COMPUTER AUTOMATION OF BUSINESS PROCESSES

T. Stoilov, K. Stoilova

Computer automation of business processes: *The paper presents the Workflow management system as an established technology for automation and controlling of business processes. The evolution of business process management systems is shortly given. The architecture and its main components is described. As an example for application of Workflow Management Systems is outlined the i-Flow Interstage Business platform.*

Key words: *Information Technology; e- services; Business systems*; *automation; Workflow Management Systems.*

INTRODUCTION

Workflow Management Systems are a mature technology for automation and control of business processes [5, 6]. One widely accepted definition of workflow comes from the Workflow Management Coalition: "Workflow is the computerized facilitation or automation of a business process, in whole or part" [7]. An extended definition in [7] works out the definition as "…a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications*."*

With the rise of the Web as the major platform for making data and services available for humans and applications a new challenge has become prevalent requiring not only the support of workflows within individual organizations, but also workflows crossing organizational boundaries referred to as interorganizational workflows [1,3].

THE EVOLUTION OF BUSINESS PROCESS MANAGEMENT

One of the leaders in development of workflow technology is Wil van der Aalst [1,2]. In his history review of the workflow technology he concludes that till the nineties there is no fundamental approach for the development of such systems in the business processes. Now the business process management systems are at the top of researchers' work. The peculiarity of the workflow management technology is the separation of process logic from application logic targeting flexible and highly configurable applications [8]. In [4] are defined seven fields as the *conceptual ancestors* of workflow management technology: office automation, database management, e-mail, document management, software process management, business process modelling, and enterprise modelling and architecture. The focus of office automation research was "to reduce the complexity of the user's interface to the (office information) system, control the flow of information, and enhance the overall efficiency of the office" [10]. "While the research interest in office automation increases since middle of the 1980, the commercial exploitation of workflow technology began between 1983 and 1985. It was fostered by advances in imaging and document management technology on the one side, and enhanced e-mail systems that extended traditional point-to-point mail routing with a predefined process map on the other side. From this first generation of workflow systems, only few vendors are still active, while the majority of the early players have been restructured through mergers and acquisitions, or dropped out of the market altogether" [8]. Mainly the terminology and glossary work of the Workflow Management Coalition (WfMC), [7] have determined the fundamental understanding of workflow management and workflow management systems. According to it the workflow management system consists of a modelling component for the creation of workflow models, functionality for the creation of workflow instances from these workflow models, and functionality for the execution of these workflow instances. Respectively the products, which implement the workflow technology, have to follow this functional consistence with appropriate software modules. The functional component for the enactment of workflow instances is called *workflow engine.*

In the sixties information systems were built on top of a small operating system with limited functionality. These systems mainly consisted of tailor-made applications. New types of product software added new functionalities since then. The nowadays operating systems offer more

functionality than in the sixties. Database management systems offer functionality, which used to be in tailor-made applications. This trend leads to the shifted emphasis from programming to assembling of complex software systems. The coding of individual modules is already an old approach. The challenge now is orchestrating and combining pieces of software from the different functional layers. Another trend is the shift from data to processes [2]. The same source points out the seventies and eighties as dominated by data-driven approaches. At that period the information technology is associated with storing and retrieving information and as a result data modelling was the starting point for building an information system. The processes had to adapt to information technology and modelling of business processes was often neglected. On the contrary today the emphasis is on processes, for instance business process reengineering.

SYSTEM'S AND TECHNOLOGICAL ARCHITECTURE OF WORKFLOW MANAGEMENT SYSTEMS

The structure of a workflow management systems consist of a development environment, named build time component and an execution environment, named run time component [9]. The development environment gives workflow designers tools for the design of workflow models, the specification of workflow relevant data structures, and the design of the organizational model, relevant to the execution of the workflow models. The modelling of processes and organization structures are mainly supported through graphical modelling tools. The specification of data structures and integration adapters is ordinary text-based and resembles to the traditional programming view. The last can be stored in a workflow model repository.

