Relatório Técnico

Sharing Scientific Experiments and Workflows in Environmental Applications

Maria Cláudia Cavalcanti
Marta Mattoso
Maria Luiza Campos
Eric Simon
François Llirbat

NCE - 18/2000

Núcleo de Computação Eletrônica

Universidade Federal do Rio de Janeiro

Sharing Scientific Experiments and Workflows in Environmental Applications

Maria Cláudia Cavalcanti[α], *Marta Mattoso*[α], *Maria Luiza Campos*[β], *Eric Simon*[δ], *François Llirbat*[δ]

{yoko,marta}@cos.ufrj.br, mluiza@nce.ufrj.br
{eric.simon, francois.llirbat}@inria.fr

[α] **COPPE Sistemas - UFRJ**

[β] **Departamento de Ciência da Computação - IM/UFRJ**

[δ] **INRIA**

Abstract

Environmental applications have been stimulating the cooperation among scientists from different disciplines. There are many examples where this cooperation takes place through exchanging scientific resources, such as data, programs and mathematical models. The LeSelect architecture supports environmental applications, where scientists may share their data and programs. We believe that besides programs and data, models, as well as experiments and workflows are scientific resources that need to be shared in environmental applications. Therefore, in this paper we propose an extension to LeSelect architecture that allows sharing of models, experiments and workflows.

***Keywords**: scientific workflows, mediation, model management, environmental application*

1 Introduction

Even though scientific experiments have traditionally evolved in isolation, nowadays scientists need to exchange their data, which are embedded in heterogeneous legacy systems. Moreover, scientists need to exchange not only data but also scientific models and their implementations (programs). Since scientific resources involve models, programs and data, integrating scientific applications can be considered a hard task. Besides models, programs and data, scientists also need to exchange their experience. Information about the applicability of a model can feedback its authors/users with more accurate model pre-conditions. Furthermore, to fully understand a model, the scientist may need to investigate previous case studies that successfully used that model. Therefore, monitoring scientific experiments is another important requirement for scientific applications, which demands some management mechanism.

A scientific experiment can be viewed as a flow of data transformations that starts from raw data and finally produces data with added scientific value. When a scientist builds a new experiment she (he) has first, to select relevant input data for the problem to be studied and then determine an adequate flow of program instances that can process the selected input data. Moreover, some scientists deal with empirical models, which imply reviewing previous experiments to choose the most relevant input data and/or program instances to solve their problems. Therefore, the user needs an experiment catalog system with query facilities.

In particular, environmental applications require the use of a large variety of information that is geographically distributed, multi-disciplinary, and managed by many different organizations. Information typically encompasses data of various kinds, scientific models that perform predictions and simulations, and the programs that implement these models. Ideally, a distributed information management system should enable scientists to publish (that is, make publicly available) their scientific data, models and programs. On the other hand, scientists and decision-makers should be able to search, select and manipulate published data, models and programs that are relevant for their experiments and decisions.

Therefore, some integration effort should bring together all these specialists allowing cooperation by sharing data and models, encompassing the already existing systems, where each group of specialists work, enabling their interaction. To accomplish this goal we need to provide solutions to three main problems: (i) how to deal with the distribution and heterogeneity of data and program sources; (ii) how to describe models; and (iii) how to monitor the distributed usage of models, programs and data.

Several technologies have been proposed to address those problems. We focused on Heterogeneous and Distributed Database Systems (HDDS), Model management systems (MMS) and Workflow management systems (WfMS). In this paper we propose the extension of an existing HDDS called LeSelect (Xhumari et al., 2000), specially developed to support environmental applications. The main idea of the extended architecture is to provide a better support for these applications, by allowing scientists to share not only their data and programs, but also models, experiments and scientific workflows. Despite the many HDDS proposals, LeSelect is unique in its features to handle environmental applications and our proposed extensions also represents an innovative contribution to these application areas.

The next section describes our motivating applications, which deal with environmental systems. Section 3 presents the technologies used to address the problem and also reviews the related works. Details on the architecture of LeSelect system can be found in the fourth section. The fifth section presents our main contribution, which is an extension to LeSelect's architecture, identifying enhancements needed to address environmental scientific applications. Finally, section 6 discusses development issues, commenting on some early results and future directions.

2 Environmental Systems

The inherent complexity of environmental systems is due to the number of elements and processes involved, and it can be addressed by specific disciplines, such as, geomorphology, climatology, geology, biology, meteorology, physics, chemistry, etc. Therefore, it is difficult to find a single environmental specialist, because a person rarely gets skilled in that many disciplines. Usually, what happens is a natural separation of the specialists, each one working on a slice of the same environmental problem. For instance, biologists work on bio-corrosion of oil pipes and oceanographers work on ocean stream behavior, but both may be involved on the same environmental problem: an oil spill from underwater pipes. They may be working at different agencies of the same company, or even in different companies, focusing on different aspects of the same problem. Essentially, these scientists work with scientific models, either developing or using them.

