An ECA Rule-based Workflow Design Tool for Shanghai Grid

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 workflow infrastructure. Workflow design tool is one core component that assists in defining workflow processes as well as providing graphical representation of process model through which users can have an easier understanding of the semantics of process. 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.

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]. Therefore, a workflow infrastructure that realizes the composition of available services to build complex computing experiments or applications function is a necessity for Shanghai Grid. Since the purpose of Shanghai Grid is to benefit the 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 to create new workflow model from scratch and load existing workflows.

The particular focus of this paper is on workflow design tool. Other issues, such as workflow engine and user portal 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. 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.

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 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

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. [2]. Although the scripting language approaches may be very convenient for skilled users, they are not really intuitive. 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. It supports the drag and drop to introduce new components. A powerful validation mechanism is
supported to assist user compose complicated workflow correctly. In order to provide more knowledge support, the right-click menu is used for showing the properties of the selected components.

The output of the model is saved in database and it can also be loaded from the database. In the following subsections, we will discuss in details the features of Workflow Design Tool for Shanghai Grid.

![Figure 1](image-url) A screenshot of the Workflow Design Tool for Shanghai Grid.

2.1. The anatomy of ECA rule-based workflow

A workflow model contains different kinds of components. 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.

- **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**
    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, ObjectVariable, XML Object and Other Document Object. Inherent Variable can be utilized to set guarding condition and act as a decision point. XML Objects generally represents the input and output parameters of services. Object Variable is one field or data item extracted from XML Object. It is usually needed when assigning values. 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 executes a corresponding action. It’s an interesting approach to use ECA rules for controlling workflow processes [3].

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, 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, users specify the proper triggering event and design normal flow or exception flow. Condition is a logical expression that must be satisfied in order to activate the action part. Action involves the Activity that need to be executed or the Event needs to be triggered.

### 2.3. Workflow Composition

Three categories of services is supported, that is, Web Services, Grid Services and legacy applications. A service repository containing the available services’ metadata exists. When user plan to create 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, user can resort to integration adapters to locate new services.

Fig 2 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. Other 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. Fig 1 displays a medical image processing workflow, which invokes the “Reverse” and “SmoothBox” services iteratively until a factor is satisfied.
After the design, this workflow can be submitted to the workflow engine for running. Fig 3 shows the ECA rule-based workflow engine. 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.

![Figure 3 ECA rule-based Workflow Engine](image)

User portal is illustrated in Fig 4. 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).

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.

![Figure 4 Workflow Processing Result](image)

### 4. Related Work

As to the workflow process modeling, most Grid workflow systems adopt the Directed Acyclic Graphs (DAG) [4] [5]. 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 [2]. However, Petri-net based workflow is hard to model uncertainty, thus it is not adaptive to the undeterministic situations which usually appears in service composition.

There are also a lot of works related to ECA rule-based workflow systems. [3] proposes a systematic method of reducing an ECA rule-based process into a simple form. [6] report the use of ECA rules to support workflows in product development. However, they are limited to traditional applications without mention the issues of web service or grid service composition.

### 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.

In the future, we plan to apply agent technologies which responsible for the dynamical discovery of suitable services at run time. Additionally, the ongoing work focuses on providing support for automatic workflow generation by the aid of artificial intelligence planning technologies.

### 6. Reference