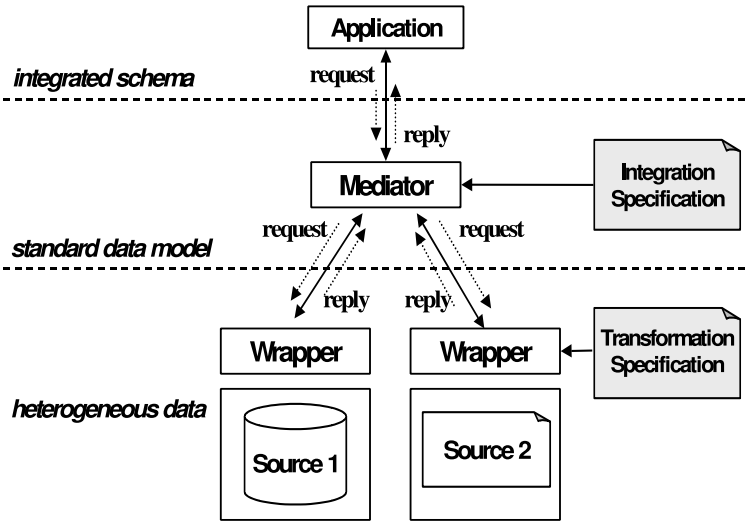

Figure 5: Wrappers and mediators

Figure 5 shows two wrappers and one mediator providing integrated access to two different data sources. The mediator brokers the requests from the applications into requests to the wrappers of the particular sources involved in the request. On receiving the replies

from the source wrappers, the mediator composes the results to return an integrated result to the application. Data transformation and mapping specifications drive the functioning of wrappers and mediators. Wrapper generators and data mapping specification languages [84] enable the specification of data integration in a more intelligible manner than using conventional programming languages to hard code wrappers and mediators.

5.3 Data Warehouses

A *data warehouse* [190, 114, 148, 42, 144] is a separated database built specifically for decision support. It provides the basis for analysis of large amounts of data, collected from a variety of possibly heterogeneous data sources. A data warehouse replicates and integrates data from sources such as relational databases maintained by on-line transaction processing systems (OLTP), spreadsheets and textual data. These sources typically run in the operational level of organizations, while data warehouses are intended for the strategic level.

Data warehousing is the activity of collecting, transforming and integrating data for consolidated analysis. This can be performed off-line with periodical updates, perhaps overnight. The separation between the data warehouse and the data sources prevents the warehouse from interfering in the functioning of the systems at the operational level and confers flexibility for data organization and processing in the warehouse. Data from the sources is first processed before being stored at the warehouse.

There are specific methods for modeling and organizing data in a warehouse – e.g. multidimensional, star, and snowflake style schemas [112] – and also for data processing and user interaction – e.g., on-line analytical processing (OLAP) [85, 94, 42, 106, 58]. Figure 6 shows the loading of data from the sources into a warehouse and their use for data analysis purposes.

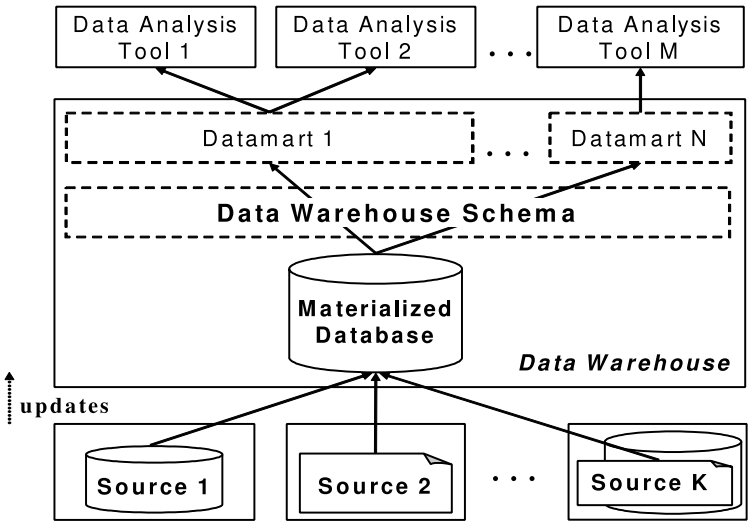

Figure 6: A data warehouse

5.4 The View Approach

Wrappers and mediators support non-materialized (i.e., abstract) integrated views for heterogeneous data, while data warehouses provide materialized views (i.e., concrete sets of copied, transformed and integrated data). In data warehouses, the unidirectional data flow, from the data sources to the warehouse repository simplifies the view update problem [100, 192, 241]. The data warehouse cannot be updated by end users. Updates done to the sources have to be periodically loaded in the warehouse to reflect them in the unified view. Figure 7 illustrates a general view-based data integration system. In this case, updates posed on the exporting views are difficulty to be performed in the lower levels, especially the original data sources. The transformations applied for data analysis purposes (e.g., data aggregation) can lead to complex problems of data lineage and view updating [51, 52].

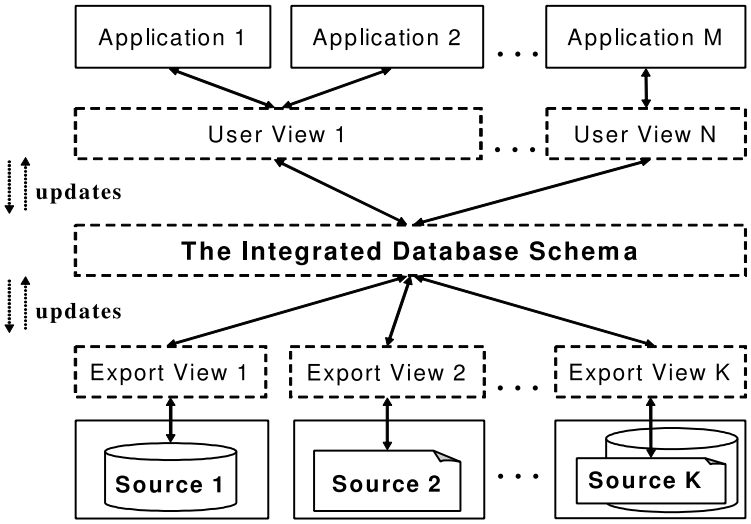

Figure 7: The view approach

Many of the techniques developed for views in heterogeneous database systems can be employed for the construction of wrappers, mediators and data warehouses. Unfortunately, integrating highly heterogeneous data and exporting them to specific data analysis tools are harder problems. They demand data transformation and management facilities beyond those provided by the current DBMSs. Views stored in warehouses also involve historical information that may not remain in the original sources. Nevertheless, several works take the view approach for the integration of heterogeneous data [18, 99, 213] and data warehouses [144].

6 The Semantic Web

The *semantic Web* [197, 72, 57, 240, 62, 22] is an emerging research area whose goal is to achieve information systems interoperability and enable a variety of sophisticated applications, by taking advantage of semantic descriptions of Web resources (data and services).

It is an infrastructure on which different applications can be developed [69]. It intends to enrich the current Web with formalized knowledge and data, that different human beings and/or computers can exchange and process.

The key requirement for the semantic Web is interoperability. Data and metadata must comply to consensual formats and conceptualizations, in order to enable their exchange and proper processing. Therefore, standards for expressing data and metadata are crucial for the semantic Web. Figure 8, adapted from [127], illustrates the semantic Web layers of standards and technologies.

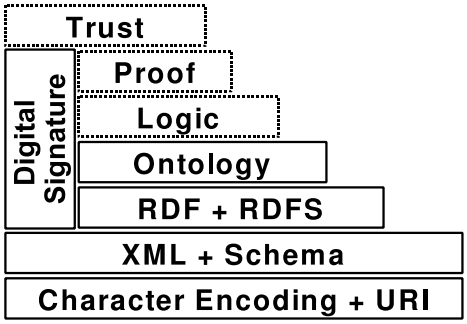

Figure 8: Layers of semantic Web standards and technologies

The lowest layer, *character encoding + URI*, provides an international standard for coding character sets (Unicode) and a means to uniquely identifying resources in the semantic Web (the URI specification [214]). The *XML* [236] layer, which includes namespaces [170] and schema definitions [236, 237], constitutes a standard syntax, with an underlying data model, to express interchangeable data and schemas. In the *RDF + RDFS* layer, *RDF* [133] allows statements associating resources with their properties. *RDFS* (RDF Schema) [27] enable the definition of vocabularies that can be referred to by the URIs in which they are published. These vocabularies can be used to associate types to resources and properties. The *Ontology* layer enriches vocabularies and supports their evolution, by extending the repertory of concepts and semantic relationships among them. Several languages for describing ontologies in the Web have been proposed to fulfill the needs of this layer [72, 88, 96, 150, 181, 55, 176].

The top layers: *Logic*, *Proof* and *Trust* are still under development. The *Logic* layer expresses knowledge by rules, while the *Proof* layer uses these rules to infer other knowledge. The *Trust* layer provides mechanisms to determine the degree of trust on inferred knowledge. *Digital Signature* permeates several layers to ensure security, by using means like encryption and digital signatures.

The remainder of this section describes the XML, RDF and ontology layers of the semantic Web in more detail, analyzing the major standards and technologies and how they interrelate.

6.1 XML

XML (eXtensible Markup Language) [236, 2] is a syntax standard, with a graph-based

data model, to represent and exchange semi-structured data. XML derives from the ISO standard SGML (Standard Generalized Markup Language) [115]. These languages are know as meta-markup languages because they allow the definition of specific markup languages. Like HTML, XML employs tags and attributes of tags to structure data. However, the structure and tags of an XML document are user defined. In XML, tags and structure are intended to describe data meaning, not data presentation as in HTML. Web servers, browsers and certain applications are able to process XML-encoded data.