According to Hagget et al. (1967; as quoted by Christofoletti, 1999), a *model* is defined as a simplified abstraction of reality that presents, in a generic way, characteristics or important relations. There are different kinds of models: data models, physical models, scientific models, etc. Scientists may use all kinds of models, but in the scope of this work the focus will be on scientific models. These models can also be classified in sub-categories, such as: probabilistic, numerical, empirical, etc. Formulas, equations, inequalities, algorithms, graphics are examples of scientific model representations.

Scientists from different disciplines have their own set of models. However, when addressing environmental problems, required models are usually composed by linked sub-models (Scott, 1996), originally from different disciplines. Therefore, a group of specialists very often interact by sharing models and data. Once they are dealing with the same environmental problem, they may have to use data and models from each other. The biologist may need to use the ocean stream data and models in order to determine if some oil pipe might have generated an oil spill. On the other hand, the oceanographer may need to use oil or pipe samples' data and biologic models to determine the oil spill cause.

Exchanging scientific models and programs leads to the problem of how to manipulate them, i.e., describe, query and execute them. Describing models is not an easy task because of the significant variety and quantity of existing models. For the same reason, the user should have some model query facility. Moreover, publishing programs means to allow their remote execution, using data from elsewhere and generating new data that should also be published. Providing this fully distributed scenario may face some technological obstacles, such as programs that do not run on some platforms, where should the results be published, how to fit input data into programs, among others.

Environmental information sources are quite heterogeneous with respect to their data processing capabilities, and their semantic meanings. First, data sources can be as varied as: relational or object-oriented databases; files; spreadsheets; web sites; or integrated application packages. Consequently, access to data is also diversified, ranging from standard languages like SQL or OQL to specific protocols and APIs. Data can also be generated on-demand by sophisticated data simulation models of physical, or biological processes and data processing techniques. In this case, input data must be provided in a specific format that depends on the implementation environment of the models. The heterogeneity of data processing capabilities leads to uniform and adaptive distributed access mechanisms.

Typical examples of environmental applications are described in the following sections.

2.1 Weather monitoring

A typical example of model and data sharing can be found in (Ailamaki et al., 1998), where soil specialists used meteorological models and data to build a composed model for preventing overnight frost damages in cranberry bogs (see Figure 1). At least three sub-models are used regularly to monitor temperatures. The first one provides a 24h forecast of the atmosphere temperature. The second one, uses this output, and generates the temperature forecast for 25 meters above the vine locations. Finally a third one takes this input and generates the temperature over the canopy level.

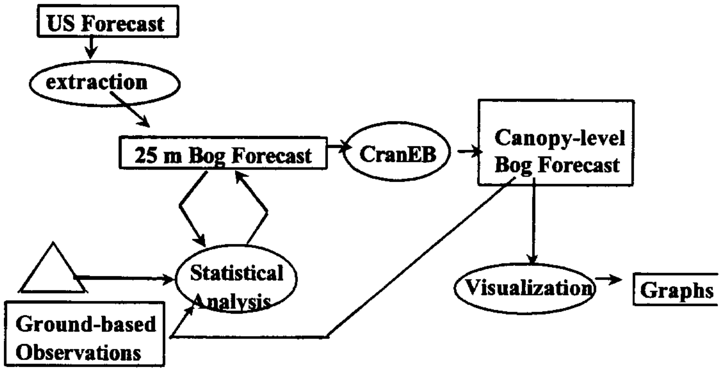

Figure 1: The cranberry workflow (Ailamaki et al., 1998)

2.2 Pollution control

Another example of model sharing can be seen in the DECAIR Project (Llirbat et al., 1999) where scientists aim at providing air pollution models with good quality input data derived from satellite data. The DECAIR application requires the collaboration of two kinds of scientists: those specialized in air quality modeling and those specialized in satellite image analysis. One difficulty of the project is to give the different air quality models high quality "EO-processed" satellite data and to automatically enforce accuracy and freshness of these data. Satellite images may be obtained from various remote sources. Moreover, depending on the quality of these images and depending on models' requirements, various treatments or programs have to be performed. Figure 2 shows a typical dataflow of DECAIR aimed application.

