VISUAL WORKFLOW EDITORS: A CRITICAL REVIEW FROM USERS' PERSPECTIVE

FLORIAN URMETZER, ASHISH THANDAVAN, VASSIL N. ALEXANDROV
Center for Advanced Computing & Emerging Technologies
University of Reading, PO Box 225, Reading RG6 6AY
[f.urmetzer,a.thandavan,v.n.alexandrov]@rdg.ac.uk

ROB ALLAN
e-Science Center, Daresbury Laboratory
Daresbury, Warrington WA4 4AD
r.j.allan@dl.ac.uk

[Received December 2005 and Accepted March 2006]

Abstract. The aim of this paper is to discuss the state-of-the-art in visual workflow editing tools for scientific applications in distributed and grid computing for the e-Sciences. The structure of the research behind this paper was a large-scale review of literature on several workflow editing tools. These tools were then installed and used to be able to contrast the literature with a user experience. The outcomes are recommendations towards bettering workflow editor interfaces and indications for further research.

Keywords: workflows, visual workflow editors, grid computing

1. Introduction

Workflow management tools support the user in designing, creating and managing the execution of workflows [12] / [13]. They enable the users to describe and perform experimental procedures in an organized, replicable and, most importantly, provable way. The tools are needed for the definition and for the visualization of the processes involved in a computational experiment [15]. There are several projects involved in the development of such visual workflow editors (Kepler, Taverna, Triana, P-GRADE). The main endeavour of these projects is to enable the non-computing specialist to handle distributed computing resources in a user-friendly way through these interfaces. The computing resources are mainly defined as web services and/or Grid technology.

There are several user groups that are making substantial use of distributed technology, for example, Astronomy, Physics and Biology. The Bioinformatics community, for instance, has the need to access different specialized large data-sets and databases, which may be related to one particular disease and compare these to another data-set [8]. These resources are highly specialized databases, which are very expensive to maintain. Therefore, one of the data-sets maybe stored in Germany and another in Japan. When bound together through processors, they can be used virtually as one data-set. Workflow management tools are helping the e-scientist to use such external resources in a flexible way and independent from the IT specialist [12].

The major challenge to these tools is that the user-base for workflow editors is mostly not specialized in computing or in the use of such complex IT systems [14]. Therefore some authors state that there may be a trade off between a highly powerful tool and a target audience that is able to handle it e.g. [7].

This paper details the important outcomes of a wider study. It shows an outline of the arguments presented in the study and concludes in recommendations to better workflow editors and further the research in the field.

2. The tools in detail

This paper will look at four tools in detail. They are Triana, Taverna, Kepler and the P-GRADE Grid portal. These tools are all visual workflow editors and three of them have been created by research projects needing such tools. They differ in the system type - where Triana, Taverna and Kepler are Java applications, the P-GRADE Grid portal is server-based. The visual representation of the workflow is different from tool to tool, as is the quality and method of user interaction.

2.1. Taverna

Taverna is a workflow editing tool which is available from [21]. This tool is a component of the myGrid project which was funded by the Engineering and Physical Sciences Research Council (EPSRC). The development of the tool was mainly driven by the requirements of biologists from the UK's life sciences community [21].

The format of storage of workflows is SCUFL (Simple Conceptual Unified Flow Language). SCUFL is an XML based workflow language. It has been specially developed because the use of a generic, standard language would have not given the opportunity to investigate key aspects and needs for a workflow language in the bio-sciences. The interface of Taverna is based on three main windows: The SCUFL Diagram window, the XSCUFL Window and the SCUFL Model Explorer (see fig. 1. The SCUFL Diagram window displays an overview of the present workflow; this window is a display-only facility and therefore not editable. The graphical display consists of nodes and links.

The nodes can be processors, inputs or outputs. Processors are a transformation entity that take data and process it. These processors can be of six different types as described in detail by [7].

1. WSDL processor - can call a web service defined in WSDL.
2. Soap lab processor - can call a complete Soap lab process.
3. Talisman processors - enables a Talisman task to be processed.
4. Nested workflow processors - are needed to implement child workflows.
5. String constant processor returns a string to an output port; for instance to a processor that needs a constant value for processing.
6. Local processor - enables the user to add local functionality like Java programs.

