Workflow Management in Geoprocessing Applications*

Mathias Weske, Gottfried Vossen†
University of Muenster, Germany

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

February 1998

Abstract

The use of computers in data-intensive and at the same time process-oriented scientific applications, e.g., in geo-sciences, is of emerging interest. However, adequate computing environments supporting such applications are still in their infancy. The workflow paradigm has meanwhile been recognized as being suited for considerably improving the situation and current practice. This paper shows why workflows can meet the requirements of such applications, and presents a specific system architecture – WASA, which is being implemented for supporting scientific application environments. A geo-scientific application is used throughout the paper to illustrate and justify the particularities of the problem and our proposed solution.

1 Introduction

Workflow management aims at modeling and controlling the execution of processes in both business applications [8, 4, 9] and scientific applications [7, 15]. 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

*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 and FAPESP grants.

view, in which collections of activities, their interactions and exchanges are modeled and supported. The exploitation of the workflow paradigm in scientific application domains such as the geo-sciences, 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 an environmentalist's expertise on process modeling with his or her 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, described in the course of this paper, is to remedy this situation. Specifically, WASA tries to take the particular requirements of these applications into account; among these, high modeling and specification flexibility as well as platform independence [1, 10] seem to be most crucial. We show how WASA can be used in supporting the complex cycle of process and data modeling in environmental-related sciences. In particular, we investigate planning processes in geo-processing applications, formalize these processes as workflows, and show how they can be supported by a prototypical workflow management system we have been developing in the context of 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 geo-processing applications by instantiating the example in it. Section 5 presents conclusions and future work.

2 The Life-Cycle of Applications in Geo-Processing

There are different types of users who work with computational tools in the domain of geo-processing: beginners, casual users, expert users and application designers. The latter are responsible for developing new tools and applications to be used by the others. Very often, expert users work closely with designers. We here look at such 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.

More specifically, we are interested in issues concerning projects related to the environment (e.g., monitoring); we will refer to these applications as *geo-applications*, denoting the fact that they deal with *geo-referenced* data. This field encompasses a large spectrum of scientific activities. This variety is due first to the fact that the term

geo-referenced concerns any data that is associated to its location on Earth and thus potentially to all existing natural phenomena and human-made artifacts on Earth. Another issue is that, for any set of such data, distinct views and needs exist, depending on the scientist/expert that is conducting the experiment (e.g., social scientist, engineer, geologist, ecologist).

Users who design geo-applications currently take advantage of a variety of computational tools which help spatial analysis and cartographic presentation. An important set of tools is offered by Geographic Information Systems (GIS), which are systems that allow storage, querying, management, and visualization of geo-referenced data.

From a macro point of view, the life-cycle of a geo-processing environmental application can be considered in four major steps: real-world modeling; geographic database specification and loading; implementation; and monitoring [12].

Real world modeling comprises *data* and *process modeling*, and corresponds to selecting, abstracting and generalizing the entities of interest to the user, showing how they vary through time. The output of this activity directs the definition of the *database*, as well as specifies the *function libraries* and *model parameters* that are to be used together with data stored in the database. *Implementation* concerns the use of databases and libraries, combining functions and producing new data, either directly by means of programs or, more frequently, using a GIS. The result of the implementation is usually a set of maps and tables, which will be used by experts to determine how to act on some situation (in our example of the next section, how to better prevent fire risks). Finally, the *monitoring* phase concerns checking the actions determined by the experts to find out whether the previous application phases were developed correctly.

Process modeling refers to constructing a mathematical model that describes operations involving the stored data representations, and includes the simulation of natural phenomena. Process modeling is based on, for a given problem, selecting the phenomena and mathematical set of equations on the phenomena that simulate the corresponding real-world situation best. Process models run on data which has been organized according to a data model. A *data model* provides the tools and formalisms needed to describe the logical organization of a database, as well as to define the allowed data manipulation operations. After the modeling stage, data is stored in some database.

The definition of the databases to be used for a given application is called *inventory* by the users. In this stage, users indicate data sources that must be collected and combined in order to present an adequate view of the reality. Once data is collected and stored, the process model is "invoked"[11], i.e., a sequence of algorithmic transformations is applied to data. This actual execution of the process model on data corresponds to the actual *implementation* of the application, which is preceded by analysis of alternatives for this implementation.