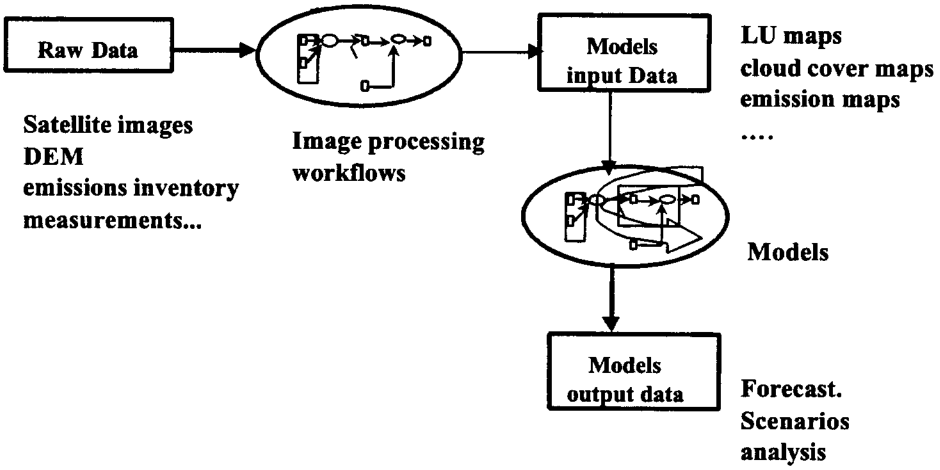

Figure 2: Pollution control workflow (Llirbat et al., 1999)

2.3 Bio-corrosion Monitoring

Bio-corrosion scientists main task is to identify bacteria as the main cause of corrosion events. Either the observation of a possible sign of bio-corrosion, a prevention study or even a simple investigation may start a new case study. First, scientists collect water, soil or pipe samples from the region under investigation. Then, laboratory analyses provide numerical data sets from these samples, such as chemical components' indexes. These data sets are then interpreted or analyzed by means of scientific models in order to derive new data, or some useful conclusion, such as: "there is evidence of a certain type of bacteria", "there is no evidence of a certain type of bacteria, some other type should be checked" or "re-sampling is needed".

Even though data may be continuously collected by distributed sensors. This is not always true for all case studies. Raw data should be invariably treated by data cleaning programs, which are based on mathematical models. Figure 3 shows a generic dataflow of a typical case study.

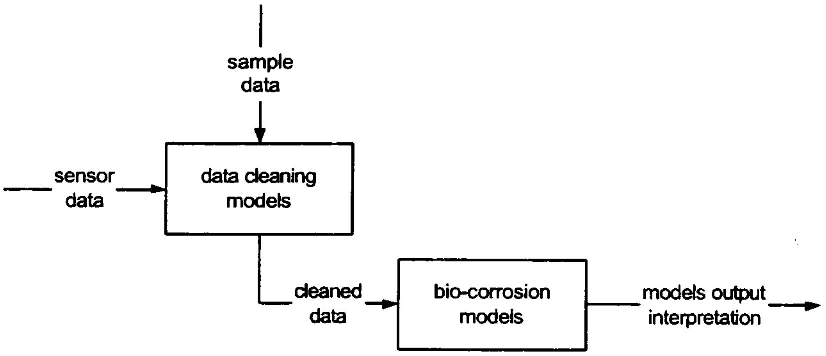

Figure 3: Biophenomena monitoring workflow

3 General Approach

There are several technologies that have been proposed in the area of databases, information systems, and cooperative information systems that can be useful to support the development of environmental scientific applications. Such technologies have addressed some of the main problems identified in these applications: heterogeneous and distributed database systems (HDDS) address the problem of integrating heterogeneous systems; model management systems address the problem of scientific models manipulation; workflow management systems address the problem of managing distributed processes, such as experiment processes. If combined, these technologies may constitute an adequate solution.

3.1 Mediation

So far, several mediator-based HDDS have been proposed: Himpar (Pires, 1997), Disco (Tomasic et al., 1998), Tsimmis (Garcia-Molina et al., 1997) and Garlic (Carey et al., 1995). The concept of information mediation, initially presented in (Wiederhold, 1992), is one of the most important contributions to the HDDS. It consists of defining an intermediate layer between information sources and applications. This intermediate layer provides an integrated view of information from queries without having to physically integrate data sources. The main advantages of the information mediation concept are: (i) to provide an integrated uniform access point to distributed and heterogeneous data sources, (ii) to provide logical and physical independence between applications and data sources in order to help the evolution of applications and respect the autonomy of the sources, and (iii) the ability to provide integrated information with added value by exploiting specific knowledge on a given application domain.

3.2 Model management

Model Management Systems (MMS) were developed to support modeling activities. Even though Decision Support Systems mainly drove their development for business applications, they are also useful for environmental scientific applications. In general, building a model involves combining models or deriving new ones out of a collection of data or other models. MMS architecture (Guariso et al., 1996)(Banerjee and Basu, 1993) includes functionality to provide model design, description, query and execution. Model classifications and description frameworks were also proposed for such systems (Banerjee and Basu, 1993(Benz and Hoch, 1999)(Gabele et al., 1999). However, most of the systems proposed so far have limitations, such as no remote model execution, no model query facility, and limited model description (aiming restricted areas). Environmental applications need systems that overcome such limitations, i.e., systems that provide, for instance, remotely model enacting, multi-discipline model descriptors and query facilities.