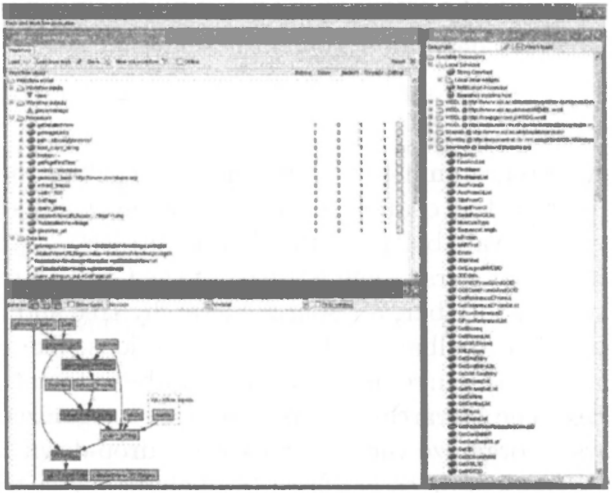

Figure 1. The Taverna interface with the toolbar on top, the hierarchy tree populated with services on the left and the editing area on the right.

The links are either data links (which show the direction of the flow of data) and coordination constraints (which control the execution, for example, of two processors). The interface supports three different visual displays - showing all ports, showing no ports or showing only those ports that have connectivity [8]. This is intended to show details of the services that are available through the tools. Normally the ports differ by the type of input and the type of processing done on them. The visualization is always organized from the top to the bottom, from the input to the outputs [20]. There is also a text window displaying a read-only version of the current workflow.

Finally, there is a reporting facility, which can handle failure reporting and the collection of provenance data. The provenance tool is based on XML document format where the details are presented to the user in a tabular format [8]. This tool has been seen to be very useful throughout the tests performed during the research.

The workflows are edited in the workflow model explorer, which is a hierarchy tree. To add a service, the service is either dragged and dropped into the workflow model explorer or right-clicked on and added. The resource hierarchy tree has a search function at the top of the window, to search for resources in highly populated trees. The linking of the services is accomplished by choosing the output of a service to be linked in the model explorer, right-clicking on it and choosing one of the possible connections that are displayed. The visualization is automatically updated to include the connection and the connection is shown in the workflow model explorer under the Connections tab.

2.2. Triana

Triana is a visual programming environment that enables the user to create workflow graphs from the connection of programming units or components [6][11]. The tool is available from the Triana project [22]. The tool was developed by Cardiff University and is a part of the GridLab project [17]. The interface consists of a tool bar, a resource hierarchy tree and a visual display area (see fig. 2). The toolbar has the main functions (like copy, paste and save) as buttons. The resource hierarchy tree can be automatically populated with web services. The hierarchy tree is at all times organized alphabetically and has six ways to organize the resources via a drop-down menu above the hierarchy tree. The default case, where the default packages are displayed, shows the 'All packages' option, where all resource packages are shown. A 'Show all tools' option displays all the tools and finally tree options show only the data, the input or the output tools. Triana's source recovery tree interface has been described as limiting. It is argued that users may want an alternative or a range of different ways to discover resources [14]. The authors found that the non-availability of a search function for the hierarchy tree slowed down the assembly processes.

It was found that only web services type processors can be displayed. This is supported by White, Jones et al. [14] stating that Triana only processes a limited number of data types.

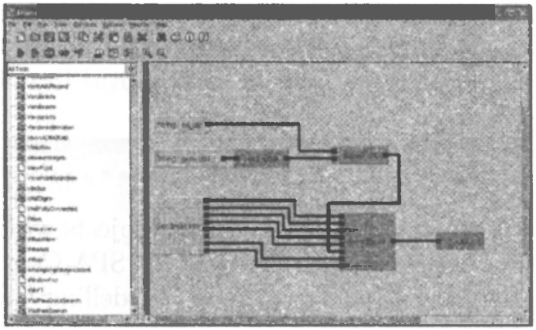

Figure 2. The Triana interface with the three interaction windows, the resources, the workflow builder and the workflow display.