The output of the implementation phase is analyzed by the users, in order to define

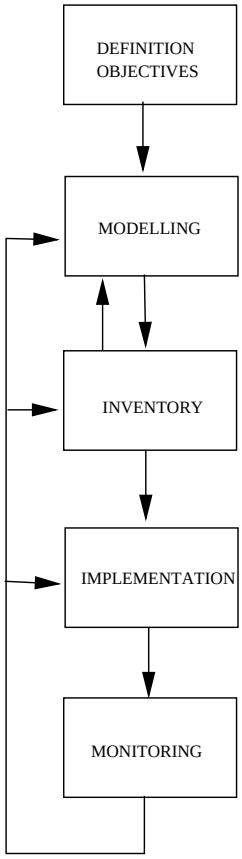

Figure 1: Life-cycle of geo-processing applications.

policies to be taken. Furthermore, once policies start to be enacted, there has to be constant *monitoring* to verify the adequacy of the policies, and this may even go back to model refinement. Each step provides feedback to the others.

Figure 1 gives an overview of the steps involved in designing and executing geo-applications, and which are described in detail in [13]. 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.

Briefly, the methodology supports user activities in specification and implementation of geo-processing applications in terms of a sequence of data and information transformations. Data sources can be of any type (files, user actions, etc). The development of an application is a process that is triggered by the need for solving an environmental problem, and whose final output is a combination of electronic data and policies and strategies that direct the implementation of the solution.

In conclusion, a common and typical geo-processing life-cycle comprises five global steps: *definition of objectives*; *modeling*; *inventory and database creation*; *implementa-*

tion and *strategy definition and monitoring*. Each step comprises several substeps or tasks, which can be accomplished by different means (automatic or manual). 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 Workflow Management

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 [9]; as the *modeling* of business processes has become a strategic goal in many enterprises, a further step is to optimize or to re-engineer them, with the goal of automation in mind. Various tools are now commercially available which support such business process re-engineering. 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 workflow management, i.e., business applications, new domains emerge, among which scientific ones play a major role [15, 1].

In general, workflows can be characterized as the automated control and management of processes in businesses and other organizations. The structure of these processes and the structure of the organization performing them are specified in *workflow models* [8, 9]. In general, workflows consist of a set of related activities which are executed by processing entities [14]. The building blocks of process models are *activities*. Activities are units of work as perceived by the modeler. Each activity includes a description of what it codes for, the data used and generated by the activity, formalized as typed input and output parameters, resp. Activities can be specified in a variety of ways, including textual descriptions (e.g., in email or a file), forms, messages, or computer programs. *Processing entities* which can perform tasks may be humans or software systems, e.g., mailers, application programs, or database management systems.

3.1 Modeling Geo-Processing Processes as Workflows

This section provides a brief analysis of how user procedures in designing and developing geo-applications can be appropriately modeled by workflows. Consider the sequence of steps depicted in Figure 1. First, this high level description can be seen as a 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 some previous

step, reflecting the fact that the modeling and interpreting of natural processes is a never-ending activity.

For these reasons, it appears appropriate to exploit workflows and workflow management in geo-applications. Clearly, there are several issues which are not covered adequately by existing commercial workflow management systems. Indeed, existing tools are geared towards business applications, whose designers have a clearer understanding of the rules governing them. In such domains, constraints are better understood and thus easier to specify. In geo-sciences, on the other hand, each application can be seen as one-of-a-kind, its specification being 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. For this reason, scientists working on geo-sciences find it hard to reuse past results, since each application is, in a sense, unique. This uniqueness of specification creates several problems to the use of standard workflow management systems. Nevertheless, there are steps that can be repeated (e.g., procedures for data collection such as digitization) and many databases can be reused, if the data therein is described appropriately. The problem, from the users' point of view, is to specify the sequences of activities in a way that will allow this reuse.

As we show in the remainder of this paper, the WASA system we have developed presents a solution to these problems. We also point out that a characteristic of geo-applications is the intensive use of GIS technology. In fact, data are stored in several source files, and their analysis is performed by means of combining GIS computational tools. Visualization of results is also provided by the GIS graphical interface, usually by means of cartographic representation. This need for invoking specific tools is also handled well by a workflow management system. As we shall see, WASA allows interspersing of human and computational activities, which are mediated by the workflow management system.