The second main part of the Workflow management system is the workflow engine or run time environment which consists of modules covering different functional aspects. Usually, the modules are hierarchically organized. The coordinator is an event handler, sending event notifications to and receiving notifications from these modules, [9]:

- The *process management facility* creates workflow instances from the workflow model. It ensures an appropriate entry to the workflow instance database. It has to obey to the execution constraints like the validity period of a workflow model.
- The *control flow manager* controls state changes of the workflow instances. It has to handle the control flow conditions.
- The *worklist handler* creates and manages access rights to work items. It handles the users' work lists and if conflicts arise, it makes a decision.
- The *user management* facility manages the access of system users to work lists and all the workflow management system. The information from the organizational repository is used by this module for determination of the workflow participants.
- The *application invocation* module controls the interaction of the workflow engine with its applications.
- The *data management* unit executes data conversion and data mapping between activity instances.
- The *history management* component logs system events in the audit trail. These events can be either system related (e. g., user log-on and logoff) or workflow instance related (e. g., activity started, activity completed).
- *Integration APIs* provide access to the workflow engine for external systems. In this way the workflow engine can be used in another application.

Figure 1 shows an example of relationships between the main components in a technical and software system, performing workflow management activities [9] Michael zur Muehlen, Workflow-based Process Controlling. Foundation, Design, and Application of Workflow-driven Process Information Systems, 2004. Logos Verlag Berlin, 2004, ISBN 3-8325-0388-9.:

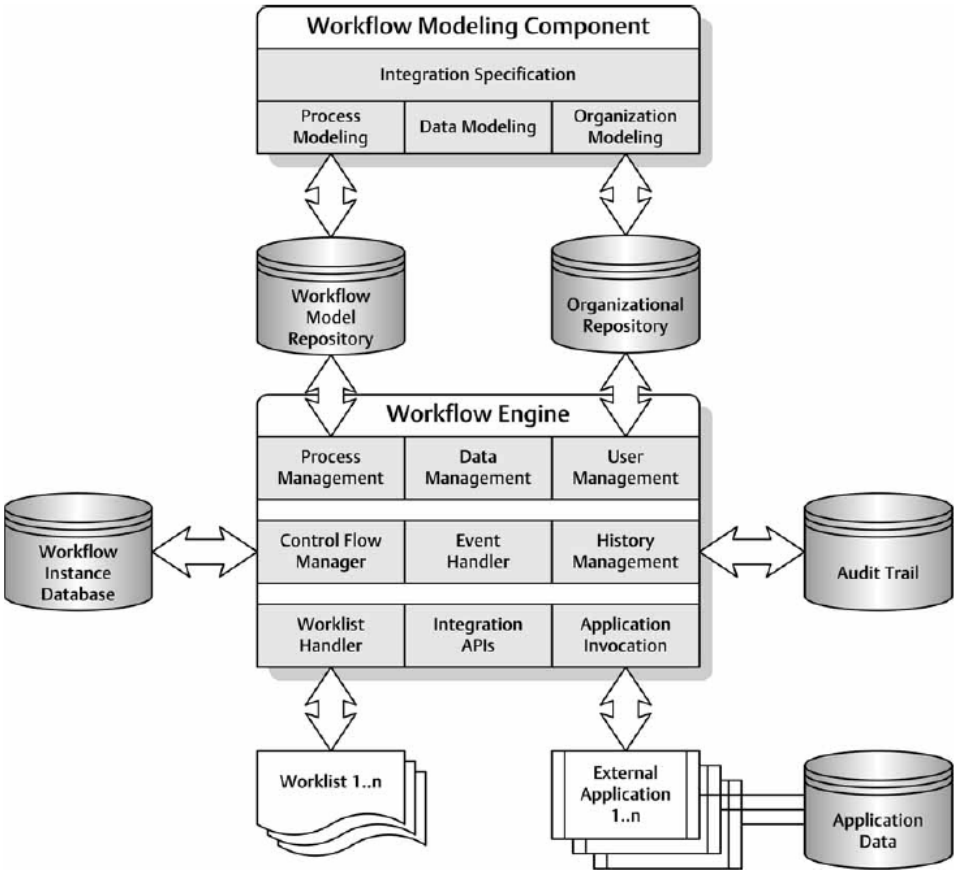

Figure 1 - Software architecture of a Workflow Application [9]

