Open Source Workflow Management Systems: A Concise Survey

Ricardo Garcês, Tony de Jesus, Jorge Cardoso* and Pedro Valente

University of Madeira, Portugal
***SAP Research, Germany**
***University of Coimbra, Portugal**

ABSTRACT

The use of open source Workflow Management Systems (WfMS) is appealing for organizations due to its low or inexistent cost and its customization capabilities. In this chapter we analyze ten different open source WfMS using a framework that offers decision makers a starting point for selecting a workflow solution. The framework is to be used as a basis for characterizing WfMS based on a set of 22 parameters.

INTRODUCTION

Nowadays, many organizations in the commercial, government and non-profit sectors benefit from the use of open source software [1]. Open source software is having a growing impact on the software industry by becoming an important competitor to commercial software [2]. According to [3], the number of suppliers offering workflow management software is estimated to be two hundred. The selection of an open source WfMS solution may be quite a difficult and complex undertaking. A sound selection requires a complete analysis of the most popular solutions available. Otherwise, it may lead to the choice of an inadequate workflow product that will not support efficiently the business processes of an organization.

According to [4], the motivations for using and developing open source software are mixed, ranging from philosophical and ethical reasons to pure practical issues. Usually, the first perceived advantage of open source models is the fact that the software is made available gratis or at a low cost. But this characteristic is not exclusive to open source software [5]. What really distinguishes open source software from software available without a fee is the access to the source code and the right to modify it, the right to redistribute the modifications and to improve the code. Each organization has its particularities, so the characteristics of open source solutions allow the customization of open source workflow systems according to the functioning and the needs of organizations.

This chapter offers an overview and comparison of ten popular open source WfMS using a comprehensive framework for decision makers, providing a starting point to the complex process of selecting an open source WfMS. In fact, this document

is intended to enable managers to better guide, justify and explain their decisions and choices.

WFMS COMPARISON FRAMEWORK

Several approaches have been proposed to compare information systems and information technologies. They have been provided by prestigious consulting companies such as Andersen Worldwide, Ernst & Young, Deloitte & Touche, Coopers & Lybrand, KPMG and Price Waterhouse.

Since workflow technologies have specific characteristics, existing approaches do not address many important perspectives. Therefore, we propose a new and more complete approach. On the one hand, we want to determine what functionalities are provided by WfMS. On the other hand, we also want to evaluate the installation and usage of WfMS, as well as the definition of workflow processes. For this reason, and as showed in Table 1, we will focus our attention on the compliance of WfMS with the WfMC reference model [6] and on two functional perspectives: *runtime* and *design time*.

		Parameters
WfMC Reference Model		Process Definition Application (Interface 1)
		Workflow Client Application (Interface 2)
		Invoked Applications (Interface 3)
		Other Workflow Enactment Services (Interface 4)
		Administration and Monitoring tools (Interface 5)
Functional Perspectives	Runtime	Research Scope
		Installation Time
		Documentation
		Platform Independent
		Easiness of Installation and Utilization
		Web Based
		Other Software Required
		Middleware Platform
		DBMS Integration
		Transactions Support
	Design Time	Process Definition Time
		Documentation
		Easiness of the Process Definition
		Web Based
		Organizational Perspective
		Workflow Language

Table 1: Overview of the framework developed in our study

WfMC reference model

One of the many principles used by the WfMC is the so-called workflow reference model [6]. This model is a general description of the architecture of a workflow management system, in which the main components and the associated interfaces are described. In the workflow reference model, the tools for constructing and designing workflows are known as process definition applications (Interface 1). Work items are offered to the employees through workflow client applications (Interface 2). By selecting a work item, an employee can begin performing a specific task for a specific case. When carrying out a task it may be necessary to start an application. All the application software that can be started from the workflow system are known as invoked applications (Interface 3). According to the WfMC reference model, workflow systems may also able interaction with other workflow engines (Interface 4). Workflow tracking, case control and staff management are supported by administration and monitoring tools (Interface 5).