3.3 Workflow management

A workflow can be defined as a set of interrelated tasks. A scientific experiment can be viewed as a workflow whose tasks are program instances that are running against scientific data input produced by a previous task. These workflows are called scientific workflows (Singh and Vouk, 1996)(Weske et al., 1996). Therefore, a scientist could use a Workflow Management System (WfMS) to describe, implement and monitor shared experiments. However, conventional WfMSs need some adjusts in order to accommodate the scientific community.

Three generations of Workflow systems were identified in (Hsu and Kleissner, 1996). The first generation systems encoded all the control flow (business processes) within the applications. Then, the second-generation systems represent workflow processes explicitly. However, workflows are still tightly coupled to the application, in the sense that these workflows are strictly used by one application, such as document routing systems that handles only documents. Generic workflow systems belong to the third generation, where workflow is independent of specific applications. In these systems, the focus is on optimizing processes, enforcing business policies, and providing audit trails and history services.

Organizations see business processes as important as data manipulated by these processes. Initially, first generation systems had to deal with data sharing problems, which have been addressed by the database and distributed systems technologies. Nowadays, there is a need to share workflows as well. Workflow management across multiple organizations requires a distributed WfMS, which consists of multiple workflow engines, application servers, and ORB-style communication servers (Gillmann et al., 2000). Distributed and multi-domain workflow systems raise another issue: the dynamic reconfiguration of workflows. The focus and complexity of these systems may constitute the fourth generation of workflow systems. Scientific Workflows are an example that fits in this generation, not only for its distribution characteristic but also for being multi-disciplinary.

3.3.1 Scientific Workflow

From the workflow point of view experiments are composed by a series of steps or tasks, which obey a certain procedural logic (precedence, loops, conditions and parallelism). In particular, the empirical nature of some experiments demands some sort of workflow tight control. Often, certain steps are not successful and have to be re-executed, leading to unexpected loops in the process. Moreover, environmental problems are usually complex and not known in advance, and hence tasks are frequently not predictable, which means that ad hoc workflows are a common practice.

In addition to WfMSs basic functionality, i.e., workflow specification, instantiation and execution, there are some extra facilities that should be considered in scientific workflows. First, scientific results should be disseminated and reused, demanding auditing facilities. This is because scientists learn from their past experiences, even if they ended up in errors. Thus, it is important to keep track of all the performed experiments, even if they have failed. Distribution and heterogeneity are also main characteristics of environmental applications; thus integration and interoperability are probably required facilities.

Evolution is another required facility for scientific workflows. Since scientific processes in general are not fully specified before they start, a scientist may decide to modify or skip steps of a workflow during its enactment. This change may simply consist of choosing an alternative program to implement a given step of the workflow. For example, various image analysis techniques can be used for a given image depending on its accuracy and the context in which the image was taken (e.g., meteorological conditions, and date). The choice of a given program instance usually depends on meta-information directly associated with the input of the program (e.g., meteorological conditions and date are meta-information). This meta-information is readily available for already existing data sets. However, it is not available for those data sets that have to be computed on-demand. In this later case, the meta-information is computed by the result of the execution of previous steps of the workflow. Thus, the choice of program instances has to be done incrementally, backward or forward, along the execution of the workflow. To support the dynamic instantiation of workflows, WfMSs need to provide a declarative language that expresses the relationship between programs and data. Then, from these expressions, it should be possible to infer which adequate program instances, and in their associated input data, can be possibly chosen to implement a specific task in a workflow.

Most of the time, WfMSs adopt a task-centric approach that is reflected by their architecture: they use a Database Management System (DBMS) to store the descriptions of tasks, and implement all workflow functionality in modules that run on top of the DBMS. However, in scientific workflows the description of processed data is as important as the description of tasks because the quality of data sets often impacts on the quality of data returned by a model run on these data sets. Since the quality of data generated along an experiment, influences the logic of the experiment, a WFMS for scientific workflow should also accommodate a data-centric approach. In (Ailamaki et al., 1998), the workflow is viewed as a web of data objects interconnected with active links that carry process description. In this proposal, the DBMS incorporates the WfMS functionality, providing benefits such as: reduced implementation effort, increased optimization opportunity and workflow management uniformity. However, even though this centralized architecture provides benefits, such as a unique access language and point of control, it might not be a good idea when considering distributed and heterogeneous environments, which is the case of environmental applications.

Another interesting work that considers scientific WfMSs can be found in (Weske et al., 1996). In this work the authors describe the WASA architecture, whose goal is to provide a supportive environment for data-intensive scientific applications. WASA's main contributions are the support for dynamic execution of tasks, by combining active and temporal database facilities, and the support for experiment re-usability and reproducibility, by means of the documentation and versioning facilities. Even though WASA can be seen as a generic architecture for scientific workflows, when considering environmental applications it is not complete, lacking facilities such as integration and interoperability.