Figure 9 presents a fragment of a XML document containing climate data, specifically water balance data (measurements of climate data, soil moisture and evaporation of this moisture). These data refer to a particular point in the earth surface, denoted by its geographic coordinates and the name of the city where that point is located. The major data element contained in this XML document, `WaterBal`, expresses the geographic position by means of the XML attributes `location`, `latitude` and `longitude`, attached to its opening tag. This data element includes several climate measures for each month. Each measure is represented by an atomic data element. The value of each measure appears between the element's opening and closing tags. For example, the value of the average temperature in January is enclosed by the tags `<Temperature>` and `</Temperature>`. This atomic data element is nested in the composite element congregating all the measures for January, delimited by the `<Jan>` and `</Jan>` tags. The default namespace associated with this XML document points to the description of its schema (presented in Figure 10), via a http address.

```
<?xml version="1.0"?>
<WaterBal xmlns="http://www.agric.gov.br/WaterBalBrotas.xml"
          location="Brotas" latitude=-22.1500 longitude=-47.5800>
      <Jan>
          <Temperature> 22.0 </Temperature>
          <AvgRainFall> 201.3 </AvgRainFall>
          <PotET> 115.4 </PotET>
          <RealET> 115.4 </RealET>
          <Stored> 125.0 </Stored>
          <WaterDeficit> 0.0 </WaterDeficit>
          <WaterExcess> 86.0 </WaterExcess>
      </Jan>
      :
</WaterBal>
```

Figure 9: An XML document for climate data (water balance)

The emergence of XML poses many challenges to academia and industry [61, 40, 223, 135]. Leading software vendors are moving toward adopting XML, either as an internal data representation model for their software or just for data exchange among different applications and platforms. The publication of data in XML format can make the Web a huge XML data source for all sorts of information.

There are many technologies being developed to explore the potential of XML (e.g., XML

query languages [2, 238, 24]). The use of XML as a data representation standard can bring many benefits for data integration [2, 135]. Furthermore, since XML is a semi-structured data model, it can lend versatility and openness to data representation and integration.

However, XML alone does not solve all the data heterogeneity conflicts. XML data sets from independent sources can present schema and semantic conflicts, even if these sources provide data about the same domain for the same application. The resolution of these conflicts requires consensual semantics to be associated with XML contents and tags. This cannot be done in one step. Interoperability requires multiple agreements on XML data modeling and terminology.

Common Schemas and Metadata Standards

DTD and *XML Schema* are schema languages for XML [134]. Schema specifications can be stored with XML data, or in a separate document, that can be referenced to by several XML documents. *DTD* (Document Type Description) [236] is part of the XML specification itself. It defines the structure of XML documents using a list of element declarations. These declarations, in the style of regular expressions, define the types of atomic XML components and the nested structure of composite elements.

XML Schema [237] offers an XML-based syntax to describe the structure and constraining the contents of XML documents. XML Schema reconstructs and extends DTD capabilities. Figure 10 presents an XML Schema description for the climate data document presented in Figure 10. The first line of this description declares the namespace for the XML Schema vocabulary. The second one states that a document conforming to this schema must have an element called `WaterBal` (the string used in its tags) of the type `WaterBalType`. An element of type `WaterBalType` includes twelve nested elements of the type `AggregValues`, to hold the climate measurements for each month of the year. `WaterBalType` also includes attributes to specify the geographic location to which the climate data refer.

Note that the schema description is not enough to ensure the correct interpretation of the XML data and support data integration. Much semantic information is missing. For example, there is no indication of the measurement units and the geographic coordinate system used in the XML and XML Schema fragments of climate data. In addition, the meaning of the data elements is not clearly specified by their tags. For example, `Temperature` probably refers to the average temperature in the month, while `AvgRainfall` refers to the average accumulated rainfall during the particular month (these averages are derived from temporal series of weather data). The meaning of certain attributes like `PotET` and `RealET` (potential evapotranspiration and real evapotranspiration, respectively) are even harder to infer, and require expert knowledge to be fully understood.

This example illustrates the need to associate consensual semantics with XML data and their markup. The use of standard schemas and metadata standards, with well documented and widely agreed meaning, can decrease this problem. General metadata standards such as Dublin Core [59] define vocabularies and the precise meaning of terms for general use, while metadata standards and standard schemas developed for specific fields help to establish some consensus inside these fields [218]. However, these standards and formats are not enough because: (i) they hinder the autonomy of information systems, (ii) they do not

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="WaterBal" type="WaterBalType"/>
    <xsd:complexType name="WaterBalType">
       </xsd:sequence>
          <xsd:element name="Jan" type="AggregValues"/>
           :
          <xsd:element name="Dec" type="AggregValues"/>
       </xsd:sequence>
       <xsd:attribute name="location" type="xsd:string"/>
       <xsd:attribute name="latitude" type="xsd:Latitude"/>
       <xsd:attribute name="longitude" type="xsd:Longitude"/>
    </xsd:complexType>
    <xsd:complexType name="AggregValues">
       <xsd:sequence>
          <xsd:element name="Temperature" type="decimal"/>
          <xsd:element name="AvgRainfall" type="decimal"/>
          <xsd:element name="PotET" type="decimal"/>
          <xsd:element name="RealET" type="decimal"/>
          <xsd:element name="Stored" type="decimal"/>
          <xsd:element name="WaterDeficit" type="decimal"/>
          <xsd:element name="WaterExcess" type="decimal"/>
       </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

Figure 10: An XML schema for climate data (climate data)

contemplate the evolution of these systems, (iii) they do not cover all types of data, and (iv) they are unsuitable to provide different views of the same data.

6.2 RDF

RDF [133, 72] is the major format for machine-processable metadata in the semantic Web. RDF is based on knowledge representation formalisms such as frames [165] and description logics [15]. The basic construct of the RDF model is the *statement* – a triple of the form *subject-predicate-object*, where *subject* refers to a resource (anything that can be denoted by a URI), *predicate* is a property of that resource, and *object* is the value of that property. The object can be a literal (e.g., a string) or another resource. An RDF statement declares a property of a resource and can also be regarded as a *resource-property-value* triple, where *resource* is used as a synonym for *subject*, *property* for *predicate* and *value* for *object*. Thus, one can stipulate an RDF triple (http://www.Embrapa.br, PART_OF, http://www.Brazil.gov.br) to indicate that the organization whose home page is accessible by the URI http://www.Embrapa.br is part of the Brazilian government.

RDFS (RDF Schema) extends RDF with classes of resources, values, and properties. An RDFS specification defines a structure of classes, properties and subclasses for a particular domain or application, similar to an object-oriented class diagram.

Figure 11, adapted from [120], illustrates the use of RDF and RDFS to describe Web resources. Two different RDF schemas, on the top of the figure, describe resources for gathering weather data (e.g. weather stations). The RDF schema on the left describes these resources from the point of view of scientists who are interested in analyzing weather data. These scientists connect their applications to data collecting devices available on the

Web (e.g. via Web services) to obtain such data. Their applications are concerned with the geographic location of the data collecting devices and how different land parcels (e.g., states, counties) are interrelated. A company responsible for the maintenance of the data collecting devices, on the other hand, has a different view of the same resources. For such a company, each device is an equipment, with category and model. Each equipment is associated with one client.

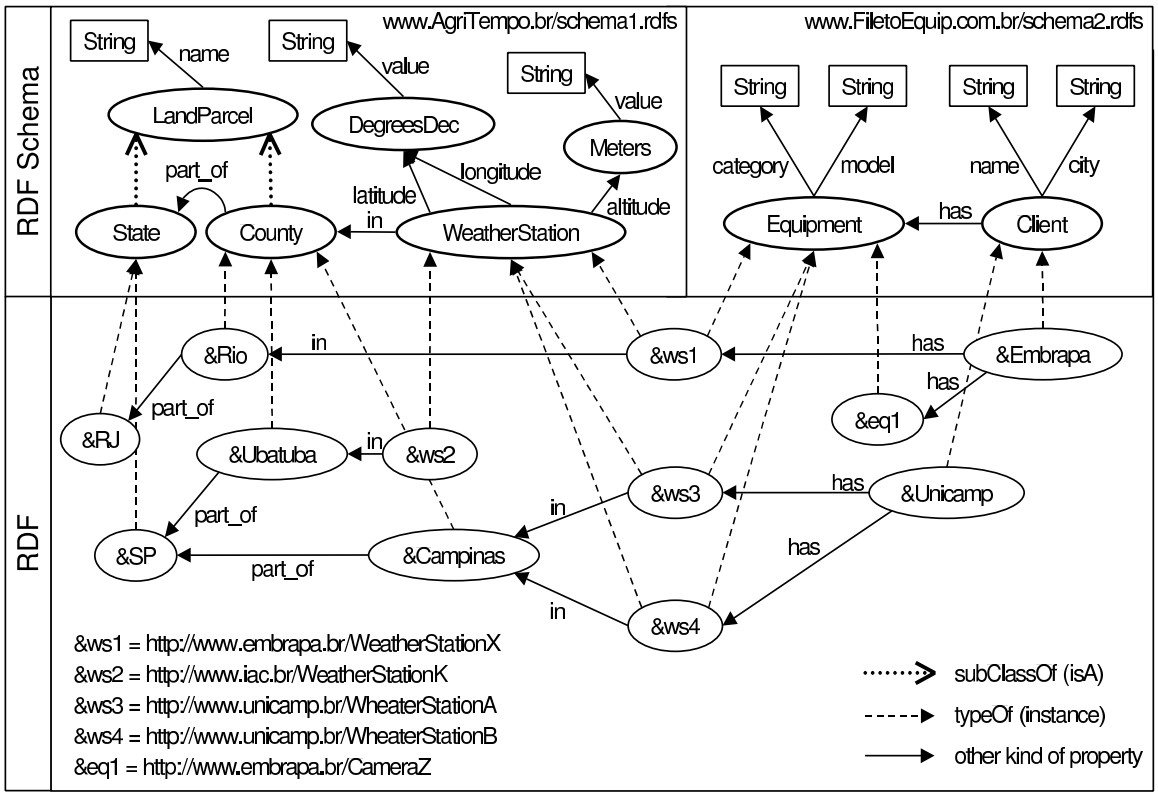

Figure 11: RDF descriptions of resources for collecting scientific data

Each resource in the unified RDF specification on the bottom of Figure 11 is an instance of some class (i.e., another resource describing its type) of one or both RDF schemas on the top. For example, the weather station `&ws1` is an instance of `WeatherStation` in the RDF schema on the left and of `Equipment` in the RDF schema on the right. `&ws1` is a shorthand for the URI `http://www.embrapa.br/WeatherStationX`. Statements involving resource instances must match statements defined at the RDFS level. For example, `&ws1` belongs to `&Embrapa` and is located in `&Rio`, a county of `&RJ` State. The URIs of land parcels and clients are omitted for simplicity.