These processors can be dragged and dropped into the workspace and connected together [10]. The connections are done in the display and editing area, by clicking an output and dragging and dropping the output onto an input. The workflow language used to save the workflow and to communicate is the Business Process Execution Language for Web Services (BPEL4WS).

The features of Triana include four major points [6]:

1. Simplified construction of Web Services, which details in a simplified discovery, composition, invocation and publication of services.
2. Execution of composite services on distributed systems.
3. Sensitivity analysis tool which enables the user to do a what-if? analysis.
4. Recording of workflows as well as automated provenance related information.

To have full functionality, Triana has some pre-requisites which are based on web services. These are service discovery methods, service composition methods, transparent execution methods and transparent publishing methods [6]. The system works on the basis of interacting components which are pluggable and modularized [10]. There are two tutorials, titled "Running a Wave Unit Remotely" and "Distributing Units Amongst OCL Servers" supplied with the installation files. The authors tried to follow these tutorials and get then to work but were not successful. One of the authors then found a letter in the users' mailing-list archive stating that the tutorials are out of date and do not work. It is therefore not obvious which features work, apart from the tested web services execution.

When executing a basic workflow, Triana shows the progress through little boxes in the workflow processors that turn black when active. Therefore the user can check on progress. There is no prevenance collection or metadata entry like in Taverna. The Triana XML file has been found to be much longer compared to SCUFL as well.

2.3. Kepler

Kepler was built by a collaboration of different projects which included SEEK (Science Environment for Ecological Knowledge), SPA Center (Scientific Process Automation), Ptolemy II (Heterogeneous Modelling and Design), GEON (Cyber infrastructure for the Geosciences) and ROADNet (Real-time Observatories, Applications, and Data Management Network). These projects found the similar need for the development of an open source tool to create, edit and manage scientific workflows. Kepler is available free of charge from the project's home page [18].

Kepler is a workflow enactment tool that has been built on top of Ptolemy II, which is a software tool supporting heterogeneous, concurrent modeling and design [16].

Kepler's interface is structured in a toolbar with the most important functions in button form, a resource recovery tool on the left hand side of the screen, including a search option and finally an editing area on the right hand side of the screen (see fig. 3).

Keplers strength are described in three parts [2]

1. It enables the user to define models of computation precisely, including the process networks model, which is dataflow oriented.
2. It has a modular programming approach that is oriented towards the production of reusable components.
3. It is described as an easy-to-use graphical user interface that allows the user to create complicated workflows in an easy manner.

The application can define a row of different processors. For example, Kepler is able to handle database queries to major database types, it handles Globus jobs, web service definition language and finally XSLT & Xquery which are both XML editing types.

Kepler is able to process different plug-ins, called Actors. These define the flow of information or the process of the workflow. The modularity of the interface is based on the hierarchical abstraction. Therefore complex models maybe shown in one block to make the model visually more structured. These blocks may be internal or external processes [3]. The Actors and other resources for workflows are stored on the left hand side of the workflow editor interface in the form of a hierarchical tree. The parts needed are dragged and dropped onto the workflow area where they can be linked up to each other by dragging the output of one actor or processor to the input of another one. All processors can be defined in detail through the interface.

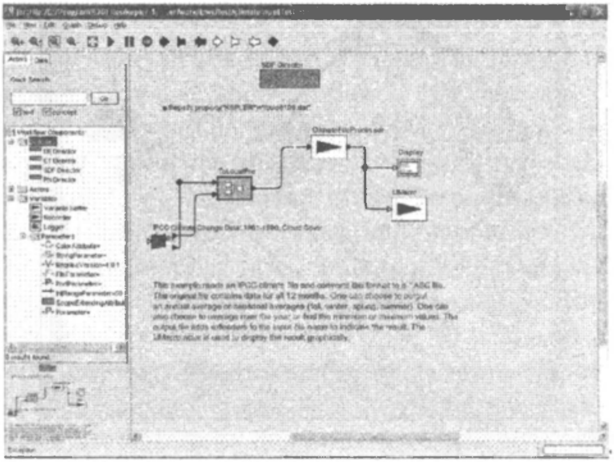

Figure 3. The Kepler interface with the tool bar on top, the hierarchy tree populated with services on the left and the editing area on the right.