4 LeSelect Architecture

In the last two years, Inria has been developing a system, called LeSelect, which is particularly appropriate to environmental applications. LeSelect is a middleware system, which implements a framework that facilitates the publication of distributed and heterogeneous data and programs (services), and provides common facilities to query published data and to invoke published programs (Xhumari et al., 2000). When considering existing mediator-based HDDS, LeSelect distinguishes itself because it provides the basic functionality for implementing environmental applications. LeSelect allows the sharing of scientific models by publishing programs that implement them. Therefore, scientists may run their experiments, by feeding these programs with remotely published data, and by using programs from multiple disciplines, which are served in sites over the Internet.

Figure 4 presents the LeSelect architecture. The intermediate layer between information sources and applications integrates information from multiple data sources without having to physically integrate them. In LeSelect, data from each data source are wrapped into a common relational model of data. This is done via a piece of code called a data wrapper, i.e., publishing information of a given type (e.g., HTML file, C program or database) requires creating a specific wrapper for it. Each data wrapper interfaces with a local mediator called LeSelect server, to form a publishing site, which is accessible from applications. When an application needs to access data from multiple data sources, it can connect itself to a LeSelect client, which provides a JDBC interface to access multiple publishing sites (LeSelect Servers) in a single SQL query. The facilities offered by the mediators and the wrappers enable the sharing of data without forcing each application to redundantly encode the data transformation and data processing parts.

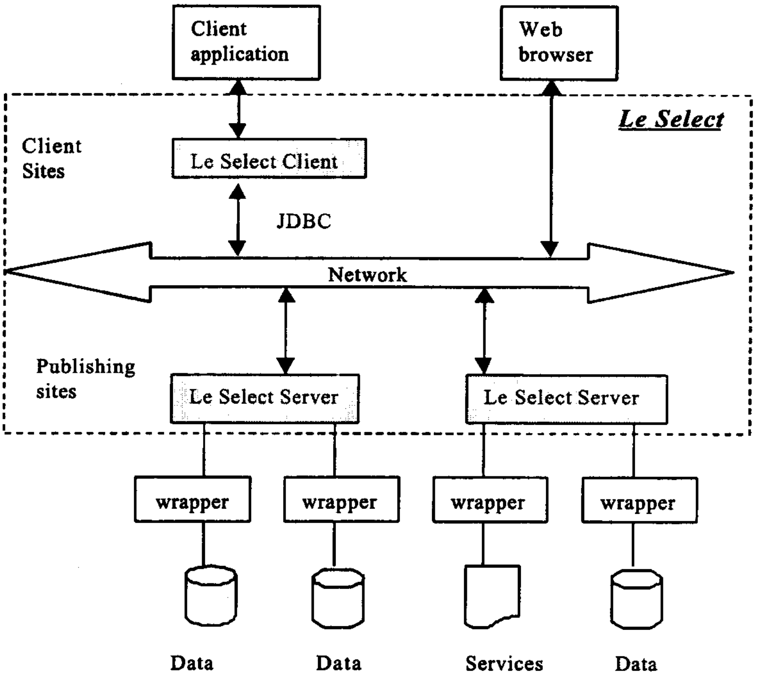

Figure 4: LeSelect Architecture (Xhumari et al., 2000)

LeSelect also enables sharing services, which are available in a specific source, via a particular kind of wrapper, which interfaces with a LeSelect server within a publishing site. A publishing site can be interfaced simultaneously with both data and service wrappers. On the other extreme of the architecture, a client application can invoke a given service that uses data from multiple publishing sites via a LeSelect Client.

Wrappers manage metadata by providing a uniform representation of data, functions and programs with an extended relational model, and manage the execution of queries on local sources. The publishing mediator (LeSelect Server) maps global queries into local queries, each for a different wrapper, and a composition query for producing the final result. It also has a runtime system to integrate the results of local queries. Global queries are expressed in an SQL-like language, which allows invoking functions or programs on data sources.

Publication sites can be organized as a hierarchy. Thus, a publication site can include a wrapper to a virtual database schema whose query-based specification can refer to information published by other publication sites. In this case, the schema corresponds to an integrated view of information published by other sites. The major advantage of this architecture is that the process of information publishing is completely decentralized via the publication sites.

LeSelect's approach contrasts with previous information mediation systems such as Garlic, Disco, Himpar and Tsimmis, with respect to the integration policy. In these systems, publishing data at some site requires that a set of view definitions should be provided in some mediator located at another site. Their goal is to provide data transparency, which means hiding integration transformation details. When there are new data to be published, sometimes it is a difficult task for the publisher to reflect the changes into view definitions. LeSelect does not automatically provide full transparency of data distribution because when building distributed SQL queries, a LeSelect client references tables by their identifier, which contains the address of the publication site where the corresponding data have been published. However, the view definition service provided by LeSelect enables the publication of virtual derived data, i.e., views. Hence, queries over the views hide the physical distribution of the underlying data from which the views are defined.