Runtime and design time perspectives

According to [6], at the highest level, all WfMS may be characterized as providing support in three functional areas:

- *build time functions* are concerned with defining and modeling workflow processes and their activities;
- *runtime control functions* are concerned with managing workflow processes in an operational environment and sequencing activities;
- *runtime interactions* are concerned with human users and other application tools for processing the various activity steps.

As we can see, these three functional areas can be summarized in two core functional perspectives: design time (associated with build time functions) and runtime (gathering runtime control functions and their interactions).

The runtime perspective, proposed by our framework, is associated with the installation and the testing of the main functionalities of a workflow solution. It is also related to the analysis of the support that is offered by the workflow solution to workflow processes (e.g. support of transactions and treatment of exceptions). The design time perspective, proposed by our framework, is associated with the task of designing a sample workflow process using the process editor. This functional perspective is also related to the ability of the process editor to easily, and in a small amount of time, help us to define a relatively complete workflow process. Given the importance of these two functional perspectives, it only seems natural that they should be analyzed before choosing a workflow solution.

COMPARISON ENVIRONMENT

We have chosen ten of the most popular and promising open source workflow systems available nowadays. The final set that we will analyze in this chapter is composed of the following WfMS: Bonita, Enhydra Shark, JawFlow, JBoss jBPM, JFolder, JOpera, OpenWFE, RUNA WFE, WfMOpen and YAWL. Before presenting the results of our comparison, it is crucial to clearly identify the environment on which the analysis of the WfMS was made. The installation and test of the workflow systems was made by two senior students in Computer Science within the scope of their final project. All the WfMS analyzed were installed and tested in a Intel Pentium M 2.00GHz computer with 1 GB memory, 100 GB disk space and running Windows XP. One way to quickly gain a good impression of a workflow management system is to work through a sample process chosen in advance. The sample workflow process used to test the WfMS' platforms was composed of 15

different tasks with multiple control structures (AND splits/joins and XOR splits/joins) and by nested workflow definitions. It also included 5 different participants.

COMPARISON OF THE 10 WFMS SELECTED

The framework specified in this chapter is now considered to compare the ten workflow systems. Table 2 offers an overview of our findings. Most of the systems are not completely compliant to the WfMC reference model. In fact, only Bonita, OpenWFE and YAWL are fully compliant. Moreover, most of the non compliant WfMS do not provide an interface to interact with other workflow enactment services. All solutions are platform independent. Two systems have been developed within the scope of a research project (YAWL and JOpera). Regarding installation and testing time, we have discovered a wide range of values that go from only 22 minutes, with OpenWFE, to 12 hours and 47 minutes with WfMOpen. One of the most important aspects that influenced the installation time was the documentation provided. We have reached the conclusion that most of the WfMS studied offer enough documentation in order to correctly install and use the system.

Regarding the installation easiness, JFolder's installation was easy, standing out from the all the other workflow systems. On the opposite end, we found Bonita, JawFlow and WfMOpen with a rather complicated installation procedure. We also discovered that only Enhydra Shark and JOpera do not offer a web based administration environment.

The process definition applications provided by five of the workflow systems analyzed offered mechanisms that allowed designing our sample process without major constraints. However, RUNA WFE, WfMOpen, Bonita, JFolder and JBoss jBPM process definition applications were quite limited and unpleasant.

The time spent to define our sample process assumes values that vary from almost 2 hours to approximately 6 hours. YAWL allowed the quickest process definition. Bonita was the one that required most time to design our sample process, 5 hours and 11 minutes.

Regarding the documentation provided, there is little or no documentation available about the process editor of several WfMS analyzed, like: JawFlow, JBoss jBPM, JFolder and WfMOpen. Finally, XPDL is the process definition language most often used by the workflow systems analyzed.

In the following subsections we will discuss each workflow management system, according to our framework, in greater detail.

