TOWARDS A NEXT GENERATION OF OPEN SCIENTIFIC DATA REPOSITORIES AND SERVICES

Catherine Houstis, Christos Nikolaou, Spyros Lalis
Computer Science Department University of Crete & Institute of Computer Science-Foundation for Research and Technology
houstis@ics.forth.gr, nikolau@ics.forth.gr, lalis@ics.forth.gr

Sarantos Kapidakis, Vassilis Christophides
Institute of Computer Science- Foundation for Research and Technology
sarantos@ics.forth.gr, christop@ics.forth.gr

Abstract

Scientific data repositories which comprise of numeric data, images, text documents audio and data along with software methods for data processing and simulators of processes where data are inputted to predict physical behavior, are found in most research institutions and organizations. The integration and visualization of data repositories into an easily accessed interoperable networked environment is needed in many disciplines for both scientific and management purposes. To satisfy these needs we present an open hybrid architecture, which combines digital library technology, information integration mechanisms and workflow-based systems. Our experience is with the THETIS[1] project, a system of data repositories of coastal zones, which focuses on supporting Coastal Zone Management of the Mediterranean Region in Europe. It will demonstrate its ability to respond to users such as scientists and local authorities that use scientific information for decision making.

1. Introduction

Scientific data repositories, maintained by various institutions, have existed for many years mostly in isolation. The improvement of communication technologies and the emergence of the Internet and the WWW [Lee94] have made feasible the electronic availability and interconnection of historically independent information sources across national borders. Despite the global network infrastructure that opens the way towards advanced data exchange, the location, retrieval and even more the combination of information from distributed sources proves to be a complex and a time consuming task. In fact, even expert users are not aware of where, what and how to look for information. It appears that - now more than ever - managing, accessing and integrating information from multiple scientific data sources is a major challenge.

For example, environmental scientists and public institutions working on Coastal Zone Management (CZM) need to extract and visualize data matching their interests from

[1] THETIS is funded by the Research on Telematics programme of the EU, project nr. F0069, April 1997.
Contact: Catherine Houstis, Institute of Computer Science, FORTH, PO Box 1385, GR71110, Heraklion, Greece. Phone: +30.81.391729 , Fax: +30.81.391601, E-mail: houstis@ics.forth.gr

several repositories, involving different scientific disciplines such as marine biology, oceanography, chemistry, geology, and engineering. They are facing today an array of commercial and custom database interfaces, computer operating systems and network protocols to be mastered in order to examine potentially relevant data. They also have to deal with almost all the aspects of the information they receives from identifying what is useful, reliable, complete or how they would put it together with other information in order to further produce scientific data.

The volume of data resulting from experimental observations and post-processing can be staggering. The amount of data collected over several years in a conventional information system can be generated within a few hours in a scientific environment (e.g. satellite pictures). Downloading entire scientific data sets over the network is simply unrealistic. Even ad-hoc browsing and retrieval engines can easily result in huge amounts of information, causing unacceptably long delays in large scale distributed systems. Database technology support to scientific applications may alleviate only some of the problems involved in scientific data computing such as the persistence of large data volumes and declarative querying facilities. The main reason is that scientific data management and processing differs considerably from business applications supported by commercial systems.

A central issue is that scientific data sets are quite complex, containing a multitude of variables across several dimensions - usually (but not necessarily) longitude, latitude, depth, and time. Hence, multidimensional scientific data cannot be represented efficiently and directly using record-oriented models (i.e., relational databases). Moreover, scientific environments do not comprise only data. A multitude of programs is used to perform sophisticated simulations of physical, chemical and biological processes or implement data processing functions. Unlike conventional transactional applications, these programs are extremely complex, usually run on special-purpose hardware and have long execution times (several hours). They range from legacy software operated under strict licensing agreements to public domain programs that have been customized by hand to the needs of each organization. Object-oriented models seams to be more appropriate for scientific applications but commercial systems are still stand alone and do not interoperate easily with existing external data and programs.

Data management and integration is further complicated for networked scientific repositories, since each legacy source or program understands its own syntax, semantics and set of operations, and has different runtime requirements. Scientific data is typically represented in a variety of forms and formats from flat files to object DBMSs. Traditional multidatabases are too rigid and labor-intensive to adapt to an open environment where new sources are integrated dynamically and they do not predict and access to data residing outside databases. Also, contrary to the explicit description of the conceptualization provided by a database (i.e., a schema), the information required for interpreting scientific data is far less precise. It is typically part of the discipline's nomenclature documented in some form of free text or hidden in the various processing programs and analytical tools. Furthermore, scientific data is continually evolving together with their conceptualizations. Because the scientist's conceptualization of the data is highly dynamic, the underlying logical schema of a data source tends to be modified frequently. For these reasons the cost and effort required to maintain a functional, conventional warehouse becomes prohibitive.

These problems are multiplied when considering large environmental information systems. Environmental questions touch upon a large number of different knowledge domains that are addressed by several autonomous subsystems with radically different semantic data models. Also, the services of environmental information systems are offered to a widely heterogeneous user community, ranging from authorities with varying technical and environmental expertise as well as to the public. Finally, scientists and researchers are themselves users of the system, accessing data that has already been produced by others and reusing available components to implement more complex data processing and simulations.

In this paper we focus on the architectural issues involved in the next generation of scientific data repositories and services. We propose an open environment with appropriate middleware architecture that combines digital library technology, information integration mechanisms and workflow-based systems to address scalability both at the data repositories level and the user level. A hybrid approach is advocated where the middleware infrastructure contains a dynamic scientific collaborative work environment and mediation to provide both the scientists and simple users the means to access globally distributed heterogeneous scientific information. In this environment, integration and visualization of scientific information is a collaborative and interactive process. Scientists are invited to "plug-and-play", by selecting which sources to access, what part of the data sets to extract, how to consolidate and aggregate data at a higher level of abstraction, which scientific models to employ for analyzing data, and what programs to invoke for data visualization. Mediation is used to offer robust data abstraction with advanced querying capabilities, which is required to build heavy-weight applications such as decision making tools. The collaborative work environment is addressing mainly the needs of the scientific community for flexible experimentation with the system components, at run time.

The remaining of this paper is organized as follows. In section 2 the new trends in global information systems are reviewed, along with existing information systems built according to these trends. In section 3, we present a hybrid architecture that addresses the main problems of networked scientific repositories, both for simple users and scientists. The discussion is by no means complete and it is still on going. The proposed design is currently being implemented in THETIS, a WWW-based system for coastal zone management of the Mediterranean Sea [Houstis97], funded by the European Union. In section 5, we compare with related work. Finally, in section 6, we conclude and set future directions.