In addition to their use in providing different views of the same resources, RDF/RDFS also help to define unified views of heterogeneous resources. For example, the weather stations of Figure 11, having different technical characteristics and belonging to different institutions, can be originally described and handled in different ways. Furthermore, their

positions can be defined in distinct systems of geographic coordinates, and the arrangement of land parcels can differ across institutions (e.g., water supply companies divide land in hydrological basins). The data provided by different weather stations can also differ in their structuring and representation (e.g., measurement units). Several layers of RDF/RDFS descriptions provide the solution for these conflicts.

The RDF/RDFS standards play the following fundamental roles in the semantic Web:

- denote relationships involving resources and resource descriptions;
- provide distinct views of the same resources, tailored for different domain or applications;
- build unified views for collections of heterogeneous resources;
- describe knowledge in terms of vocabularies of concepts and the semantic relationships among these concepts.

The XML/RDF Mismatch

RDF/RDFS can be expressed using XML syntax. However, many XML handling facilities are not appropriate for handling RDF. XML and RDF/RDFS are both based on directed graphs, but have different models. The RDF/RDFS model is a directed graph in which labeled nodes represent resources or literals and labeled directed edges represent properties linking resources to the values of their properties. The edges of the RDF graph-based model are unordered and their labels define properties. The XML semi-structured data model, on the other hand, is more hierarchical. The labeled nodes of the XML model represent data elements or attributes, and its directed edges represent nesting and reference relationships between data elements. In the XML model edges are unlabeled and the outgoing edges of a node have a total order. Patel-Schneider and Siméon [187, 186] point out problems resulting from this mismatch between the XML and RDF/RDFS models. They propose a semantic foundation for the Web, based on model theory, to reconcile XML and RDF information sources.

Handling RDF/RDFS

XML query languages, such as XQuery [238, 2], are not suitable for RDF, due to the models' mismatch. Thus, several languages and tools have been developed specifically for querying RDF metadata. Jena [119, 229] is a popular toolkit for handling RDF triples. It allows navigation in RDF triples through an application program interface (API) or the RDQL query language, an implementation of SquishQL [162]. Nevertheless, procedural languages for handling RDF triples and their components are cumbersome. For many applications, a template-based declarative language would be more appropriate. RQL (RDF Query Language) [120] is a declarative language for querying RDF according to its graph model. RQL adapts functionality of query languages for semi-structured and XML data [2], to provide functional constructs, in the style of OQL [36], for uniformly querying RDF/RDFS.

Sesame [28] is a server-based architecture for storing and querying large quantities of metadata in RDF/RDFS, with support for RQL and concurrency control. Most of the current facilities for handling large RDF repositories, including Jena and Sesame, rely on relational or object-oriented database management systems to provide persistence and scalability [119, 229, 68, 147, 120, 28]

6.3 Ontologies

Ontologies [215, 96, 97, 157] are shared conceptualizations of knowledge about delimited domains. An ontology organizes definitions and interrelationships involving a set of concepts (e.g., entities, attributes, processes). It captures the meaning of classes and instances from a universe of discourse, by arranging the symbols (e.g., words, expressions, signs) referring to them, according to semantic relationships [228].

An ontology entails or embodies a particular viewpoint of a given domain. This viewpoint must be *shared* by a group of individuals, formed according to factors like geographic proximity, cultural background, profession, interests or involvement in particular enterprises. These people establish agreements with respect to their views of the world and the symbols used to communicate their views. Ontologies can be explicit or implicit, formal or informal. However, they must be *explicit* and *formal*, to be represented and processed by computers.

There is no convention with respect to the form of a machine-processable ontology. A simple type hierarchy, specifying classes and their subsumption relationships, like a taxonomy, is an ontology. Even a relational schema can serve as an ontology, by specifying the possible relationships and integrity constraints in a database.

Ontologies constitute a means to structure knowledge to support information retrieval and interoperability [96]. The shared knowledge carried in ontologies enable precise stipulation and resolution of queries [98, 198, 108, 13, 7, 169, 121] and information brokering [122, 158] in open environments. Ontologies also help data integration, particularly the investigation of correspondences between elements of heterogeneous data sources [13, 156, 21, 157]. Related research proposes the development of information systems components by translating ontologies into object-oriented hierarchies to implement these systems, giving rise to the concept of Ontology-Driven Information Systems [97, 79].

The following paragraphs describe the currently proposed means to describe, develop and manage ontologies in the semantic Web. Sections 7 and 8 include more specific discussions of the use of ontologies in semantic Web applications.

Ontology Specification Languages

Several languages and formalisms have been proposed to express knowledge in ontologies [88, 96]. DAML+OIL and OWL are some of the most prominent ontology languages for the semantic Web. They extend the RDF/RDFS vocabulary and enrich expressiveness for delineating ontologies (e.g., to express disjunction of classes and other constraints). DAML+OIL [150] combines the basic constructs and syntax of DAML-ONT (DARPA Agent Markup Language) [55, 72] with OIL's (Ontology Inference Layer) [176] frame-based mod-

eling primitives [165] and formal semantics and reasoning services, based on description logics [15].

OWL (Web Ontology Language) [181] is a W3C candidate standard recommendation. It is intended to describe classes and relations that are inherent in Web documents and applications. OWL carries influences of DAML+OIL, among other languages and formalisms. Like OIL, OWL comes in three different flavors, with increasing expressiveness and complexity.

Descriptions of other ontology languages appear in [72, 88, 186]. The relationship and integration of XML with ontology representation languages and formalisms is addressed in [13, 7, 187, 186, 6, 126].

Ontologies Development and Management

The development of ontologies is a laborious and error prone task, especially if it is done by hand. Ontology engineering tools [175, 209, 90] can automate parts of this task and hide the idiosyncrasies of the ontology specification languages and formalisms. These tools can offer graphical interfaces, facilities for knowledge acquisition (e.g., legacy data set conversion and incorporation in the ontology), remote access to knowledge repositories and means to check the quality and consistency of the specifications produced.

Protégé [175, 209, 90, 70] is an example of an open-source graphic tool for ontology editing and knowledge acquisition. It can be extended with plugins to incorporate new functionality. Available plugins allow, for example, the development and exchange of ontology specifications in a variety of formats, including DAML+OIL and OWL.

Methodologies and guidelines for developing ontologies appear in [96, 208, 19]. They help to enhance productivity and to improve the quality of the ontologies developed. Methods and tools for automatically extracting ontologies from text documents and semi-structured data are proposed in [83, 56, 166, 167, 169].

The spreading of ontologies for different domain and applications leads to interoperability problems among diverse ontologies. Proposed solutions for this problem involve ontology composition algebras and graph-based models for ontologies articulation [71, 168, 118, 228, 226, 227].

Finally, Jess [80] and Algernon [109] are examples of inference engines for the semantic Web. These engines handle RDF/RDFS specifications and related formats as rules formalizing declarative knowledge. They apply inference to derive other knowledge from the base knowledge present in ontology specifications. These engines can be plugged to an ontology editor such as Protégé or simply process RDF/RDFS exported by such a tool.

7 Web Services

A *Web service* [73, 204, 35, 233] is a software module accessible through the Internet. Web services are usually self-describing and independent. They communicate with clients and other services via messages, over standard Web protocols. Each Web service can be identified by a URI and exposes a XML interface to allow its discovery and invocation across the Web.

The Web services technology is based on the notion of building new applications by combining network-available services. The services participating in distributed processes cooperate to achieve some goal, by exchanging messages and coordinating their executions. It enables interoperability of information systems, while allowing decoupling and just-in-time applications integration. The resulting cooperative systems are potentially self-configuring, adaptive and robust, because they can allow the dynamic incorporation of alternative services and avoid single points of failure. Furthermore, implementing systems components as Web services reduces complexity, as application designers do not have to worry about platform and implementation details, which are encapsulated by the Web services interfaces.

7.1 Architecture and Basic Standards

A service oriented architecture postulates cooperation of software components with three distinct roles: service providers, service requesters and service brokers. A *service provider* holds the implementation of one or more services and manages the public interfaces that make these services available on the Web. A *service requester* is the party that has a need to be fulfilled by some published service. It can be a human user accessing services through a console or Web browser, an application program or another Web service. The *service broker* provides a searchable repository of service descriptions, where service providers publish their services and service requesters find descriptions and binding information to access services contemplating their particular needs.

Service providers, requesters and brokers communicate using standard technologies. There are many standards currently under development to allow language and platform independent implementation of Web services [128, 211]. Figure 12 outlines the layers of standards and technologies supporting Web services-based applications.

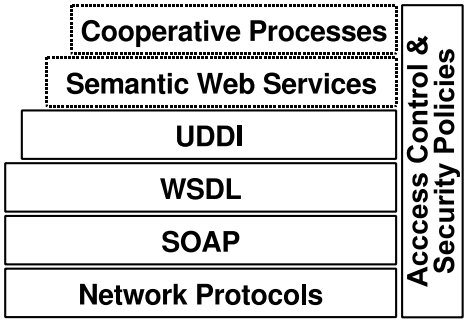

Figure 12: Layers of Web services standards and technologies

The *Network Protocols* layer provides the basic communication facilities and protocols (e.g., HTTP). *SOAP* [26] is a lightweight protocol for services to exchange XML-encoded messages and make procedure calls over the Internet. Messages can be routed along a message path. SOAP provides enveloping facilities to describe the intent of a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. Though SOAP was originally designed to use HTTP as the transport protocol, it can run on other

network protocols such as FTP, SMTP or even raw TCP/IP sockets. SOAP is extensible, allowing different communication models such as one-way, request-response and multicast. In addition, SOAP is not tied to any language or component technology.

