WOODSS and the Web: annotating and reusing scientific workflows

C. B. Medeiros, J. Perez-Alcazar, L. Digiampietri, G. Z. Pastorello Jr, A. Santanche
R. S. Torres, E. Madeira and E. Bacarin

Institute of Computing – University of Campinas – UNICAMP CP 6176, 13084-971 Campinas, SP, Brazil

{*cmbm,jperez,luciano,gilberto,santanch,rtorres,edmundo,bacarin*}*@ic.unicamp.br*

Abstract

This paper discusses ongoing research on scientific workflows at the Institute of Computing, University of Campinas (IC - UNICAMP) Brazil. Our projects with bio-scientists have led us to develop a scientific workflow infrastructure named WOODSS. This framework has two main objectives in mind: to help scientists to specify and annotate their models and experiments; and to document collaborative efforts in scientific activities. In both contexts, workflows are annotated and stored in a database. This "annotated scientific workflow" database is treated as a repository of (sometimes incomplete) approaches to solving scientific problems. Thus, it serves two purposes: allows comparison of distinct solutions to a problem, and their designs; and provides reusable and executable building blocks to construct new scientific workflows, to meet specific needs. Annotations, moreover, allow further insight into methodology, success rates, underlying hypotheses and other issues in experimental activities.

The many research challenges faced by us at the moment include: the extension of this framework to the Web, following Semantic Web standards; providing means of discovering workflow components on the Web for reuse; and taking advantage of planning in Artificial Intelligence to support composition mechanisms. This paper describes our efforts in these directions, tested over two domains – agro-environmental planning and bioinformatics.

1 Introduction

Scientific activities involve complex multidisciplinary processes and demand cooperative work from people in distinct institutions. While the Web provides a good environment for this kind of work, on the other hand it introduces the problems of data, process and tool proliferation. Challenges in this scenario include how to understand and organize these resources and provide interoperability among tools to achieve a given goal, as well as appropriate mechanisms to document, share, deploy, construct and re-execute scientific experiments.

Scientific workflows [23] are being used to help solve some of these questions. In particular, their execution must allow for issues such as human curator intervention, flexibility in specification to accommodate distinct approaches to a problem, deviation from the specification while the workflow is being executed and unfinished executions.

Our work with environmental and bioinformatics applications have shown us that scientific workflows are moreover a good *documentation* paradigm. In other words, workflows not only help specification and execution, but keeping track of evolution of experiments and their annotation. We are testing these concepts using WOODSS (WOrkflOw-based spatial Decision Support System) [7, 11, 15, 19], a scientific workflow infrastructure developed at IC - UNICAMP. Originally conceived for decision support in environmental planning, it has evolved to an extensible environment that supports specification, reuse and annotation of scientific workflows and their components.

We are now extending this work to the Web, supporting flexibility in workflow design, experiment annotation, customization and reuse. This paper describes this work, being centered on the following issues: the database of workflow building blocks, that induces a workflow design methodology; the use of component technology to encapsulate these blocks, enabling their reuse and composition; and the exploration of advances in planning in AI to support automatic and semi-automatic composition of workflows. We briefly mention our testbed efforts in two domains - agro-environmental planning and bioinformatics.

2 Overview of WOODSS

WOODSS was initially conceived to support environmental planning activities. It was implemented on top of a commercial Geographic Information System (GIS) and has been tested in several contexts, mostly within agro-environmental applications.

The original idea was to dynamically capture user interactions with a GIS in real time, and document them by means of scientific workflows, which could then be re-executed invoking the GIS functions.

In environmental applications, user interactions

with a GIS express models for solving a given problem. Fig. 1, adapted from [11] illustrates the interaction modes of WOODSS. In the first mode, users interact with the GIS, to implement some model, whose execution is materialized into a map. WOODSS processes this interaction and generates the corresponding scientific workflow specification, stored in the Workflow Repository. In the second interaction mode, users access WOODSS' visual workflow editing and annotation interface to query, update and/or combine repository elements, annotating and constructing workflows that can be stored in the repository and subsequently executed invoking the GIS. This allows workflow evolution and lets scientists find out about previous solutions to a similar problem. The geographic database contains domain-specific data.

The present version of WOODSS is implemented in Java^{TM}, with the repository in POSTGRESQL. It contains scientific workflows (and parts thereof) and associated annotations (keywords, free text, metadata and domain ontologies [7]). The graphical interface allows user-friendly workflow editing and manipulation. More details appear in [11, 19].