LeSelect relies on well-established open standards for interoperability. Network communication between LeSelect components is assured via a CORBA protocol, although other means of communication are also possible. That is, JDBC statements between LeSelect components (clients or servers) are embedded into CORBA/IIOP messages.

Richer metadata can also be provided via LeSelect. Information such as the author and creation date of a published data set or program, the meaning of some values of a given data set column, the units of measures, etc., are examples of semantic metadata. LeSelect allows the attachment of this information to the published data set or program, by using XML format.

Publishing data and programs, running experiments using remote published data and programs, and publishing optional metadata related to published data and programs, are some of the facilities that distinguishes LeSelect as a good platform choice to implement environmental scientific applications. However, aiming at a broader solution we propose some extensions, which are described in the next section.

5 Extending LeSelect Architecture

This proposal aims to provide a generic solution for environmental scientific applications where multidisciplinary scientists can (i) share scientific data, programs, models, experiments and workflows (ii) monitor and collect scientific experiments and (iii) compose and analyze workflow instances.

Even though LeSelect fulfills many requirements of environmental applications, it may benefit from some extensions. An alternative on this direction would be to work on LeSelect's user interface. LeSelect users have to know where are the programs and data, i.e., which are the exact addresses of these programs and data. Elsewhere (Houstis et al., 1999) there is some ongoing work on providing a search engine where the users can freely search keywords over a list of indexed LeSelect servers. However, after identifying a program and its input data, in order to run this program on a given data set, the user has to compose a statement such as:

```
Job execute //cacuia.nce.ufrj.br/Kusnetzoval-0
  input data set is Select * from
   //www.cenpes.br/sample/cabiunas
```

LeSelect allows non-structured metadata about published programs and data to be also published. Those metadata may bring more semantics to these programs and data. However, the publisher may opt not offer them. We believe there is a need to provide other means for publishing more semantic metadata about those programs and data. A step in this direction could be to publish models as an abstraction of one or more programs, by using LeSelect's wrapper components.

Experiments are also an important resource to the scientific community that could provide more semantics to programs and data. Viewed as a log of a program run, an experiment describes it in terms of the location of the input/output data set and parameters' values. More information can be added, such as the author, date, status, and also the interpretation of the experiment. Based on published experiments, users can get an idea of the usefulness of programs, understanding how to use them and their input/output data sets. For instance, consider that the job example given before really took place at a LeSelect server site. Suppose a scientist named Dexter ran a program called `Kusnetzoval-0` that is published in site `cacuia.nce.ufrj.br`. Dexter used data published from elsewhere (`www.cenpes.br`) as its input. When the program finished, it had the input data set transformed into some output data set, which is then published in the same site where the program resides. According to the relational LeSelect publishing style, this experiment could be expressed in terms of the following relations, which are expressed using the metadata framework presented in (Galhardas et al., 1998):

Metadata

```
predicates(experiment)
attributes(experiment,program, Url)
attributes(experiment,parameters-values, Array of Real)
attributes(experiment,input-datasets, Array of Url)
attributes(experiment,ouput-datasets, Array of Url)

predicates(exp-interpretation)
attributes(exp-interpretation, author, String)
attributes(exp-interpretation, date, Date)
attributes(exp-interpretation, status, String)
attributes(exp-interpretation, interpretation, String)
```

Facts

```
experiment("http://cacuia.nce.ufrj.br/leselect/kusnetzoval-0",
          (3.5; 6.1),
          ("http://www.sat.gov/leselect/sample/cabiunas"),
          ("http://www.site.com/leselect/criticalAreas.img")
          )
exp-interpretation("Dexter", 10/05/2000, "S",
                   "Absence of dangerous areas")
```

Two or more scientific experiments can be interconnected, i.e. they may take part on a previously conceived sequence. Let us consider two experiments, e_1 and e_2, where the output data set of e_1 is used as the input data set of e_2. Thus, there is an experiment e_3 that results from the composition of e_1 and e_2. However, if the experiments e_1 and e_2 took place on different LeSelect servers, it would be difficult to realize that they belong to a more complex experiment, named e_3. Figure 5 shows an example of a complex experiment.

