Workflow Management in Geoprocessing Applications*

Mathias Weske, Gottfried Vossen†
University of Muenster, Germany

Claudia Bauzer Medeiros, Fatima Pires‡
University of Campinas, Brazil

Abstract

This paper presents a system which is being developed in the University of Münster to support scientific application environments. This system – WASA – is based on taking advantage of workflows to document and monitor the execution of scientific applications. A geoprocessing application is used throughout the paper to illustrate and justify the specificity of the problem and our proposed solution.

1 Introduction

Workflow management aims at modeling and controlling the execution of processes in business applications [9, 4, 10]. It has gained increasing attention recently, since it allows combining a data-oriented view on applications, which is the traditional one for an information system, with a process-oriented view, in which collections of activities, their interactions and exchanges are modeled and supported. The exploitation of the workflow paradigm in geoprocessing applications, however, has rarely been studied yet; the goal of this paper is to remedy this situation. In particular, we will show, using environmental control and monitoring as a case study, how workflow management can prove useful, since it helps combine environmentalists' expertise on process modeling with their need for appropriate data management.

While a number of workflow management systems for business applications are already commercially available, systems for scientific applications exist at best as research prototypes. One goal of the WASA project [18, 11] is to give support to such applications. In [1] we outlined the main aspects WASA should support in order to allow its working in a GIS context – e.g., execution of partial workflows, or specifying a workflow from a case. In this paper, we investigate planning processes in geoprocessing applications, formalize these processes as workflows, and show how they can be supported by a prototypical workflow management system under development in the WASA project.

The remainder of this paper is organized as follows. Section 2 gives an overview of the life-cycle of design and development of typical applications in geoprocessing. Section 3 shows how workflow management can be exploited to support these activities. This section uses a real-life example as motivation to the paper, which concerns the development of a map of fire risks for a given region. Section 4 describes the WASA prototype and shows how it can be used in geoprocessing applications by instantiating the example in it. Section 5 presents conclusions and future work.

2 The Life-Cycle of Applications in Geoprocessing

There are different profiles of GIS users, varying from beginners to application designers. We here look at geoprocessing applications from the point of view of designers, i.e., people who are knowledgeable in the application domain (e.g., biologists, ecologists, soil scientists) and, at the same time, know how to take advantage of available computational tools embedded in the GIS.

More specifically, we are interested in problems related to the development of applications in the environmental area (e.g., monitoring); we will refer to these applications as *geo-applications*, denoting the fact that they deal with *geo-referenced* data. Users who design geo-applications currently take advantage of a variety of computational tools which help spatial analysis and cartographic presentation, usually embedded within some GIS.

The design of a geo-application, from a software engineering viewpoint, presents some particularities (e.g., see [6]). First, it is both data and process-intensive. Second, designers seldom worry about reuse (of data or of code), being under the impression that applications are developed on a case-to-case basis and therefore reuse is impossible. Indeed, the specification of each application is tailored to a specific region of the Earth's surface, for a given set of goals, to be acted upon by a distinct group of agents. This makes each application unique in the designer's mind. Third, in spite of designers' expertise, they do not often worry about documenting their application. Thus, even when reuse would be possible, it cannot be enforced for lack of documentation. As we shall see, WASA helps fill this gap, thereby fostering reuse and consequent savings in time and money.

*This work was partially supported by BMBF Germany and CNPq Brazil, within a bilateral cooperation on Database Technology and Knowledge-Based Systems, under Grant No. 30.3.I1A.6.B

†Lehrstuhl fuer Informatik, University of Muenster, Steinfurter Strasse 107, D-48149 Muenster, Germany. E-mail {weske,vossen}@helios.uni-muenster.de

‡IC - UNICAMP - CP 6176, 13081-970 Campinas SP, Brazil. E-mail cmbm@dcc.unicamp.br. Research partially funded by CNPq, PROTEM, FAPESP and the SAI PRONEX grants.

