An ECA Rule-based Workflow Design Tool for Shanghai Grid[1]

Lin Chen Minglu Li Jian Cao Yi Wang
School of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200030, P.R. China
{chen.lin, wangsuper}@sjtu.edu.cn {li-ml, cao-jian }@cs.sjtu.edu.cn

Abstract

Service integration, the ultimate goal of Shanghai Grid, has created a necessity for more efficient and effective workflow infrastructure. Workflow design tool is one core component that assists in defining workflow processes as well as providing a graphical representation of process model through which users can have an easier understanding of the semantics of process models. This paper discusses a workflow design tool for Shanghai Grid that has the following features. First, it combines graphical process representation and ECA rules in controlling Grid workflow process. Second, integration adapter of the Grid Workflow system is presented to facilitate the composition of all possible services. Finally, this tool supports hierarchical graph definition that allows workflow coursing and refinement. In this way, it extends the scope of resource sharing and offers a well-layered view for complicated workflow. Design principle and implementation details of workflow design tool for Shanghai Grid are also given in this paper.

Keywords: ECA rules, Grid Workflow, Service Composition

1. Introduction

Shanghai Grid, one of the top five grand projects in China, is an ongoing city grid project to enhance the digitalizing of city. It aims at constructing metropolis-area information service grid infrastructure and establishing an open standard for widespread upper-layer applications from both research communities and official departments [1]. As many other grid project, Shanghai Grid tries to address the challenge of integrating services spread across distributed, heterogeneous dynamic virtual organisation [2]. Meanwhile, Shanghai Grid has its own features, that is, it principally serves average people, which distinguish it from other grid project dedicated to scientific domain.

To achieve this goal, three levels of interoperability and integration need to be dealt with. At the first level, individuals, organizations and business companies can publish their information or resources as Grid services or Web services, which can be accessed directly by public through the Grid Portal [1]. Then at the second level, a collaborative workflow infrastructure allowing users to describe the interactions between services and compose new workflow out of existing services is required to build complex computing experiments or applications consisting of thousands of tasks and services. Last but not least, these workflows can be published as special services and act as building blocks, which allows user to reuse or share with other users. These are just what the third level supposed to think about.

Therefore, a workflow infrastructure that realizes the function of both the second and third level is a necessity for Shanghai Grid. Since the purpose of Shanghai Grid is to benefit the

[1] This paper is supported by 973 project (No.2002CB312002) of China, grand project (No.03dz15027) and key project (No.025115033) of the Science and Technology Commission of Shanghai Municipality

scientists as well as unskilled user who may lack the expertise, a graph-based workflow design tool is required to provide a GUI environment, which assists users quickly, and easily to create new workflow model from scratch, load existing workflows to perform a complicated service-oriented computing over the Grid, or compose more complicated workflow based on legacy workflows [3]. Besides the modeling tool, a workflow execution engine and web-based grid portal are also indispensable parts.

The particular focus of this paper is on workflow design tool. Other issues, such as workflow engine and user portal in Shanghai Grid are only discussed briefly. In this paper we designed and implemented a workflow-modeling tool for Shanghai Grid, which has three main features. First, it applies ECA rules to control workflow process. ECA Rules has a sound theoretical basis and can be very flexible in model design and process execution [4]. Second, integration adapter of the Grid Workflow system is presented to facilitate the composition of all possible services. Finally, Finally, this tool supports hierarchical graph definition that allows workflow coursing and refinement. It extends the scope of resource sharing and offers a well-layered view for complicated workflow.

The remainder of this paper is structured as follows. In Section 2, we provide detailed description of the ECA rule-based workflow design tool. Section 3 demonstrates the effectiveness and validness of it through the combined usage with workflow execution engine and user portal. Related work is addressed in Section 4. Finally, Section 5 closes the paper with some brief concluding remarks and future research directions.

2. Workflow Design Tool

Many complex applications require the creation of a collaborative workflow. In Scientific field, such applications include climate modeling, high-energy physics, structural biology and so on. For average people, various information services, travel plan etc. are such applications. By using the workflow approach, various customized services can be provided through the coordinated use of numerous distributed and heterogeneous resources.