The workflows are saved using an XML language called Modeling Mark-up Language (MoML). This XML language does not however include versioning and indexing of information and is described to be the only way of provenance as well [1].

2.4. P-GRADE Grid Portal

The Laboratory of Parallel and Distributed Systems in the MTA-SZTAKI Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary has developed P-GRADE, a workflow solution for complex grid applications. Their tool is intended to help the user in designing, executing, monitoring the different stages of execution and visualizing the progress. While P-GRADE was originally a stand-alone application, the P-GRADE Grid Portal is a portal-based solution and therefore runs on a web server. The portal version will be discussed in this paper. The files required to install the portal are available from MTA-SZTAKI [23]. The installation is however very complicated, because of operating system requirements and some rather specific pre-requisite software - Condor v6.4.2 (or v6.4.7), Globus Toolkit v3.2, GridSphere v1.0.1, Apache Ant v1.6.1, Java 2 Platform SDK v1.4.2_04, Apache Tomcat v4.1.27 and C libraries (libpng, libgd) [19]. The

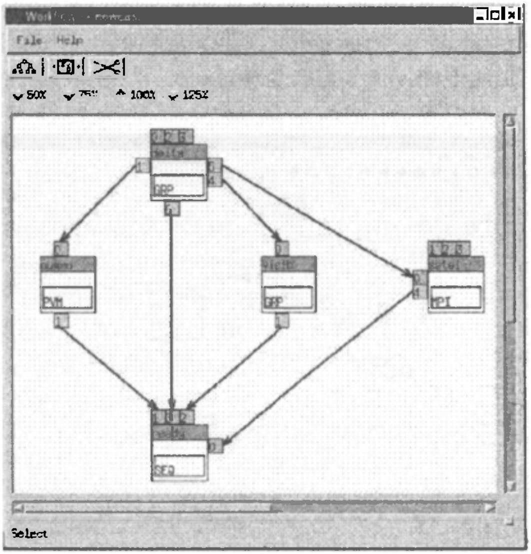

Figure 4. The P-GRADE interface showing the Petri net based workflow representation. Each processor has input and output ports. The green are input ports and the gray are output ports.

portal's interface is based on the P-GRADE graphical programming environment [5] (see fig. 4). The components of a workflow can be either sequential (C / Fortran) or parallel (PVM (Parallel Virtual Machine) / MPI (Message Passing Interface) / P-GRADE job). P-GRADE uses the hybrid GRAphical Process NEt Language (GRAPNEL) internally.

Communication between jobs is expressed via input and output ports which are defined when creating the port on the processor. The ports are then linked up to each other by dragging the output port to the input port.

The P-GRADE Grid Portal has a monitoring and visualisation facility as well as interoperability with different systems on heterogeneous Grid platforms, for example, Condor or Globus [5]. They can however be stored on the server site as well as the user end.

As the P-GRADE Grid portal is portal-based, the user only has to install a version of Java Webstart on his/her computer. The client accesses the portal via a standard web browser and logs in. Java Webstart will then download and start the application enabling the user to work with the system. All the proxy credentials for grid sites are managed via a MyProxy server through the portal. Therefore the portal-based system can be seen as very much user-friendly and very advantageous for grid systems.

3. The tools compared and analyzed

In this section, the authors would like to look at the tools from a comparative point of view. Therefore they would like to mention the major points encountered throughout the wider study and briefly discuss them.