ACM GIS '98 11/98 Washington, D.C., USA
© 1998 ACM 1-58113-115-1/98/0011...$5.00

In order to do that, it is important to understand designers' work procedures and state them in the appropriate software engineering terms. From a macro point of view, the life-cycle of a geo-application can be considered in five major steps: definition of objectives; real-world modeling; inventory (geographic database specification and creation); implementation; and monitoring [13, 15]. These steps closely match those needed to develop decision-support applications. The *definition of objectives* consists in specifying what the goals of the application are to be, and what phenomena should be considered in order to attain the goal. Real world *modeling* comprises *data* and *process modeling*. Data modeling corresponds to selecting, abstracting and generalizing the entities of interest to the user, showing how they vary through time. Process modeling refers to constructing a mathematical model that describes operations involving the stored data representations, and includes the simulation of natural phenomena. The output of this activity directs the *inventory* phase, which consists in the definition of the *geographic database*, as well as specification of the function libraries and model parameters that are to be used together with data stored in the database. *Implementation* concerns the use of these databases and libraries, combining functions and producing new data, either directly by means of programs or, more frequently, using a GIS [12]. The result of the implementation is usually a set of maps and tables, which will be used by experts to determine policies to follow in some situation (in our example of the next section, how to better prevent fire risks). Finally, the *monitoring* phase concerns checking policy implementation (to both ensure its correct implementation and to correct possible errors in previous phases).

These steps constitute the basis of an environmental application design methodology, which is now being used with success in helping users specify their applications in order to maximize data reuse. As shall be seen in the next section, these steps can naturally be modeled by *workflows*, and can then be supported automatically by a workflow management system.

3 Exploiting Workflows in Geo-applications

3.1 Modeling Geo-applications as Workflows

Workflow management combines influences from a variety of disciplines, including cooperative information systems, computer-supported cooperative work, groupware systems, and active databases. Its major application area has so far been in the business field [10], as the *modeling* of business processes has become a strategic goal in many enterprises. Once the modeling and specification of business processes has been completed, they can be verified, optimized, and finally brought onto a workflow management system. Besides the traditional field of business applications, new domains emerge, among which scientific ones play a major role [18, 1, 8].

Workflow models are a formalism that support process specification [9, 10]. In general, workflows consist of a set of related *activities* which are executed by processing entities [16]. Activities are units of work as perceived by the modeler. Each activity includes a description of the data used and generated by the activity. *Processing entities* which can perform tasks may be humans or software systems, e.g., mailers, application programs, or database management systems.

Consider now the sequence of steps described in section 2

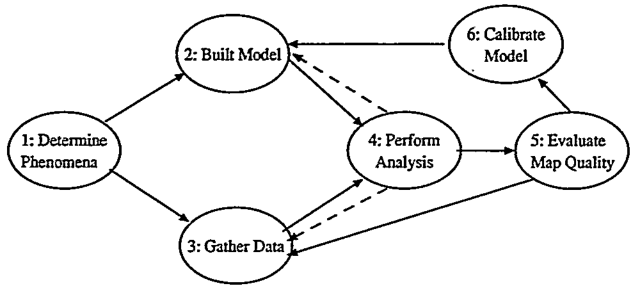

Figure 1: Top-level geo-workflow.

under the workflow paradigm. First, this high level description can be seen as a workflow specification where activities are expressed as a sequence of discrete steps with clear procedural order, well-determined input and output, and specific execution constraints. Second, many of these activities (e.g., inventory) demand intervention and cooperation of several (human) agents, whereas others can be completely automated (e.g., when GIS functions are invoked). Third, the output of each step can be used as input to subsequent steps, or be a feedback to re-execute some previous step, reflecting the fact that the modeling and interpreting natural processes is a never-ending activity. ¿From now on, we refer to the workflows describing these activities as *geo-workflows*, to differentiate them from other types of workflow.

The first (Definition of Objectives) and last (Monitoring) phases (see Section 2) are essentially human activities. However, human activities can also be monitored by a workflow facility, as human actors that perform them can signal to the computer that they are executing some task, which in turn helps documenting the entire development process.

Geo-workflows differ from standard workflows in the type of data (geo-referenced), analysis operations (geo-region sensitive) and constraints they must handle. Another distinguishing characteristic of geo-workflows lies in the *agents* and *roles* involved in their execution. The handling of geo-referenced data is essentially multidisciplinary, and therefore the execution of these workflows is frequently conducted in a collaborative way.

