Oriental Journal of Computer Science & Technology Vol. **2**(1), 01-10 (2009)

XML-Based Process Definition Tool for a Workflow Management System (WFMS)

AMIN Y. NOAMAN and NADA M. AL-MANI

Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, P.O. Box 80221, Jeddah 21589 (Saudi Arabia)

(Received: February 02, 2009; Accepted: March 25, 2009)

ABSTRACT

A Workflow is an automation of a business process. In general, the process model describes the activities performed and their relations. It could be defined by using the process definition tool, which is one of the Workflow Management System (WFMS) components. Technically, fixed workflow structure is a major problem that could be faced by any WFMS. It is lacking of reusability and flexibility needed during runtime, i.e. once a workflow has started; changes to system and process structure are no longer feasible. This research proposes a framework that supports designing and implementing of a general purpose process definition tool with a graphical user interface. The business process modeled by this tool presented internally in an XML schema file. The elements of this schema are the activities of the model which will be parsed and transformed to a graphical notation whenever loading the model. In addition, the defined process in whole could be used as an activity in other business model into the interface. Then, the process will be automated on process engine that has empowered with the required flexibility to dynamically modifying the structure of a running instance.

Keywords: XML, Workflow Management system.

INTRODUCTION

With the growth of the business enterprises, the need to coordinate, schedule and monitoring the business process is growing as well. Workflow management is a powerful tool to improve efficiency of organizations. A workflow management system (WFMS) is "a software system that transforms the explicit representation of a workflow model into an internal and executable format and provides an operational environment for workflow model execution and the administration and monitoring of workflow model instances.[1].

Workflow technology has been used for decades to automate various processes. In fact, workflow technology has begun in the mid of 1970s. The first approaches towards the automation of business processes were part of the office automation prototypes at Xerox Parc (Officetalk, developed by Skip Ellis and Gary Nutt) and Wharton (SCOOP, developed by Michael Zisman). Research during the 1980s placed more emphasis on process models[2]. During the last few years workflow has been focus on activity in terms of products, standards, and research work. It is no longer just some sort of planned document routing. Currently workflow technology and process support lies at the center of modern information systems architectures. This research is focusing on modeling aspects of WFMS.

The generation of workflow management system is acquired by meeting business experts to get the system requirements and formulating proper workflow specifications. Because this task traditionally is carried out by programmers, who have different view to the business process from business side, it has to be repeated until achieving the right requirements. By focusing on number of features specified by the business side, the final WFMS could be released. But technically, it has a

fixed structure of workflows. It lacks of reusability and flexibility needed during runtime, i.e. once a workflow has started; changes to system and process structure are no longer feasible.

Using XML as an internal format for the process model could alleviate number of problems. The activities of the model will be presented as XML elements. When loading the schema into the designer, XML elements will be transformed into the corresponding visual notation. The fixed workflow structure is solved by creating flexible workflow architecture. Number of processes models in the organization could be changed by adding, deleting or modifying existing models without affecting the others execution. In addition, a specific instance of a process definition could be edited during execution without suspending or modifying the other instances of the same process definition.

This paper is organized as follows. In section 2, a general background about WFMS and business process modeling is presented. Section 3, will review some of the existing WFMS tools. Then the proposed XML-based process definition tool will be presented in section 4. Future works will be discussed in section 5. Finally, section 6 concludes this work.

Background Work

Workflow Management System

Simply, workflow management is controlling flow of the work in any domain. It manages the execution of business processes. Usually the process definition which is a representation of what should happen is created, and then the automation of this model is taking place. There is a distinction between process definitions and process instances, the process definition deals with the types of business data or resources while the process instance deals with specific instances of business data[3].

Business Process Modeling

Any organization should describe its business process into a series of models. The model represents the real world tasks usually take place within the organization to reach a certain business objectives. Each process and task contains some activities. An activity is a single logical step in the process, for example making a payment or not making a payment is an activity. It could be manual or automated activity[3,4,5]. The automated activities then could be carried out by a workflow management system[5]. Fig. 1 is an example of a workflow model.

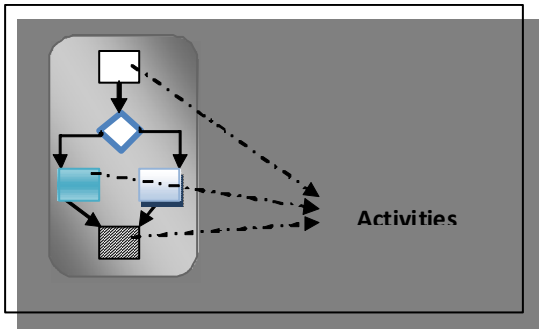

Fig. 1: A workflow model

Generally, business modeling is an essential step for any organization for several reasons. First, it helps facilitate a common understanding of the business that the organization is engaged in. Second, it helps to identify areas of business that can be improved either through automation or for business process reengineering. Third, it helps to highlight business areas that organization doesn't yet deal with or that the organization is very weak in[3].

Business modeling is achieved in WFMS by using process definition tool. It is one of the WFMS's components.

Process Definition Tool

The process definition tool is used to capture and create the process definitions that are going to be automated by the workflow engine. Typically this tool could be a text editor, or graphical interface that allows users to build the process by dragging and dropping objects onto a design canvas. These objects could be viewed as a library of reusable components that contains the commonly used entities within any process definition. This library is called a meta-model[6]. In order to give an accurate meaning to a business process, modeling the building blocks of the business model is an essential process when developing a process

definition tool, i.e. meta-modeling. It is a model of a model. The meta-model serves as a language that defines the syntax and semantics for activities that construct any model[7]. However, the business process model could be viewed from various perspectives or aspects depending on the kind of information required. From a high-level point of view, workflow aspects can be characterized by simple questions:

1) Functional Aspect: What is being done?
   The functional aspect holds the sequence of activities to be carried out, i.e. says what should be done during a particular workflow instance execution[7].
2) Behavioral Aspect: When and under which conditions will it be done?
   The behavioral aspect of workflow, also known as dynamic aspect, specifies when the process is being performed. It provides sequencing and control information about the process[7,8]. To achieve this, different constructors that permit flow of execution control are used.
3) Information Aspect: Which data will be used?
   The information aspect is represented by data flow between workflow activities. It is an important aspect of workflow since it allows workflow management systems to control the transfer of data as generated or processed by workflow activities[6].
4) Organizational Aspect: Who will perform it?
   The Organizational aspect represents who is performing the process through its application and after defining the organizational structure[7].

   These four aspects are the key aspects of workflow modeling. Depending on the application domain addressed, different additional workflow aspects can be defined. For example, applying the flexibility aspect to workflow models if the application needs modification of the models during runtime.

Related Works

Different workflow systems investigated several areas of computer science. Each of these systems provided a workflow solution based on specific techniques to the problem that it tries to solve. The work presented in this paper proposes a flexible framework with no fixed number of business processes. The flexibility of the framework will be applied also on execution time where the model of a running instance could be modified graphically without affecting other running instances of the same business process. The idea discussed in this paper relates to many existing ideas but has distinguished features. For example, Globus[11] is a framework that is mainly about defining the business process by using graphical user interface. It supports Java code generation for these models, and allows programmers to automate the generated code. In contrast to the current work, the target user for this framework is the programmer rather that the business user. In addition, the models are not flexible to be modified during the execution.

DynaFlow is a research that describes a Dynamic Workflow Model (DWM) and a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes[12]. It has a graphical editor for creating process models using DWM as the underlying model and enables the specification of dynamic properties associated with a business process model. Compared to the current work, DynaFlow supports modeling and flexible execution environment. But their work doesn't provide an instance-level modification. They rely on the concept of schema modification, where workflow schema will be modified before loading and executing instances, and whenever the workflow schema modified, all the referenced instances will be modified as well, so there is no real instance-level modification. Add to this, their work doesn't support graphical user interface for modifying running instances. The need for a graphical based instance model modification is as important as defining process models graphically.

XML is used in some researches as a specification to develop a methodology for constructing a distributed workflow systems[13]. The specifications are integrated with an agent-based distributed middleware for workflow implementation. This middleware enforces coordination and security constraints specified in the XML description of the plan. The agent based middleware interprets the XML description of the environment, and transparently initiates agents to other users' nodes when operations need to be performed to ensure

workflow requirements. Other researches such as the one done in[14], uses XML to provide the workflow management system with updateable XML views of relational data. In this research, XML supports extendible generic business data access in workflow model from any source in a uniform way.

Xml-based Process Definition Tool For (Wfms)

Several researched has used XML in the field of workflow management system for different purposes. In current work, XML is used as the underlying file format for the process definition to support the flexibility needed either during building workflow system or even during executing instances.

System Components

The structure of the solution is specified and planned along with the architecture and the behavior. The main components of the model are the process definition tool, the form builder tool, and the workflow engine tool (process engine and dynamic modification) see Figure 2. According to their functionalities, these components are arranged into two modes, design mode and run mode.

The model scenario is that the business analyst will define the data entry forms and the required processes in the design mode. The roles that are authorized to initiate processes will be defined as well. Then in the run mode, the user, process initiator, will select a process to start. The system will start the process if the signed in role is authorized to start the process. In the run mode also, the dynamic modification could take place for a running instance. Fig. 2 demonstrates the model scenario. It is enumerated to reflect the scenario.

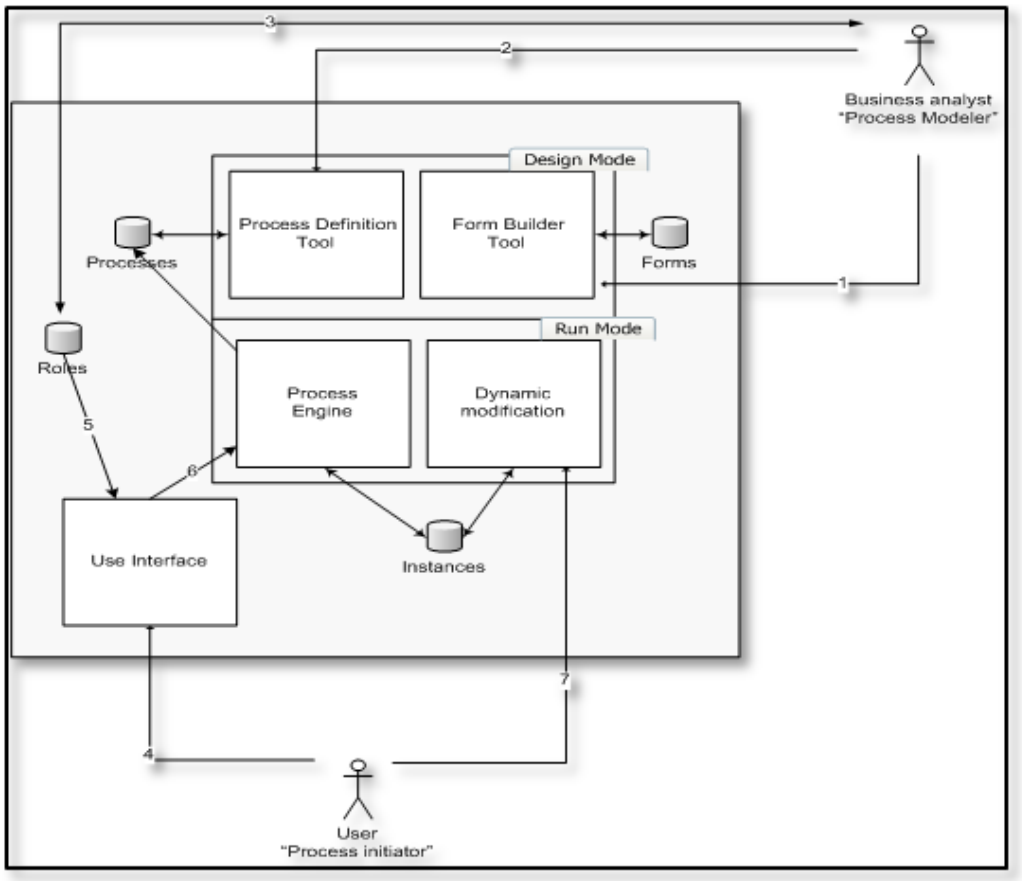

Fig. 2: Overview of the system components

System Implementation

As mentioned, the model mainly consists of two modes, the design and run modes. The design mode supports design and persists of forms and workflow models. In particular, it provides facilities to view and modify forms and workflow models. The run mode, on the other hand, contains a runtime engine that coordinates the execution of workflow processes model. Furthermore, the runtime supports dynamic modification for a running instance. The following subsections are further described the design mode and the run mode.

The Design Mode

Fig. 3 shows the design mode. It contains three main tools: the form builder, the process definition tool and the role definition. In this mode, the data entry forms and the business processes are designed. The accessible roles to initiate a process are defined as well.

The form builder supports the functionality of designing the data entry forms. The generated file is XML file that contains all the added controls with their properties. Fig. 4 shows designing a simple

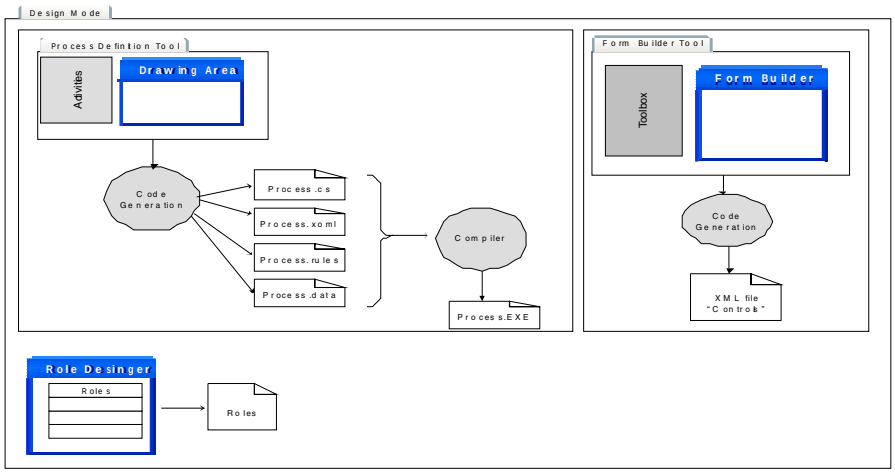

Fig. 3: Design mode - an extended model

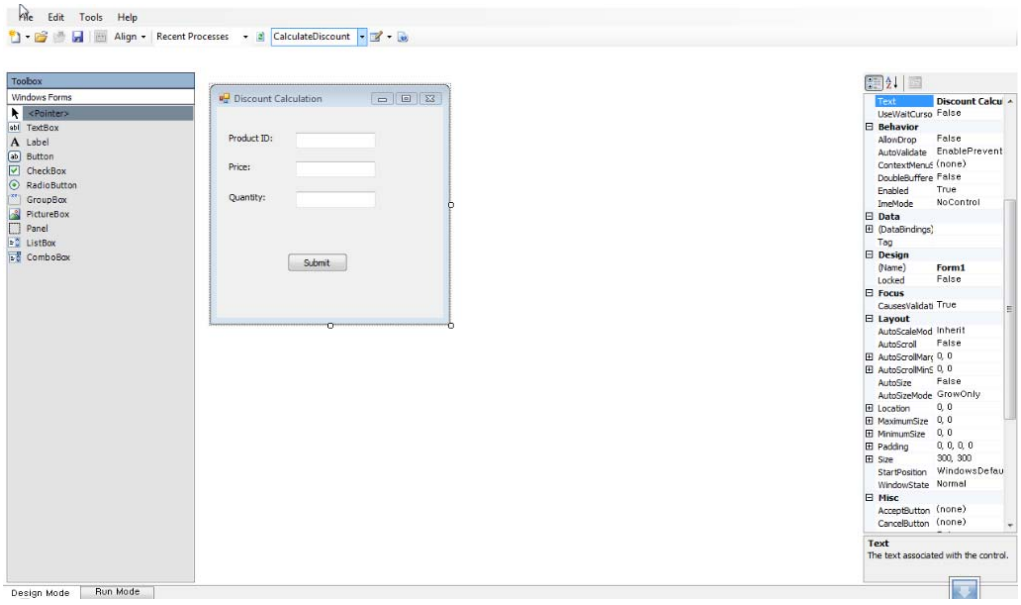

Fig. 4: The Form Builder

business process using the form builder. The presented business process calculates the discount of a specific order depending on predefined business rules.

The process definition tool is the main tool that supports designing of process definition. It consists of two main components design area (drawing area) and activities toolbox. It supports different functionalities such as authoring and editing workflow models. The tool also includes a compilation feature for validating and compiling the authored models. There are three major functionalities of the tools:

1) **Loading activities toolbox:** It is used to display the activities that can be dropped into the designer. To create and load a toolbox, the first step is to implement the Toolbox Service and integrating it with a designer. The next step is to look for items and activities then put them in the toolbox container. The activities are placed in external text file and loaded during runtime. The activities toolbox is not fixed as the user is allowed to add an activity to the toolbox by importing (*.dll) file of this activity without the need to reinstall the application.
2) **Loading Design area:** One of the advantages of windows workflow foundation that make it a good candidate is that it provides API for implementing a process designer. The classes DesignSurface, Workflow Designer Loader and Workflow View make up the designer surface and help in managing the designer components aspects of workflow.
3) **Code generation:** The tool will generate different files to represent the business process model. When the user chooses to save the process, code-beside and a serialized representation of the process definition will be generated by the code generation modules. Four files will be generated by the process definition tool, Extensible Application Markup Language (XAML) or Extensible Orchestration Markup Language (XOML), Code-beside, rules and data files.

Recalling the discount calculation example, Figure 5 shows the business process model that containing the activities required for calculating the discount then show up the result. It is a simple business process that has activities such as the policy activity containing the business rules that govern the discount calculation and an activity to show up the result in a message box to the user. For example, if the required quantity is more than ten, then the discount will be 20%, otherwise it will be 10%.

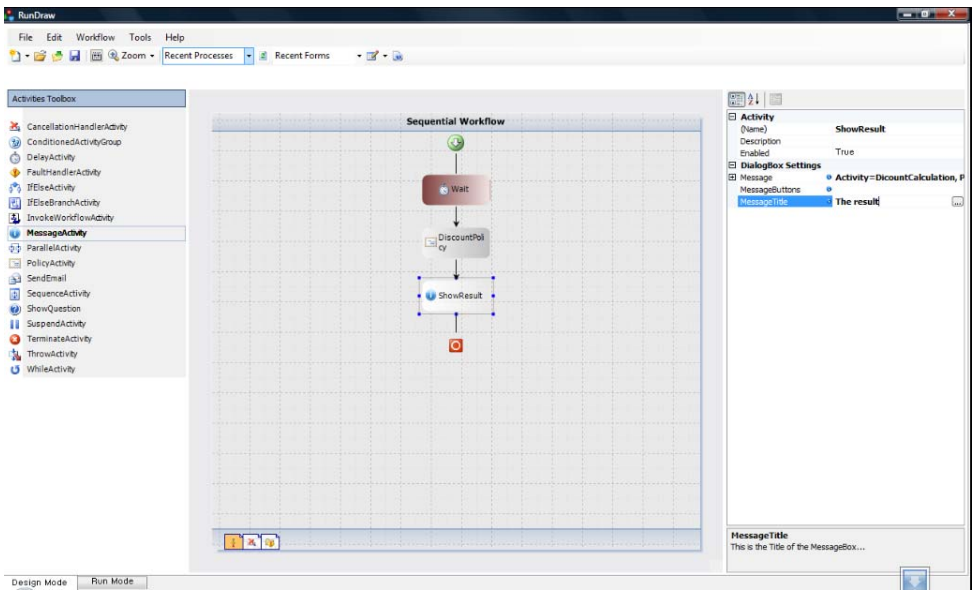

Fig. 5: Calculating discount business process

The Run Mode

Beside the design mode, the run mode acts as the workflow engine for the application. It provides the functionalities needed for executing business process, monitoring and modifying running instances. Figure 6 shows the run mode model. It consist of the process engine and the dynamic modification. The run mode maintains a list of the available processes and let the user select and start one. If the user is authorized to initiate the process, the startup form, if exists, will be launched to gather the required data. The process will start execution and the data will be passed by runtime to the workflow model.

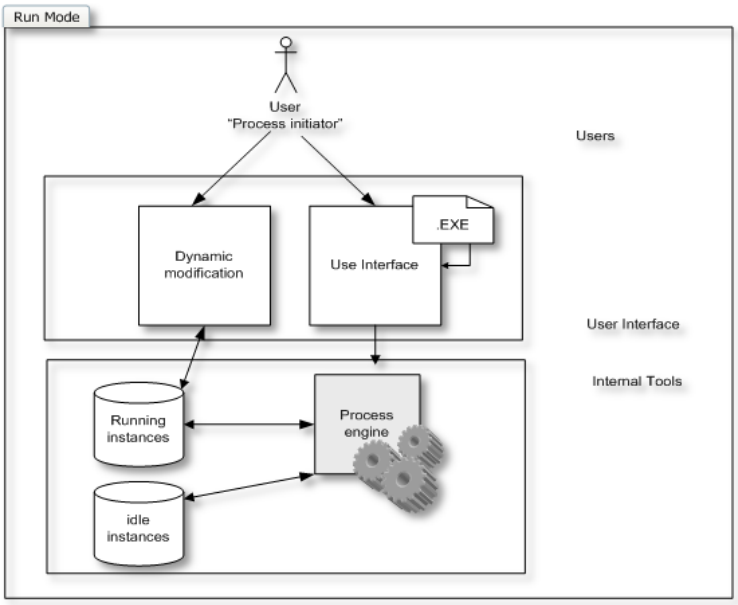

Fig. 6: Run mode – an extended model.

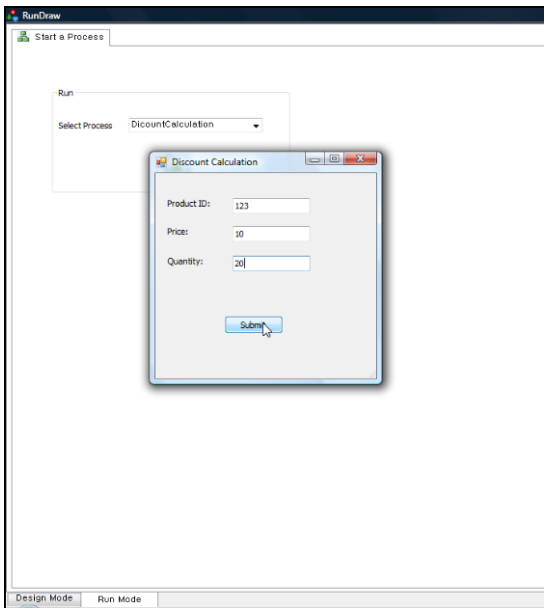

Fig. 7: Running discount calculation business process

Fig. 7 is a snapshot from the discount calculation example that represents the startup form after requesting to start executing a new instance of the business process.

The dynamic modification in the run mode means that you can add and remove activities from a workflow, either on a per-instance basis, so you can add an activity to a running workflow instance or on a per-process manner where modifying, removing or adding a new workflow model to the workflow project.

Windows Workflow Foundation provides APIs that make dynamic modification available at instance level by inserting or deleting activities or changing the properties of existing activities. Through dynamically modifying individual instance of a business process, the downtime that would occur in typical procedural code due to recompilations will be reduced. Furthermore, workflow adaptation occurs on the level of workflow instances, which reaches far beyond the flexibility one would have with procedural workflow coding. The dynamic modification methodology consists of the following two phases:

Phase (1): Defining WorkflowChanges object

At the stage of applying dynamic modification, the black box of the workflow definition has to be opened in order to apply changes. WorkflowChanges class contains a set of proposed changes for the running workflow instance. WorkflowChanges object needs to be obtained by passing in the IRootActivity, the running workflow process definition, to the constructor[9].

Phase (2): Changing the transient workflow

Transient workflow maintains the workflow definition as activities collection. Adding a new activity to a transient workflow is accomplished by the activities collection's Add and Insert methods. Let's look to the transient workflow as an XML object, and it is not more than this. The XML elements are the workflow activities. So adding a new activity during runtime is as adding new XML element to the document. However, dynamic modification is an asynchronous operation, so the best way to ensure that update goes in at the right time is to apply when the workflow is "idle". Therefore, the run mode will not permit any changes until suspend the required instance. In addition, the ApplyWorkflowChanges method could be applied to a single instance of a workflow. They are not permanent changes to a given workflow type and will not be propagated to the current or future instances of that type.

Number of researches had been done on the field of workflow flexibility. One of these researches is the one that is done in[10]. The researchers stated that the need for a graphical based workflow model modification is as important as defining process models graphically. So, the proposed design for dynamic modification tries to load the running instance into a designer by using the same loading designer algorithm with some modifications. It gets the current workflow definition for the running instance from the workflowRuntime by calling GetWorkflowDefiniton, and then passes it to the workflowloader to act as the current process definition. The changes done on the model will be monitored, and a list of added, modified and deleted activities will be maintained. Getting up-to-date definition by flushing the designer will help in knowing the index, or the position, of the changes. When the user chooses to apply the changes, this list will be added to the transient workflow and the final model will be validated and applied, by calling ApplyWorkflowChanges, to act as the referenced model for the current running instance.

Fig. 8 Shows the progressing of an instance of the discount calculation business process. The right panel contains the drawn business process and the set of activities that are being executed are listed in the left panel. If the process executed already, it will be italic and signed by one to show that it has been executed once. When dynamic modification is required, the process should be suspended first then the modification could take place in any activity that is waiting to be executed like changing in the discount policy for this current customer to be 50% even if the quantity is less than[10].

Future Works

Model analysis and optimization are considered as one of the main future works. The concept of dynamic modification could be applied

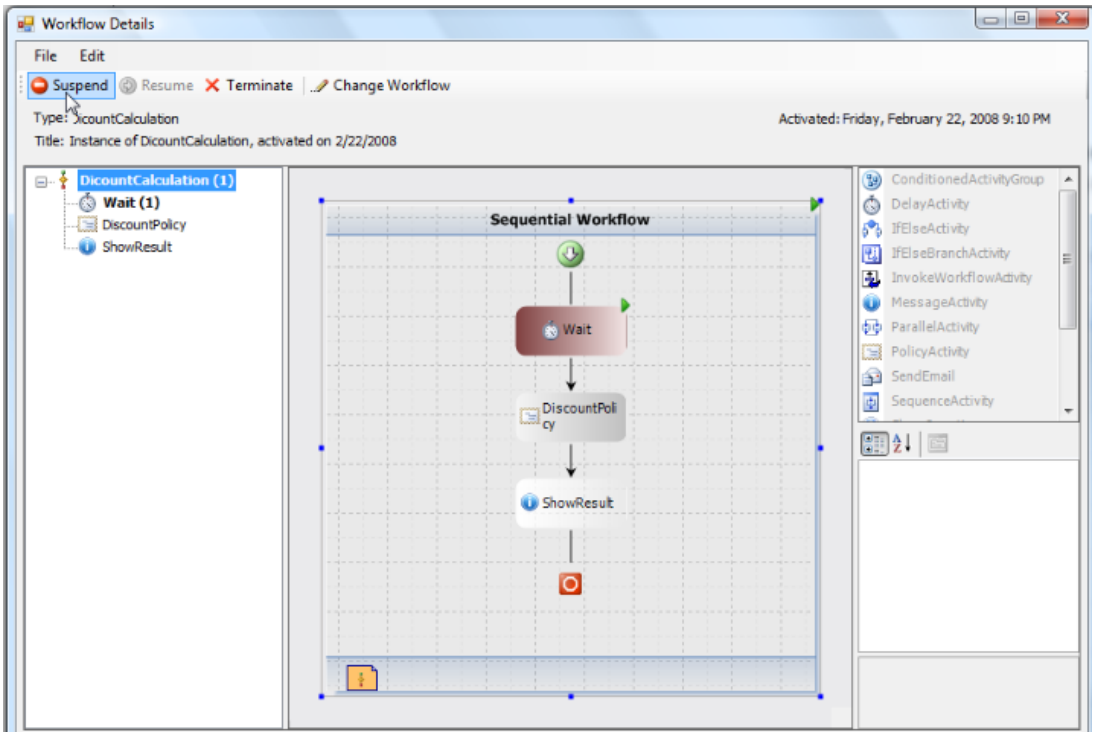

Fig. 8: A running business process.

on the model multiple times, then analyze the results to get the optimal path for the process, i.e. process optimization. Model optimization tries to evaluate the efficiency of the flow route and mark potential improvements, such as the removal of steps, or the reengineering of a part or the whole of the flow to shorten execution time and to improve efficiency of business process. Another improvement that could be done on the current work is to apply the dynamic modification to solve the problem of versioning. When the original process definition is modified, a new executable format version of the process (process.exe) will be generated by the compiler. But the running instance will still run with the old definition. Therefore, the benefit of dynamic modification could be applied by dynamically updating all running instances, if applicable, to reflect the new process definition. And finally, there is a plan to support exporting the current process definition to one of the standard format, ex BPEL, which gives the flexibility to the process definition to be used by other engines.

CONCLUSION

Workflow management systems (WFMSs) have been introduced to support modeling, monitoring and controlling the execution of the workflow. In this paper, mainly aspects related to the modeling area were addressed. The process models in a workflow management system are faced with several problems either during the build-time or run-time. One of these problems is getting the right definition of the process. Another problem is the ability to modify the process model during runtime.

The primary contribution of this work is to provide a comprehensive, flexible framework for modeling and executing workflow process definitions. The generation of the process definition is achieved by using a Graphical User Interface based tool to model and execute the business process. In this case, the business experts can detect errors in the modeling easily and earlier. The

generated file format that represents the process definition is based on XML. It was proved that using XML as an internal format to the process definition will give that system the required flexibility either during building the workflow project or even during running instances. Whenever the process definition is loaded into the interface, the XML file will be parsed and each activity will be transformed to the corresponding graphical notation. Any business process could be modified, add new business process or delete existing without affecting the executing of the other business process in the organization. In addition, the system has the reusability feature, where any defined workflow model could be embedded and used by other as an XML element in the process model schema. XML format also will give the required flexibility during runtime. The user is able to modify a running instance by blocking the current instance, make the required changes to the corresponding XML definition and finally apply the changed and resume the blocked instance from the point it were stopped.

REFERENCES

1. Emin C. Dengi: An improved model and architecture of workflow process management, *Ph.D thesis*, the Middle East technical university, (1998).
2. A. Ailamaki, Yannis E. Ioannidis, Miron Livny: Scientific Workflow Management by Database Management, In the 10th International Conference on Scientific and Statistical Database Management, pp. 190-199, (1998).
3. Hafedh Mili, Guitta Bou Jaoude, Éric Lefebvre, Guy Tremblay, Alex Petrenko: Business Process Modeling Languages: Sorting Through the Alphabet Soup. Montreal Information Research Center, Canada, (2004).
4. G. Alonso, D. Agrawal, A. El Abbadi, and C. Mohan.: Functionality and Limitations of Current Workflow Management Systems. IEEE Expert, 12(5), (1997).
5. Michael zur Muehlen: Managing Business Process, American Conference of information System, New York, (2004).
6. M. Kradolfer: A Workflow Metamodel Supporting Dynamic, Reuse-Based Model Evolution, Ph.D. Dissertation, Department of Information Technology, University of Zurich, (2000).
7. Simeon Petkov: A Multi-Meta-Model Process Execution Environment, Master Thesis, The Vienna University of Technology, (2005).
8. S. Petkov, E. Oren, A, Haller: Aspects in Workflow Management, Digital Enterprise Research Institute, National University of Ireland, Galway, (2005).
9. Todd Kitta, Professional Windows Workflow Foundation, Wiley Publishing, Indianapolis, USA (2007).
10. B. De Smet, K. Steurbaut, S. Van Hoecke, F. De Turck, B. Dhoedt: Dynamic workflow instrumentation for Windows Workflow Foundation, The Second International Conference on Software Engineering Adv. (ICSEA'07), Cap Esterel, French, (2007).
11. Francisco Hern´ andez; Purushotham Bangalore; Jeff Gray and Kevin Reilly: A Graphical Modeling Environment for the Generation of Workflows for the Globus Toolkit, International Conference on Software Engineering, 1 - 5, (2005).
12. J. Meng, Stanley Y.W. Su, H. Lam, A. Helal, J. Xian, X. Liu, S. Yang: A Dynamic Inter-Organizational Workflow Management System, *Int. J. Business Process Integration and Management*, **1**(2): (2006).
13. Anand R. Tripathi, Tanvir Ahmed, Vineet Kakani and Shremattie Jaman: Implementing Distributed Workflow Systems from XML Specifications. Technical Report, Department of Computer Science, University of Minnesota, (2000).
14. Johann Eder, Marek Lehmann, Nick Amirreza Tahamtan, "Generic Data Access in XML Based Workflow Management Systems" Proceedings of The 18th International Conference on Advanced Information Systems Engineering. CAiSE'06 Forum, Luxembourg, (2006).