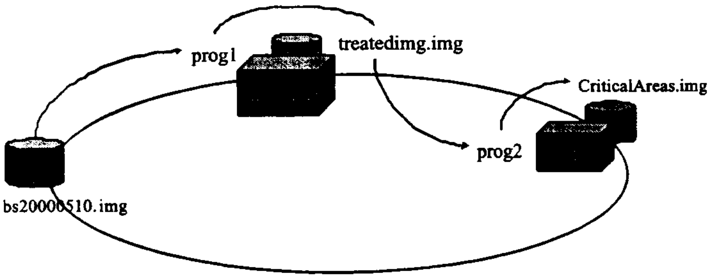

Figure 5: Complex experiment example

Publishing such complex experiments via LeSelect demands more than just a wrapper functionality. A new component would be added to the architecture LeSelect as the responsible for collecting simple experiments and for composing complex ones. After identifying these experiments, the same component could be responsible for publishing them over the Internet. Figure 6 shows how the new component (Collector) would be placed in the architecture LeSelect. Hence, complex experiments can be seen as workflow instances. After analyzing the frequency of some experiments, the publisher may wish to publish their related workflow schemas.

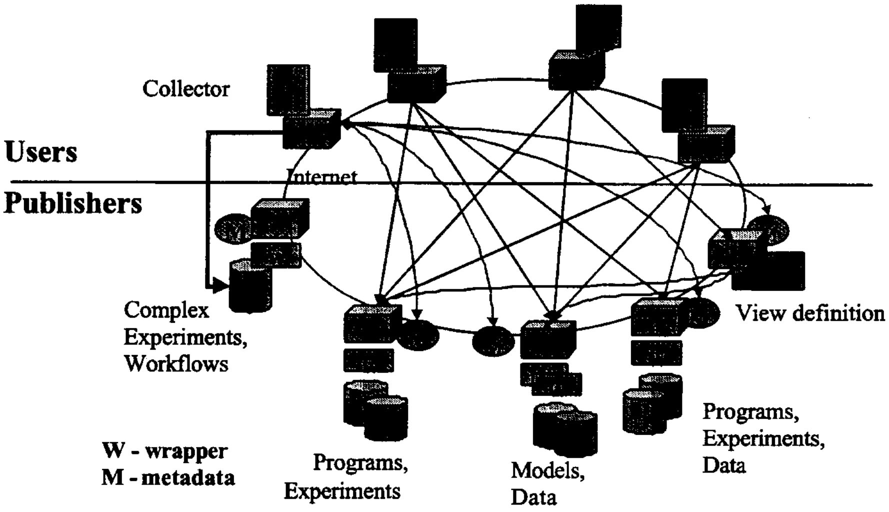

Figure 6: Extended LeSelect Architecture

In (Llirbat et al., 2000) the authors propose a formal model for describing workflows, which includes model and experiment descriptions. Aiming at workflow schema descriptions, the authors provide a program type description that is used to represent each program type involved in the workflow schema. The program type description provides information on program's input, output and constraints. We believe that with a few changes this formalism could be used to describe models. Furthermore, the authors also provide a complex experiment descriptor called experiment snapshot, which represents an instance of a workflow schema. Experiment snapshots represent complex experiments step-by-step, specifying, which simple experiments are finished and which are not.

6 Conclusions

This paper proposes an architecture extension aiming at environmental scientific experiments. LeSelect architecture focuses on environmental applications, thus it is used as the basis for the proposal. Components of the extended architecture, such as model and experiment wrappers, are under development.

A generic proposal is presented, however a real case study on Bio-corrosion scientific application has been used to support our work on the elicitation of requirements. In (Altoé et al., 2000) there is a description of the system used as a case study, which is called System for Interpretation and Modeling of Bio-phenomena (SIMBio). It aims at supporting Bio-corrosion scientists on identifying bacteria as the main cause of corrosion events. Based on data collected through this system, our final goal would be to publish their data, programs and models via a prototype of the proposed architecture.

Although human aspects are a central issue in workflow-based applications, they were not treated in this paper. Considering the complexity of environmental applications, we have concentrated our efforts in more objective aspects. However, we recognize that is an important research direction, and we have already included human aspects in our future work.

Metadata plays a critical role in environmental applications, which are built over multi-disciplinary systems. Another interesting research direction would be to extend LeSelect architecture to provide a better metadata support. Different approaches on providing metadata standards aim to support interoperability between different vendors' products by defining metamodel standards for a core set of metadata types (OMG, 1997)(MDC, 1999). Another related issue concerns addressing semantic heterogeneity. Ongoing research points to achieving a common agreement on the terminology used in a multi-domain shared area. For each domain, a pre-defined ontology is defined, composed by a vocabulary of terms and a specification of their relationships, forming a semantic net (Wiederhold, 1994).

7 References

Ailamaki, A., Ioannidis, Y., Livny, M., "Scientific Workflow Management by Database Management", In: **Proc. of 10th International Conference on Scientific and Statistical Database Management**, pp. 190-199, Capri, Italy, July, 1998.

Altoé, F., Campos, M.L. - '**SIMBio - Sistema de Interpretação e Modelagem de Biofenômenos**", Technical Report, NCE, UFRJ, 2000.

Banerjee, S., Basu, A. '**Model Type Selection in an integrated DSS environment**", Decision Support Systems, vol. 9, pp.75-89, 1993.