Nowadays, there are three main approaches to achieve the definition of workflow. It may be based on scripting languages, on graphs, or on a mixture of both. [5]. Although the scripting language approaches may be very convenient for skilled users, they are not really intuitive. So it is not applicable for the Shanghai Grid whose goal is to server the public.

In our approach, we use an ECA rule-based workflow approach to realize the graphical definition of workflows with only few basic graph elements. Fig.1 shows a screenshot of the Workflow Design Tool, a Java application providing a graphical user interface for the composition of various services, such as Web Service, Grid Service, legacy applications.

Workflow Design Tool for Shanghai Grid is very flexible and simple to use. It supports the drag and drop to introduce new components to the workflow model. To assist the user to compose a complicated workflow, a powerful validation mechanism is supported. It can verify the input information on the spot. A special event algorithm is proposed and realized to deal with the validation of model when transform the graphical representation into ECA rules. In order to provide more knowledge support of components, the right-click menu is used for showing the properties of the selected components. The Output of the Workflow Design Tool is saved in database and it can be loaded from the database. We plan to save the

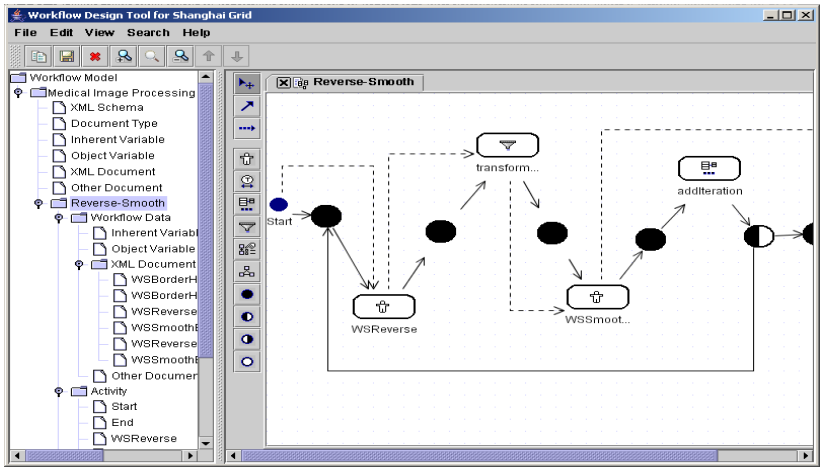

Figure 1 A screenshot of the Workflow Design Tool for Shanghai Grid. The Workflow Design Tool includes a menu bar and tool bar (*top*), a navigation tree to show the hierarchical structure of all the entities in the workflow (*left*), an element tool bar for drawing different workflow nodes and links (*middle*), and a composition panel for ECA rules based workflows (*right*). The Workflow Design Tool supports drag and drop to introduce new components to the workflow model.

output of workflow model as XML format file. This function is under development.

In the following sections, we will discuss in details the features of Workflow Design Tool for Shanghai Grid.

2.1 The anatomy of ECA rule-based workflow

A workflow model contains different kinds of components. Fig.2 shows the class view of all the main components and their relationships. Generally speaking, a workflow model can be divided into four categories of components, that is, activity, link, logic node and data object.

- Activity

A workflow model consists of at least one StartActivity and EndActivity acting as the entrance and exit of a workflow. To compose various services available in Shanghai Grid environment, several specific activity nodes are designed for invoking Web Services, Grid Services and legacy applications. Users can use these nodes to specify the services they want to invoke. When composing workflows in an open world, we have no control over the data types used by the component services. It is entirely likely that a service identified by a user as being suitable does not use the same type as the preceding service in the workflow. SetValueActivity, TransformXMLActivity are therefore designed to bridge the gap of data type between different services. Besides, there are CompositeActivity and SubworkflowActivity nodes that support hierarchical graph definition and thus allow workflow coursing and refinement. In this way, it offer the further level resource sharing and provides a well-layered view for complicated workflow.

- Logic node

There are four kinds of logic node in our design tool, that is, AND-AND, AND-OR, OR-AND, and OR-OR. These logic nodes are logical connectors used to control the logic relationships among different activities nodes. Each logic node contains a trigger event and at least one ECA rule branch. Detailed description of ECA rule and how to transform graphical model into ECA rule is in the following section.