3.2 Geo-Applications as Scientific Experiments

This section provides an example of using workflows to specify a geo-application, here treated as experiment in geo-sciences. As remarked previously, there is a wide spectrum of such applications. In spite of all this diversity, scientific experiments involving geo-referenced data are basically composed of three main steps: *data gathering* (inventory) *data analysis* (modeling) and *production of output* (implementation). These three activities form the starting point of any workflow for describing these experiments. They constitute the kernel of the methodology sketched in the previous section, and correspond to the activities that are more prone to automation. The Monitoring and the Definition of Objectives phases (see Figure 1) are essentially human activities that precede and follow the automated steps. Yet these activities can still 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 turns help documenting the entire development process.

From now on, we refer to the workflows describing these activities as *geo-workflows*, to differentiate them from other types of workflow. This sequence of activities – data gathering, analysis and obtaining of results – is, of course, essentially applicable to any scientific application. Wherein then lie the differences? First, the data handled in geo-workflows must include at least one geo-referenced data set; second, the analysis activities performed must comprise some sort of spatial inferencing and computation; third, the solution approach is sensitive to the geographic region being studied and the temporal framework; finally, the result is often also geo-referenced, generally producing one or more maps.

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. It is true that this also applies to a scientific framework in general, but the actors and roles found in geo-workflows are often very dissimilar from each other, thereby requiring a better supporting computing environment for group work.

We stress that, from our point of view, a large spectrum of geo-applications can be actually treated as scientific experiments on geo-referenced data. Indeed, when dealing with real-world phenomena, the building and testing of models is performed empirically, by teams of scientists, who often need to rely on their experience in order to define adequate modeling variables and parameters. Furthermore, application output (the experimental results) is prone to interpretation and, in several cases, are not predetermined from the start of the execution of an application. For instance, a study to correlate the influence of different anthropic factors in the pollution of a given area can be performed by an application that, given relevant geo-referenced data, will produce maps and correlation tables. It is only on interpreting maps and tables that the persons responsible for the application (designers and expert users) will be able to draw conclusions not only about the correlation, but also about the hypotheses that guided the application development. Again, in this sense, geo-applications can be treated as scientific experiments. Literature on developing such studies (e.g., see [5]) confirm this analogy between geo-application and scientific experiment.

3.3 A Geo-Workflow Case Study

We will analyze a specific instance of a geo-workflow next. This case study corresponds to an adaptation of an experiment which was conducted to evaluate fire risks in the county of Piracicaba, Brazil [3], which required developing a geo-application using a commercial GIS. The geographic area concerned is a natural preserve belonging to the Forestry Research Institute of the State of São Paulo. Though this preserve is

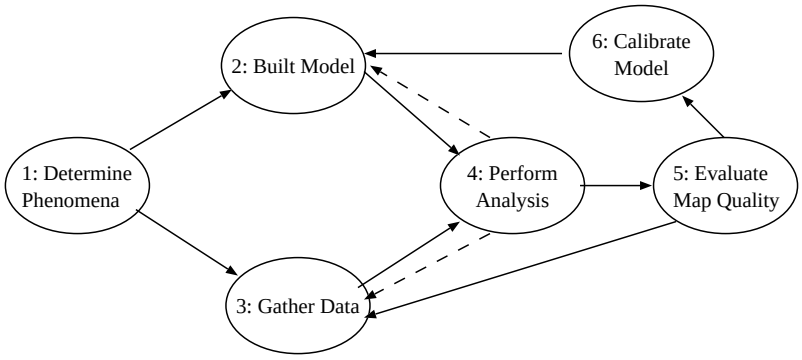

Figure 2: Top-level geo-workflow.

dedicated to different sorts of experimental research (e.g., on biology), it contains some recreation areas. 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 2 shows the top-level geo-workflow that was used to specify the three basic activities mentioned (in Figure 2, these activities are numbered 3, 4 and 5, respectively). We should point out that, for the purpose of this paper, we will often make simplifications on assumptions and activities performed during an experiment, since our goal is to present a typical geo-workflow (and not a scholarly description of the experiment, to be re-used by experts in environmental research).

This geo-workflow in question 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, performed by a group of experts, 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. The specification of data and parameters to be considered characterize this geo-application as a particular empirical experiment.