Bonita

Bonita was developed in 2003 by a team of 14 engineers, of which, Miguel Valdes Faura, Brice Revenant and François Charoy were the project leaders [7]. The current version is 2.0 and was released in June, 2006. Bonita is a complete workflow system that provides functionalities to handle long-running, user-oriented workflows and business processes. It allows to dynamically modifying the definition of a running process in order to take into account events that were not planned. This workflow solution also takes benefit from several services that the integration with a J2EE application server provides, such as transactions, role-based authentication and connection with external information systems.

			Bonita	Enhydra Shark	JawFlow	JBoss jBPM	JFolder	JOpera	OpenWFE	RUNA WFE	WfMOpen	YAWL
WfMC Reference Model		Process Definition Application (Interface 1)	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
		Workflow Client Application (Interface 2)	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
		Invoked Applications (Interface 3)	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
		Other Workflow Enactment Services (Interface 4)	Yes	Yes	No	No	No	No	Yes	No	No	Yes
		Administration and Monitoring tools (Interface 5)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Functional Perspectives	Runtime	Research Scope	No	No	No	No	No	Yes	No	No	No	Yes
		Installation Time	1h56m	6h11m	8h15m	1h09m	1h15m	1h56m	22 m	2h20m	12h47m	49 m
		Amount of Documentation Found	Comprehensive	Basic	Inexistent	Comprehensive	Basic	Comprehensive	Comprehensive	Comprehensive	Basic	Comprehensive
		Platform Independent	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
		Easiness of Installation	Simple	Simple	Complex	Simple	Trouble-free	Simple	Simple	Simple	Complex	Simple
		Easiness of Utilization	Complex	Simple	Simple	Simple	Complex	Complex	Simple	Simple	Complex	Simple
		Web Based	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
		Other Software Required	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
		Middleware Platform	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
		DBMS Integration	Comprehensive	Comprehensive	Comprehensive	Comprehensive	Basic	Basic	Comprehensive	Comprehensive	Basic	Basic
		Transactions Support	Exception Handling; Rollback	Exception Handling	None	Exception Handling; Rollback	Exception Handling	Exception Handling	Exception Handling; Rollback	Exception Handling; Rollback	Exception Handling	Exception Handling
	Design Time	Process Definition Time	5h11m	2h24m	3h03m	2h45m	4h25m	2h26m	5h15m	3h57m	3h03m	1h55m
		Web Based	Yes	No	No	No	Yes	No	Yes	No	No	No
		Amount of Documentation Found	Comprehensive	Comprehensive	Inexistent	Inexistent	Inexistent	Basic	Inexistent	Comprehensive	Inexistent	Comprehensive
		Easiness of the Process Definition	Complex	Simple	Complex	Complex	Complex	Simple	Simple	Complex	Complex	Simple
		Organizational Perspective	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No
		Workflow Language	XPDL	XPDL	XPDL	jPDL and BPEL	Developed its own XML Based Language	JOpera Visual Composition Language	Developed its own XML Based Language	jPDL language	XPDL	YAWL

Table 2: Overview of the ten WfMS proposed

1. WfMC reference model. Bonita is fully compliant to the WfMC reference model specification.

2. Runtime perspective. Bonita's installation and testing took 1 hour and 56 minutes. The documentation provided by the developers is comprehensive, allowing us to install the software without facing any major problem. However, its web-based user interface is unpleasant and not very user friendly. Bonita requires the installation of JDK 1.4, JOnAS Application Server with Tomcat, Jakarta Ant and a DBMS (database management system). It works upon the middleware platform, Java Message Service, in order to exchange data and events. Bonita offers an easy integration with most database management systems. It also supports exceptions treatment and rollback during process execution.

3. Design time perspective. In order to correctly define our sample workflow process we have spent approximately 5 hours. The graphical editor provided by Bonita is web-based. It is a Java Applet that allows us to design processes by dragging and dropping each activity. We have found a great amount of documentation about this process editor, but some details are not clearly explained. For instance, when defining a process, its sub processes should be defined first. For this reason the definition of our sample workflow process was quite complex. The Applet supports the definition of the organizational model, allowing for the specification of participants and roles. Bonita implements the Workflow Management Coalitions's XPDL (XML Process Definition Language).