3.2 A Geo-Workflow Case Study

We now analyze a specific instance of a geo-workflow, adapting an experiment which was conducted to evaluate fire risks in the county of Piracicaba, Brazil [3]. The geographic area concerned is a natural preserve belonging to the Forestry Research Institute of the State of São Paulo. The county of Piracicaba is densely populated, highly industrialized, and surrounded by sugar cane plantations. All these factors contribute to creating fire risks in the preserve. The experiment was dedicated to producing a "fire risk map", i.e., a map classifying the preserve into regions according to the probability of presenting a fire hazard.

Figure 1 shows the top-level geo-workflow that was used to specify application, and where Activities 1 through 6 cover the five phases of the life-cycle. This geo-workflow originated from the need to solve the problem to "determine fire risk", for the area described by "natural preserve at 22^0 S, $47^0 32'$ W". The statement of the problem can be considered to be the event which triggered the execution of the geo-workflow. The description and location of the area of study are essential for determining what variables should be considered in the experiment. Activity 1 was performed by a group of experts and produced the specification of the phenomena that should be analyzed for this specific problem, time frame and geographic location: relief, land use, vegetation, hydrography, roads, and historical records of fires for the region.

Next, two activities were launched in parallel: *Model specification*, corresponding to Real-world modeling (Activity 2) and *data gathering* (Activity 3), corresponding to Inventory. *Data gathering* was performed in parallel by several teams of people, as often happens in geoprocessing activities. It consisted of checking for already available data sources and, if necessary, performing field work to collect additional data.

Model specification is a highly empirical activity, and produces a system of equations which mathematically describe the dynamics of the real world. Model building for this type of problem is achieved on a trial-and-error basis, by answering the following questions [7]:

- What are the spatial units considered and their interactions?
- How do relative sizes and locations of these units affect the variables and ecosystem factors considered?

The model built for the experiment under consideration consisted of a sequence of weighted average calculations, to combine the different data sources defined during Activity 1. The weights correspond roughly to the importance a given factor would have in fire propagation – e.g., fire risk increases with proximity to humans and decreases where vegetation is more dense; areas with native vegetation are less prone to fires than areas which have been replanted.

Activity 4 (*Analysis*) corresponds to the beginning of the implementation phase, and consists of computing the model using the data gathered. In the example, this computation used the *map overlay* method, in which each data source is transformed into a map with eventual rechecking (as shown by the feedback dotted lines in Figure 1). The result of the analysis activity was a fire risk map, which was the input to Activity 5 (*Assess map quality*), already part of Monitoring of the life-cycle. In this specific case, the map was considered to be acceptable within the specified error margin, and therefore the application development was concluded. Quality assessment consists of checking the result against some control data. In the problem studied, it was done visually by a team of experts. In more complex cases, this would be done automatically by again invoking GIS functions, or spatial software/geo-statistics libraries.

Very often, Activity 5 indicates that the analysis result is not satisfactory and therefore Activities 2 or 3 may have to be executed again. The re-execution of Activity 3, in particular, is preceded by Activity 6 *Model calibration*, which consists of making adjustments to the model (e.g., by changing parameter weights)

Before proceeding to a refinement of the workflow, let us examine the agents and roles involved in the execution of this geo-workflow. Data sources and files are discussed in Section 3.3 in the refinement of Activity 3. There were three kinds of human actors: technicians, experts on environmental modeling, and fire fighters. The latter participated in Activities 1 and 5 in a consulting role. Technicians were basically employed in data gathering and in running programs in the analysis phase. Environmental experts played consulting and specification roles in all activities but Data gathering (activity 3). Computerized procedures were used as actors in Activities 3 and 4.