Next, two activities were launched in parallel: *Model specification* (Activity 2) and *data gathering* (Activity 3). *Data gathering* was performed in parallel by several teams of people, as often happens in geo-processing activities. It consisted of checking on already available data sources and, if necessary, performing field work to collect additional data. The activities involved in data gathering are specified according to the nature of data

desired, using different devices at distinct scales, and may include manual work (e.g., map digitization).

Model specification is a highly empirical activity. In the geo-sciences, models provide the basis for a simulation of the dynamics of the real-world, being a description of the key processes involved in such dynamics. A *model* in this sense usually consists of a system of equations which mathematically describe such behavior. The "execution" of a model, for a specific set of data, is called *analysis* (i.e., application of specific functions to data collections). Models are influenced by the type of study conducted, the temporal and spatial scales considered, and the goals of the experiment. Some examples of the models found in environmental research can be found in [6]. Such models range from deterministic to heuristic, and may vary from a merely qualitative appraisal of phenomena to a specification of complex equations. Model building for this type of problem is achieved on a trial-and-error basis, by answering the following questions [6]:

- 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 on 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. For instance, 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; fire propagates faster along steeper inclines, and so on. Some weights were time-dependent, e.g., in valley regions fire risks were deemed higher during the day period than at night.

Activity 4 (*Analysis*) 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 and the maps are progressively superimposed to form the final result. The computations performed by an overlay can be described, at a high level of abstraction, as a sequence of (weighted) matrix additions. Map overlays are usually performed automatically, using a GIS, but intermediate results must be checked in order to interrupt the experiment and refine the model or gather more data (i.e., go back to Activities 2 and/or 3, as shown in Figure 2 by the dotted lines).

The result of the analysis activity was a fire risk map, which was the input to Activity 5 (*Assess map quality*). In this specific experiment, the map was considered to be acceptable within the specified error margin, and therefore the experiment was concluded. Quality assessment consists of checking the result against some control data. In the problem studied, it consisted of checking the map produced against maps

created from historical fire data, to evaluate the accuracy of the fire risk map produced by Activity 4. Quality assessment 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.

We point out that it is not always the case that such experiments work at first shot. 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).

We also point out that, from a computational point of view (and as far as this geo-workflow is concerned), this experiment was considered concluded with the production of the fire risk map. In reality, such a map would next be analyzed by a different set of experts and local government authorities in order to determine preventive measures to be taken to diminish fire hazards. Policy determination and follow-up are activities that are often found in environmental experiments. They were not included in the geo-workflow since they were outside the control of the group that conducted the experiment.

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 3.4 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.

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.

3.4 Refinements of the Top-Level Workflow

Figure 2 can be refined into a variety of sub-workflows. We will here indicate only how to refine Activities 3 and 4, but all others could have been equally decomposed. Activity 1, for instance, is performed by combining sequences of meetings of experts interspersed with getting data on other fire risk analyses conducted in areas with similar characteristics. The goal of such an activity is to use past experience to help determine the adequate sources and world dynamics model to be applied.

Data gathering (Activity 3) requires launching several independent geo-workflows, each dedicated to collecting data of a different nature. Figure 3 shows the refinement of

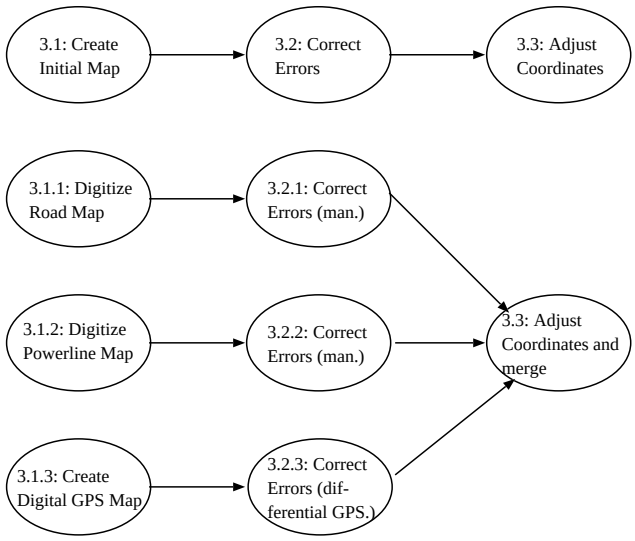

Figure 3: Refinement of Activity 3 and details of eliciting road map file.