Our extensions to this framework, described in the next sections, can be sketched in terms of Fig. 1. First, the geographic database is replaced by sets of databases for scientific application domains – e.g., genomics, agriculture. Second, instead of the functions of a proprietary GIS, we consider invoking third party applications/services (the "execute workflow" arrow). The "capture interaction" arrow can be replaced, for specific applications, by monitoring modules that generate the appropriate workflow specifications. Third, the Workflow Repository contains annotated (sub)workflows and their building blocks, designed according to a specific model (see Sect. 3) and encapsulated into our reuse unit Digital Content Components – DCC, see Sect. 4.

3 WOODSS' data model

This section gives an overview of the base data model used for representing scientific workflows in WOODSS; more details are available in [14, 15]. The workflow terms used here have the usual meaning, e.g., (i) a workflow is a model of a process; (ii) a process is a set of inter-dependent steps needed to complete a certain action; (iii) an activity can be a basic unit of work or another workflow; (iv) a transition establishes a dependency relation between two activities; (v) an agent is responsible for the execution of a basic unit of work; (vi) a role specifies the expected behavior of an agent executing an activity; and so on.

The model supports the storage of several abstraction levels of a workflow in a relational database. This induces a methodology for correctly constructing workflows, whereby users must first specify types of workflow building blocks, next combine them into abstract specifications (*abstract workflows*), and finally instantiate these specifications into executable (sub)workflows (*concrete workflows*).

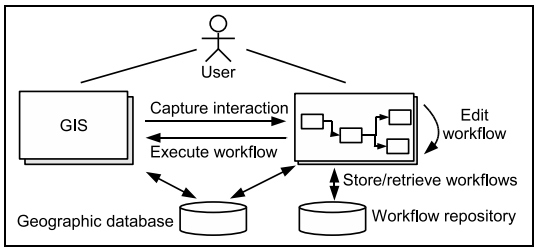

Figure 1: WOODSS interactions - adapted from [11]

The building blocks can be split in three major levels, which can be refined into many intermediate levels. The three levels reflect the methodology – see Fig. 2. The first level corresponds to definitions of data and activity types, and roles to be fulfilled by agents. Fig. 2(a) shows two activity types (*Combine Maps* and *Classify*), typical of environmental tasks. Activity types are specified in terms of their interfaces, which are based on the previously defined data types. In the figure, *Combine Maps*'s interface has two inputs, *I1* and *I2*, and one output *O1*.

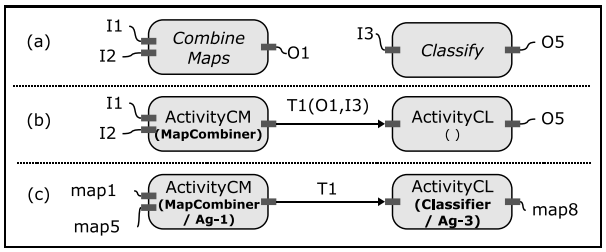

Figure 2: Levels of workflow specification

Activity and data types are used to create blocks at the second abstraction level, where a workflow structure – the abstract workflow – is specified, through transitions and dependencies among activities – e.g. the activity labeled *ActivityCM* is of type *Combine Maps* – see Fig. 2(b). At this level, it is also possible to refine activity types by associating roles to them – e.g., *ActivityCM* is associated with role *MapCombiner*. The transition *T1* connects interface elements *O1* and *I3*. The specification of types and of abstract workflows capture the notion of *workflow design* independent of execution aspects (agents and data connections), allowing design reuse, as will be seen in Sect. 4.

The last level involves creating an executable version of an abstract workflow – i.e., the concrete workflow. This is achieved by associating agents with activities and actual data sources with activities' interfaces. Fig. 2(c) shows *map1* and *map5* data sources as the input parameters for the *ActivityCM* activity, which can be executed by invoking the specific GIS *Ag-1*.

We point out four important issues covered in our model. First, the distinction among the levels is not so clearcut, since a characteristic of scientific workflows is their incremental construction – see Sect. 4. Second, the specification of an activity separating its interface from its functionality is the basis of the mechanism for allowing definition of arbitrary nesting of sub-workflows. Indeed, a sub-workflow is just another activity, accessed via its interface, and whose specification is encapsulated. A workflow/activity thus references any other workflow/activity specification inside the database, without violating the encapsulation. Third, at any moment scientists can attach annotations to the building blocks. Finally, blocks are encapsulated into DCCs – see Sect. 4.