Enhydra Shark

Enhydra Shark was developed by Enhydra.org community in 2003. It is an extendable and embeddable Java workflow engine framework completely based on WfMC specifications [8]. Shark can be used as a simple Java library in a servlet, a swing application, or in a J2EE container. The current version of Enhydra Shark is 2.3 which was released in November, 2008.

1. WfMC reference model. Shark is completely conformal to the WfMC reference model.

2. Runtime perspective. Enhydra Shark's installation and testing took 6 hours and 11 minutes. The documentation provided by the developers was quite straight forward, allowing for a relatively simple installation of the software. In order to properly administrate the workflow system, we should use a commercial administration tool. However, this is not mentioned in the documentation, and this application is not available to download in the project's homepage. This workflow system does not offer a web based environment. The administration/client application is very user friendly, allowing for a quite easy testing. This workflow system works upon a middleware platform (CORBA). Shark provides an easy integration with most database management systems and offers mechanisms that support the exceptions treatment during a process execution.

3. Design time perspective. It took 2 hours and 24 minutes to define our sample workflow process. The documentation provided for the workflow process editor is quite comprehensive. This workflow solution provides, by default, a graphical editor very similar to JPEd (used with WfMOpen) called Together Workflow Editor (TWE). It is very practical and easy to use, assuming itself as a complete and interesting editor. TWE supports the design of the organizational perspective and the workflow language used is XPDL.

JawFlow

JawFlow was been developed by Vincenzo Marchese in October, 2006 and currently it is in version 3.0. JawFlow is a workflow engine partially conformal to WfMC directives and completely written in Java. It can be customized using activities written in Java or in any scripting language supported by the Bean Scripting Framework [9]. To deploy, test and run JawFlow, we have used the JBoss application server. However, there are no code dependencies to JBoss.

1. WfMC reference model. JawFlow only offers an embedded administration (interface 5) and client application (interface 2). It does not offer a process definition application (interface 1). This workflow system also does not offer interfaces to invoke other application (interface 3), or to interact with other workflow enactment services (interface 4).

2. Runtime perspective. It took 8 hours and 15 minutes in order to correctly install and test this WfMS. The documentation provided by the developer is very poor making the installation process quite complex. The process administration environment is web-based and relatively easy to use. JawFlow requires JDK 1.5, Jakarta ant, JBoss and a DBMS. This workflow system works upon a middleware platform (Java RMI and CORBA). JawFlow can be integrated with any database management system and offers mechanisms that support error handling during the execution of a workflow process.

3. Design time perspective. This workflow engine does not provide a process editor. Any editor supporting XPDL can be used. In our case, we have used JPEd (used with WfMOpen). For this reason, the results presented it the Table 2 are identical to the ones that are described in the WfMOpen design perspective.

JBoss jBPM

JBoss jBPM is a flexible and extensible workflow management system. The JBoss jBPM' core component is the plain Java software for managing process definitions and the runtime environment for execution of process instances [10]. Its last release is version 3.2.3.

1. WfMC reference model. JBoss jBPM offers an administration/client application. JBoss jBPM is also able to interact with other applications. However, it is not able to interact with other workflow engines. This workflow system also offers a process definition application.

2. Runtime perspective. It took 1 hour and 9 minutes in order to correctly install and test a working version of jBPM. The documentation provided was comprehensive. This was the main factor for making the installation and usage of this workflow solution quite simple. Its web based administration/client application is poor in terms of features offered. For this reason, the use of this workflow solution should require the creation and implementation of a customized client and administration application. jBPM requires the installation of JDK and Eclipse with the JBoss IDE plugins. This workflow system works upon a middleware platform (Java RMI or CORBA). It offers mechanisms that make jBPM portable across the most popular databases and supports an effective treatment of transactions, allowing exceptions treatment and rollback during process execution.