2. Trends in Global Information Systems

Research on access and integration of heterogeneous information spread over a number of distributed sources has attracted a great attention during past years. Providing the necessary infrastructure for information highways has become a major challenge of computer scientists working in the areas of Information Retrieval Systems (IRS), Database Management (DBMS) and Knowledge Representation and Reasoning Systems

(KRRS). In particular in the US work in these areas has been actively supported by a number of initiatives, notably the Digital Library Initiative [DLI], the Intelligent Integration of Information Initiative [I3] and Knowledge Sharing effort [KSI].

Distributed systems architects' work on middleware software architectures and tools appropriate for network-centric applications depend on the various degrees of interoperability and integration intelligence needed between the user applications and the data repositories [Wiederhold92]. Some of the most recent work addresses scientific collaborative work environments, which emphasize dynamic architectures, and satisfy user services on demand [Marazakis97, Nikolaou97]. In this case, the problem of identifying "relevant" data sets and models is reduced to a matchmaking process, an approach that can be used for other large-scale distributed applications, electronic commerce environments, distributed systems management environments, and office automation workflow.

However, it is digital library and mediation based systems that have attracted the attention of researchers. The former focuses mainly on interoperability and openness while the latter emphasizes integration and incremental development. Also, while metadata plays a central role in both cases, in digital libraries metadata are searched directly by keyword or full-text retrieval tools, while mediator-based systems mostly use a declarative query language [Buneman97] for expressing searches. The second approach is usually present in mediator based architectures.

Digital Libraries projects address several issues related to the management of large collections of documents distributed over the Internet. These issues concern the heterogeneity of users, information repositories, services, and payment mechanisms, as well as in metadata forms and formats used to index documents by structure, semantics, and content (including multimedia information such as images, maps, video, etc.). Several digital library projects are already coming to the end of their first development phase at several Universities of US and New Zealand: Illinois [IllinoisDL], Berkeley [BerkeleyDL], Stanford [StanfordDL], Michigan [MichiganDL], Carnegie-Mellon [CMUDL], Santa Barbara [UCSBDL] and Waikato [WaikatoDL].

The projects of Illinois and Berkeley are developing DL services for multiple user groups (i.e. for educational, entertainment needs, etc.). Illinois DL focuses on semantic indexing and retrieval of SGML documents using appropriate concept spaces. Berkeley's DL focuses on natural language, optical recognition and geo reference techniques to improve native indexing of local sources especially for multimedia documents. A project with similar motivations is the Alexandria DL, currently underway at the University of California at Santa Barbara. Compound document processing and reusability of information and applications is one of the Berkeley DL objectives. A multivalent document model is considered in order to superimpose multiple levels of contents to a document with associated behavior programs, possibly distributed over the Web. The projects of Michigan and Stanford focus on the development of appropriate gateway/mediation services. They achieve interoperability between different network (e.g., HTTP, TELNET, etc.) or task specific protocols (Z39.50, ODBC, etc.) employed by the information sources as well as between co-operative agents. They elicit the representations required to handle heterogeneity of data (and metadata) forms and formats for different search services. Moreover, they handle representations for the identification and location of relevant information sources to user requests. Finally, the Waikato DL project has developed a system for full-text indexing and retrieval of very large collections of texts. A CSCW technique is also used to enable collaboration during location and gathering of documents.

Information Integration Systems focus on the design and implementation of mediator-based architectures between end-users and data sources, varying from document repositories to relational or object-oriented databases and knowledge bases. More precisely, the emphasis is given to mediator models and languages, advanced query processing and optimization, as well as specifications of wrapping interfaces for the translation of queries and results between the mediator and sources. Related to this approach are projects like TSIMMIS [TSIMMIS], MANIFOLD [MANIFOLD], GARLIC [GARLIC], DISCO [DISCO], HERMES [HERMES], SIMS [SIMS], and InfoSleuth [INFOSLEUTH], which are developed at several Universities and Research Institutes of the US and Europe.

GARLIC and DISCO rely on the object model and query language of the ODMG (ODL and OQL) standard with the former focusing on uniform access of heterogeneous multimedia data sources, and the latter on query processing when data sources may not return answers due for instance to network problems. MANIFOLD and TSIMMIS elicit two alternative integration architectures as far as the consideration of sources viewed by the mediator [Hull97] and the related query processing strategies [Ullman97]. The former extends a description logic model (CLASSIC) to integrate structured relational databases and legacy systems besides Web interfaces. The latter proposes a self-describing model and query language (OEM and LOREL) for the integration of structured and semi-structured data sources such as various document repositories (html pages, WAIS sources, etc.). It must be stressed that the above systems encompass operational discrepancies of information sources (i.e., heterogeneity of query languages) by describing and using the various native query capabilities of the sources at the mediator level. In this way they avoid the "least common denominator approach" in source's functionality followed in most of the digital libraries systems described previously. Finally, HERMES, and SIMS address mainly domain modeling and reasoning issues using appropriate KRRS technology (e.g., HKB, LOOM). They provide an intelligent integration of information [Wiederhold96] by inferring new knowledge from existing data. They also dynamically select relevant information sources, and perform semantic query rewriting in order to optimize the execution of queries in local sources. To achieve dynamic location of relevant information sources, InfoSleuth integrates the technology of Intelligent Agents and emerging standards such as KQML/KIF [Patil92].

3. User Requirements of Environmental Systems

The variety of user requirements of environmental systems is illustrated by the example of Coastal Zone Management (CZM) of scientific data repositories as encountered in the THETIS system. CZM is a methodology for the holistic management of all coastal resources with the ultimate aim of promoting sustainable development of the coastal zones. It recognizes that pollution problems transcend political boundaries and so, to be effective, CZM requires the integration of multinational data repositories, data management and data visualization across many scientific disciplines such as marine biology, oceanography, chemistry and engineering.

Even at its simplest form, CZM involves three major users:

An End User (e.g., general public, policy maker) who needs to locate and extract data that matches his interest, or appropriate data servers to retrieve data of the desired level of quality. For example, a user may need to access the rating of beaches in his town. Then, he asks why his town is not considered a safe beach. As a result he gets a definition of a safe beach that is understandable to him, i.e., at the appropriate level of detail, and the data that the definition depends on. For instance, safety may be defined as a collection of criteria such as expected height of waves, and presence of sharks. Then this user may want to find out who, and when, collected the data about the presence of sharks near his beaches. He puts a high value on the accessibility, interpretability and usefulness of data.

A Broker (e.g., environmental scientist, and public authority administrator) maintains the servers for end users. For instance, a broker may have to write programs to access measurement databases, administrative inquiries, remote sensing data, and geographical databases to construct a map of France that indicates the quality of beaches. Also, she writes programs to improve the reliability of data using consolidation techniques. Generally, a broker must find the data necessary for each new program that he writes, and each new program may use multiple data sources. Each data source requires a unique program to extract the data for the new program from the data source.