Workflow blocks are serializable in WS-BPEL, in order to make them available on the Web. The selection of an XML-based language to represent workflows took into consideration current standard proposals. In spite of shortcomings for representing workflows, WS-BPEL turned out to be the most suitable to our representation needs [14]. More details can be found in [14, 15].

4 Reuse

Workflow building blocks – types, abstract and concrete workflows – are stored in a database for documentation and reuse. WOODSS considers two kinds of workflow reuse: that of reusing a workflow design, and that of reusing executable workflow elements. To facilitate their discovery and combination, we encapsulate these reusable units into *Digital Content Components (DCCs)* [17], special content managing units that we have been using to build applications.

DCCs are self-contained stored entities that may comprise any digital content, such as pieces of software, multimedia or text. Their specification takes advantage of Semantic Web standards and ontologies, both of which are used in their discovery process. In our workflow repository, we differentiate between two kinds of DCC – *design* and *executable* DCCs. A design DCC contains an abstract process specification, represented as a WOODSS abstract workflow (see Sect. 3). An executable DCC is a component ready to be used in some execution and encapsulates any kind of executable process description, as

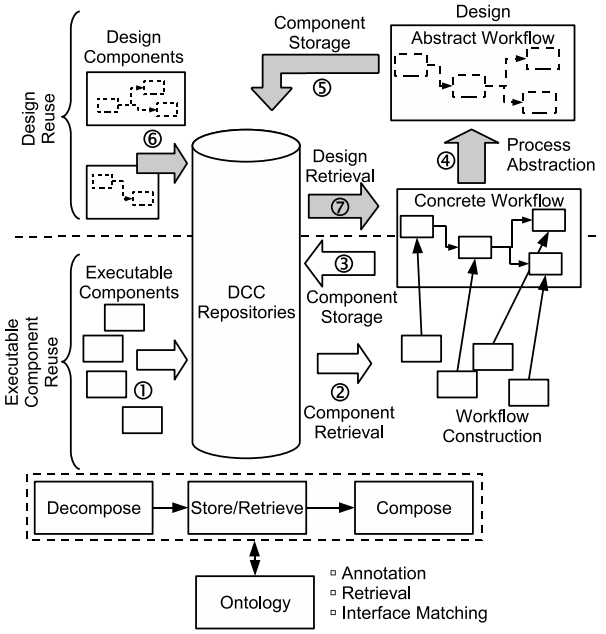

Figure 3: Reuse levels.

the concrete workflows (see Sect. 3).

DCC interfaces have been designed to semantically describe how DCCs can be accessed and combined. Interface description uses WSDL (syntactic specification) and OWL-S (`www.daml.org/services`) (semantic specification via ontologies). DCCs and Web services are homogeneously treated in our model, since they have WSDL interfaces and are able to handle the same protocols. Semantically enriched descriptions have a key role as they [18]: (i) enable similarity-based search that explores ontological relationships; (ii) support an unambiguous and more precise verification and assistance during workflow building.

Fig. 3 shows the two kinds of workflow reuse supported by WOODSS via encapsulation of building blocks into DCCs – executable components and design reuse.

Executable components reuse, represented by the cycle ①–②–③, is based on executable DCCs. This kind of DCC encapsulates third party software ① or a concrete workflow ③. Workflows are produced or modified by retrieving and composing previously stored executable DCCs ②.

In *design reuse*, represented through the cycle ④–(⑤/⑥)–⑦ we are interested in the abstract workflow inside the design DCC. Design DCCs can result from a process abstraction ④, stored in the repository ⑤; or can be produced by third party designer ⑥. Furthermore, a design DCC can be used in the

construction of a concrete workflow ⑦.

Scientific workflows can, by nature, be designed and modified on the fly during their execution. Thus, it is important to point out that the distinction, in the figure, of separate reuse cycles for design and executable DCCs has only didactical purposes, since we can intersperse both. Indeed, an abstract workflow, inside a design DCC, can have portions that are concrete specifications. Nonetheless, it still maintains its characteristics of design DCC, and may be executed considering only the concrete portions. During its execution, scientists can convert the abstract portions into concrete specifications, filling in the missing details. Other possible updates are: modification of the workflow structure (e.g., by changing control structures, deleting or including DCCs); and annotations.