WSDL (Web Services Definition Language) [234] is a XML-based format for describing Web services. WSDL specifies what a Web service does, where it is located and how it is invoked. In WSDL, a service is regarded as a set of related endpoints called ports. The ports of a service can communicate with ports of other services via messages, that can contain either document-oriented or procedure-oriented information. The abstract definitions of ports and messages are separated from their network deployment and data format bindings. This allows the reuse of abstract definitions: port types that define sets of operations supported by ports, and data types that define the data being exchanged. A concrete data format and protocol specification for a port type constitutes a reusable binding. WSDL can work in conjunction with SOAP, HTTP GET/POST or MIME.

UDDI (Universal Description, Discovery and Integration) [212] is a set of standard XML schemas, SOAP messages and API specifications to build catalogs for finding specific Web services. UDDI provides information about business (e.g., name, description, contact), services offered and particular standards used to bind with these services. It also provides identifiers and various taxonomies to describe business (e.g., related industry, products and services, geographical region). A UDDI registry is itself a Web service, providing facilities to create, modify, delete and query service descriptions. These registries can be public or private. IBM and Microsoft provide public UDDI registries. Service providers only have to register to one of these public registries, since updates to any of them are replicated in the others on a daily basis.

The two top layers of Figure 12 refer to the semantic and functional aspects of Web services integration. These layers are still under development with many proposals from industry and academia. The *semantic Web services* layer employs semantic Web technologies, such as ontologies, to support Web services discovery, selection and composition, according to the needs of specific domains or applications. The *Cooperative Processes* layer concerns the coordinated execution of Web services in cooperative processes across organizational boundaries. Finally, *Access Control and Security Policies* can be enforced in any Web services implementation layer.

7.2 Cooperative Distributed Processes enabled by Web Services

Semantic Web Services

Semantic Web services [151] are associated with well-defined semantics to express their functional properties, capabilities, applicability and ontological relationships, in order to enable their utilization in cooperative processes over an open and distributed environment. Research in this area rely on semantic Web ideas and technologies [111, 240, 95, 202, 149, 195, 203, 14, 183, 34, 151, 33].

The capabilities of registries such as UDDI and languages like WSDL are not enough to support services discovery [183]. DAML-S (or DAML-services) [14] is an extension of the DAML ontology specification language for Web services. It includes mechanisms to de-

scribing, discover, select, activate, compose, and monitor Web resources. The work of [183] employs DAML-S for services discovery, presenting an algorithm to match service requests with the profile of advertised services, based on the minimum distance between concepts in a taxonomy tree. Cardoso and Sheth [34] present metrics to select Web services for composing processes. These metrics take into account functional and operational features such as the purpose of the services, quality of service (QoS) attributes, and the resolution of structural and semantic conflicts. McIlraith *et al.* [151] use agent programming to define generic procedures involving the interoperation of Web services. These procedures, expressed in terms of concepts defined with DAML-S, do not specify concrete services to perform the tasks or the exact way to use available services. Such procedures are instantiated by applying deduction in the context of a knowledge base, which includes properties of the agent, its user, and the Web services.

Topology and models have been proposed to enable cooperation and composition of services. Schlosser *et al.* [195] propose a graph topology, determined by a globally know ontology, to speed up communication of Web services in a peer-to-peer system. Maximilien and Singh [149] present a model for gathering and assessing information relative to the use of Web services to determine their trustfulness. Sirin *et al.* [202] presents a prototype to guide a user in the dynamic composition of Web services. Finally, Grüninguer [95] show how an ontology for process specification languages can serve as a semantic foundation for the composition of Web services.

Web Services Coordination

Nowadays, there is a myriad of proposals concerning the interoperability and synchronization of Web services [216, 103, 20, 76, 189, 231, 160]. Examples of Web services composition languages include BPEL4WS (BEA, IBM, Microsoft) [231], WSFL (IBM) [235], BPML (BPMI), XLANG (Microsoft), WSCI (BEL, Intalio, SAP, Sun), XPDL (WfMC), EDOC (OMG) and UML 2.0 (OMG). Some challenges of these technologies are: (i) reducing the amount of low-level programming necessary for the interconnection of Web services (e.g., through declarative languages), (ii) providing flexibility to establish interactions among growing numbers of continuously changing Web services during run time, and (iii) devising mechanisms for the decentralized and scalable transaction control for cooperative processes running on the Web. Much of the current technology for syncronizing processes are based on centralized control, even if the the execution is distributed. This centralization is inappropriate for Web systems, for reasons of autonomy and scalability. Thus, in oppositon to techniques to orchestrate services, Web-based workflows require technology to allow service to choreography their executions, based on agreed upon protocols.

Van der Aalst [216] compares the major candidate standards for Web services composition and synchronization. He points out problems related with the lack of formal semantics, expressiveness, complexity and adequacy of these proposals. [216] suggests the incorporation of well-established process modeling techniques in a single standard for Web services composition. The use of Petri-nets for this purpose is considered in [103, 217, 171]. Activity models appear in [82, 142, 141, 140, 139].

8 Applications and Supporting Environments

Semantic Web applications take advantage of knowledge, represented in proposed standards like RDF, to leverage automated means to describe, organize, discover, select and compose Web resources for the solution of a variety of problems. The most usual approach is to define semantic markup based on some ontology, and use them to integrate and provide unified access to data and services, typically via Web portals. There are many examples of this approach in the literature [108, 198, 98, 13].

Some experimental systems possess distinctive features. Edutella [173] is a Peer-to-Peer infrastructure using RDF metadata to facilitate access to educational resources. In Edutella, each peer holds a set of resources and has an RDF repository of resource descriptions, to allow querying its contents at the storage layer (e.g., SQL) or user layer (e.g., RQL). Peers can be heterogeneous in their internal organization and the query language they provide. The common data model and the exchange language of Edutella enables a standard interface for posing queries to specific peers or communities and find resources across the network.

Piazza [102] is an infrastructure to provide interoperability of data sources in the Web, by mapping their contents at the domain level (RDF) and the document structure level (XML), and addressing the interoperation between these levels. The mappings are specified declaratively for small sets of nodes. A query answering algorithm chains these mappings together to obtain relevant data from across the network.

Papers focusing specifically scientific applications of the semantic Web and Web services include [206, 145, 159, 89, 39]. Some scientific applications refer to particular fields such as bioinformatics [30, 138, 37, 205, 101, 91], earth sciences [17, 222] and the environment [16, 38, 146]. The *grid* – a platform for coordinated resource sharing through the Internet, increasingly used for scientific data processing – and the semantic Web have mutual characteristics and goals [89]. Both operate in a global, distributed and dynamic environment, and both need computationally accessible and sharable metadata to support automated information discovery, integration and aggregation.

POESIA [74] introduces the concept of ontological coverages – tuples of terms taken from a multidimensional ontology – which are used to describe the utilization scope of data and processing resources, particularly in agricultural sciences. The partial ordering among these descriptors enable the organization, discovery, and reuse of resources. POESIA also includes mechanisms, based on ontologies, workflows and activity models, to semantically orient the composition of Web services in cooperative distributed processes [74] and help to trace the information flow across these processes [75].

Web services development and execution platforms are described in [77, 49, 125, 230, 161]. Bandholtz [16] propose the use of Web services to share ontologies and describes the implementation of a service network for this purpose.

8.1 Scientific Workflows

Scientific work is typically based in experiments [39]. Sometimes scientists rely on simplified models of real world phenomena to found their investigation, and use vast amounts of data to corroborate their results. The technological development has generated a great availabil-

ity of data, from a variety of heterogeneous sources, that scientists can use to enhance their experiments. Moreover, scientists can exchange models and computer programs implementing these models. Although scientific work can vary among diverse people, disciplines and organizations, it can benefit a lot from data and systems interoperability.

Scientific Workflows [37, 12, 153, 220, 8] use workflow technology [117, 110, 53] to manage scientific work. They regard scientific experiments as complex processes with intricate data transformations and information flow. These processes may encompass automatic and manual activities. The data and execution dependencies among these activities can be very complex, yielding interoperability and synchronization problems. Many scientific processes are distributed, in order to enable cooperation of different groups and foster reuse of partial results. Therefore, semantic Web service technologies are fundamental to implement these processes in an open environment encompassing different platforms.

Scientific processes differ from business processes in several aspects. Scientific work demands freedom to try alternative ways of doing things. The sequence of steps (and even the goal, sometimes) is not totally known in advance. The scientist perform some task and decides on the further steps only after evaluating the previous ones. Specific subjects in scientific processes management include documentation [219] and reorganization [141] of these processes.

The exploitation of the workflows paradigm for managing scientific processes has been exploited in specific domains such as bioinformatics [30, 37, 205, 155] and geoinformatics [196, 222, 154, 10, 116, 239, 11]. For instance, Cavalcanti *et al.* [37] combines metadata support with Web services in a framework to support scientific workflows and apply this framework to structural genomics. Seffino *et al.* [196], on the other hand, use scientific workflows to describe and reuse patterns of geographic data processing in agricultural and environmental applications.

8.2 Geographic Information Systems Interoperability

Geographic information systems (GIS) [3, 152, 45] manage data referring to geographic entities or phenomena. These data are geo-referenced, i.e., they carry some indication of the geographic location. A GIS provides specialized basic facilities to process geographic data, being useful for information extraction, planning and decision support, among other kinds of applications.

The GIS market is characterized by proprietary formats that make interoperability hard to achieve. Many formats have been proposed for exchanging geographic data [177, 199, 9]. However, scientists have progressively found out that standard formats are not enough to strengthen GIS interoperability [92]. The conversion of data through these formats often results in information loss, incorrect interpretation of data and poor information quality [47]. It happens because formats for geographic data exchange are mainly concerned with syntax, structure and the geometry of geographic objects. Even GML (Geography Markup Language) [177] do not ensure the correct interpretation of data, because it does not take into account the semantics and the behavior of geographic objects.