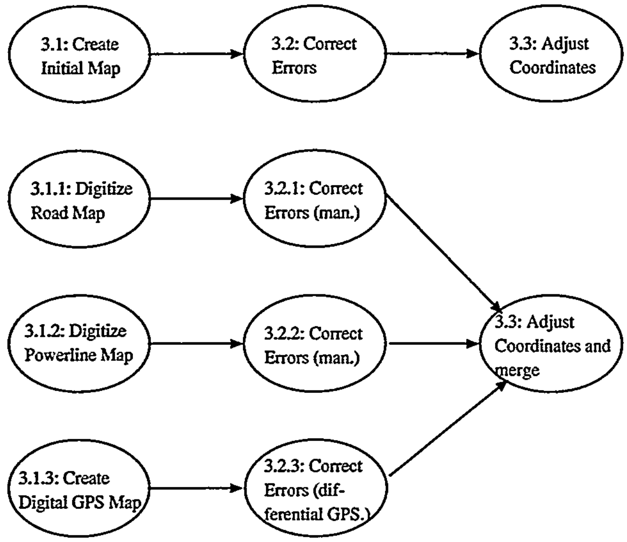

Figure 2: Refinement of Activity 3 and details of eliciting road map file.

The execution of each activity involves choosing among several acceptable alternatives. For instance, Activity 5 might have been executed automatically by programs, rather than visually by human experts. Therefore, this same workflow might have had different executions (instantiations). In WASA, this would characterize storing distinct *geo-workflow models* for the same experiment.

3.3 Refinements of the Top-Level Workflow

Figure 1 can be refined into a variety of sub-workflows. We will here indicate only how to refine Activity 3, but all others could have been equally decomposed. Data gathering (Activity 3) requires launching several independent geo-workflows, each dedicated to collecting data of a different nature. Figure 2 shows the refinement of this activity and its instantiation for production of the "road map" input data file.

For this type of problem, practically all data gathering tasks are subdivided into producing a basic map (Activity 3.1), correcting errors (Activity 3.2) and adjusting coordinates (Activity 3.3). Error detection is an integral part of spatial information processing. Understanding and limiting errors at this stage (data gathering) is fundamental to controlling the quality of the result (during Activity 5). The execution of these three sub-activities varies widely according to the data sources, scale, devices, etc.

In our case study, vegetation and hydrography were already available in digital media and thus did not need to go through all steps of gathering. The other data sources had to be created (e.g., scanning) in order to allow the application to run.

An example of a data file that was especially created for this experiment was the "road map" file. In this case, three different data sources were processed: highway paper maps (provided by the municipality), a power line paper map (provided by the local electric power company) and a pathways digital map (generated by walking or riding along existing small paths using a differential GPS). High voltage lines imply the existence of small paths directly underneath, that must always be kept clean of vegetation, in order to allow line repairs and maintenance checks. This is an example of a very common activity in geo-referenced data gathering – using a given data source (here, power lines) to derive another kind of data (pathways).

Experimental procedures are described at length in [3] and a more complete description of workflow refinement appears in [22]. We conclude this section by a few remarks. First, the procedures described are highly simplified, since our goal is to give an overview of a geo-workflow. Second, this type of experiment is highly dependent on the expertise of the researchers who define the initial parameters (data sources) and the relevant process model. This knowledge cannot be embedded "in" the workflow. As we will see next, in WASA it is stored apart using a mixture of knowledge base technology and textual documentation. Finally, most of the activities, especially 1, 2, and 5, are highly dependent on collaborative work. Again, this typically requires additional tools.

4 The WASA Contribution

As we said in the Introduction, computerized support for scientific environments has not yet come across an exploitation of workflow management, though such environments could profit considerably from this technology. The WASA environment tries to fill this need, by providing scientists with a workflow-based environment, whose goal is to support scientists document and develop their experiments, focusing on applications in the natural sciences and in laboratory environments.

4.1 The WASA Prototype

A first prototype of WASA has been implemented using Java and a commercial relational database system [19, 23]. This prototype has been tested on various cases. The workflow engine is the core part of the architecture; it aims at enhancing the flexibility of existing workflow management systems while providing a high degree of platform independence [18]. The term "flexibility" refers to the ability of users (or system administrators) to change workflow models while workflows execute (also known as *dynamic modification* [2]). Furthermore, the prototype supports flexible workflow modeling by allowing to reuse pre-existing component workflow models in multiple other workflow models. In the previous geo-workflow, this would mean re-using part of the specification in other applications (e.g, to define fire risk maps for other regions).