A Data Provider (e.g., biologist, geologist, physicist, oceanographer, etc.) collects data, and wants to distribute them as widely as possible. For instance, a data provider may manually add his data to an existing database through a standard form-based entry program. Data can also be collected using automatic sensors that directly transmit their data to an associated system. In this case, the provider has to verify the quality of data and eliminate erroneous measurements. To do this, he needs to use specific programs for data analysis and interpretation and access other systems for comparing data with other related data.

It becomes evident that users of environmental systems have widely different expertise and requirements. Data providers may have no particular knowledge in the environmental problem being addressed but still offer data that is used to calculate important evaluation parameters. Hence, it is important for the system to support inclusion of new data collections, in a straightforward way. Brokers may not be concerned with the management and calibration of individual data sources but combine several sources to produce value added information necessary by advanced decision support applications. Thus powerful abstraction mechanisms are needed, which simplify development and maintenance of intermediate data manipulation components. On the other hand, scientists who are strongly interested in using available data in an ad-hoc manner, or wish to experiment with simulations and visualization tools, value flexibility and openness. Finally, end users merely want to get specific information out of the system using well-defined interfaces and without being aware of the underlying complexity.

4. An Architecture for Environmental Systems

Standardization efforts have been intensified to provide building blocks for global information exchange. The recommendations of the workshops organized by NSF about scientific database management and projects [French90, Chu93], was a first important step towards standardization of some aspects of modern scientific computing. It has adopted standards for binary data encoding and APIs, like NSSDC CDF [CDF], Unidata NetCDF [NetCDF], or NCSA HDF [HDF]. It has also become evident that interoperability, integration, and collaboration, i.e., the ability to interpret, share and manipulate data and programs from multiple autonomous sources transparently, is the main requirement towards Global Scientific Data Repositories and Services over these repositories [NOAA97, Karp95, Davinson95, Letosky94, Maier94].

To address the interoperability, information integration and management issues emerging in the next generation of scientific data repositories and services, a hybrid architecture is proposed. It combines elements from the digital library technology, mediator-based systems and rapid prototyping environments to achieve the desired functionality. The system architecture consists of three main object types: data collections, mediators, and programs. The role of each object and the way in which they interact with each other is described in the following.

This advocated architecture is currently being implemented to build the THETIS system for managing coastal zones of the Mediterranean Sea [Houstis97]. THETIS can be viewed as a digital library of scientific data repositories that addresses the frequent requirement of scientists, engineers and decision-makers to access, process, combine and subsequently visualize data collected and stored in different formats and held at different locations.

4.1 Data Collections

Data collections represent data that are physically available, and which are housed in the system's repositories (figure 1). As the name suggests, a data collection is not a simple data object but comprises numerous data items. Items belonging to the same collection have common structural and semantic properties that allow them to be treated in a collective way, rather than as separate objects. However, individual data items of a collection can have different contents. Data collections are strictly local to a repository, hence all data items belonging to a certain data collection are stored in the same repository. A single repository may contain several data collections.

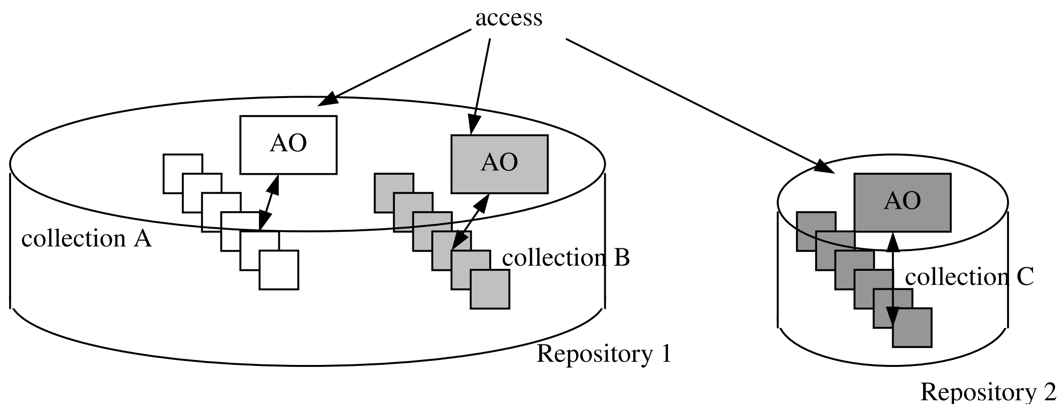

Figure 1. Data collections and repositories

Each data collection also comes with a set of operations, which provide the means for accessing its contents. The access operations essentially combine the model according to which items are organized within the collection and the mechanisms used to effectively retrieve data. Access operations are collection specific so that different data collections are likely to have different operations. This holds even for data collections that reside in the same repository.

Data collections typically contain numbers (values) and text (descriptions). Numbers are usually strictly ordered sets of values, representing measurements, parts of pictures, etc. Every value has an exact meaning that may depend on its position in the data. These data items are strictly structured and can be read and used directly by programs that know their format. Texts are unstructured or loosely structured information that may be partially parsed by programs, to derive information, but are mostly used as descriptions directed to the user. The most usual computer use of their content is display and participation on search based according to some criteria.

4.2 Mediators

Mediators are intermediary software components that stand between applications and data collections. They encapsulate one or more data collections, possibly residing in different repositories, and provide applications with higher-level abstract data models and querying facilities (figure 2).