3. Design time perspective. To correctly define our sample workflow process, using the Eclipse-based tooling available for BPEL, we have spent 2 hours and 45 minutes. The lack of documentation about JBoss jBPM' process editor reflected negatively upon the ease of the process definition. But this was not the only problem faced. In fact, another problem found was the decision building block (XOR-

split) had to be directly implemented in the code. Because JBoss jBPM uses BPEL in order to define processes, the definition of sub processes is not supported. This results complex workflow diagrams, which are difficult to analyze and understand. The definition of our sample process was therefore quite difficult. This process definition editor supports the specification of the organizational perspective. JBoss jBPM supports two process definition languages: jPDL and BPEL. jPDL is a process language to implement business processes and workflows in Java. BPEL provides process orchestration which is the ability to combine web services into a process execution flow.

JFolder

JFolder (also known as PowerFolder) was developed by Gary Steinmetz in 2004 and is in version 1.1. It is a business application development studio and server that uses a XML based language in order to define workflow processes that run within a J2EE environment. Development and administration takes place through a web browser. JFolder contains features like security, persistence, email, file management and data access [11].

1. WfMC reference model. JFolder offers administration and monitoring tools as well as a workflow client application. However, it is not able to interact with other applications and with other workflow engines. This WfMS also offers a process definition application.

2. Runtime perspective. JFolder installation and testing took 1 hour and 25 minutes. The amount of documentation provided by the developers is sufficient, allowing us to install the software without facing any major problem. Its web-based administration environment is quite unpleasant, becoming very often confusing. This fact makes this workflow solution unattractive from an administration point of view. JFolder requires J2EE, Jakarta Ant and JBoss. This workflow system works upon the middleware platform. The documentation does not indicate if is possible to integrate JFolder with other than its default DBMS (hsqldb). JFolder offers mechanisms that support error handling during the execution of the workflow process.

3. Design time perspective. It took 4 hours and 25 minutes in order to design our sample workflow process. A poor documentation is available for this editor. This tool provides a very limited web-based process editor. It is not based on a "drag and drop" idea. This situation makes it harder to add or edit elements of the diagram. There is also no automated mechanism to save the process definition. All these aspects made the design of our workflow process quite hard and complex. It does not support the definition of roles and participants (organizational perspective). The JFolder process editor uses a XML-based proprietary language in order to define workflow processes. This language does not support the definition of sub processes.

JOpera

JOpera is built as a collection of plugins for Eclipse. It is a service composition tool that offers a visual language and an execution platform for building workflow processes. It includes a graphical modeling environment, a light-weight execution engine, and also a set of powerful debugging tools which natively supports the iterative nature of service composition. JOpera has a wide range of applications and implications: from rapid development of service-oriented business applications to classical workflow management and business process automation [12]. JOpera plugin for Eclipse 1.9.11 is the latest release of this system.

1. WfMC reference model. JOpera offers an administration and monitoring tool. It is able to interact with other applications. This workflow system also offers a process definition application.

2. Runtime perspective. The JOpera system was developed with research purposes. Its installation and testing took 1 hour and 56 minutes. The comprehensive documentation provided by the developers has allowed us to install the software without facing any major problem. However, the environment offered, based on Eclipse workbench, is not a practical and user friendly management environment. This poor management environment makes this workflow solution unattractive from a usage point of view. JOpera requires the installation of Java JDK and Eclipse. JOpera provides integration with the most popular DBMS and supports a simple exception handling model.

3. Design time perspective. In order to correctly define our sample workflow process we have spent 2 hours and 26 minutes. Enough documentation related with the graphical editor is provided. In spite the fact that the definition of the process is quite simple, JOpera process editor is quite repetitive, making the definition of our sample process longer. Another problem found is that the processes being designed quickly became confusing and it was difficult to analyze/identify the transitions between tasks. This limitation added to the fact that it does not support the organizational perspective allows us to say that this is a very unattractive process editor. The workflow language used by JOpera is JOpera visual composition language.