Example for application of Workflow Management Systems : i-Flow Interstage Business platform

i-Flow Interstage Business platform is a customizable Business Process Management engine for web-centric applications. The system has 4-tier architecture: repository, process logic, web tier, user interface [8]. The web tier contains servlet components running in a web server. The main process logic is in the business process management tier. The fourth tier contains the database directory, document management as well as connectivity to other systems. The main tool within the Interstage suite is the Development Manager Client. It defines workflow types (called workflow templates in i-Flow) and manages the execution of workflows. The Administration Client has the responsibility for administering processes and templates.

A. *Functional aspect*. The tasks, performed in the process, the operators representing the workflow activities are components of i-Flow workflow templates. The workflow template has one start node and at least one end node. Workflow templates can be organized in super and respective sub-workflows.

B. *Behavioral aspect*. i-Flow provides a graphical user interface to create process templates. The units below are available for designing the sequential behaviour [8].

- *Arrow*: They are used to link two activities to denote the flow of events.
- *Conditional* Node: This node realize an exclusive OR or an AND. A default outgoing arrow must be chosen to allow process ending if conditions are not satisfied. The order for conditions' evaluation can be determined. For conditions' defining are applied standard JavaScript expressions.
- *Complex Conditional* Node: The behavior of this node is similar to the Conditional Node. Here the expressions for defining conditions can include multiple incoming data items.

- *OR* Node: It can be used as a split construct (it behaves as a parallel split that executes all subsequent activities simultaneously or in any order) or as a join construct (the OR node merges two or more branches without synchronization: the subsequent activity is started as soon as one branch is finished, but when the subsequent activity finishes all active branches are terminated).
- *AND* Node: It can be used as a split construct (it is identical to the behaviour of the OR node). When it is a join construct it is synchronizing: the process waits until all activities that lead to this node are completed, before starting the subsequent activity.
- *Subprocess* Node: The workflow reaching this node set process control to a sub-process, and the node enters a Waiting-for-Sub-process state. When the sub-process completes, control returns to this node, and all ensuing activities connected to the node are activated simultaneously. Both run-time and design-time sub-processes are used to break complex tasks into a hierarchy of easier-to-handle units.
- *Remote Sub-process* Node: This node corresponds to the sub-process node with the only difference that the sub-process resides in another workflow engine. i-Flow supports other i-Flow servers, Collaboration Ring or other SWAP compatible process engines so that it is possible to start a remote sub-process from the local i-Flow Server. The local process waits for the remote process to complete, and incorporates the results of the remote process back into the local process.
- *Chained Process* Node: When the workflow reaches this node, another process is activated similarly to a sub-process. The difference is that the node does not enter a suspended state, and the chained process operates as an independent entity.
- *Delay* Node: When the workflow reaches this node, the process pauses until the first timer attached to the node fires.
- *Triggers*: Triggers either start processes or make choices on activities in response to Data Event files (added to a particular directory by a data source external to i-Flow).
- *Exit* Node: Identifies the end of the process. Every process must have at least one Exit Node.

C. *Informational aspect.* i-Flow uses the principle of globally shared data. All data items defined within a workflow type are shared between the tasks. Only between super and sub-workflows a distinct data mapping has to be defined to pass data elements back and forth. External data passed to i-Flow in the prologue or epilogue actions of any node are always mapped to so called User-defined process attributes (UDAs). UDAs are variables, which hold values in a running process, set by a user through the use of a form, script, or a JavaAction. Triggers allow to map external data from XML files to UDAs. By loading a XML schema file mappings can be defined from each element to defined UDAs for the activity.