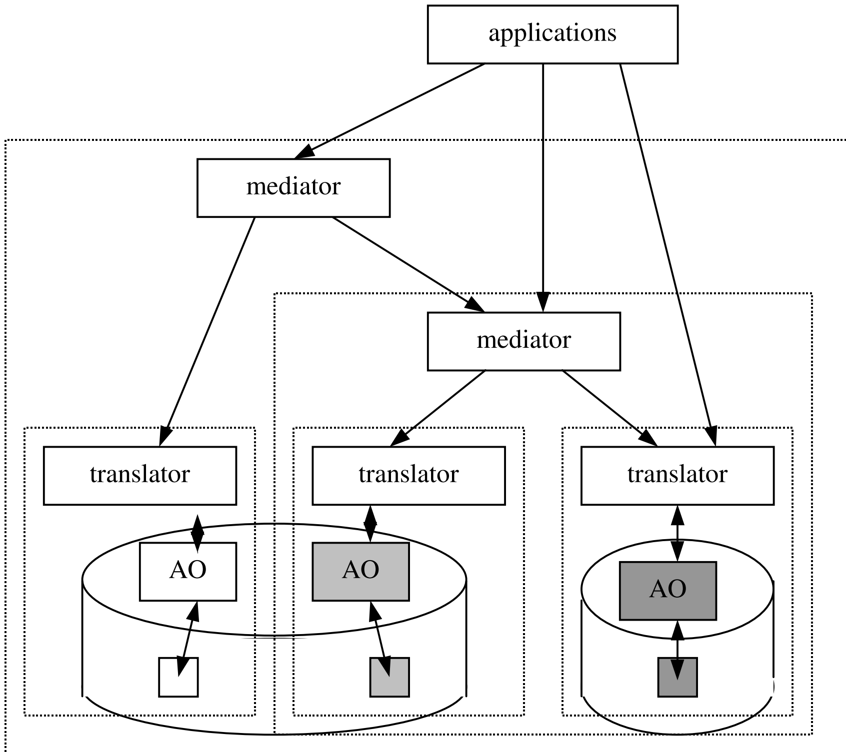

Figure 2. A hierarchy of mediators

The new functionality is introduced in a transparent way, i.e., by hiding all operational incompatibilities and functional differences among the various data collections. To achieve this, each query submitted to a mediator is translated into several low-level queries, using the native query models of the underlying data collections. Then, the sub-queries are forwarded to the data collections, and the results are collected. For data collections with weak querying capabilities, several consecutive queries may be required to produce a result that can be combined, at a semantic level, with the results from other data collections. Moreover, conversions at the data level and additional processing is typically needed to integrate the results received from the individual data collections in order to produce the final answer, which is returned to the application.

Mediators are introduced to encode complex tasks of consolidation, aggregation, analysis, and interpretation, involving several data collections. Thus mediators incorporate considerable *expertise* regarding not only the kind of data that is to be combined to obtain value-added information, but also the exact transformation procedure that has to be followed to achieve semantic correctness. In the simplest case, a mediator degenerates into a translator that merely converts queries and results between two different data models.

Because of the abstraction offered by mediators, mediators themselves may be viewed as abstract data collections. Consequently, novel and more demanding data abstractions can be built incrementally, by implementing new mediators on top of other mediators and data collections. This approach is well suited for large systems, because new functionality can be introduced without modifying or affecting existing components and applications.

4.3 Programs

Programs are procedures, described in a computer language that implement pure data processing functions. They may be used in combination with other data to perform complex calculations and produce results that can either be used by other programs, or can be shown to the user.

It is important to notice that programs are functionally different from mediators. First of all programs do not offer a query model through which data can be accessed in an organized way, instead they generate data in very primitive forms (e.g. streams or files).

Second, programs usually perform computationally intensive calculations and thus may exhibit long execution times (e.g. several hours). In contrast, data manipulations within mediators are done instantly and queries are answered within a few seconds. It is therefore costly to perform the same calculation more than once, whereas a query can be submitted to a mediator several times without inflicting a great performance penalty.

Third, program components can be very difficult to transport and execute on platforms other than the one where they are already installed. Numerical processing software that runs on special-purpose hardware like multiprocessors or vector computers is a typical example. Mediators, on the other hand, are portable and can be installed on any server of the system. Hence, program components are more like abstract processors rather than abstract data collections. In fact, several program components can be combined with each other to perform complex processing functions. This is achieved by directing the output port of one program into the input port of another, thereby creating a program pipeline (figure 3).

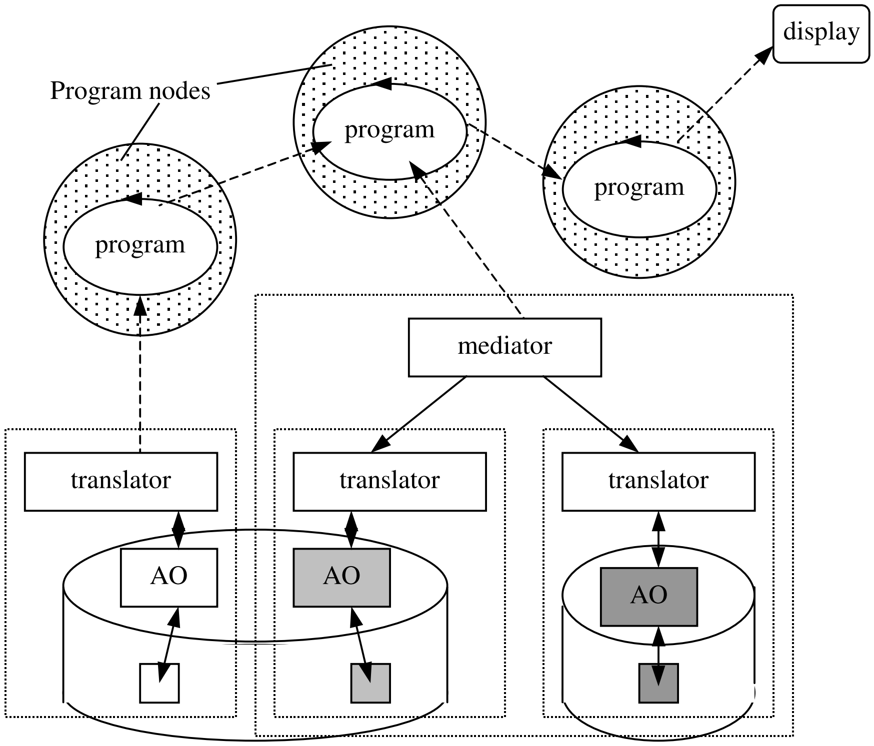

Figure 3. A program pipeline

Notably, programs can use existing data collections, mediators, as well as other programs to read input data. Special viewer programs can be put at the end of pipelines to present the result to the user, using appropriate visualization methods. Hence, the simplest pipeline is formed with a single viewer program that merely reads data from a source and displays it on the screen. Viewer programs usually run on the user's machine as dynamically loaded browser applets.

4.4 Metadata

Metadata means "data about data" or "information about information". In general, we classify metadata into two types:

Fully machine parsable and usable information or "machine understandable information". These metadata fields can be different in nature. Some fields (like titles) contain mostly terminology, some fields contain uninterpreted keywords (e.g. names like a creator name) and other fields contain more structured or accurate information (e.g. numeric values, dates, and format or protocol specifications).

Free-format (additional) information. Fields that mainly contain free text (like a free-text description field), are usually not machine parsed, but can be searched for keywords. For example, in an experiment we can store experimental conditions that cannot be classified under the other existing metadata fields, which describe the experiment. The free-format information can also be searched by the computer programs. The results from these searches are not very accurate, but are statistically usable: for example, the existence of the word "temperature", or a specific text pattern containing the word, may indicate with some certainty that there is temperature information. Information derived from free-format fields can either be processed once and be stored in other, derived, machine parsable fields, or be searched dynamically every time it is needed.

Of most importance to computers is the machine understandable information, because programs can directly process and use it efficiently, while free-format information is normally only presented to the user. For this reason, system designers always try to structure the metadata information into as many fine grained pieces as possible, so that programs can make better use of it.

The availability of fine-grained metadata gives better quality results on searching, using the same human effort but requires more human effort on the metadata creation.

The need for more functionality based on metadata and structure to the data has lead to many, evolving approaches. The more structured the metadata are, the more flexible and accurate information retrieval methods are possible - but also the more difficult it is to classify and handle the metadata and data. Depending on the desired system functionality, and the available supporting effort, the most appropriate compromise in structure is used.