- Control flow and data flow

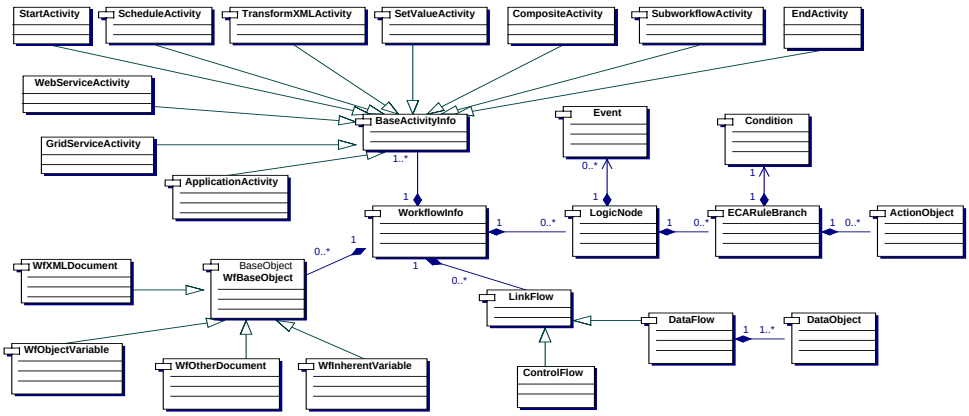

Figure2 Class view of the components of ECA rule-based Workflow

Control flow links the activity node and logic node. It passes the event to subsequent logic node and thus contributes to the control of the process. Data flow, however, specifying the input and output information exchanged between activities, can set the read/write right on each information and the event that triggers the data flow.

- Data object

In our workflow modeling, we define four categories of data definition for the control and exchange of data, that is, *Inherent Variable*, *Object Variable*, *XML Object* and *Other Document Object*. *Inherent Variable* can be utilized to set guarding condition and act as a decision point. *XML Objects* are generally used to represent the input and output parameters of services. *Object Variable* is one field or data item extracted from *XML Object* or *Other Document Object*. It is usually needed when assigning values from one field of XML Object to another. Other Document Object is an abstract representation of documents formats data except XML document, such as word, pdf, rtf and so on. Through this definition, our workflow modeling supports various data formats that can be exchanged between different services.

2.2 ECA Rules in workflow

An ECA (Event-Condition-Action) model is originally used in active database systems. When an event occurs and a condition turns out to be true, the active database execute a corresponding action. It's an interesting approach to use ECA rules for controlling workflow processes [4].

In our workflow design tool, we provide a graphical process representation for human user to grasp the actual process conveniently. Meanwhile, we transform the graphical model into a set of ECA rules during modeling, so that our workflow execution engine is able to control its execution automatically.

Table 1 shows the ECA rules for typical workflow patterns. The event is a triggering statement, which specifies the event associated with an activity. Each activity is associated with six possible events in our workflow design tool, that is, Initialized, Started, EndOf, Overtime, Aborted, Error. Table 1 only summarizes the ECA rules triggered by EndOf Event. Other event-driven rules are similar to these. When designing a workflow model, user can specify the proper triggering event and design the following flow, including normal flow and exception flow. The condition is a logical expression that must be satisfied in order to activate the action part. The action involves the Activity that need to be executed or the Event needs to be triggered.

Table1 ECA Rules for Basic Workflow Pattern

Workflow Pattern	Example	Event	Condition	Action
Serial		EndOf(A)	null	Activity.B
AND-Join		EndOf(A)AND EndOf(B)	null	Activity.C
AND-Split		EndOf(A)	null	Activity.B AND Activity.C
OR-Join		EndOf(A) OR EndOf(B)	null	Activity.C
OR-Split		EndOf(A)	*a* == *true*	Activity.B
			b == *true*	Activity.C
OR-Join OR-Split		EndOf(A) OR EndOf(B)	*a* == *true*	Activity.C
			b == *true*	Activity.D
Iteration		EndOf(A)	*a* == *true*	Activity.A
			b == *true*	Activity.B

Note: AND-AND Logic Node; AND-OR Logic Node; Activity; OR-AND Logic Node; OR-OR Logic Node; Control Flow; *a*, *b* Condition Expression

The Iteration workflow pattern, for example, is implemented as the related ECA rules in Table 1.When activity A is finished, an EndOf(A) event occurs. The workflow execution engine detects this event, and then it checks the condition part. If the condition part says that *a* is equal to true, the workflow engine begin to execute the action part according to ECA rule, that is, the Activity A. Otherwise, if the *b* equals true, the workflow engine leaves the iteration and executes Activity B. The execution of other ECA rules is similar to this Iteration Pattern. In this way, the ECA rule-based workflow can be executed automatically.