One issue that is discussed in the literature from the workflow tools' projects, is that of the representation of the workflow. Direct Acyclic Graph (DAG) is used in Taverna and Direct Cyclic Graphs is used in Triana; these are discussed throughout the tools' literature and have been briefly touched upon in this document. A pictorial representation of single workflow entities has been discussed by other authors like Hernandez and Bangalore [4]. The arguments presented in the literature were not found to be supportive by the authors for the definition of a good representation. This was mainly due to the lack of publications detailing with comparative user testing and user opinions on the graphical representation of workflows. The authors agree that there may not be only one solution and therefore there may be the need to implement multiple interface representations of workflows to choose from.

The author strongly recommends testing workflow interfaces with potential users of the e-Science community in a structured way. This would then enable a further definition of interface needs and preferences of users.

The storage and presentation of services and processors available to the user was found to be in the form of a hierarchy tree in all tools. This form of organization was seen as usable by the authors, but only if the hierarchy tree is not overfilled with services. Additionally the organization in multiple layers and most important search mechanisms for the hierarchy tree are highly recommended by the authors.

However there should be another form of service retrieval and keeping, because the hierarchy tree has been discussed as not ideal by the research community [14]. There is therefore scope for further research into visualising the retrieval and keeping of services available to the user.

The editing mechanisms range from editing in a hierarchy tree that displays the workflow entities, like in Taverna, to editing the entities together graphically. There is however no evidence in the literature that supports any of the editing methods. The authors argue that a multiple approach to editing facilities is probably the best solution, leaving the decision to the user. A multiple way of editing workflows is however not implemented in any of the workflow tools. This means that there should be other ways of linking processes than those explored by the projects.

There are several features that must be included in e-Science workflow tools to support their user community.

1. There is a need to have meta data to describe the workflow to other users. This may include outcomes and links to publications and search words to be used in repositories.
2. This meta data should be included in the workflow language, because of the danger of a disconnection of the description and the workflow script.
3. The use of provenance collection tools was confirmed to be useful. Therefore information of the workflow execution is collected during the runtime of the workflow and later presented to the user for storage. This provenance information can then be used to validate the experiment at some later date.
4. Server-based tools are found to be advantageous, because proxy and networking problems can be overcome and access to computing resources can be managed from the server side by computing specialists rather than by users on their individual computers. In addition, deployment problems when updating versions of the interface and changing of the computer on the client side are overcome. The problems mentioned are not overcome by using an enactment engine. However the enactment engines as well as a server-based tool overcome the problem of long-running workflows being able to be executed remotely from the user's computer.
5. A common workflow scripting language would allow a workflow created by one editor to be opened and edited in another. Users can then use their preferred workflow editor knowing that they can share their workflow descriptions with their peers.
6. An easy install process would also be highly advantageous.

4. Conclusions

In conclusion, there are two ongoing problems that re-occurred throughout the research done. The first was the missing definition of an e-Science workflow script language and the second was the total absence of work towards user tests with workflow interfaces within the e-Science community.

REFERENCES

[1] Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludaescher, B. and Mock, S.: *Kepler: Towards a Grid-Enabled System for Scientific Workflows.* Workflow in Grid Systems Workshop in GGF10 - The Tenth Global Grid Forum, Berlin, Germany, 2004.

[2] Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B. and Mock, S.: *Kepler: An Extensible System for Design and Execution of Scientific*

Workflows. 16th International Conference on Scientific and Statistical Database Management (SSDBM), Santorini Island, Greece, 2004.

[3] BHATTACHARYYA, S. S., BROOKS, C., CHEONG, E., DAVIS, J., GOEL, M., KIENHUIS, B., LEE, E. A., LIU, J., LIU, X., MULIADI, L., NEUENDORFFER, S., REEKIE, J., SMYTH, N. TSAY, J., VOGEL, B., WILLIAMS, W., XIONG, Y ZHAO, Y. and ZHENG, H.: *Volume 1: Introduction To Ptolemy II.* Brooks, C., Lee, E.A., Liu, X. Neuendorffer, S., Zhao, Y. and Zheng, H. eds. Ptolemy II: Heterogenous Concurrent Modeling and Design In Java, 2004.