OpenWFE

OpenWFE is an open source workflow engine that has been developed by Lukas Eder and Nicolas Modryzk. It is a complete Business Process Management suite with four components: an engine, a worklist, a client application and a host for automatic agents. It is written in Java, but features access libraries for languages such as Python, Perl, and Ruby, C# (.NET), PHP and Pnuts [13]. OpenWFE is based on a distributed and web-friendly infrastructure. It offers mechanisms that allow persistence, automated form generation and workflow administration. In November 2006, OpenWFE 1.7.2 was released.

1. WfMC reference model. OpenWFE is completely conformal to the WfMC reference model.

2. Runtime perspective. OpenWFE' installation and testing took only 22 minutes mainly due to its intuitive and user friendly environment. The comprehensive documentation provided by the developers allows us to install the software without facing any major problems. The only problem found during the installation was that the documentation available mainly described the installation of the system for a Linux operating system. The web-based administration tool and client application were very user friendly and simple to use. OpenWFE installation requires JDK and JRE in order to work properly. This workflow system works upon a middleware platform (Java RMI). It is also able to be integrated with all of the most important database systems and it supports an effective treatment of transactions, allowing exceptions treatment and rollback during process execution.

3. Design time perspective. Using the graphical editor provided, we have spent 5 hours and 15 minutes in order to correctly define our sample workflow process. Workflow processes are designed in their own XML based language. The lack of documentation of the process editor made this definition process quite long. Droflo is a very limited web based process editor. In fact, it is not based on a "drag

and drop" idea. This situation makes it harder to add or edit element of the flow diagram. Another problem found is that in order to save the XML code generated the user has to copy it and then paste it in a text document. In other words, the editor does not have any option to perform this action. This process editor is so unpractical that in most situations it is much easier to define the workflow process directly using XML. The definition of our sample process was, therefore, quite hard. It supports the definition of the organizational perspective.

RUNA WFE

The Runa Consulting Group has released RUNA WFE, an open source workflow/business process management environment for jBoss jBPM engine. It is an end user solution for business process management, written in Java, which provides a rich web interface containing a work list handler, a process monitor and a form player. It also supports the interaction with external applications [14]. This workflow solution most recent update is RUNA WFE 2.2 (November 2008).

1. WfMC reference model. RUNA WFE offers an administration/workflow client application and also supports the interaction with other applications. However it is not able to interact with other workflow engines. Moreover, this WfMS offers a process definition application. RUNA WFE is partially conformal to the WfMC model because it does not interact with other workflow enactment services.

2. Runtime perspective. The installation and testing of RUNA WFE took 2 hours and 20 minutes. The comprehensive documentation provided was sufficient to install and test this system without facing any major problem. The friendly web based administration/client application offered also contributed to an easy testing. RUNA WFE requires the installation of JDK. This workflow system works upon the middleware platform. It offers an easy integration with the most popular database management systems and also supports an effective treatment of transactions, allowing exceptions treatment and rollback during process execution.

3. Design time perspective. The definition of our sample workflow process, using RUNA GPD (a process editor for RUNA that sits upon Eclipse workbench), took 3 hour and 57 minutes. The documentation provided was comprehensive, describing several workflow process definition examples. This workflow system does not allow the definition of sub processes, which results in the creation of complex and confusing workflow diagrams. It also requires the direct implementation of the user forms; which may become quite hard for inexperienced users. For these reasons, the definition of our workflow process was quite complicated. It supports the definition of the organizational perspective. The workflow language used by RUNA WFE is jPDL.

WfMOpen

WfMOpen is a J2EE based implementation of a workflow engine. The workflow component is based on a set of Java interfaces that defines API for workflow management facility. It may be used as the core for any process based application implementation and is well suited in providing solutions for business process management related jobs [15]. In May 2008, the most recent update, WfMOpen 2.2, was released.

1. WfMC reference model. WfMOpen is only partially conformal to the WfMC model, because it does not interact with other workflow enactment services.