Reuse via DCC is comparable to software engineering's software component reuse, with two advantages: (i) reuse is not limited to program code, but also to design encapsulated inside components; (ii) interfaces provide semantically enriched descriptions. Moreover, we also allow encapsulation of complex data within DCCs, providing thus an unified model for design, process and data. Data DCC are outside the scope of this paper – see [17].

Finally, we point out that workflow building blocks have two kinds of annotation mechanisms. First, users can associate information to them while constructing or executing a workflow (see Sect. 3) – e.g., providing pointers to documentation such as bibliography, reasons for adopting a model, and even commenting on success rate of a given experiment. Such annotations can be attached to (sub)workflows, activities, agents, data, data types, etc. Second, DCC interfaces provide semantic annotations that are used for searching and composing the appropriate workflow blocks (using among others ontology alignment solutions) [18].

5 Case studies

5.1 Agro-Environmental Management

WOODSS' workflow base started from environmental studies, and evolved to agro-environmental planning with help of experts from the Brazilian Agriculture Ministry. Agro-environmental management combines environmental (preservation) and agricultural (exploitation) issues, thus presenting an interesting challenge to scientific workflow specification. Examples of factors involved include regional topography, soil properties, climate, crop requirements, social and environmental issues. Data sources include sensors such as weather stations and satellites, and may be stored in a variety of databases, with different spatial, temporal and thematic scopes [8].

Examples of an abstract workflow specification is, for instance, a sub-workflow that computes an estimate of soil erosion for an area given a crop's needs. This would use, among others, sub-workflows such as those indicated in Fig. 2(b). This can become a concrete sub-workflow if actual data instances are provided, and agents are defined (e.g., a Web service). This sub-workflow can be composed with another one, that for instance evaluates which parts within the area have to be left alone in order to decrease erosion risk.

Within the agricultural domain, we are now investigating the interesting problem of integration of scientific workflows into agricultural supply chains [2], thereby breaching the gap between business and scientific workflows. A supply chain is a network of heterogeneous and distributed elements – retailers, distributors, transporters, storage facilities and suppliers that participate in the sale, delivery and production of a particular product.

The workflows that run within an agricultural supply chain are mainly business workflows. They are concerned with controlling production processes, and selling and delivery of goods. Although most chain activities are business driven, scientific workflows can be needed in these environments.

For instance, precision farming, land use and climate monitoring and forecasting can be supported by scientific workflows. The result of such workflows can feed planning activities within business supply chain workflows – e.g., to forecast harvest productivity and thus distribution logistics within the supply business chain. On the other hand, business workflows can produce feedback to such scientific workflows. For example, market demand for a given produce may require expanding its production at the start of the chain, which will influence factors such as area planted, water consumption and fertilizer utilization. These factors will then affect the ecological balance of the regions involved; calculating this impact involves drawing upon scientific workflows.

5.2 Planning in Bioinformatics

The appropriate composition of blocks to construct a concrete or abstract workflow is a challenge. Artificial Intelligence research in *planning* (also called plan synthesis) concerns the (semi-)automatic creation of a sequence of actions to meet a specific goal. This research domain appeared in the sixties, heavily influenced by work on automated theorem proving and operations research. Nowadays, such techniques are employed in various domains – e.g., robotics, manufacturing processes, satellite control [9]. In particular,

a new trend is to use them to solve the problem of automatic composition of Web Services [10].

We are exploring this kind of initiative to support the semi-automatic or automatic composition of (reusable) workflow blocks for bioinformatics. Bioinformatics experiments are, most of the times, composed of several related activities, so they can be modeled as workflows [4].

Scientific workflows are being utilized in *in silico* experiments at the Laboratory for Bioinformatics (LBI) (`www.lbi.ic.unicamp.br`) at IC – UNICAMP. LBI was the first Brazilian bioinformatics laboratory, being responsible for the coordination of the assembly and annotation of the *Xylella fastidiosa* genome [6]. At present, LBI is dedicating efforts in the specification and implementation of a framework for the management of bioinformatics scientific workflows [5]. For instance, a genome assembly abstract workflow is composed by, among others, sequence filters and sequence assembly activities. This can next become a concrete workflow through instantiation of data and using tools such as *crossmatch* and *phrap*.

A large part of bioinformatics workflows are designed manually, using simple resources like script languages and invocation of Web services. However, manual composition is a hard work and susceptible to errors. Therefore, it is necessary to design tools to support the composition of bioinformatics services in an automatic or interactive (semi-automatic) way [12].