The importance of establishing a semantic basis for geographic data representation and management has been recognized in several papers [64, 188, 50, 79, 146, 232]. Córcoles

et al. [50] describes an approach for integrating geographic data, based on mappings between ontologies and XML schemas. They present an ontology to support the creation and exchange of semantic descriptors for geographic resources (XML documents containing geographic data). The descriptors and the links among them and the resources themselves are both expressed in RDF. It enables a unique language for querying GML documents, without knowledge of their structure.

Ontologies for the integration of geographic data appear in [79, 146, 232]. Fonseca *et al.* [79] employs ontologies to define classes for developing geographic applications. Their applications rely on ontology servers and mediators to access their data sources. It allows, for example, loading data instances from heterogeneous data sources, using a schema defined by one ontology.

GIS interoperability also requires additional levels of integration such as commonality of systems behavior and system-user interaction. The adoption of a common geographic data model [210, 25] or at least a framework to unify heterogeneous models [46] constitutes one ingredient to achieve this goal.

9 Conclusions

Integration of heterogeneous data has been one of the greatest challenges in database research. The advent of the Web is pushing the demand for solutions, and reformulating this problem into a more complex setting – the discovery, selection and composition of data and services. Solutions for all these problems involve versatile standards and enriching the Web with semantics, in order to allow interoperability while embracing diversity.

The Web is becoming the common platform for implementing cooperative distributed systems. The semantic Web and workflows based on the collaboration of services across the Web, are expected to expand the role of computers to support human activities in a variety of fields. In this open distributed environment, data processing and semantics cannot be dissociated, because the meaning of data depends on the whole process employed to produce them. Technology to support the idealized systems is under fast development, in areas ranging from knowledge management to Web services development and composition. Concrete applications must be developed in the near future to fulfill end users' expectations.

This survey has outlined the research on information systems interoperability, from work on interconnection of relational databases, to the most recent developments in semantic Web services. The major contributions are: (1) describing and comparing proposed standards and architectures; (2) categorizing heterogeneity and proposed solutions; (3) discussing specific needs related with data and services integration, particularly for scientific applications.

Acknowledgments

This work was partially supported by Embrapa, CAPES, CNPq and the MCT/PRONEX-SAI and the CNPq WebMaps projects. Thanks to professors Ana Carolina Salgado, Caetano

Traina Júnior, Célio Cardoso Guimarães and Edmundo Madeira, who provided several suggestions for the improvement of this work.

References

[1] S. Abiteboul. Querying semi-structured data. In *Proc. ICDT Conf.*, volume 1186 of *LNCS*, pages 1–18. Springer-Verlag, 1997.

[2] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web – from relations to semistructured data and XML*. Morgan Kaufann, San Francisco, CA, 2000.

[3] T. Abraham and J. F. Roddick. Survey of spatio-temporal databases. *GeoInformatica*, 3(1):61–995, 1999.

[4] B. Adelberg. NoDoSE - a tool for semi-automatically extracting semi-structured data from text documents. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 283–294. ACM Press, 1998.

[5] C. D. Aguiar. Heterogeneous database integration into urban planning applications. Master's thesis, Department of Computer Science, State University of Campinas, Brazil, 1995. (in portuguese).

[6] L. Ahmedi, P. J. Marrón, and G. Lausen. Ontology-based access to heterogeneous XML data, 2001.

[7] L. Ahmedi, P. J. Marrón, and G. Lausen. Ontology-based querying of linked XML documents, 2002.

[8] A. Ailamaki, Y. E. Ioannidis, and M. Livny. Scientific workflow management by database management. In *Proc. Conf. on Statistical and Scientific Database Management*, pages 190–199. IEEE Computer Society, 1998.

[9] J. Albrecht. Geospatial information standards a comparative study of approaches in the standardisation of geospatial information. *Computers & Geosciences*, 25:9–24, 1999.

[10] G. Alonso and C. Hagen. Geo-opera: Workflow concepts for spatial processes. In *Advances in Spatial Databases - 5th Intl. Symp. on Large Spatial Databases (SSD)*, volume 1262 of *LNCS*, pages 238–258. Springer-Verlag, 1997.

[11] Gustavo Alonso and Amr El Abbadi. Cooperative modeling in applied geographic research. In *CoopIS*, pages 227–234, 1994.

[12] I. Altintas, S. Bhagwanani, D. Buttler, S. Chandra, Z. Cheng, M. Coleman, T. Critchlow, A. Gupta, Wei Han, L. Liu, B. Ludäscher, Calton Pu, R. Moore, A. Shoshani, and M. A. Vouk. A modeling and execution environment for distributed scientific workflows. In *Proc. Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, pages 247–250. IEEE Computer Society, 2003.

[13] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Ontology-based integration of xml web resources. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 117–131. Springer-Verlag, 2002.

[14] A. Ankolekar, M. H. Burstein, J. R. Hobbs, O. Lassila, D. Martin, D. V. McDermott, S. A. McIlraith, S. Narayanan, M. Paolucci, T. R. Payne, and K. P. Sycara. DAML-S: Web service description for the semantic web. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 348–363. Springer-Verlag, 2002.

[15] F. Baader, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[16] T. Bandholtz. Sharing ontology by web services: Implementation of a semantic network service (SNS) in the context of the german environmental information network (gein). In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 189–201, 2003.

[17] T. Barclay, D. R. Slutz, and J. Gray. TerraServer: A spatial data warehouse. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 307–318. ACM Press, 2000.

[18] C. Batini, M. L., and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, 1986.

[19] C. Behrens and V. Kashyap. The "emergent" semantic web: An approach for derivation of semantic agreements on the web. In *Proc. Semantic Web Working Symposium (SWWS)*, 2001.

[20] B. Benatallah, Q. Z. Sheng, and M. Dumas. The self-serv environment for web services composition. *IEEE Internet Computing*, 7(1):40–48, 2003.

[21] S Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[22] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[23] E. Bertino. A view mechanism for object-oriented databases. In *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 580 of *LNCS*, pages 136–151. Springer-Verlag, 1992.

[24] Angela Bonifati and Stefano Ceri. Comparative analysis of five XML query languages. *SIGMOD Record*, 29(1):68–79, 2000.

[25] K. A. V. Borges. Geographical data modeling: Extension of OMT for spatial applications. Master's thesis, Department of Computer Science, Federal University of Minas Gerais, Brazil, 1997. (in portuguese).

[26] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer. W3C's Simple Object Access Protocol (SOAP). `http://www.w3.org/TR/SOAP/` (as of October 2003).

[27] D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema, 2003. `http://www.w3.org/TR/rdf-schema/` (as of October 2003).

[28] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF schema. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 54–68. Springer-Verlag, 2002.

[29] P. Buneman. Semistructured data. In *16th ACM Symposium on Principles of Database Systems (PODS'97)*, pages 117–121, 1997.

[30] D. Buttler, M. Coleman, T. Critchlow, R. Fileto, Wei Han, C. Pu, D. Rocco, and Li Xiong. Querying multiple bioinformatics information sources: Can semantic web research help? *SIGMOD Record*, 31(4):59–64, 2002.

[31] D. Buttler, L. Liu, and C. Pu. A fully automated object extract system for the web. In *Proc. Intl. Conf. on distributed Computing Systems (ICDCS)*. IEEE Press, 2001.

[32] D. Buttler, L. Liu, C. Pu, H. Paques, W. Han, and W. Tang. OminiSearch: A method for searching dynamic content on the web. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, page 604. ACM Press, 2001.

[33] A. Maedche C. Bussler, D. Fensel. A conceptual architecture for semantic web enabled web services. *SIGMOD Record*, 31(4):24–29, 2003.

[34] J. Cardoso and A. Sheth. Semantic e-workflow composition. Report, LSDIS Lab, Computer Science Dep., Univ. of Georgia, 2002.

[35] F. Casati and U. Dayal (editors). Special issue on web services. *IEEE Data Engineering Bulletin*, 25(4), 2002.

[36] R. G. G. Cattell, D. Barry, M. Berler, D. Jordan, C. Russel, O. Schadow, T. Stanienda, and F. Velez. *The Object Data Standard - ODMG 3.0*. Morgan Kaufmann, 2000.

[37] M. C. Cavalcanti, F. A. Baião, S. C. Rössle, P. M. Bisch, R. Targino, P. F. Pires, M. L. Campos, and M. Mattoso. Structural genomic workflows supported by web services. In *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, pages 45–49. IEEE Computer Society, 2003.

[38] M. C. Cavalcanti, M. Mattoso, M. L. Campos, F. Llirbat, and E. Simon. Sharing scientific models in environmental applications. In *Proc. ACM Symposium on Applied computing (SAC)*, pages 453–457. ACM Press, 2002.

[39] M. C. Cavalcanti, M. Mattoso, M. L. Campos, E. Simon, and F. Llirbat. An architecture for managing distributed scientific resources. In *Proc. Intl. Conf. on Scientific*

and Statistical Database Management (SSDBM), pages 47–55. IEEE Computer Society, 2002.

[40] S. Ceri, P. Fraternali, and S. Paraboschi. XML: Current developments and future challenges for the database community. In *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1777 of *LNCS*, pages 3–17. Springer-Verlag, 2000.

[41] D. Chang and D. Harkey. *Client/Server Data Access with Java and XML*. John Wiley & Sons, 1998.

[42] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1):65–74, 1997.

[43] V. Christophides, S. Cluet, and J. Siméon. On wrapping query languages and efficient XML integration. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 141–152. ACM Press, 2000.

[44] S. Cluet, C. Delobel, J. Siméon, and K. Smaga. Your mediators need data conversion! In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 177–188. ACM Press, 1998.

[45] G. Câmara, M. A. Casanova, A. Hemerly, G. C. Magalhães, and C. B. Medeiros. *Anatomy of Geographical Information systems*. State University of Campinas, Brazil, 1996. (in portuguese).

[46] G. Câmara, A. M. V. Monteiro, J. A. Paiva, J. Gomes, and L. Velho. Towards a unified framework for geographical data models. *Journal of the Brazilian Computing Society*, 7(1), 2000.

[47] G. Câmara, R. Thomé, U. Freitas, and A. Monteiro. Interoperability in practice: Problems in semantic conversion from current technology to open GIS. In *Intl. Conf. on Interoperable GIS*, 1999.