2. Runtime perspective. It took 12 hours and 47 minutes in order to correctly install and test a working version of WfMOpen. The information available in the documentation provided was confusing, dispersed over the document and in

many aspects insufficient. This was the main reason for making the installation and testing of this workflow solution very complex. Moreover, besides the fact that it is poorly documented, the web-based management environment offered is in many aspects quite user unfriendly. WfMOpen requires the installation of JDK and JBoss. This workflow system works upon a middleware platform (Java RMI, CORBA and SOAP). The database integration is achieved only using the default DBMS of this workflow system. It offers build-in solutions for handling exceptions during a process execution.

3. Design time perspective. To correctly define our sample workflow process using JPEd, we spent 3 hours and 3 minutes. The lack of documentation about JPEd reflected negatively upon the ease of the process definition, making it quite complex. After understanding how it works, JPEd becomes very practical and easy to use. It supports the organizational perspective. WfMOpen uses XPDL with some extensions to define workflow processes.

YAWL

The YAWL system is an open source workflow solution based on the YAWL (Yet Another Workflow Language) language, designed by Wil van der Aalst, Lachlan Aldred, Marlon Dumas and Arthur ter Hofstede, members of the Faculty of Information Technology of Queensland University of Technology. The project designers developed this new language by taking Petri nets as a starting point and adding mechanisms to allow for a more direct and intuitive support of the workflow patterns identified [16]. YAWL provides direct support for all of the workflow patterns and offers mechanisms that allow persistence, automated form generation and workflow administration [17]. YAWL supports the control-flow perspective, data perspective, and is able to interact with web services. The last version of the system, version 2.2, was released in November 2008.

1. WfMC reference model. YAWL system is completely conformal to the WfMC reference model specifications.

2. Runtime perspective. The YAWL system was developed for research purposes. YAWL installation and testing took only 49 minutes. The documentation provided by the developers is comprehensive, describing in detail each step of the installation and allowing us to install the software without facing any major problems. In fact, the installation of the software was simple. This workflow system provides a web based administration/client application that is very user friendly and easy to use. In order to work properly, YAWL system installation requires JRE and Apache Tomcat. This system is compatible with a middleware platform: SOAP. The database integration provided does not support some of the most popular DBMS available. It only offers integration with PostrgreSQL as an alternative to Hypersonic. It allows exceptions treatment during process execution.

3. Design time perspective. Using the graphical editor provided, which is not web-based, we spent 1 hour and 55 minutes in order to correctly define our sample workflow process. The provided documentation related to the editor was comprehensive. The definition of our sample process was simple. In fact, the process definition editor uses a small set of elements to design a process, simplifying its analysis. It is also based on a "drag and drop" idea. This situation makes it easier to add or to edit elements of the flow diagram. However, one of the major drawbacks of this workflow solution is that it does not support the organizational perspective. For this reason, we are not able to associate participants or roles to a task.

RELATED WORK

Aalst et al. [16] offers a comparison of the functionality of 15 workflow languages based on a set of workflow patterns. We have a different objective since our aim is to evaluate the main features offered, the easiness of the installation and use of WfMS as well as the easiness of the definition of workflow processes. The research on runtime and design time perspectives of workflow systems is very limited. However, these two perspectives have been somewhat and indirectly addressed by academic papers. In [21], Murray offers a case study that analyses the implementation of a commercially available healthcare workflow system in two hospitals' settings. The framework proposed also includes a parameter with the same aim as our parameter named *organizational perspective*. It also proposes the parameter *ease of use of the WfMS* which is similar to our parameter *easiness of utilization*. The research developed by Stoilova and Stoilov [22] addresses problems related to the assessment and comparison of workflow management systems. The paper proposes an evaluation template composed by eight categories. The functional category is composed by some parameters equivalent to the ones that we have used. These parameters are: *modeling process definition*, *workflow client application*, *integration with other workflow engines (supported standards)* and *administration and monitoring*. The paper also proposes another evaluation category: *usability*. This category is related with our parameter: *easiness of utilization*.