In order to support these issues, we have defined a system architecture that allows semi-automatic and automatic composition. This architecture is based in the SHOP2 [20] domain independent planner. It takes advantage of the WOODSS workflow component repository to select elements to construct abstract and concrete workflows, using DCC interface specification for service description and matching.

6 Related Work

Scientific workflows appear increasingly in the literature. One trend is concerned with showing how specific kinds of systems support application domains such as bioinformatics or geosciences. Another issue is their execution, raising questions such as transaction management or grid initiatives (e.g., [22]). Grid architectures support flexible execution options, assigning each part of a workflow for execution in a distinct node, considering factors such as availability or computational power.

Our work in extending WOODSS to the Web adds another dimension to the notion of flexibility in workflow handling. Not only does it allow distinct executable components to be stored in distributed sources on the Web, but it also allows their dynamic reuse, adaptation and configuration. Furthermore, it introduces the notion of design components, which are also stored and can be retrieved and reused to construct abstract workflow specifications.

The issue of composition presents several challenges to supporting scientific activities. We consider composing DCCs (design and executable) at two levels: manual or semi-automatic composition (using AI techniques). Planning in AI has recently considered workflow composition [10]. Interactive modeling and specification is another new trend, answering the need of customizing and tailoring workflows for a specific goal [12, 20]. In general, most efforts are oriented to business workflows. An exception is [12], a pioneer framework to support interactive composition of domain independent scientific workflows based in ontologies and planning. Our model extends this proposal by supporting interactive and automatic composition of DCCs using planning strategies.

There are several proposals for composition and serialization on the Web via languages such as XPDL (XML Process Definition Language) or WS-BPEL [13]. The latter is based on using a Web services infrastructure. This facilitates the standard utilization of several distributed heterogeneous activities and data sources. We have also adopted WS-BPEL as the standard to export the content of a workflow component, even though we had to extend the language to support specific workflow structures [14].

One of our major contributions lies in reuse. Reuse technologies can be divided in two major groups [3]. One of these groups is based on *composition technologies*, adopted in our reuse approach. Composition technologies consider two players: *components* and *composition*. Components are akin to our executable DCCs, serving as building blocks in the new product construction. Compositions (our design DCCs) connect components establishing relationships between them and defining how they will work together.

The WOODSS architecture also considers these two levels, extending them to workflow building and reuse activities. The DCC model, furthermore, allows seamless integration of mechanisms to reuse design and executable elements, whereas other approaches in the literature require distinct formalisms to handle these problems. The use of this model enables workflows to invoke not only Web services, but also any kind of facility encapsulated within a DCC.

Finally, workflow execution on the Web is not our main concern at this stage. Initiatives such as CHIMERA [1], MyGRID [21, 22] worry about execution aspects. Nonetheless, WOODSS' workflows can be executed by serializing the workflow specification into a WS-BPEL document, which can be submitted

to execution on a WS-BPEL execution engine such as the one in `www.activebpel.org`.

7 Concluding Remarks

This paper presented the ongoing research on the WOODSS scientific workflow framework. Originally conceived for supporting scientific work in environmental planning, it is now being extended to distributed web-based applications for other scientific domains. The core of this work is based on creating repositories containing workflow specifications at several abstraction levels. These specifications are encapsulated in DCCs, which provide a standard search and composition interface based on metadata and semantic annotations linked to domain ontologies.

Scientists can query repositories to reuse and compose workflow elements at the design and execution stages. This allows, for instance, comparing various versions of workflows that solve a given problem.

The main contributions are: (i) discussion of issues concerning a data model that induces a methodology for workflow design; (ii) presentation of the DCC reuse model, which integrates the two different levels of workflow production – specification and execution. We support furthermore several requirements of scientific workflows, such as on-the-fly intervention and modification. The data model and WS-BPEL export facilities have been implemented [14] and DCCs manual composition has been implemented for map management in environmental applications [16]. Ongoing work concerns porting the graphical interface to the Web, creating DCCs for bioinformatics applications and checking the suitability of SHOP2 to composition, first in bioinformatics and subsequently to other domains. Another activity is the study mentioned in Sect. 5.1 of interoperation of business and scientific workflows for agriculture.

Acknowledgments. This work was partially financed by CNPq, CAPES, FAPESP and UNIFACS, CNPq WebMaps and AgroFlow projects, and a Microsoft eScience grant.