Loosely speaking, the WASA prototype consists of a workflow engine, a database server and workflow clients. It is based on a generic, layered architecture – the WASA architecture – shown in Figure 3. The architecture relies on the fact that scientific experiments, specified as workflows, are stored in the system's database as workflow models. Models are instantiated at each workflow execution. Essentially, this is a client/server architecture, where the server reads workflow models from the underlying database, controls the execution of workflows, and performs other important services like role resolution. Internally, it is composed of the workflow engine as core and the database server which accesses application data stored in the database. Both components are connected to the database by a JDBC interface, and the database contains workflow-related data (like workflow models and role descriptions) as well as application-specific data.

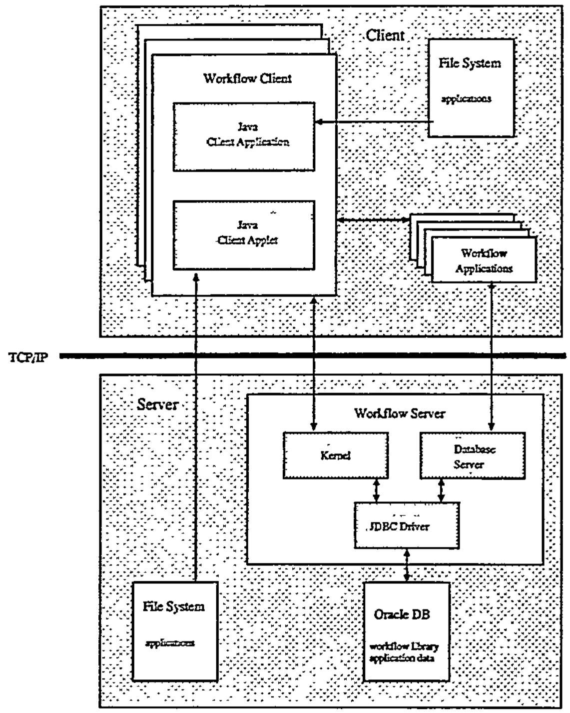

Figure 3: WASA Architecture.

Users access the workflow system using workflow clients. The basic functionality of a workflow client is to inform users (agents in general) of activities to perform. We have implemented two types of workflow clients: Clients can be (i) stand-alone Java applications, or (ii) Java applets which are interpreted by Web browsers. We now comment on the respective properties of these alternative implementations. Since the Java byte code of an applet can be transferred when the workflow client is started (by accessing the workflow client URL), the applet version of a client requires a Web browser on the client side only.

For each activity of a workflow, the corresponding workflow model holds execution information. For atomic activities, there are two options: either the activity is to be performed using a software system (e.g., a GIS) without involvement of a person, or a person is responsible for executing the activity. The former activities are called automatic, while the latter are manual activities. Persons executing manual activities have their workload controlled by the workflow engine, by means of a *work item list*. Each person has a list of tasks (work items) for execution. When a manual activity is started, the workflow engine assigns some person to execute that activity, and sends a work item on the work item list of that person. The person selects that item from his/her work item list, and an application program is started on the workstation of that person. When the manual activity is completed the person notifies the system, which then decides on the next activity to start. When the workflow terminates, the person who started the workflow is notified.

We mention that we are currently working on a second version of the WASA prototype, which is based on object technology and which allows flexible and distributed workflow executions. Its conceptual model is described in [20]. While the primary aims of flexibility and platform independence are still in place, the new WASA prototype is based on a CORBA infrastructure, which allows persistent workflow executions and the integration of application objects in workflow applications using CORBA interface definitions and Object Request Broker functionality [14, 17].

4.2 Using WASA for Geo-Workflows

We now analyze how the geo-workflow in the example can be executed in WASA. Activity 1 (determine phenomena) is a manual activity performed by experts (scientists) and fire fighters. Model building (Activity 2) is done in a cooperative way, but while in most real-life situations experts start from scratch to determine the appropriate model, using WASA they can try to find out about previously designed applications, by browsing the database of workflow models, or by simply inspecting the workflow models that are retrieved by the workflow server for starting. In particular, "research on other applications" includes browsing the workflow database to retrieve geo-workflows built to document the execution of similar experiments.