[48] OMG's Common Object Ranguageesquest Broker Arquitecture (CORBA). `http://www.omg.org/CORBA` (as of October 2003).

[49] O. Corcho, A. Gómez-Pérez, M. Fernández-López, and M. Lama. ODE-SWS: A semantic web service development environment. In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 203–216, 2003. `http://www.cs.uic.edu/~ifc/SWDB/` (as of November 2003).

[50] J. E. Córcoles, P. GonzÃ¡lez, and V. L. Jaquero. Integration of spatial XML documents with RDF. In *Proc. Ibero American Conference on Web Engineering (ICWE)*, volume 2722 of *LNCS*, pages 407–410. Springer-Verlag, 2003.

[51] Y. Cui and J. Widom. Practical lineage tracing in data warehouses. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 367–378. IEEE, 2000.

[52] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In *Proc. VLDB Conf.*, pages 471–480. Morgan Kaufmann, 2001.

[53] B. Curtis, M. Kellner, and J. Over. Process Modeling. *Communications of the ACM*, 35(9):75–90, 1992.

[54] A. S. da Silva, P. Calado, R. Vieira, A. H. F. Laender, and B. A. Ribeiro-Neto. *Keywork-Based Queries over Web Databases*, pages 74–92. IRM Press, 2003.

[55] The DARPA Agent Markup Language (DAML). `http://www.daml.org/` (as of August 2003).

[56] H. Davulcu, S. Vadrevu, and S. Nagarajan. Ontominer: Bootstrapping and populating ontologies from domain specific web sites. In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 259–276, 2003. `http://www.cs.uic.edu/~ifc/SWDB/` (as of November 2003).

[57] Y. Ding, D. Fensel, M. Klein, and B. Omelayenko. The semantic web: yet another hip? *Data & Knowledge Engineering*, 41(2/3):205–227, June 2002.

[58] B. Dinter, C. Sapia, G. Hofling, and M. Blaschka. The OLAP market: state of the art and research issues. In *ACM 1st Intl. Workshop on Data Warehousing and OLAP (DOLAP'98)*, pages 22–27, 1998.

[59] Dublin Core Metadata Initiative. `http://www.dublincore.org/` (as of October 2003).

[60] D. Suciu (ed.). Special issue on management of semistructured data. *SIGMOD Record*, 26(4), 1997.

[61] A. Y. Halevy (editor). Special issue on XML data management. *IEEE Data Engineering Bulletin*, 24(2), 2002.

[62] G. Weikum (editor). Special issue on organizing and discovering the semantic web. *IEEE Data Engineering Bulletin*, 25(1), 2002.

[63] R. Miller (editor). Special issue on integration management. *IEEE Data Engineering Bulletin*, 25(3), 2002.

[64] M.J. Egenhofer. Toward the semantic geospatial web. In *Proc. ACM GIS*, 2002.

[65] G. Ehmayer, G. Kappel, and S. Reich. Connecting databases to the web: A taxonomy of gateways. In *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, pages 1–15. IEEE Computer Society, 1997.

[66] A. K. Elmagarmid and C. Pu. Introduction: Special issue on heterogeneous databases. *ACM Computing Surveys*, 22(3):175–178, 1990.

[67] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, Menlo Park, CA, 1994.

[68] F. Esposito, L. Iannone, I. Palmisano, and G. Semeraro. RDF core: A component for effective management of RDF models. In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 169–187. VLDB endowment, 2003. `http://www.cs.uic.edu/~ifc/SWDB/` (as of November 2003).

[69] J. Euzenat. Research challenges and perspectives of the semantic web. *IEEE Intelligent Systems*, 17(5):86–88, 2002.

[70] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: a tool for collaborative ontology construction. *Intl. Journal of Human Computer Studies*, 46(6):707–727, 1997.

[71] D. Fensel. Ontology-based Knowledge Management. *IEEE Computer*, 35(11):56–59, 2002.

[72] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster (editors). *Spinning the Semantic Web*. MIT Press, 2003.

[73] C. Ferris and J. Farrel. What are web services? *Communications of the ACM*, 46(6):31, 2003.

[74] R. Fileto, L. Liu, C. Pu, E. D. Assad, and C. B. Medeiros. POESIA: An ontological workflow approach for composing web services in agriculture. *The VLDB Journal*, 12(4):352–367, 2003.

[75] R. Fileto, L. Liu, C. Pu, E. D. Assad, and C. B. Medeiros. Using domain ontologies to help track data provenance. In *Proc. Brazilian Symposium on Databases*, pages 84–98, 2003.

[76] D. Florescu, A. Grünhagen, and D. Kossmann. XL: An XML programming language for web wervice specification and composition. In *Proc. WWW Conf.*, pages 65–76. ACM Press, 2002.

[77] D. Florescu, A. Grünhagen, and D. Kossmann. XL: A platform for web services. In *Proc. Conf. on Innovative Data Systems Research (CIDR)*, 2003.

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

[79] F. T. Fonseca, M. Egenhofer, P. Agouris, and G. Câmara. Using ontologies for integrated geographic information systems. *Trans. in GIS*, 6(3):13–19, 2002.

[80] E. Friedman-Hill. JESS – the rule engine for the java platform. `http://herzberg.ca.sandia.gov/jess` (as of August 2003).

[81] A. L. Furtado, K. C. Sevcik, and C. S. dos Santos. Permitting updates through views of data bases. *Information Systems*, 4(4):269–283, 1979.

[82] A. Gal. Semantic interoperability in information services: Experiencing with CoopWARE. *SIGMOD Record*, 28(1):68–75, 1999.

[83] A. Gal, G. Modica, and H. Jamil. OntoBuilder: Fully automatic extraction and consolidation of ontologies from web sources. In *Intl. Conf. on Conceptual Modeling*, 2003.

[84] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.

[85] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database System Implementation*. Prentice Hall, Upper Saddle River, NJ, 2000.

[86] F. Gingras and L. V. S. Lakshmanan. nd-sql: A multi-dimensional language for interoperability and olap. In *Proc. VLDB Conf.*, pages 134–145. Morgan Kaufmann, 1998.

[87] F. Gingras, L. V. S. Lakshmanan, I. N. Subramanian, D. Papoulis, and N. Shiri. Languages for multi-database interoperability. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 536–538. ACM Press, 1997.

[88] A. Gómez-Pérez and O. Corcho. Ontology specification languages for the semantic web. *IEEE Intelligent Systems*, 17(1):54–60, 2002.

[89] C. Goble and D. De Roure. The grid: An application of the semantic web. *SIGMOD Record*, 31(4):65–70, 2002.

[90] C. A. Goble, D. L. McGuinness, R. Möller, and P. F. Patel-Schneider. OilEd a reasonable ontology editor for the semantic web. In *Intl. Description Logics Workshop*, volume 49 of *CEUR Workshop Proceedings*, 2001.

[91] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal*, 40(2):532–551, 2001.

[92] M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. Kottman. *Interoperating Geographical Information Systems*. Kluwer, 1997.

[93] L. Gravano, P. G. Ipeirotis, and M. Sahami. QProber: A system for automatic classification of hidden-web databases. *ACM Transactions on Information Systems (TOIS)*, 21(1):1–41, 2003.

[94] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 152–159. IEEE, 1996.

[95] M. Grüninguer. Applications of PSL to semantic web services. In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 217–230, 2003. `http://www.cs.uic.edu/~ifc/SWDB/` (as of November 2003).

[96] M. Gruninger and J. Lee (eds.). Special issue on ontologies applications and design. *Communications of the ACM*, 45(2):39–65, 2002.

[97] N. Guarino. Formal ontology and information systems. In *Proc. Intl. Conf. on Formal Ontologies in Information Systems (FOIS)*, pages 3–15. IOS Press, 1998.

[98] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proc. WWW Conf.*, pages 700–709. ACM Press, 2003.

[99] A. Gupta, H. V. Jagadish, and I. S. Mumick. Data integration using self-maintainable views. In *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1057 of *LNCS*, pages 140–144. Springer-Verlag, 1996.

[100] A. Gupta and I. S. Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Engineering Bulletin*, 18(2):3–18, 1995.

[101] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, and W. C. Swope. Discoverylink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, 40(2):489–511, 2001.

[102] A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Piazza: Data management infrastructure for semantic web applications. In *Proc. WWW Conf.*, pages 556–567. ACM Press, 2003.

[103] R. Hamadi and B. Benatallah. A petri net-based model for web service composition. In *Proc. Australasian Database Conf. (ADC)*, pages 191–200. Australasian Computer Society, 2003.

[104] J. Hammer, H. Garcia-Molina, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom. Information translation, mediation, and mosaic-based browsing in the TSIMMIS system. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, page 483. ACM Press, 1995.

[105] T. Härder, G. Sauter, and J. Thomas. The intrinsic problems of structural heterogeneity and an approach to their solution. *The VLDB Journal*, 8(1):25–43, 1999.

[106] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 205–216. ACM Press, 1996.

[107] W. Hasselbring. Information system integration. *Communications of the ACM*, 43(6):33–38, 2000.

[108] S. Haustein and J. Pleumann. Is participation in the semantic web too difficulty? In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 448–453. Springer-Verlag, 2002.

[109] M. Hewett. Algernon in java. `http://smi.stanford.edu/people/hewett/research/ai/algernon/` (as of August 2003).

[110] D. Hollingsworth. *The Workflow Reference Model*. Workflow Management Coalition, January 1995.

[111] I. Horrocks and J. Hendler, editors. *Intl. Semantic Web Conf.(ISWC)*, volume 2342 of *LNCS*, Sardinia, Italy, June 2002. Springer-Verlag.

[112] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual data warehouse modeling. In *2nd Intl. Workshop on Design and Management of Data Warehouses*, 2000. `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/` (as of November 2000).

[113] D. K. Hsiao and M. N. Kamel. Heterogeneous databases: Proliferation, issues, and solutions. *IEEE Tran. on Knowledge and Data Engineering*, 1(1):45–62, 1989.