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 only issue in the case of these two data types was identifying the appropriate files, as far as geographic region and time frame were concerned. On the other hand, 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 created for this experiment was the "road map" file. Let us briefly examine how this data set was produced. In this case, three different data sources were processed and combined (see the lower part of Figure 3): a highway paper map (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[1]). Why would a power line map be used to help generate a road map? In fact, high voltage lines imply the existence of small paths directly underneath, that must always be kept clean of vegetation, in order

[1]Global Positioning System – an electronic device that, used in conjunction with signals sent from orbiting satellites, gives a very accurate description of one's location.

Figure 4: Initial refinement of analysis activities.

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).

Activity 4 (Analysis) is also broken into Activities 4.1 (Coordinate transformation), 4.2 (Scale homogenization) and 4.3 (Model application) as shown in Figure 4. Activities 4.1 and 4.2 are performed sequentially to adjust all the data sets produced by Activity 3 to the same spatial framework. For instance, the study was conducted on scale 1:10,000, but the vegetation data available was stored at scale 1:25,000. Therefore, special algorithms had to be applied to the vegetation map in order to convert it to the desired scale.

Activity 4.3 (Model application) is one of the most complex parts of the analysis. It includes sub-activities such as data generalization, classification and spatial clustering, some of which can be executed automatically, while others may require human monitoring. It also often requires deriving data. In the example, experts had to derive a "land-use" data map from available data sources. Land use denotes the partitioning of a region in areas according to the predominant use of the area (e.g., industrial, agricultural, mining), and often needs the intervention of experts who know the area of study. Land use maps are often obtained by examining available socio-economic and natural resources data. Another instance of data derivation at this stage was the construction of an intermediate "solar exposition" map. Given relief and cardinal orientation data, one can determine the degree of exposition to solar rays of a given area, which in turn helps defining regions where vegetation will be drier than others (and thus more prone to fires), having been more exposed to the sun.

We omit further workflow refinements, since they proceed vastly along the lines of the refinements described above. Experimental procedures are described at length in [3]. We conclude this section by a few remarks. First, the analysis procedure described is highly simplified. Analysis is the gist of all scientific geo-experiments. 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. Its major focus is on applications in the natural sciences and in laboratory environments.

In this section we first describe the WASA architecture as well as the operation of its system. We then look into the application discussed in the previous section, showing how it can be supported by WASA, pointing out the relevant system characteristics which distinguish it from other workflow systems.

A first prototype of WASA has been implemented using Java and a commercial relational database system [17, 16]. This prototype has been tested on various cases, and we are now refining it in order to extend its functionality. We briefly comment on a prototypical implementation of the core part of the WASA system, namely the workflow engine. This component aims at enhancing the flexibility of existing workflow management systems while providing a high degree of platform independence [15]. 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.

4.1 System Architecture

The design decisions of supporting flexibility, providing dynamic modeling capabilities, and delivering platform independence have led to the system architecture shown in Figure 5. Loosely speaking, the WASA protoype consists of a workflow engine, a database server and workflow clients. 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.