Data gathering (inventory) is represented by assigning tasks to different technicians which are to provide the desired data sources. This requires putting work items in the lists of different people/departments, e.g., "digitize map of Highway 82" is a typical work item specification that may be sent to a technician in the geo-referencing data processing department. We remark that these people or departments may be situated at different sites, and that the entire procedure of task assignment and execution may proceed remotely, being monitored by WASA.

Once each data file is created, the workflow manager is notified, and the next step (analysis) begins only after all data gathering tasks are signaled as completed. Analysis is a task performed by a person who is knowledgeable about the GIS being used. This person will receive the work item "execute model X using data A, B, C" and will then invoke GIS functions combining these data sources according to the model specification, producing a map. When this is completed, the map is stored in a file and the workflow engine is notified of this (work item is taken from the list). The map is sent for expert analysis once this is finished.

We finally point out a few important issues. First, this automation of scheduling of procedures optimizes execution of tasks in parallel. This is very important, for instance, in activities involving production of electronic data (e.g., digitization of paper maps), since the work item distribution allows technicians to organize their daily work by choosing from this list tasks according to their duration or priority. At the same time, this allows the execution of several applications within a given organization at the same time, each of which following distinct task scheduling policies. Second, the existence of a workflow database allows documentation of the tasks involved in the execution of a given application, which is in itself very useful. Third, this documentation, expressed in terms of executable workflows, will allow repeated execution of a given set of steps. What is even more interesting from a geo-application point of view, this will also allow reusing parts of the application specification to design and implement similar experiments. Using again the fire hazard example, the workflow can be used to direct application developers to create applications for areas where similar conditions exist (i.e., weather, vegetation, human occupation etc).

5 Conclusions

The main goal of this paper has been to show, through a detailed case study taken from a real-life empirical experiment, that workflow management is a reasonable technology to exploit in the area of geo-processing. Indeed, the typical tasks comprising any experiment in that domain can adequately be cast in the form of a workflow model, which is capable of appropriately capturing the relevant process as well as data aspects. However, commercial workflow management systems will vastly fail to support experimental environments, due to the fact that they are based on a compilation instead of an interpretation approach. In other words, they require complete workflow specifications to be compiled into executable code whose execution is then controlled by the workflow engine. Geo-applications, as exemplified by our case study, require workflows based on an interpretative approach.

Clearly, a variety of issues remain to be resolved. One of them is to actually *build* a workflow-intensive environment for geo-processing applications, which integrates devices and procedures throughout an application's life-cycle. Considering that technology in geo-data gathering and processing encompasses a wide range of sophisticated devices, ranging from palm-top to mainframe machines, and including high-resolution graphical devices or satellite-based instruments, an *integrated* environment in which a workflow engine acts as the core component is not easy to build. This has both technical as well as conceptual reasons. For example, palm-top computers are far from being able to act as workflow clients. Moreover, the variety of software tools already in use in geo-applications is difficult to interface to a workflow system, since they are all based on distinct protocols or languages. A way out of this situation could be to construct the workflow system around object-oriented component software, an effort currently undertaken in the group at the University of Muenster. Currently, a new version of the WASA prototype is implemented, which makes extensive use of object-oriented technology in design and implementation [21].

Another issue is to obtain a collection of "prototypical" workflow models that arise from a larger number of geo-processing experiments, in order to enable casual users (e.g., environmental specialists or biologists) to build their work lists with the help of the computer. To this end, we envision a repository of geo-workflows representing a large collection of past experiments into which novel users can do some form of "mining" in order to grasp a handle on their specific tasks. It therefore seems that the introduction of workflow management into the field of geo-processing applications is not only fruitful, but has only just begun.

References

[1] C. Bauzer Medeiros, G. Vossen, M. Weske. *GEO-WASA: Combining GIS Technology with Workflow Management.* In Proc. 7th Israeli Conference on Computer Systems and Software Engineering, Herzliya, Israel 1996, 129–139, IEEE Computer Society Press, Los Alamitos, CA.

[2] C. Ellis, K. Keddara and G. Rozenberg: *Dynamic Change Within Workflow Systems.* In Proc. Conference on Organizational Computing Systems (COOCS), Milpitas, CA 1995, 10–22.