[114] W. H. Inmon. *Building the Data Warehouse*. John Wiley and Sons, New York, 1996.

[115] International Organization for Standardization. Standard generalized markup language (SGML), 1986. ISO 8879.

[116] Y. E. Ioannidis, M. Livny, A. Ailamaki, A. Narayanan, and A. Therber. ZOO: A desktop emperiment management environment. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 580–583. ACM Press, 1997.

[117] S. Jablonski and C. Bussler. *Workflow Management. Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press, 1996.

[118] J. Jannink, P. Mitra, E. Neuhold, S. Pichai, R. Studer, and G. Wiederhold. An algebra for semantic interoperation of semistructured data. In *IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, pages 86–100, 1999.

[119] Jena semantic web toolkit. `http://jena.sourceforge.net/` (as of September 2003).

[120] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A declarative query language for RDF. In *Proc. Intl. World Wide Web Conf.*, pages 592–503. ACM Press, 2002.

[121] V. Kashyap and A. Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In Michael P. Papazoglou and Gunter Schlageter, editors, *Cooperative Information Systems*, pages 139–178. Academic Press, San Diego, 1998.

[122] V. Kashyap and A. Sheth. *Information Brokering Across Heterogenous Digital Data*. Kluwer Academic Publishers, 2000.

[123] W. Kent. The many forms of a single fact. In *IEEE COMPCON*, pages 438–443, 1989.

[124] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991.

[125] S. Kleijnen and S. Raju. An open web services arquitecture. *ACM Queue*, 1(1):39–46, 2003.

[126] M. Klein, D. Fensel, F. van Harmelen, and I. Horrocks. The relation between ontologies and xml schemas, 2001. `http://www.ep.liu.se/ea/cis/2001/004/` (as of November 2003).

[127] M. R. Koivunen and E. Miller. W3C semantic web activity, 2001. `http://www.w3.org/2001/12/semweb-fin/w3csw` (as of November 2003).

[128] H. Kreger. Fulfilling the web services promise. *Communications of the ACM*, 46(6):29–34, 2003.

[129] R. Krishnamurthy, W. Litwin, and W. Kent. Language features for interoperability of databases with schematic discrepancies. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 40–49. ACM Press, 1991.

[130] A. Labrinidis and N. Roussopoulos. Generating dynamic content at database-backed web servers: cgi-bin vs. mod-perl. *SIGMOD Record*, 29(1):26–31, 2000.

[131] L. V. S. Lakshmanan, F. Sadri, and I. N. Subramanian. SchemaSQL - a language for interoperability in relational multi-database systems. In *Proc. VLDB Conf.*, pages 239–250. Morgan Kaufmann, 1996.

[132] L. V. S. Lakshmanan, S. N. Subramanian, N. Goyal, and R. Krishnamurthy. On query spreadsheets. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 134–141. IEEE, 1998.

[133] O. Lassila and R. R. Swick. Resource Description Framework (RDF): Model and syntax specification, 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222` (as of November 2003).

[134] Dongwon Lee and Wesley W. Chu. Comparative analysis of six XML schema languages. *SIGMOD Record*, 29(3):76–87, 2000.

[135] Alon Levy. More on data management for XML, 1999. `http://www.cs.washington.edu/homes/alon/widom-response.html` (as of November 1999).

[136] W. Litwin and A. Abdellatif. Multidatabase interoperability. *IEEE Computer*, 19(12):10–18, 1986.

[137] W. Litwin, L. Mark, and Nick Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22(3):267–293, 1990.

[138] L. Liu, D. Buttler, T. Critchlow, W. Han, H. Paques, C. Pu, and D. Rocco. Bioseek: Exploiting source-capability information for integrated access to multiple bioinformatics data sources. In *Intl. Symp. on BioInformatics and BioEngineering (BIBE)*, pages 263–274. IEEE Computer Society, 2003.

[139] L. Liu and R. Meersman. The building blocks for specifying communication behavior of complex objects: An activity-driven approach. *ACM TODS*, 21(2):157–207, 1996.

[140] L. Liu and C. Pu. Activityflow: Towards incremental specification and flexible coordination of workflow activities. In *Intl. Conf. on Conceptual Modeling (ER)*, volume 1331 of *LNCS*, pages 169–182. Springer-Verlag, 1997.

[141] L. Liu and C. Pu. Methodical restructuring of complex workflow activities. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 342–350. IEEE, 1998.

[142] L. Liu and C. Pu. A transactional activity model for organizing open-ended cooperative activities. In *Hawaii Intl. Conf. on Systen Sciences (HICSS)*, 1998.

[143] L. Liu, C. Pu, and Wei Han. XWrap: An XML-enabled wrapper construction system for web information sources. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 611–621. IEEE Press, 2000.

[144] D. B. Lomet and J. Widom (eds.). Special issue on materialized views and data warehouses. *IEEE Data Engineering Bulletin*, 18(2), 1995.

[145] K. Sattler M. Gertz. Integrating scientific data through external, concept-based annotations. In *Proc. VLDB Workshop on Efficiency and Effectiveness of XML Tools and Techniques (EEXTT)*, volume 2590 of *LNCS*, pages 220–240. Springer-Verlag, 2002.

[146] D. S. Mackay. Semantic integration of environmental models for application to global information systems and decision-making. *SIGMOD Record*, 28(1):13–19, 1999.

[147] A. Matono, T. Amagasa, M. Yoshikawa, and S. Uemura. An indexing scheme for RDF and RDF schema based on suffix arrays. In *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pages 169–187. VLDB endowment, 2003. `http://www.cs.uic.edu/~ifc/SWDB/` (as of November 2003).

[148] R. Mattison. *Data warehousing: strategies, technologies and techniques*. John Wiley and Sons, New York, 1996.

[149] E. M. Maximilien and M. P. Singh. Conceptual model of web service reputation. *SIGMOD Record*, 31(4):36–41, 2002.

[150] D. L. McGuinness, R. Fikes, J. Hendler, and L. A. Stein. DAML+OIL: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5), Sep 2002.

[151] S. A. McIlraith, T. C. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, 2001.

[152] C. B. Medeiros and F. Pires. Databases for GIS. *SIGMOD Record*, 23(1):107–115, March 1994.

[153] C. B. Medeiros, G. Vossen, and M. Weske. WASA - a workflow-based architecture to support scientific database applications. In *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, volume 978 of *LNCS*, pages 574–583. Springer-Verlag, 1995.

[154] C. B. Medeiros, M. Weske, and G. Vossen. GEO-WASA - combining GIS technology with workflow management. In *Israeli Conf. on Computer-Based Systems and Software Engineering*, pages 129–139, 1996.

[155] J. Meidanis, G. Vossen, and M. Weske. Using workflow management in dna sequencing. In *CoopIS*, pages 114–123, 1996.

[156] E. Mena, A. Illarramendi, V. Kashyap, and . P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[157] E Mena, V. Kashyap, A. Illarramendi, and A. P. Sheth. Managing multiple information sources through ontologies: Relationship between vocabulary heterogeneity and loss of information. In *KRDB*, number 4 in CEUR-WS.org, 1996.

[158] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Domain specific ontologies for semantic information brokering on the global information infrastructure, 1998.

[159] G. A. Mihaila, L. Raschid, and A. Tomasic. Locating and accessing data repositories with web semantics. *VLDB Journal*, 11(1):41–57, 2002.

[160] T. Mikalsen, S. Tai, and I. Rouvellou. Transactional attitudes: Reliable composition of autonomous web services. In *Proc. Workshop on Dependable Middleware-based Systems (WDMS)*, 2002.

[161] G. Miller. The web services debate – .NET versus J2EE. *Communications of the ACM*, 46(6):64–67, 2003.

[162] L. Miller, A. Seaborne, and A. Reggiori. Three implementations of SquishQL, a simple RDF query language. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 423–435. Springer-Verlag, 2002.

[163] R. J. Miller. Using schematically heterogeneous structures. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 189–200. ACM Press, 1998.

[164] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Report HPL-2002-57, HP Labs, Palo

Alto, CA, 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html (as of December 2003).

[165] M. Minsky. A framework for representing knowledge. In *The Psycology of Computer Vision*, pages 211–277. McGraw-Hill, 1975.

[166] M. Missikoff, R. Navigli, and P. Velardi. Integrated approach to web ontology learning and engineering. *IEEE Computer*, 35(11):60–63, 2002.

[167] M. Missikoff, R. Navigli, and P. Velardi. The Usable Ontology: An Environment for Building and Assessing a Domain Ontology. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 39–53. Springer-Verlag, 2002.

[168] P. Mitra, G. Wiederhold, and M. L. Kersten. A graph-oriented model for articulation of ontology interdependencies. In *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1777 of *LNCS*, pages 86–100. Springer-Verlag, 2000.

[169] G. A. Modica, A. Gal, and H. M. Jamil. The use of machine-generated ontologies in dynamic information seeking. In *Proc. Intl. Conf. on Cooperative Information Systems (CoopIS)*, volume 2172 of *LNCS*, pages 433–448. Springer-Verlag, 2001.

[170] Namespaces in XML 1.1. http://www.w3.org/TR/xml-names11/ (as of October 2003).

[171] S. Narayanan and S. A. McIlraith. Simulation, verification and automated composition of web services. In *Proc. WWW Conf.*, pages 77–88. ACM Press, 2002.

[172] M. Neiling, M. Schaal, and M. Schumann. WrapIt: Automated integration of web databases with extensional overlaps. In *Proc. Intl. Conf. on Web Databases and Web Services*, volume 2593 of *LNCS*, pages 184–198. Springer-Verlag, 2002.

[173] W. Nejdl, B. Wolf, Changtao Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P networking infrastructure based on RDF. In *Proc. WWW Conf.*, pages 604–615. ACM Press, 2002.

[174] S. Nestorov, S. Abiteboul, and R. Motwani. Extracting schema from semistructured data. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 295–306. ACM Press, 1998.

[175] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2002.

[176] Ontology inference layer (OIL). http://www.ontoknowledge.org/oil/ (as of November 2003).

[177] OpenGIS Consortium. Geography markup language (GML). http://www.opengis.net/gml/02-069/GML2-12.html (as of October 2003).