References

[1] K. M. Anderson, R. N. Taylor, and E. J. Whitehead Jr. Chimera: hypermedia for heterogeneous software development enviroments. *ACM Transactions on Information Systems (TOIS)*, 18(3):211–245, 2000.

[2] E. Bacarin, C. B. Medeiros, and E. Madeira. A Collaborative Model for Agricultural Supply Chains. In *Proc COOPIS 2004*, volume LNCS 3290, pages 319–336, 2004.

[3] T. J. Biggerstaff and C. Richter. *Reusability framework, assessment, and directions*, pages 1–17. ACM Press, 1989.

[4] M. C. Cavalcanti, R. Targino, F. Baião, S. C. Rössle, P. M. Bisch, P. F. Pires, M. L. M. Campos, and M. Mattoso. Managing structural genomic workflows using Web services. *Data & Knowledge Engineering*, 53(1):45–74, 2005.

[5] L. A. Digiampietri, C. B. Medeiros, and J. C. Setubal. A framework based in Web services orchestration for bioinformatics workflow management. In *Proc III Brazilian Workshop in Bioinformatics*, 2004.

[6] A. J. G. Simpson et al. The genome sequence of the plant pathogen Xylella fastidiosa. *Nature*, 406(1):151–157, 2000.

[7] R. Fileto. *The POESIA Approach for Services and Data Integration On the Semantic Web*. PhD thesis, IC–UNICAMP, Campinas–SP, 2003.

[8] R. Fileto, L. Liu, C. Pu, E. Assad, and C. B. Medeiros. POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture. *VLDB Journal*, 12(4):352–367, 2003.

[9] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning, Theory and Practice*. Elsevier, 2004.

[10] *Proc. of the Workshop on Planning and Scheduling for Web and Grid Services*, June 2004. `http://www.isi.edu/ikcap/icaps04-workshop/` (as of 2005-07-11).

[11] D. Kaster, C. B. Medeiros, and H. Rocha. Supporting Modeling and Problem Solving from Precedent Experiences: The Role of Workflows and Case-Based Reasoning. *Environmental Modeling and Software*, 20:689–704, 2005.

[12] J. Kim and Y. Gil. Towards Interactive Composition of Semantic Web Services. In *Proc. of the AAAI Spring Symposium on Semantic Web Services*, march 2004.

[13] Oasis-Open. OASIS Web Services Business Process Execution Language Technical Committee. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel` (as of 2005-07-11).

[14] G. Z. Pastorello Jr. Publication and Integration of Scientific Workflows on the Web. Master's thesis, UNICAMP, 2005. In Portuguese.

[15] G. Z. Pastorello Jr, C. B. Medeiros, S. Resende, and H. Rocha. Interoperability for GIS Document Management in Environmental Planning. *Journal on Data Semantics*, 3(LNCS 3534):100–124, 2005.

[16] A. Santanchè and C. B. Medeiros. Geographic Digital Content Components. In *Proc. of VI Brazilian Symposium on GeoInformatics*, November 2004.

[17] A. Santanchè and C. B. Medeiros. Managing Dynamic Repositories for Digital Content Components. In *EDBT 2004 Workshops*, volume LNCS 3268/2004, pages 66–77, 2004.

[18] A. Santanchè and C. B. Medeiros. Self describing components: Searching for digital artifacts on the web. In *Proc. of XX Brazilian Symp. on Databases*, October 2005.

[19] L Seffino, C. B. Medeiros, J. Rocha, and B. Yi. WOODSS - A Spatial Decision Support System based on Workflows. *Decision Support Systems*, 27(1-2):105–123, 1999.

[20] E. Sirin, B. Parsia, D. Wu, J. A. Hendler, and D. S. Nau. HTN planning for Web Service composition using SHOP2. *Journal of Web Semantics*, 1(4):377–396, 2004.

[21] R. D. Stevens, H. J. Tipney, C. J. Wroe, T. M. Oinn, M. Senger, P. W. Lord, C. A. Goble, A. Brass, and M. Tassabehji. Exploring Williams-Beuren syndrome using myGrid. *Bioinformatics*, 20(suppl_1):i303–310, 2004.

[22] The myGrid Consortium. myGrid: Middleware for in silico experiments in biology. `http://www.mygrid.org.uk/` (as of 2005-07-11).

[23] J. Wainer, M. Weske, G. Vossen, and C. B. Medeiros. Scientific Workflow Systems. In *Proc. of the NSF Workshop on Workflow and Process Automation Information Systems*, 1996.