[3] S.F. B. Faray and C. A. Vettorazzi: Evaluation of Fire Risks in Forest Areas using a GIS. Technical report, ESALQ-USP, Department of Rural Engineering, 1996. In Portuguese.

[4] D. Georgakopoulos, M. Hornick, A. Sheth. *An Overview of Workflow Mangement: From Process Modeling to Workflow Automation Infrastructure.* Distributed and Parallel Databases, 3:119–153, 1995.

[5] M. Goodchild and B. Parks and L. Steyaert (Editors). *Environmental Modelling with GIS*, Oxford University Press, 1993.

[6] S. Gordillo, F. Balaguer and F. DasNeves. *Generating the Architecture of GIS Applications with Design Patterns*, Proceedings ACM GIS Conference, 1997

[7] C. Hunsacker, R. Nisbet, D. Lam, J. Browder, W. Baker, M. Turner, and D. Botkin. Spatial Models of Ecological Systems and Processes: the Role of GIS. In M. Goodchild, B. Parks, and L. Steyaert, editors, *Environmental Modelling with GIS*, pages 248–260. Oxford University Press, 1993.

[8] Y. Ioannidis (ed.). *Special Issue on Scientific Databases.* Data Engineering Bulletin 16 (1) 1993

[9] S. Jablonski, C. Bußler. *Workflow-Management: Modeling Concepts, Architecture and Implementation* International Thomson Computer Press, 1996.

[10] F. Leymann, W. Altenhuber. *Managing Business Processes as an Information Resource.* IBM Systems Journal 33, 1994, 326–347.

[11] J. Meidanis, G. Vossen, M. Weske. *Using Workflow Management in DNA Sequencing.* In Proc. 1st IFCIS International Conference on Cooperative Information Systems (CoopIS), Brussels, Belgium 1996, 114–123, IEEE Computer Society Press, Los Alamitos, CA.

[12] T. Nyerges. *Understanding the Scope of GIS: its Relationship to Environmental Modelling* In M. Goodchild and B. Parks and L. Steyaert (editors): Environmental Modelling with GIS, Oxford University Press, 1993, pp75–93.

[13] J. L. Oliveira, F. Pires and C. Bauzer Medeiros. An Environment for Modelling and Design of Geographic Applications. In *GeoInformatica*, Kluwer Academic Publishers, 1(1):29-58, 1997

[14] OMG. *CorbaServices: Common Object Services Specification.* (available from www.omg.org)

[15] F. Pires. *A Computational Framework for Modeling Environmental Applications.* Ph.D. Thesis, University of Campinas, Institute of Computing, December 1997.

[16] M. Rusinkiewicz, A. Sheth. *Specification and Execution of Transactional Workflows.* In: W. Kim (ed.), *Modern Database Systems — The Object Model, Interoperability, and Beyond*, Addison-Wesley 1995, 592–620.

[17] J. Siegel. *Corba – Fundamentals and Programming.* John Wiley, 1996.

[18] G. Vossen, M. Weske. *The WASA Approach to Workflow Management for Scientific Applications.* NATO ASI Workshop, Istanbul, August 12–21, 1997. To appear in Springer ASI NATO Series.

[19] G. Vossen, M. Weske, G. Wittkowski. *Dynamic Workflow Management on the Web.* Technical Report No. 24/96-I, University of Muenster, Germany 1996.

[20] M. Weske: *Object-Oriented Design of a Flexible Workflow Management System.* Second East-European Symposium on Advances in Databases and Information Systems ADBIS'98, Sept 8-11 1998, Poznan, Poland. Springer Lecture Notes in Computer Science 1475, pp 119–131. Berlin: Springer 1998

[21] Weske, M., Kuropka, D., Hündling, J., Schuschel, H.: *Design of a Flexible Workflow Management System for CORBA Architectures* (in German). Technical Report Angewandte Mathematik und Informatik 17/97-I, Universität Münster, 1997

[22] M. Weske, G. Vossen, C. Bauzer Medeiros and F. Pires. *Workflow Management in Geoprocessing Applications.* Technical Report No. 04-98-I, University of Muenster, Germany 1998.

[23] G. Wittkowski. *Design and Implementation of a Workflow System in Java (in German).* Diploma Thesis, University of Muenster, Germany 1996.