The Dublin Core and the Warwick Framework [DCa, DCb] are a first step on metadata classification. They define a loose structure that is not difficult to maintain, and gives reasonable good search results.

However, XML [XMLa, XMLb] is becoming increasingly adopted as a common syntax for expressing structure in data. Now the resource Description Framework (RDF), a layer on top of XML, provides a common basis for expressing semantics by providing a set of specifications which allow metadata applications to be combined, and to operate with a common way of expressing the semantics which they share. Applications which allow programs to combine data logically are built using RDF (and therefor XML) and this enhances the modularity and extensibility of the Web. This is essential to its rapid future growth, multiplying together the strengths of new, independently developed, applications.

Then, searching is based exclusively, or primarily, on metadata. The user specifies keywords or other values that should match specific fields of interest, expressing his preferences, and the search engine locates the corresponding objects matching them.

For example, a user may have data for the height of the waves in Crete. The data may consist of quadruples of numbers: the time of the measurement, the two coordinates of the measurement and the height of the wave at this specific position and time (or a related time interval). The metadata for these specific data store location (they all refer to Crere), type (they express height of waves), quality (what is the measurement precision, quality and condition of the instruments, how often did we take measurements), storage method (ascii or proprietary, value alignment), etc.

Metadata is used to introduce the notion of *context* associated with a particular object. A formal definition of context requires a formal logic approach, which is not suited for use by an interdisciplinary group of scientists. Thus, in addition to a formal context, which is understood by the machine, each instance of data, mediator, or program is *paired* with a piece of text that provides an *informal interpretation.* In THETIS, each such pair is called a *dyad* (figure 4). Dyads that pair metadata instances with text are particularly useful for searching. In [Millard96] a set of classes is offered to describe the structure of typical environmental data objects such as maps, measuring series simulations, etc. Additional user defined classes can be defined and inherit the properties of the system-defined classes. Therefore, new classes can be added according to specific application needs, such as coastal zones, to offer a more detailed description of data sources.

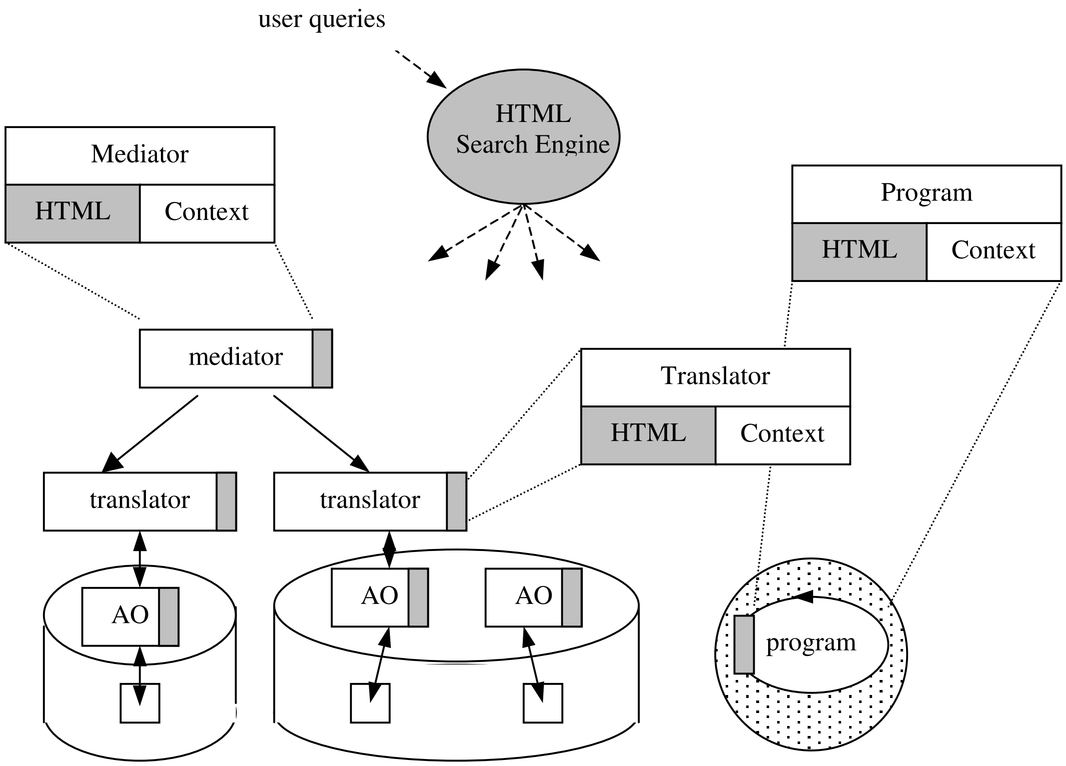

Figure 4. THETIS Objects as Dyads

Every object in the system has metadata associated with it. Thus the user (or any program) can browse through all registered objects, by issuing corresponding meta-searches expressing the properties of the objects to be retrieved. Since programs are also considered as objects, they are also described via metadata. Metadata are the key issue in combining data and producing interesting results: only when we know all significant details about our data we will be able to combine them and use them with minimal extra human effort.

The human readable metadata component of each object is an HTML document describing the component and the implementation of the data, mediation and computation. The HTML document provides a means (through indexing engines) for locating the corresponding objects. A data source, for instance a database system, exports a scheme, data, and query processing encapsulated as context. All of these elements are described in the associated HTML document. The document has sufficient information to permit direct browsing of the data by an (intelligent) browser that understands the query language supported by the data source. A translator provides conversion of queries between two different query languages-the language supported by the data source, and the language in which a mediator expresses queries to the data source. This functionality is again captured by context elements that are also described in the corresponding HTML document. Mediators encode the tasks of consolidation, aggregation, analysis and interpretation. The associated HTML document describes the integration process used and the type of the results. Some mediators may support the invocation of the computation used to generate the data. All mediators conform to the *same language* for queries, metadata, and data. Program objects are described in the same way, allowing the HTML part to describe the scientific models used, the run time requirements of the executable, and the type of the input and output.

4.5 End User Functionality: Navigation, Workflows and Applications

Access to this networked system, comprising several autonomous subsystems that provide the various data collections, mediators and programs, is given through a single web-based interface. However, the form of the interface changes as the user activates various functional elements.

Navigation is a key function in the system allowing users to browse through the object collections in order to locate objects of interest, which may be data sets, mediators, applications, or programs. This is achieved by invoking the metadata search engine that locates objects based on their metadata descriptions and provides the user with a list of descriptions and links to the corresponding objects. If the user selects the link to an object, all available information (metadata) about the object is presented, together with a list of available options that can be used to access or activate it. For instance, if the object is a data collection, a detailed description of its properties is given, along with the method provided for retrieving its contents. If the object is a mediator, the data model according to which data can be queried is returned. In both cases, the system interface dynamically adapts itself according to the selected objects, enabling the user to access (part of the) data located in the underlying repositories in a transparent way.