2.3 Workflow Composition

The workflow design tool for Shanghai Grid supports three methods for user to construct workflows. Firstly, user can create a workflow process from scratch with the user-friendly GUI. The second method enables users to search previously shared workflows, then reuse them or make small modifications. Finally, users can import existing workflows as part of their workflows. It's generally called sub-workflow. In this way, this tool extends the scope of sharing and reuse.

Our workflow system currently supports three categories of services: Web Services, Grid Services and Java applications. A service repository containing the available services' metadata exists. When user plan to creates an activity invoking service, a service selection panel appears which loads all the available services definition from the service repository and allows user to select proper service and operation. However, when no suitable service exists in the service repository, user can resort to integration adapters to locate new services. There are three kinds of integration adapter presented to facilitate the composition of those three categories of possible services.

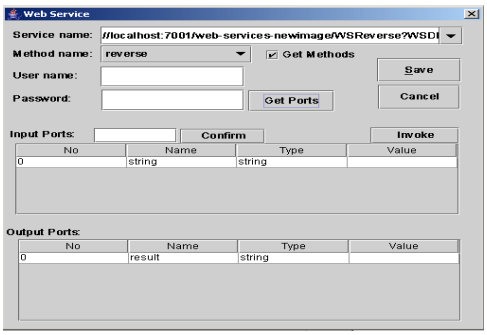

Figure 3 Integration Adapters for Web Service

Fig 3 shows the integration adapter for Web Services. User can input a service WSDL Documents links, then the adapter explore the WSDL document, extract services, port types,

input/output message's schemas and save them within the service repository for facilitating share and reuse of available services. Another integration adapters work in similar way.

3. Case Study

For illustration purpose, the example detailed below will use a simpler medical image-processing domain. The user wants to process a medical image through a series of algorithms. Each image-processing algorithm is wrapped by Web Service. So user needs to construct a workflow, which composes these image-processing Web Services according to user's requirement. Fig 1 displays a medical image processing workflow, which invokes the "Reverse" and "SmoothBox" services iteratively until a factor is satisfied. Except the two activities invoking Web Services, there are other assistant activities, which responsible for the transformation of XML document and the assignment of data values for the match of different services' input/output parameter. Then this workflow applies serial and iterative workflow pattern to realize the workflow composition.

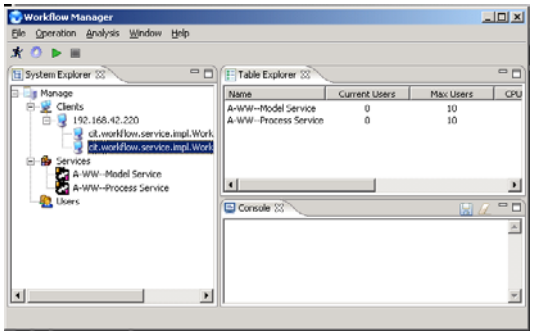

Figure 4 ECA rule-based Workflow Engine

When finishing the design of the medical image processing workflow, this workflow can be submitted through user portal to the workflow engine for running. Fig 4 shows the ECA rule-based workflow engine for Shanghai Grid. It's supposed to deploy and start model service and process service firstly, which make preparations for the execution of forthcoming workflow request. When the workflow request is coming, workflow engine obtains a copy of the workflow model from database and then takes charge of the specific invocation and routing according to ECA rules.

User portal is illustrated in Fig 5. From this portal, user can specify the workflow and the medical image needs to be processed, then submit the request to workflow engine and wait for the result. Fig 5 shows the primitive image (left) and the result image (right).