Benz, J., Hoch, R. - '**ECOBAS - Model Interchange Format Reference Manual**" - ECOBAS_MIF version 3.0, http://dino.wiz.uni-kassel.de/ecobas/syntax_mif/syntax2_mif.ps, 1999.

Christofoletti, A., '**Modelagem de Sistemas Ambientais**", Editora Edgard Blucher, 1999.

Gabele, T., Benz, J., Hoch, R. - "Standardization of model documentation: Usage of ECOBAS model documentation system - a short introductory manual", in **ECOMOD Newsletter**, ISEM, also in http://ecomod.tamu.edu/ecomod/isem.html, June1999.

Galhardas, H., Simon, E., Tomasic, A., "**A Framework for Classifying Scientific Metadata**", Proceedings of AAAI'98, 1998

Garcia-Molina, H., Papakonstantinou, Y., Quass, D. et al. - "The TSIMMIS Approach to Mediation: Data Models and Languages" In **Journal of Intelligent Information Systems**, http://www-db.stanford.edu/tsimmis/publications.html, 1997.

Gillmann, M., Weissenfels, J., Weikum, G., Kraiss, A. - "Performance and Availability Assessment of the Configuration of Distributed Workflow Management Systems", to appear in Proc. of 7th **International Conference on Extending Database Technology**, EDBT'2000, Konstanz, Germany, March, 2000.

Guariso, G., Hitz, M., Werthner, H. "**An Integrated Simulation And Optimization Modelling Environment For Decision Support**", Decision Support Systems, Vol.16, pp. 103-117, 1996

Hagget, P., Chorley, R. J. - "Models, Paradigms and New Geography", in **Models in**

Geography, London, Methuen & Co., 1967. *Apud* [1].

Houstis, C., Nikolaou, C., Lalis, S., Kapidakis, S., Christophides, V., Simon, E., Tomasic, A. - "Towards a Next Generation of Open Scientific Data Repositories and Services" - **In CWI Quarterly**, Vol. 12, No.12, Special Issue on Digital Libraries, Amsterdam, June, 1999.

Hsu, M., Kleissner, C. - "ObjectFlow: Towards a Process Management Infrastructure", In **Distributed and Parallel Databases**, vol. 4, pp. 169-194, 1996.

Hull, R., Llirbat, F., Simon, E., Su, J., Dong, G., Kumar, B., Zhou, G., "Declarative Workflows that Support Easy Modification and Dynamic Browsing", In: **Proceedings of ACM International Joint Conference on Work Activities Coordination, WACC'99**, February, San Francisco, CA, USA, pp.69-78, 1999

Llirbat, F. et al., **DECAIR** - Technical Report, INRIA, 1999.

Llirbat, F.; Simon, E, Berroir, J. - **Specifying Scientific Experiments By Means of Declarative Workflows** - to be published, 2000

MDC (Metadata Coalition) - **Open Information Model**, version 1.1, August, 1999

OMG (Object Management Group) - "**Meta-Object Facility**" - OMG TC document cf/97-01-01, Linnæus Project, DSTC, January, 1997

Pires, P., "**HIMPAR, Uma Arquitetura para Interoperabilidade de objetos Distribuídos.**" M.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997.

Scott, E. M. - "Uncertainty and Sensitivity Studies of Models of Environmental Systems", in **Proceedings of 1996 Winter Simulation Conference**, (eds.) J. Charnes, D. Morrice, D. Brunner and J. Swain, San Diego, CA, USA, December, 1996.

Singh, M.P., Vouk, M.A. - "Scientific workflows: scientific computing meets transactional workflows," **Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems** : State-of-the-Art and Future Directions, Univ. Georgia, Athens, GA, USA; 1996, pp. 28-34.

Tomasic, A., Rachid, L., Valduriez, P., "A Data Model and Query Processing Techniques for Scaling Access to distributed Heterogeneous Databases in Disco." In: **IEEE Transactions on Knowledge and Data Engineering**, Volume 10, Number 4, July 1998.

Weske, M., Vossen, G., Medeiros, C. "**Scientific Workflow Management: WASA Architecture and Applications** ", Schriften zur Angewandten Mathematik und Informatik 03/96-I, Universität Münster, 1996.

Wiederhold, G. "Mediators in the architecture of future information systems", In: **IEEE Computer**, v.25, pp. 38-49, 1992.

Wiederhold, G., "Interoperation, Mediators and Ontologies" In: **Proceedings International Symposium on Fifth Generation Computer Systems (FGCSOB94)**, Workshop on Heterogeneous Cooperative Knowledge-Bases, Vol.W3, pages 33-48, ICOT, Tokyo, Japan, Dec. 1994

Xhumari, F., Amzal, M., Manolescu I., Simon, E. - "LeSelect: a Middleware System for Publishing Autonomous and Heterogeneous Information Sources", Technical Report (to be published), INRIA, 2000.