Data collections, mediators and programs are also put together to offer the user two advanced modes of use: applications and workflows. Workflows are essentially program pipelines that can be set up at run time. To create a workflow, components of interest have to be discovered using the system navigation and searching facilities. Several searches can be performed, using various selection criteria, to narrow the scope of previous searches, or to find additional components with different functionality. When the appropriate components have been identified, a workflow is defined by interactively *linking* the corresponding handles with each other on the screen. Based on the data flow indicated and the type of the components selected, the system checks for incompatibilities and restrictions. It then creates an execution context for running the workflow, sets up the network connections among the various components, and starts execution.

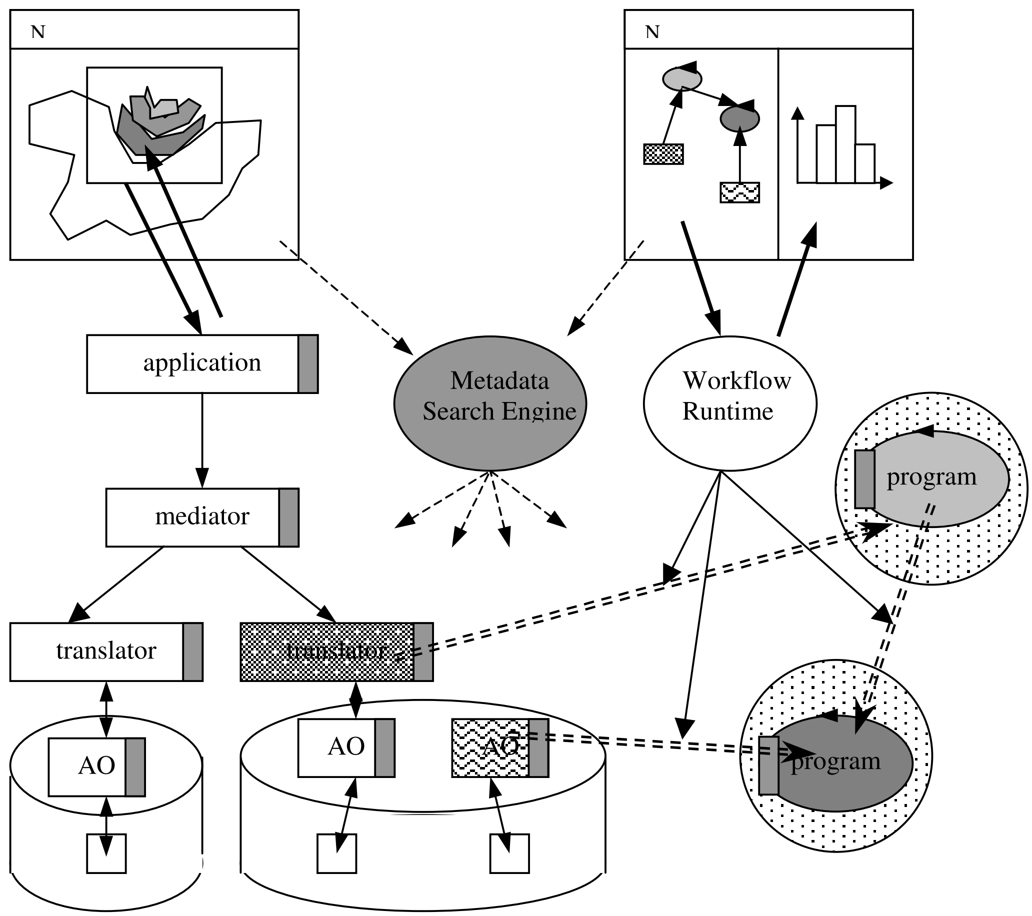

Figure 5. Finding and using system components

Applications are software systems tailored to a specific information-processing task. Each application uses a well-defined subset of the system's data collections, combines their content in a certain way, and provides value-added services to a known user group. Data flow and data processing of applications is static, because it is designed at implementation time, and is not modified unless the requirements of the application change. Consequently, all data-oriented application functions are introduced via carefully designed mediators. Applications, just like all other system components, have metadata and can be searched for in a similar manner.

This approach greatly enhances flexibility. On one hand, robust data abstraction with powerful querying capabilities, which is provided by mediators, simplifies integration of information and promotes incremental development of complex applications. On the other hand, with workflows, several independent components can be dynamically linked together, at run time, to experiment with novel combinations, which are not supported by the main mediation architecture. In fact, workflows can be used to test processing patterns that are useful for a particular user community, before embodying them into the system as mediators and applications.

5. Related Work

With the widespread use of the Internet and the increasing concern for the environment, several systems have been developed to simplify access of scientific data, and to support environmental management processes. A few typical approaches are outlined below.

The FERRET system [Hanklin] offers on-line access to an extensive climatological database containing data sets that numerous institutions have collected or calculated over several years. Data includes global surface marine observations, upper air analyses, satellite observations, ship drift derived surface ocean currents, heat flux climatologies, rainfall records, and other observed and derived fields. Web access to the FERRET program is achieved via an HTML graphical user interface that sends user selection over the network to the web server. In turn, the server invokes a set of CGI scripts to activate the FERRET program that performs the necessary calculations and returns the result, in the form of an HTML document, to the browser. FERRET supports interactive data visualization and extraction through option menus, push buttons, radio buttons, and text input fields. With these controls a user can specify the choice of data set, variable name, view, and location of interest in space and time. The location and time may be further refined through a map-based graphical pan and zoom option and through text input fields used to specify precise locations. FERRET is an excellent environment for unifying and browsing gridded data that may be defined on different grids. Graphical displays may be created in a straightforward way by combining fields of the available data sets. A collection of basic transformations ("filters") is provided, including derivatives, integrals, statistics, missing-value fillers, smoothers, value locators, and others, that may be applied symmetrically along any axis of a variable. New variables can be defined interactively as mathematical transformations of variables from data sets, and more complex expressions can be built through hierarchical variable definitions. Multidimensional command syntax is also used to designate arbitrary rectangular regions in space and "if-then-else" logic permits calculations to be applied over arbitrarily shaped regions. An assortment of data smoothers and data gap fillers compliments the normal range of mathematical analysis tools. Thus FERRET is essentially a sophisticated interface for visualizing local data sets. It does not support development of mediators and applications nor does it allow for simulation programs to be combined with each other.