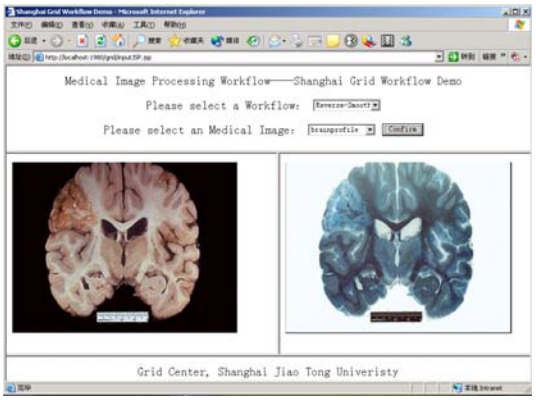

Figure 5 Workflow Processing Result

This ECA rule-based workflow design tool together with the workflow engine has already been a vital part of Shanghai Grid and utilized frequently. It proves to be very flexible and powerful, which can satisfy the disparate need of scientists as well as average people.

4. Related Work

Although Grid workflow systems do not have a long history compared to other grid technologies in the Grid Computing community [6], there are quite a few works related to workflow tools in Grid environment that have been proposed and used by researchers. WebFlow [7] and Common Component Architecture (CCA) [8] are two early projects in this field. WebFlow pays emphasis on integrating applications whereas CCA focuses more on composing disparate and coarse-grain

high performance components into a running application. Many current Grid workflow systems, such as GridAnt [9], Triana [10], Symphony [11], XCAT [12], GridFlow [13], and Ptolemy II [14] inherited the strategy of components/Services composition. All these Grid workflow systems provide a graphical user interface and a script-like language for users to organize and describe the workflow process. Some Other Grid workflow systems, such as McRunjob [15], Traverna [16], GriPhyN [17], and Kepler [18] focus on the integration of programs/applications. UNICORE [19] and DAGMan [20] are resources management system and process planning/scheduling system that could also provide workflow control functionality in Grid environment.

As to the workflow process modeling, most Grid workflow systems adopt the Directed Acyclic Graphs (DAG) [10] [11] [18] [19]. However, DAG has no cycle circuits in its model, it's not applicable to explicitly express loops. Another modeling approach, Petri Nets, is gradually being introduced into Grid workflow systems [6] [21] [5]. They use Petri Net modeling method to describe user tools and depict the characteristics of dynamism of Grid job scheduling.

In this paper, we propose a method of using ECA rules for controlling Grid workflow process. User can conveniently construct a graphical process model based on ECA rule. The workflow design tool transforms the graphical model into a set of ECA rules, so that the workflow execution engine is capable of controlling its execution automatically. Besides, integration adapter is proposed to facilitate the composition of all possible services and sharing resources conveniently. Existing Grid workflow systems, however, do not provide such approaches.

5. Conclusion and Future Work

This paper describes a workflow design tool to realize the service integration and workflow reuse in Shanghai grid. The goal is to solve the complicated business logic or scientific application problems, which need interaction between services and compose new workflow out of existing services. The design principle and implementation details are given in this paper. An image process workflow model designed by this tool is implemented and combined with the workflow engine and user portal to illustrate the validation and usefulness of this architecture.

In the future, we plan to apply agent technologies which responsible for the dynamical discovery of suitable services at run time so that the process can be more flexible and it further simplify the design process, conduct research on transaction based exception-handling mechanism in Grid environment. Additionally, the ongoing work focuses on providing support for automatic workflow generation by the aid of artificial intelligence planning technologies.

Reference

[1] Y Li, M Li, J Yu, L Cao, a workflow services middleware model on Shanghai Grid, Proceedings of Services Computing, 2004 IEEE International Conference on SCC 2004, pp. 366-371

[2] Ian Foster, Carl Kesselman, Jeffrey Nick, and Steven Tuecke. The physiology of the grid: an open grid services architecture for distributed systems integration. http://www.globus.org/research/papers/ogsa.pdf, January 2002.