D. *Operational aspect.* i-Flow provides communication with external applications over Java Actions. A Java Action is a call to a static method on a specified Java class. The Java action is configured to pass UDA values as parameters, and the result of the method is assigned to a UDA. A Java Action can be invoked when a process starts, when a node is activated or completed, when a timer expires, at time of role resolution and when the process ends.

E.*Organizational aspect.* At design time template owners have the option of assigning process ownership. Ownership can be assigned to any i-Flow role or set of users. Activities are assigned to a role, a user, or a set of users. If more than one user is assigned, each assignee receives a separate activity in the Activity List. Every user has its own profile, defining the way the user is notified when a work-item is assigned and the association of a DMS directory for its user ID. The reassignment Java action allows reassigning tasks. A task can either be manually reassigned or automatically to a user or users with the predefined "Assign Task to User Action". A work-item owner can also reassign the task to any other role, whereas the new owner can accept or reject the reassignment. The directory adapter implements an i-Flow specific interface to expand a group into

a list of individuals. The enactment engine uses this at runtime to determine work assignments. The ability to read the directory or to expand groups into a list of individuals is exposed by the Model API. i-Flow provides adaptors to standard LDAP directory such as Sun ONE Directory Server, to Microsoft Active Directory and to Microsoft Windows native user/group support.

CONCLUSIONS

A general task of the development of the workflow system in the business activities is the implementation of principles of the automatic control for business systems. These systems doesn't consist pure technical components, but they integrate both human and human-computer activities and non-automatic interactions. Thus the implementation of the principles of the automation will benefit the exploitation behavior of the business systems.

REFERENCES

[1] [Aalst, 1999] W.M.P. van der Aalst, *Process-Oriented Architectures for Electronic Commerce and Interorganizational Workflow*. Information Systems, Vol. 24/8, 1999.

[2] [Aalst et al., 2003] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, Workflow patterns. D*istributed and Parallel Databases*, 14(1), .5-51, 2003

[3] Dayal U., M. Hsu, R. Ladin, *Business Process Coordination - State of the Art, Trends, and Open Issues*, 2001. Proc. of the 27th VLDB Conference

[4] Jablonski S. and C. Bussler, *Workflow Management: Modelling Concepts, Architecture and Implementation,* 1996. International Thomson Computer Press, London.

[5] Kappel G., S. Rausch-Schott, W. Retschitzegger, *A Framework for Workflow Management Systems Based on Objects, Rules and Roles,* 2000. ACM Computing Surveys Electronic Symposium on Object-Oriented Application Frameworks, M.E. Fayad (guest editor)

[6] Leymann F. and D. Roller, *Production workflow: concepts and techniques,* 2000. Prentice-Hall

[7] WfMC, Glossary, Terminology and Glossary, 3rd Edition. Document No WFMC-TC-1011. Workflow Management Coalition. Winchester, 1999.

[8] M. zur Muehlen. Organizational management in workflow applications - issues and perspectives. *Information Technology and Management*, 5, 2004, 271-291

[9] Michael zur Muehlen, Workflow-based Process Controlling. Foundation, Design, and Application of Workflow-driven Process Information Systems, 2004. Logos Verlag Berlin, 2004, ISBN 3-8325-0388-9.

[10] Ellis, C. A.; Nutt, G. J., Office Information Systems and Computer Science, 1980. In: ACM Computing Surveys, 12 (1980) 1, pp. 27-60.

Contacts:

Prof. DSc. Todor Stoilov, Head of Department "Hierarchical Systems", Institute of Computer and Communication Systems, Bulgarian Academy of Sciences e-mail: todor@hsi.iccs.bas.bg

This research is partly supported by the European Commission, project №FP6-027178 VISP, and National Scientific Fund of Bulgaria, project № ВУ-МИ-108/2005.