[178] R. Orfali and D. Harkey. *Client/Server Programming with Java and CORBA*. John Wiley & Sons, 2 edition, 1998.

[179] R. Orfali, D. Harkey, and J. Edwards. *The Essential Distributed Objects Survival Guide*. John Wiley & Sons, 1996.

[180] A. M. Ouksel and A. P. Sheth. Semantic interoperability in global information systems: A brief introduction to the research area and the special section. *SIGMOD Record*, 28(1):5–12, 1999.

[181] Web ontology language (OWL) version 1.0. `http://www.w3.org/TR/2003/WD-owl-ref-20030221/` (as of November 2003).

[182] M. T. Ozsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, San Ysidro, CA, 1999.

[183] M. Paolucci, T. Kawamura, and K. P. Sycara T. R. Payne. Semantic matching of web services capabilities. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 333–347. Springer-Verlag, 2002.

[184] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proc. ICDT Conf.*, pages 251–260. IEEE Press, 1995.

[185] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *CACM*, 41(5):166–178, 1998.

[186] P. Patel-Schneider and J. Siméon. Building the semantic web on XML. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 147–161. Springer-Verlag, 2002.

[187] P. Patel-Schneider and J. Siméon. The yin/yang web: Xml syntax and rdf semantics. In *Proc. Intl. Conf. on World Wide Web*, pages 443–453. ACM Press, 2002.

[188] G. R. B. Pinto, S. P. J. Medeiros, J. M. de Souza, J. C. M. Strauch, and C. R. F. Marques. Spatial data integration in a collaborative design framework. *Communications of the ACM*, 46(3):86–90, 2003.

[189] P. F. Pires, M. R. F. Benevides, and M. Mattoso. Building reliable web services compositions. In *Proc. Intl. Conf. on Web Databases and Web Services*, volume 2593 of *LNCS*, pages 59–72. Springer-Verlag, 2002.

[190] V. Poe, P. Klauer, and S. Brost. *Building a Data warehouse for Decision Support*. Prentice Hall, 1998.

[191] C. Pu, K Schwan, and J. Walpole. Infosphere project: System support for information flow applications. *SIGMOD Record*, 30(1):25–34, 2001.

[192] D. Quass and J. Widom. On-line warehouse view maintenance. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 393–404. ACM Press, 1997.

[193] Ahmed R, P. De Smedt, W. Du, W. Kent, M. A. Ketabchi, W. Litwin, A. Rafii, and Ming-Chien Shan. The pegasus heterogeneous multidatabase system. *IEEE Computer*, 24(12):19–27, 1991.

[194] S. Raghavan and H. Garcia-Molina. Integrating diverse information management systems: A brief survey. *IEEE Data Engineering Bulletin*, 24(4):44–52, 2001.

[195] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. A scalable and ontology-based p2p infrastructure for semantic web services. In *Proc. Intl. Conf. on Peer-to-Peer Computing (P2P)*, pages 104–111. Australasian Computer Society, 2002.

[196] L. A. Seffino, C. B. Medeiros, J. V. Rocha, and Bei Yi. WOODSS - a spatial decision support system based on workflows. *Decision Support Systems*, 27(1-2):105–123, 1999.

[197] W3C's Semantic web Activity. `http://www.w3.org/2001/sw/` (as of July 2003).

[198] U. Shah, T. Finin, and J. Mayfield. Information retrieval on the semantic web. In *Proc. Intl. Conf. on Information and Knowledge Management (CIKM)*, pages 461–468. ACM Press, 2002.

[199] S. Shekthar, S. Chawla, S. Ravada, A. Fetterer, Xuan Liu, and Chang tien Lu. Spatial databases - accomplishments and reseach needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 1999.

[200] A. P. Sheth. Semantic issues in multidatabase systems - preface by the special issue editor. *SIGMOD Record*, 20(4):5–9, 1991.

[201] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[202] E. Sirin, J. A. Hendler, and B. Parsia. Semi-automatic composition of web services using semantic descriptions. In *Proc. Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI)*, pages 17–24. ICEIS Press, 2003.

[203] T. Sollazzo, S. Handschuh, S. Staab, and M. Frank. Semantic web service architecture – evolving web service standards toward the semantic web. In *FLAIRS Conf. – Special Track on Semantic Web*, pages 425–429, 2002.

[204] M. Stal. Web services: beyond component-based computing. *Communications of the ACM*, 45(10):71–76, 2002.

[205] L. Stein. Creating a bioinformatics nation. *Nature*, 417(6885):119–120, 2002.

[206] E. Stolte, C. von Praun, G. Alonso, and T. Gross. Scientific data repositories – designing for a moving target. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 349–360. ACM Press, 2003.

[207] M. Stonebraker. Implementation of integrity constraints and views by query modification. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 65–78. ACM Press, 1975.

[208] Y. Sure, J. Angele, and S. Staab. OntoEdit: Guiding ontology development by methodology and inferencing. In *Intl. Conf. on Cooperative Information Systems (CoopIS)*, volume 2519 of *LNCS*, pages 1205–1222. Springer-Verlag, 2002.

[209] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. Ontoedit: Collaborative ontology development for the semantic web. In *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pages 348–363. Springer-Verlag, 2002.

[210] N. Tryfona and C. S. Jensen. Conceptual data modeling for spatiotemporal applications. *GeoInformatica*, 3(3):245–268, 1999.

[211] A. Tsalgatidou and T. Pilioura. An overview of standards and related technologies in web services. *Distributed and Parallel Databases*, 12:135–162, 2002.

[212] Universal description, discovery and integration of web services (UDDI). `http://www.uddi.org/` as of October 2003.

[213] J. D. Ullman. Information integration using logical views. In *Proc. ICDT Conf.*, volume 1186 of *LNCS*, pages 19–40. Springer-Verlag, 1997.

[214] Naming and addressing (URIs, URLs, ... `http://www.w3.org/Addressing/` (as of October 2003).

[215] M. Uschold and M. Gruninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[216] W. M. P. van der Aalst. Don't go with the flow: Web services composition standards exposed. *IEEE Intelligent Systems*, 18(1):72–85, 2003.

[217] W. M. P. van der Aalst, A. Hirnschall, and H. M. W. Verbeek. An alternative way to analyze workflow graphs. In *Advanced Information Systems Engineering (CAiSE)*, volume 2348 of *LNCS*, pages 535–552. Springer-Verlag, 2002.

[218] V. Ventrone and S. Heiler. Semantic heterogeneity as a result of domain evolution. *SIGMOD Record*, 20(4):16–20, 1991.

[219] A. Voisard, C. B. Medeiros, and G. Jomier. Database support for cooperative work documentation. In *COOP*, 2000.

[220] J. Wainer, G. Vossen, M. Weske, and C. B. Medeiros. Scientific workflow systems. In *NSF Workshop on Workflow and Process Automation: State of the Art and Future Directions*, 1995.

[221] J. Wang and F. H. Lochovsky. Data extraction and label assignment for web databases. In *Proc. Intl. Conf. on World Wide Web (WWW)*, pages 187–196. ACM Press, 2003.

[222] M. Weske, G. Vossen, C. B. Medeiros, and F. Pires. Workflow management in geo-processing applications. In *ACM-GIS*, pages 88–93, 1998.

[223] J. Widom. Data management for XML: Research directions. *IEEE Data Engineering Bulletin*, 22(3):44–52, 1999.

[224] G. Wiederhold. Views, objects, and databases. *IEEE Computer*, 19(12):37–44, 1986.

[225] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(1):38–49, 1992.

[226] G. Wiederhold. An algebra for ontology composition. In *Monterey Workshop on Formal Methods*, pages 56–61, 1994.

[227] G. Wiederhold. Interoperation, mediation, and ontologies. In *Intl. Symp. on Fifth Generation Computer Systems (FGCS) – Workshop on Heterogeneous Cooperative Knowledge-Bases*, pages 33–48, 1994.

[228] G. Wiederhold and J. Jannik. Composing diverse ontologies. Report, Stanford University, 1998.

[229] K. Wilkinson, C. Sayers, and H. Kuno. Efficient RDF storage and retrieval in jena2. In *Proc. Intl. Workshop on Semantic Web and Databases*, pages 131–150. Humboldt-Universität, 2003.

[230] J. Williams. The web services debate – J2EE versus .NET. *Communications of the ACM*, 46(6):59–63, 2003.

[231] P. Wohed, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. Pattern based analysis of BPEL4WS. Report FIT-TR-2002-04, Queensland University of Technology, Queensland, Australia, 2002.

[232] M. F. Worboys and S. M. Deen. Semantic heterogeneity in distributed geographic databases. *SIGMOD Record*, 20(4):30–34, 1991.

[233] The W3C web services activity. `http://www.w3.org/2002/ws/` (as of October 2003).

[234] Web services description language (WSDL) 1.1. `http://www.w3.org/TR/wsdl` (as of October 2003).

[235] Web Services Flow Language (WSFL 1.0). `http://www.ibm.com/software/solutions/webservices/pdf/WSFL.pdf` (as of October 2003).

[236] W3C's Extensible Markup Language (XML) 1.o (second edition). `http://www.w3.org/TR/REC-xml` (as of October 2003).

[237] XML Schema part 0: Primer. `http://www.w3.org/XML` (as of October 2003).

[238] W3C's XML Query Activity. `http://www.w3.org/XML/Query` (as of October 2003).

[239] E. Ioannidis Y, M. Livny, A. Ailamaki, A. Ranganathan, A. Therber, M. Yuin, M. Anderson, and J. Norman. Managing soil science experiments using ZOO. In *Proc. Conf. on Statistical and Scientific Database Management*, pages 121–124. IEEE, 1997.

[240] N. Zhong, J. Liu, and Y. Yao (eds.). Special Issue – In Search of the Wisdom Web. *IEEE Computer*, 35(11):27–76, 2002.

[241] Yue Zhuge, H.r Garcia-Molina, J. Hammer, and J. Widom. View maintenance in a warehousing environment. In *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pages 316–327. ACM Press, 1995.