Users access the workflow system using workflow clients. The basic functionality

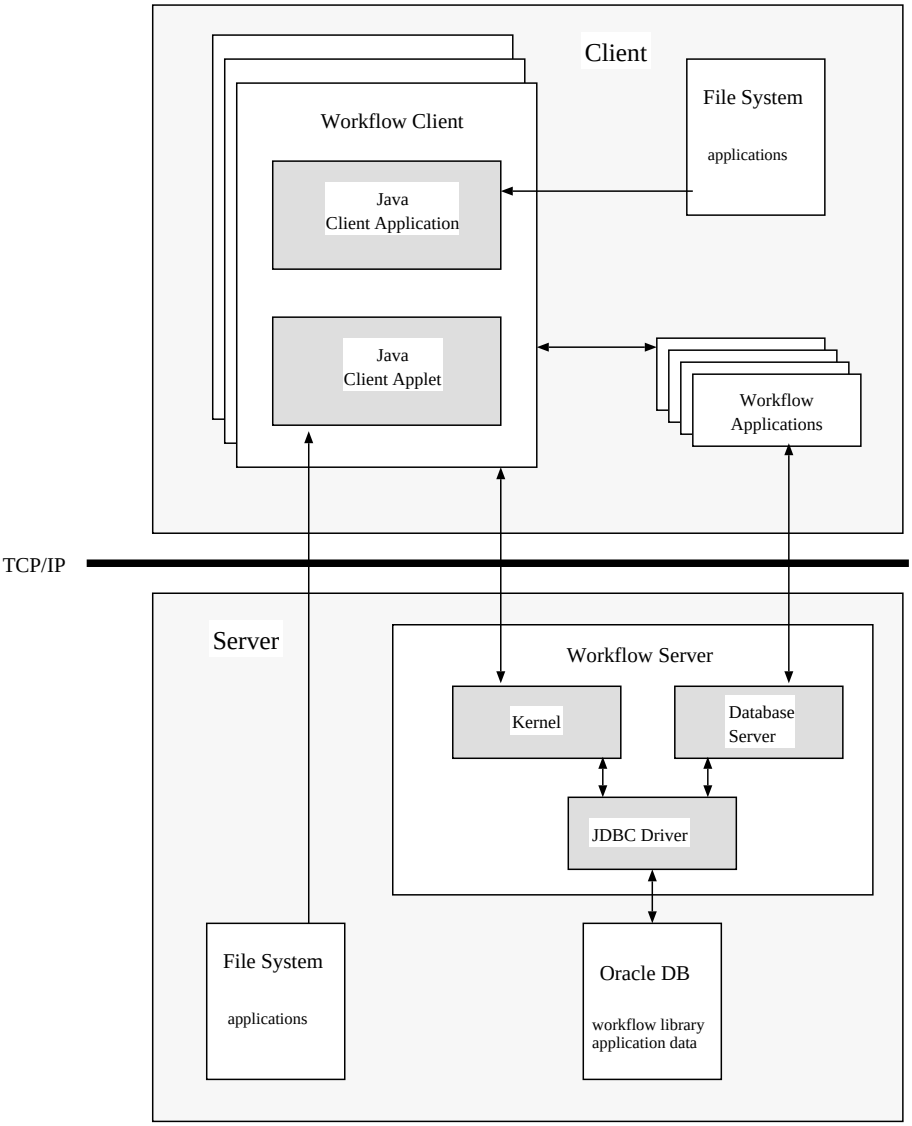

Figure 5: System Architecture.

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. Due to security restrictions, local data accesses are not allowed for Java applets. Stand-alone Java applications do not have this restriction. In particular, workflow clients can access the local file system and may start arbitrary application programs to implement activities. However, they have to be installed on the client host before they can be used, which requires the Java byte code of the application and a Java interpreter. In both cases, the communication between the workflow server and the workflow clients is based on the TCP/IP protocol.

4.2 System Operation

When the workflow server is started, it reads all workflow models from the database and displays those which can be started immediately. In general, a workflow model, which can be seen as a labeled, directed graph, can be started if it does not have any (pending) incoming control connectors. By selecting the workflow model, a new workflow instance is created. At any point in time, the system may control multiple instances of a given workflow model. Hence, the system supports concurrent execution of multiple workflows.

For each activity of a workflow, the corresponding workflow model holds execution information. For atomic activities, there are two options: either the activity is implemented using a software system (e.g., a database system) 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, helping the system identify responsibilities and bottlenecks. When a manual activity is started in a workflow instance, 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.

The workflow server creates a log file which includes information on the creation and termination dates of the workflow and its activities and role resolution information. This historical information on workflow executions can be used to analyze and optimize workflows.

4.3 Using WASA for Geo-Workflows

We next analyze how the example geo-workflow can be executed in WASA. Activity 1 – determine phenomena – is a manual activity performed by experts (scientists) and fire fighters. Model building 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 previous experiments, 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 experiments" includes browsing the workflow database to retrieve geo-workflows built to document the execution of similar experiments. Thus, in order to allow reproducibility, WASA requires that these procedures be monitored in detail.

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 the person responsible for 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 develop a GIS program combining these data sources according to the model specification. When this program is completed, the result is stored in a file and the workflow engine is notified of this (work item is taken from the list). Once the map is produced (analysis result) the workflow execution is finished, and the map is sent to experts for analysis.

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 experiments 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. As the case study we have described inevitably supports, basing experimental workflows on an interpretative approach is definitely more suited.

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 experiment'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 *component software*, an effort currently undertaken in the group at the University of Muenster.

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] 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.

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

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

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

[10] 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.

[11] 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.

[12] 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

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

[14] 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.

[15] 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.

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

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