The Environmental Decision Support System (EDSS) [Fine96] is a full-fledged system, which supports air quality modeling and management. The main component of the system, the chemistry and transport model mimics the chemical reactions that take place among chemical species in the atmosphere and follows these elements as they move about. This component is not just a model, but a complete modeling framework, where major physical and chemical phenomena fit into well-defined classes or categories. Contributing scientists write science "modules" as a collection of related subroutines. This is done in a conventional programming language, observing simple guidelines to guarantee compatibility with other modules of the same class. An Input/Output applications programming interface provides an easy-to-learn, easy-to-use programming interface to files for the air quality model developer and model-related-tool developer. It provides a variety of data structure types for organizing the data, and a set of access routines that offer selective direct access to the data in terms meaningful to the modeler.

EDSS users can create new models by placing the modules to be used into a configuration file. The selected modules are then obtained, compiled and linked automatically. To support the definition of complex simulations involving hundreds of programs, EDSS allows the user to specify the links among programs via a graphical point-and-click interface. The programs and the files used and produced by them can exist on almost any computer accessible over the network. For example, a user might choose to run computationally demanding simulation programs on a Cray supercomputer, while running some scientific visualization programs on a local SGI workstation. The system instructs the various computers to run the programs, either simultaneously or sequentially as determined by the configuration of the links among the programs. The user is informed of the simulation's progress by the changing colors and icon styles. The results of simulations can be visualized by selecting variables from a suitably formatted data file. A variety of the most often used visualizations can be used, such as color contour atlas plots, three-dimensional mesh plots, and line, bar, and scatter graphs. The system supports animation of several visualizations at the same time. It is also possible to select data based on regions and times of interest, and to use simulation variables in formulas to create more complex expressions. Several "standard" analysis programs are also available to examine model data. Finally, EDSS features a strategy development tool that demonstrates how air quality models, optimization, and various computer-based analysis tools can be used to make the design of air quality management strategies more efficient and allow more comprehensive consideration of such issues as cost, equity, and uncertainty. The tool provides four major capabilities: inventory and control strategy analysis, interactive strategy development, strategy performance testing, and optimization-based strategy development. EDSS exhibits similar functionality as the workflows proposed in our architecture, but in a controlled environment. There are certain guidelines to be followed when implementing modules that are incorporated into the program collections and the system's machines have been configured to allow remote installation, execution and monitoring of programs. In contrast to EDSS, our goal is to support workflows in an open environment, using completely autonomous subsystems connected to each other via the Internet. Also, EDSS focuses heavily on the dynamic combination of simulation programs whereas the THETIS architecture equally supports the design of value added data abstractions and the incremental development information integration applications.

The Distributed Oceanographic Data System (DODS) [Gallager95] also deals with several heterogeneous repositories. Essentially it defines a data access protocol that includes both a functional interface to data systems and a data model for representing data on these systems. It is designed to integrate already existing user applications and resource management systems. DODS models data analysis programs as some body of user written code linked with one or more API libraries that present specialized interfaces through which data access is achieved. User programs are split at the library interface, and by adding suitable interprocess communication layers, a client-server architecture is introduced to hide the differences of the underlying data sources and provide the user transparent access to this data. Calls to an API are forwarded to DODS data servers, which are implemented as web servers. Each data set is referred to via a URL, which contains the name of a CGI script to be executed when a request arrives. Depending on the request type, the server returns a textual description of the data set contents or the values of data variables in binary form. The primary role of the textual descriptions is to provide a client library with information concerning the operations that can be applied to data and the way binary data is to be decoded. This information is parsed, at run time, to determine the invocations and data conversions, which need to be done for each API call. Users via a standard browser can also view it. Hence, DODS adopts a similar approach for introducing metadata information as proposed in our architecture. The main difference is that the DODS supports a point-to-point communication between applications and data repositories, so that aggregation and consolidation of data is performed by the application rather than by mediators. However, the primary goal of DODS is to reuse a wide range of legacy analysis and data processing applications that are already implemented for local data. In the THETIS system, only a few applications exist so that the cost of rewriting code to take advantage of mediators is acceptable.

Other federated architectures for environmental information systems have been proposed, that implement the middleware layer using a combination of WWW and CORBA technology [Koschel96]. CORBA is used to deal with the limitations of WWW, namely statelessness and the inability to query more than one source at the same time. Browser requests are sent to web servers acting as CORBA clients, which in turn access the underlying repositories, perform the desired data manipulation and processing functions, and return results back to the browser as HTML documents. Communication among the CORBA client and CORBA server objects, hiding the internal implementation of repositories, is achieved via external C function calls with embedded SQL. The entire localization and invocation process of the middleware objects is handled transparently by the CORBA system. Hence, in this case, CORBA clients act as mediators. In our approach, however, mediator components are not mere translators; they have their own data models that can provide for new, powerful data abstractions with advanced querying capabilities that need not be invoked through functions. Moreover, we advocate transparent, instead of hidden, mediation. Since mediators carry with them metadata describing their capabilities, developers and scientists can interactively discover and use functionality that is available in the system. On the contrary, this is difficult to achieve for CORBA objects whose interface is known only to the programs that use them.

6. Conclusion

Integrated information systems have the capacity of spurring new economic growth and development that is unprecedented in the areas for which they are built. This is due primarily to their ability to provide *new services*. The way the Internet and the WWW have changed our ways of thinking and approaching growth, integration (in this term we compress the entire system architecture) is a new vehicle for economic growth in the information market place. It has taken a sizeable investment so far to build the data repositories (data, data models, and data analysis and interpretation models) and the expertise (people and organizations) to complete this task. Now we also have available a common communication network (Internet) and a common platform (WWW), in addition the expertise, and state of the art software tools to integrate. Thus it makes very good economic sense to efficiently put to use all of these by means of integration which at the same time has the capacity of making information easily available to a great deal more users than in the past.

Building systems that integrate scientific repositories to provide value-added information and to promote combination of resources is a difficult task. Legacy systems, computationally intensive simulation programs, and data intensive analysis and visualization tools mainly due to the complexity of scientific data, the inhomogeneous design of repositories and the limitations pose difficult problems. To address these problems, a hybrid architecture has been proposed that combines digital library techniques with mediation approaches and workflow systems. It allows various functional components to be added in a straightforward way, without modifying existing components and applications. Mediation is used to achieve incremental information integration and to provide end-user applications with the appropriate data abstractions. Digital library and workflow technology is used to support discovery and flexible combination of the existing system components, in a "mix and match" environment of mediators and programs. This architecture is currently being implement in the THETIS system, a coastal zone management environment for the Mediterranean Sea.

Acknowledgement

We are in debt to Anastasia Anastasiadh for her support in putting this paper together. We thank her for her invaluable contribution.

References

[BerkeleyDL] The Berkeley Digital Library project. http://elib.cs.berkeley.edu

[Buneman97] P. Buneman, L. Raschid and J. Ullman: "Mediator Languages- a Proposal for a Standard". In SIGMOD Record Vol. 26(1), pages 39-44,1997.