[3] M.H. Eres, G.E. Pound, Z. Jiao, J.L. Wason,etc. Implementation and Utilization of a Grid-enabled Problem Solving Environment in Matlab, Future Generation Computer Systems (In Press)

[4] Joonsoo B, Hyerim B, and etc. Automatic Control of Workflow Processes Using ECA Rules, IEEE transactions on knowledge and data

engineering, vol. 16, No.8, pp. 1010-1023, August 2004

[5] Hoheisel, Andreas, "User Tools and Languages for Graph-based Grid Workflows", In Proceedings of Workflow in Grid Systems Workshop in GGF10, at Berlin, Germany, March 2004.

[6] Zhang, Shaohua, Ning Gu, and Saihan Li, "Grid workflow based on dynamic modeling and scheduling", In Proceedings of International Conference on information Technology: Coding and Computing (ITCC2004), Vol. 2, at Las Vegas, Nevada, April 5-7, 2004, pp.35-9.

[7] Bhatia, D., V. Burzevski, M. Camuseva, G. Fox, W. Furmanski, and G. Premchandran, "WebFlow - A Visual Programming Paradigm for Web/Java Based Coarse GrainDistributed Computing", Concurrency: Practice and Experience, Vol. 9, No. 6: pp. 555-77, 1997.

[8] Common Component Architecture, Available from http://www.cca-forum.org.

[9] Laszewski, Gregor von, Kaizar Amin, Mihael Hategan, Nestor J. Zaluzec, Shawn Hampton, and Albert Rossi, "GridAnt: A Client-Controllable Grid Workflow System", InProceedings of 37th Hawaii International Conference on System Science, at Island of Hawaii, Big Island, Jan. 5-8, 2004.

[10] Shields, Matthew and Ian Taylor, "Programming Scientific and Distributed Workflow with Triana Services", In Proceedings of Workflow in Grid Systems Workshop in GGF10, at Berlin, Germany, March, 2004.

[11] Lorch, Markus and Dennis Kafura, "Symphony - A Java-based Composition and Manipulation Framework for Computational Grids", In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002), at Berlin,Germany, May 21-24, 2002.

[12] Krishnan, S., R. Bramley, D. Gannon, M. Govindaraju, J. Alameda, R. Alkire, T. Drews, and E. Webb, "The XCAT Science Portal", In Proceedings of Supercomputing (SC2001),at Danver, Colorado, Nov. 10-16, 2001.

[13] Cao, J., S. A. Jarvis, S. Saini, and G. R. Nudd, "GridFlow: Workflow Management for Grid Computing", In Proceedings of 3rd International Symposium on Cluster Computing and the Grid, at Tokyo, Japan, May 12-15, 2003, p. 198.

[14] The Ptolemy II Project, Available fromhttp://ptolemy.eecs.berkeley.edu/ptolemyII.

[15] Graham, G. E., D. Evans, and I. Bertram, "McRunjob: A High Energy Physics Workflow Planner for Grid Production Processing", In Proceedings of Computing in High Energy and Nuclear Physics, at La Jolla, California, March 24-28, 2003.

[16] Oinn, Tom, Matthew Addis, Justin Ferris, Darren Marvin, Mark Greenwood, Tim Carver,Anil Wipat, and Peter Li, "Taverna, Lessons in Creating a Workflow Environment for the Life Sciences", In Proceedings of Workflow in Grid Systems Workshop in GGF10, at Berlin, Germany, March, 2004.

[17] Deelman, Ewa, James Blythe, Yolanda Gil, and Carl Kesselman, Workflow Management in GriPhyN, In Grid Resource Management: State of the Art and Future Trends, Edited by Nabrzyski, Jarek, Jennifer M. Schopf, and Jan Weglarz, 2003.

[18] Altintas, Ilkay, Chad Berkley, Efrat Jaeger, Matthew Jones, Bertram Ludaescher, and Steve Mock, "Kepler: Towards a Grid-Enabled System for Scientific Workflows", InProceedings of Workflow in Grid Systems Workshop in GGF10, at Berlin, Germany, March, 2004.

[19] Erwin, D. W. and D. F. Snelling, "UNICORE: A Grid Computing environment", Lecture Notes in Computer Science, Vol. 2150: pp. 825-34, 2001.

[20] DAGMan (Directed Acyclic Graph Manager), 2002,Available from http://www.cs. wisc.edu/condor/dagman.

[21] Grid-Flow: A Grid-Enabled Scientific Workflow System with a Petri Net-Based Interface