[4] HERNANDEZ, F., BANGALORE, P., GRAY, J. and REILLY, K.: *A Graphical Modelling Environment For The Generation Of Workflows For The Globus Toolkit.* Workshop on Component Models and Systems for Grid Applications, Held in conjunction with ICS 2004: 18th Annual ACM International Conference on Supercomputing, Saint-Malo, France, 2004, Springer Verlag.

[5] LOVAS, R., DÓZSA, G., KACSUK, P., PODHORSZKI, N. and DRÓTOS, D.: *Workflow Support for Complex Grid Applications: Integrated and Portal Solutions.* 2nd European Across Grids Conference, Nicosia, Cyprus, 2004.

[6] MAJITHIA, S., SHIELDS, M., TAYLOR, I. and WANG, I.: *Triana: A Graphical Web Service Composition and Execution Toolkit.* IEEE International Conference on Web Services 2004.

[7] OINN, T., ADDIS, M., FERRIS, J., MARVIN, D., GREENWOOD, M., GOBLE, C. WIPAT, A., LI, P and CARVER, T.: *Delivering Web Service Coordination Capability to Users.* 2004, http://decweb.ethz.ch/WWW2004/docs/2002p2438.pdf

[8] OINN, T., ADDIS, M., FERRIS, J. MARVIN, D., SENGER, M. GREENWOOD, M., CARVER, T GLOVER, K., POCOCK, M. R., WIPAT, A. and LI, P *Taverna: a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, 20 (17), 30453054.

[9] SHIELDS, M.: *Triana User Guide*, Cardiff, 2004,
http://www.trianacode.org/docs/index.html

[10] TAYLOR, I. SHIELDS, M. and WANG, I.: *Resource Management of Triana P2P Services.* Weglarz, J., Nabrzyski, J., Schopf, J. and Stroinski, M. eds. Grid Resource Management, Kluwer, 2003.

[11] TAYLOR, I., SHIELDS, M., WANG, I. and PHILP, R.: *Grid Enabling Applications Using Triana. in Workshop on Grid Applications and Programming Tools,* Seattle, 2003, GGF Applications and Test beds Research Group (APPS-RG). GGF User Program Development Tools Research Group (UPDT-RG).

[12] THURSTON, C.: *Go with the workflow.* Scientific Computing World, September/October 2004 (78).

[13] WORKFLOW MANAGEMENT COALITION: *The Workflow Management Coalition Specification: Terminology & Glossary.* Winchester, 1999, http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v1013.pdf

[14] WHITE, R., JONES, A., PITTAS, N., GRAY, A., XU, X., SUTTON, T., BROMLEY, O., CAITHNESS, N., FIDDIAN, N., CULHAM, A., BISBY, F., BHAGWAT, S.,

BREWER, P YESSON, C. and WILLIAMS, P *Building a Biodiversity Problem-Solving Environment.* All Hands Meeting (AHM 2004), Nottingham, 2004.

[15] WROE, C., LORD, P MILES, S., PAPAY, J. MOREAU, L. and GOBLE, C.: *Recycling Services and Workflows through Discovery and Reuse.* www.mygrid.org, Manchester and Southampton, 2004, http://www.ecs.soton.ac.uk/ lavm/papers/ahm04-wroe.pdf.

[16] WWW.BERKELEY.EDU: Ptolemy II, 2004, http://ptolemy.eecs.berkeley.edu/ptolemyII/.

[17] WWW.GRIDLAB.ORG: Gridlab - a grid application toolkit and testbed. 2005.

[18] WWW.KEPLER-PROJECT.ORG: Kepler Project, 2004.

[19] WWW.LPDS.SZTAKI.HU: How to install P-GRADE PORTAL, SZTAKI, Budapest, 2004, http://www.lpds.sztaki.hu/pgportal/manual/install/ PORTAL_installation_an_Introduction.html.

[20] MYGRID. *Taverna User Guide*, 2004.

[21] TAVERNA.SOURCEFORGE.NET: Welcome to Taverna, 2004

[22] WWW.TRIANACODE.ORG: The Triana Project, Cardiff University, Cardiff, Wales, UK, 2003.

[23] WWW.LPDS.SZTAKI.HU: MTA-SZTAKI Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary