

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Web-enabled Architecture of Workflow Management System for Heterogeneous Environment

¹Khurram Shahzad and ²Khalid Rashid

¹Department of Computer Sciences, ²Faculty of Applied Sciences,
International Islamic University, Islamabad, Pakistan

Abstract: In this study architectural framework of workflow management system for heterogeneous and distributed environment has been proposed which incorporates web-enabled independent interface for clients to execute workflows. Some of the drawbacks and limitations of the traditional approaches are discussed; then architecture of flexible and platform independent simple workflow management system has been presented which is based on Java and internet technologies. Workflow engine and clients are implemented in Java. Workflow models are stored in relational database and the workflow engine accesses these models using JDBC interface. Standard browsers are used as web based clients to access the workflow system via HTTP protocol.

Key words: Business process, workflow, activity, process execution, workflow engine, transition, workflow participant

INTRODUCTION

Workflow management is one of the areas that in recent years have attracted the attention of many researchers, developers and users. It has primarily evolved from the need to understand, to organize and, often ultimately, to automate the processes on which a business is based. Workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve an overall business goal. This computerized facilitation or automation of a business process, in whole or part is called workflow. Where as a system that completely defines, manages and executes “workflows” through the execution of software whose order of execution is driven by a computer representation of the workflow logic is called Workflow Management System^[1].

A workflow is an abstraction of business process. It consists of activities, which correspond to individual process steps and agents, which execute these activities. An agent may be an information system (e.g., a database system), a human (e.g., a customer representative) or combination of both (e.g., a human using a software program). The workflow specification may also specify execution durations for both the activities and the process itself^[2].

WORKFLOW MANAGEMENT

The basic concepts of workflow management system can be best introduced using the definitions provided by

the reference model of the Workflow Management Coalition (WFMC), an international organization leading the efforts to standardize workflow management product. It defines workflow as:

The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules^[1].

Workflow management includes a build time and a run time dimension imposing different requirements each^[3]. The build time requirements comprise all issues related to modeling of control and data flow, like graphical or formal representation, ease of use, readability, abstraction and modularity, or correctness (Fig. 1).

The run time requirements are related to the activation, enactment and termination of workflows. The system should be extendible to be able to handle future unforeseen application scenarios. Workflow definitions may change over time and thus workflows should be dynamically customizable and adaptable. Continuously growing application areas and increasing organizational coverage call for a scalable system. The system should be open that it can run in a distributed heterogeneous computing infrastructure^[3].

Workflow management aims at modeling and controlling the execution of process in both business and scientific applications. It has gained increasing attention recently, since it allows combining data-oriented view on applications, which is traditional for an information system, with a process-oriented one in which activities and their occurrence over time are modeled and supported properly^[2].

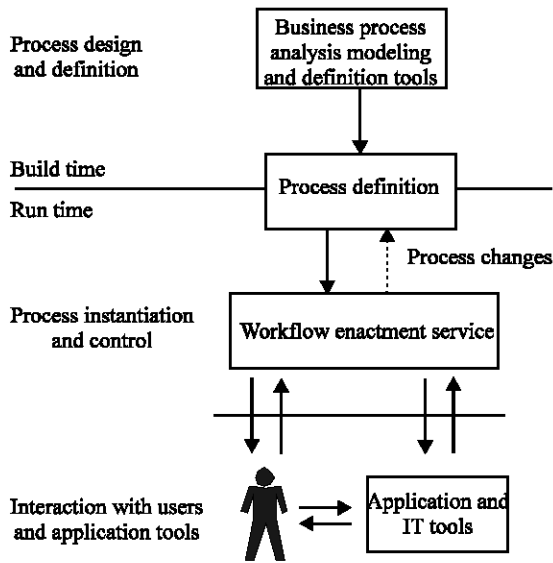


Fig. 1: Workflow system characteristics

CURRENT STATE OF WORKFLOW MANAGEMENT SYSTEMS

According to the Workflow Management Coalition (WFMC) reference model^[1], a workflow consists of an engine, application agents, and administration and monitoring tools (Fig. 2). The process definition tool is visual editor which is used to define the specification, i.e., the schema, of workflow process. The same schema can

be used for creating multiple instances of the same business process at a later time. The workflow engine and the tools communicate with a workflow database to store and update workflow relevant data, such as workflow schemas, statistical information, and control information required to execute and monitor the active process instances^[2].

LIMITATIONS OF EXISTING SYSTEMS

The state of the art in workflow management has been determined so far by the functionality provided in commercial systems. Paradoxically, this has been the major source of limitations. Many products were developed without clear understanding of the user requirements and these products were quiet unprepared to meet the demands placed upon them by eager users. Issues such as performance, scalability or reliability are hardly ever considered in these areas, an unfortunate characteristic inherited by the workflow products.

Current workflow systems are not designed to be embedded into earlier applications easily. This causes problem for large scale enterprise deployments, independent software vendors and system integrators who need comprehensive business process management facility for their applications. Current homegrown, quick and dirty workflow engines are dead-end.

Moreover existing systems are almost incompatible. In spite of the efforts of the workflow management coalition, current products incorporate in the design very

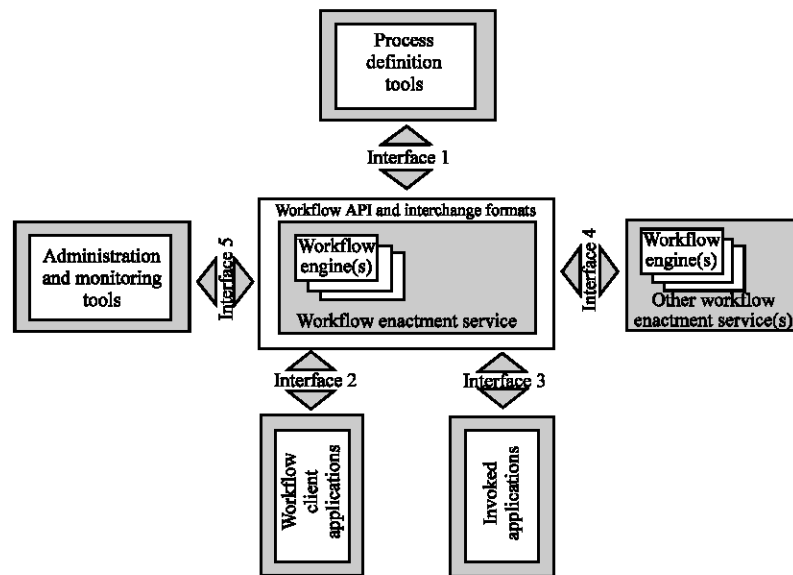


Fig. 2: WFMC reference architecture

concrete and exclusive interpretations of the world that makes particularly impossible to federate different systems. These incompatibilities are not just the syntax or the platform, but the very interpretation of the workflow execution. In most cases, the workflow is so tied to the underlying support system that it is not feasible to extend its functionality to accommodate other workflow interpretations^[4].

Workflow specifications may be ambiguous or contain inflexible semantics. Visual workflow programming languages sometimes lack representations for timing and execution constraints of workflows^[5].

SYSTEM ARCHITECTURE

This workflow management system provides support in two functional areas. Build time and Runtime control. The build time functions support the definition and modeling of workflow processes. Internally all elements of workflow engine/server can reside on the same node or be transparently distributed across many nodes. This architecture can grow incrementally from a small compact system to a large system that can handle hundreds of processes per hour.

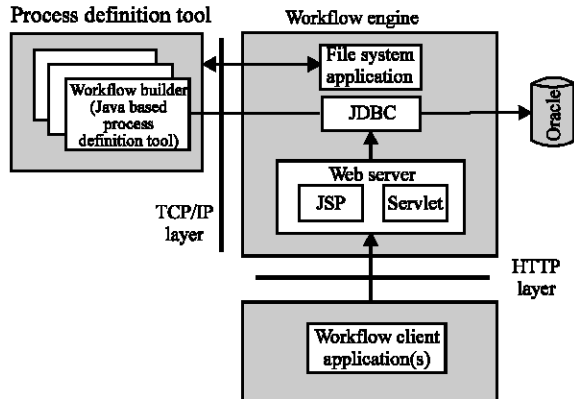


Fig. 3: System architecture

The Runtime control functions are implemented in workflow client interface, which handle the execution of processes. Runtime control is built on top of persistence storage which allows system to recover from failure. In this recovery process data is not lost like if it is present in live thread. Navigational logic of end user between activities governs workflow. Within the workflow management system two components are considered, storage server and navigation server. These are referred to as the workflow engine. Navigation server is also called web server. This interacts with users as well as with other invoked applications (Fig. 3).

Workflow client functionality is delivered through a web browser. Web server reads workflow models from database, controls the execution of workflows and then generates web based interfaces related to the activity for specific end user and renders it to him. Users access the workflow system using web based client.

SYSTEM IMPLEMENTATION

Following design decisions have been adopted to achieve main goals of flexibility and platform independence.

1. Thin client architecture has two parts; Process definition tool, named Workflow Builder and the HTTP based clients. Workflow Builder is written in Java. Java byte code can be interpreted on a large variety of platforms including the Sun Solaris and Windows platforms. This allows to execute the code on a large variety of systems without recompiling it.
2. All other client functionalities like starting the process, modifying process at run time, access tasks and check status and history, is delivered through web browser.
3. Workflow models are currently stored in a relational database and the workflow server accesses these models using a JDBC interface^[6]. Using this middleware component, different underlying relational database products can be used without changing the code.

CONCLUSION AND FUTURE WORK

In this study current workflow management systems have been analyzed which lead us to conclude that current systems are inflexible, lack any standardization across any products and are not designed to be embedded into earlier application easily. Then architecture of flexible and platform independent simple workflow management system is presented for heterogeneous and distributed environment.

This proposed system incorporates all of the components necessary for adding process layer to business applications. The architecture is designed to be distributed on heterogeneous environment. Java is used as the workflow process definition tool and internet technologies are used as workflow client applications. This research and development is step towards building scalable and reliable distributed workflow management systems.

As already mentioned, workflow management can be considered an important direction in the future

development of information systems. In future focus will be on reflexivity that allows a workflow component to programmatically examine, analyze, create, and manipulate its own process and data as part of automatable tasks within the executing workflow. We will evolve and optimize workflow management system through measurement, tracking, and reuse of generalizeable process fragments to improve the ability of the workflow to adapt new applications and uses.

REFERENCES

1. David, H.W., 1995. The workflow reference model. Document No. TC00-1003. Document Status-Issue 1.1, Workflow Management Coalition 2 Crown Walk Winchester Hampshire, UK SO22 5XE, <http://www.wfmc.org/standards/model.htm> pp: 6-8.
2. Euthimous, P. and M. Rabinovich, 1997. Predictive workflow management. AT and T Labs-Research 180 Park Avenue Florham Park, NJ 07932, pp: 1-2.
3. Michael, W. and T. Illmann, 1998. Using java for the coordination of workflows in the world wide web. Faculty of Computer Science, University of Ulm, <http://medien.informatik.uni-ulm.de/forschung/publikationen/interaktion98.pdf> pp: 2-4.
4. Alonso, G., D. Agrawl, A. El. Abbadi and C. Mohin, 1996. Functionality and limitations of current workflow management systems. IEEE Expert J., pp: 14-17.
5. Peter, J.K., G.A. Bolcer and M. Bergman, 1998. Requirements for supporting dynamic and adaptive workflow on the WWW. University of California, Irvine Irvine, CA <http://citeseer.ist.psu.edu/kammer98requirements.html> pp: 2-6.
6. Gottfried, V., M. Weske and G. Wittkowski, 1996. Dynamic workflow management on the web. <http://citeseer.ist.psu.edu/vossen96dynamic.html> pp: 10-15.