[CDF] The CDF standard. http://nssdc.gsfc.nasa.gov/cdf

[Chu93] W. Chu, A. Cardenas, and R Taira editors: "Proceedings of the NSF Scientific Database Projects", Boston, Massachusetts, February1993, AAAS, NSF.

[CMUDL] The Carnegie-Mellon Digital Library project. http://informedia.cs.cmu.edu

[DISCO] The DISCO system. http://rodin.inria.fr/disco

[DLI] The Digital Library Initiative. http://www.cise.nsf.gov/iis/dli_home.html

[Fine96] S. Fine and J. Ambrosiano, “The Environmental Decision Support System: Overview and air quality application”, Preprints, Symposium on Environmental Applications, January 28-February 2, Atlanta, GA, American Meteorological Society, 152-157, 1996. http://www.iceis.mcnc.org/pub_files/fine1996a.pdf

[French90] J.C. French, A.K. Jones, J.L. Pfaltz: "Scientific Database Management Final Report", NSF Workshop on Scientific Database Management, DSC Univ. of Virginia, Charlottesville, August 1990. http://www.cs.virginia.edu/~french/papers/sdbpapers.html

[Hanklin] S. Hankin, J. Davidson, and D.E. Harrison, “Web Visualization and Extraction of Gridded Climate Data with the FERRET Program”. http://www.pmel.noaa.gov/ferret/ferret_climate_server.html

[HDF] The HDF standard. http://hdf.ncsa.uiuc.edu

[HERMES] The Hermes system. http://www.cs.umd.edu/projects/hermes

[Houstis97] C. Houstis, C. Nikolaou, M. Marazakis, N. Patrikalakis, J. Sairamesh, A. Thomasic, “THETIS: Design of a Data Repositories Collection and Data Visualization System for Coastal Zone management of the Mediterranean Sea”, invited publication, D-Lib magazine, The Magazine for Digital Library Research, November 1997, http://www.dlib.org/dlib/november97/thetis/11thetis.html

[Hull97] R. Hull: "Managing semantic heterogeneity in databases: A theoretical perspective", PODS'97, May 12-14, Tucson Arizona.

[I3] The Intelligent Integration of Information Initiative. http://mole.dc.isx.com/I3

[IllinoisDL] The Illinois Digital Library project. http://dli.grainger.uiuc.edu

[INFOSLEUTH] The InfoSleuth system. http://www.mcc.com:80/projects/infosleuth

[Gallager95] J. Gallagher and G. Milkowski, Data Transport Within The Distributed Oceanographic Data System, Fourth International World Wide Web Conference, December 11-14, Boston, Massachusetts, USA, 1995. http://www.w3.org/Conferences/WWW4/Papers/67/

[GARLIC] The GARLIC system. http://www.almaden.ibm.com/cs/showtell/garlic

[Karp95] P. Karp: "A Strategy for Database Interoperation", Journal of Computational Biology, Vol 2, No 4, 1995, pp. 573-583.

[DAVINSON95] S.B. Davidson, C. Overton and P. Buneman: "Challenges in Integrating Biological Data Sources: Journal of Computational Biology, Vol. 2, No 4, 1995, pp. 557-572.

[Koschel96] A. Koschel, R. Kramer, R. Nikolai, W. Hagg,, and J. Wiesel, “A Federation Architecture for an Environmental Information System incorporating GIS, the World Wide Web, and CORBA”, Proceedings Third International Conference/Workshop on Integrating GIS and Environmental Modeling, National Center for Geographic Information and Analysis (NCGIA), Santa Fe, New Mexico, USA, January 1996 http://129.13.100.24/usrr3/ftp/pub/www/Personen/hagg/publications/ncgia96_koschel.html

[KSI] The Knowledge Sharing Initiative. http://www-ksl.stanford.edu/knowledge-sharing

[Lee94] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen and A. Secret: "The Word-Wide Web", CACM, and 37(8), pp. 76-82, August 1994.

[Letovsky94] S. Letovsky and M. Berlyn: "Issues in the Development of Complex Scientific Databases", Proceedings of the twenty-seventh Annual Hawaii International Conference on System Sciences, Volume V, pp. 5-14, Maui, Hawaii, January 1994.

[Marazakis97] M. Marazakis, D. Papadakis, and C. Nikolaou: “The Aurora Architecture for Developing Network-Centric Applications by Dynamic Composition of Services”, Technical Report TR 213, FORTH/ICS, 1997.

[Maier94] D. Maier and D. M. Hansen: "Bambi Meets Godzilla: Object Databases for Scientific Computing", In James C. French and Hans Hinterberger, editors, Seventh International Working Conference on Scientific and Statistical Database Management, pp. 176-184,Charlottesville, VA, September 1994. IEEE Computer Society Press.

[MichiganDL] The Michigan Digital Library project. http://www.si.umich.edu/UMDL

[Millard96] K. Millard,"A study of the Information Requirements for Coastal Zone Management" Smith Systems Engineering and HR Wallingford report, conducted for the British National Space Centre, 1996.

[MANIFOLD] The MANIFOLD system. http://www.research.att.com/~levy/imhome.html

[NetCDF] The NetCDF standard. http://www.unidata.ucar.edu/packages/netcdf

[Nikolaou97] C. Nikolaou, M. Marazakis, D. Papadakis, Y. Yeorgiannakis, J. Sairamesh, "Towards a Common Infrastructure for Large-Scale Distributed Applications". Conference proceeding EuroDL 97. http://www.ics.forth.gr/~maraz/euroDL97.ps

[NOAA97] NOAA: "Coastal Ocean Data Workshop Final Report", Harbor Branch Oceanographic Institution Fort Pierce, Florida, March 11-13, 1997. http://www.nodc.noaa.gov/coast/fr.html

[Patil92] R. Patil, R. Fikes, P. Patel-Schneider, D. Mckay, T. Finin,T. Gruber and R. Neches: "The DARPA Knowledge Sharing Effort: Progress Report", In Charles Rich, Bernhard Nebel, & William Swartout, Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, Cambridge, MA, Morgan Kaufmann,1992.

[StanfordDL] The Stanford Digital Library project. http://www-diglib.stanford.edu

[SIMS] The SIMS system. http://www.isi.edu/sims

[TSIMMIS] The TSIMMIS system. http://www-db.stanford.edu/tsimmis

[UCSBDL] The Santa Barbara Digital Library project. http://alexandria.sdc.ucsb.edu

[Ullman97] J. Ullman: "Information Integration using Logical Views"ICDT'97 Delphi's, Greece, pages 19-40, January 1997.

[WaikatoDL] The Waikato Digital Library project. http://www.cs.waikato.ac.nz/cgi-bin/nzdlbeta/gw

[Wiederhold92] G. Wiederhold: "Mediators in the architecture of future Information Systems", IEEE Computer, and pages 38-49, March 1992.

[Wiederhold96] G. Wiederhold (editor) : "Intelligent Integration of Information", Kluwer Academic Publishers 1996.