RELEVANCE AND VALUE

The selection of an adequate workflow system to manage the business processes of an organization is an important and complex decision that depends on several aspects. The decision is significant due to the wide and heterogeneous set of WfMS available, either commercial or open source. The use of open source solutions may become very advantageous for organizations since source code as well as the right to modify it allows organizations to address specific requirements. Moreover, there are many success cases using this type of software. Nowadays, open source software is used extensively in the industry. The recent acceptance of Linux and the Apache project are excellent examples of this phenomenon. Due to the success of open source solutions, open source workflow systems have, therefore, become particularly interesting and appealing to IS and IT decision makers. From a set of open source WfMS currently available, we have chosen the most popular and, in our opinion, most interesting WfMS to be analyzed and compared. The framework proposed in this chapter for comparing open source WfMS is based on the WfMC reference model and on the runtime and design time perspective of workflow systems. This chapter offers an important study for industry decision makers by providing a starting point to the complex process of selecting an open source WfMS.

ACKNOWLEDGEMENTS

This work has been support by Foundation for Science and Technology (FCT), POCTI-219 and FEDER. The support of SAP Research, CEC Dresden, Germany, is also gratefully acknowledged.

REFERENCES

1. TDG, *Open Source Software: Case Studies Examining Its Use*. 2003, The Dravis Group.
2. Peeling, N. and J. Satchell, *Analysis of the Impact of Open Source Software*. 2001, QinetiQ.

3. Aalst, W.M.P.v.d. and K.v. Hee, *Workflow management: models, methods, and systems*. 1st edition ed. 2002: MIT Press Cambridge, MA, USA.
4. EWLS, *Free Software / Open Source: Information Society Opportunities for Europe?* 2000, European Working group on Libre Software.
5. Kenwood, C.A., *A Business Case Study of Open Source Software*. 2001, The MITRE Corporation.
6. Hollingsworth, D., *The Workflow Reference Model*. 1995, Workflow Management Coalition.
7. BONITA. *BONITA: Workflow Cooperative System*. 2007 [cited 21. May 2007]; Available from: http://bonita.objectweb.org.
8. Shark. *Enhydra Shark: Java Open Source workflow engine based on XPDL*. 2007 [cited 22.05.2007]; Available from: http://www.enhydra.org/workflow/shark/index.html.
9. jawflow. *jawflow: Java Workflow Manager*. 2007 [cited 22.05.2007]; Available from: https://www-304.ibm.com/jct03004c/servers/solutions/finder/solution/overview.jsp?solution_id=soq74085540080014002%7C30.
10. jBPM. *JBoss jBPM*. 2007 [cited 22.05.2007]; Available from: http://www.jbpm.org/.
11. JFolder. *JFolder - Application development and deployment platform*. 2006 [cited 22.05.2007]; 1.1 Alpha:[Available from: http://www.powerfolder.org.
12. JOpera. *JOpera Project: Process Support for more than Web Services*. 2004 [cited 22.05.2007]; Available from: http://www.iks.ethz.ch/jopera.
13. OpenWFE. *OpenWFE - open source workflow engine*. 2007 [cited 22.05.2007]; Available from: http://www.openwfe.org/.
14. Runa. *RUNA WFE*. 2007 [cited 22.05.2007]; Available from: http://runawfe.sourceforge.net/.
15. WfMOpen. *WfMOpen*. 2005 [cited 22.05.2007]; Available from: http://wfmopen.sourceforge.net.
16. Aalst, W.M.P.v.d., et al., *Workflow Patterns*. Distributed and Parallel Databases, 2003. **14**(1): p. 5-51.
17. YAWL. *YAWL: Yet Another Workflow Language*. 2007 [cited 22.05.2007]; Available from: http://yawlfoundation.org/product/index.php.
18. Fitzgerald, B. and T. Kenny. *Open Source Software can Improve the Health of the Bank Balance - The Beaumont*. in *24th International Conference on Information Systems (ICIS)*. 2003. Seattle.
19. Scacchi, W. *OpenEC/B: electronic commerce and free/open source software development*. in *The 5th Workshop on Open Source Software Engineering, part of The 27th International Conference on Software Engineering (ICSE 2005)*. 2005. St. Louis, Missouri: ACM Press, New York, NY, USA.
20. Gurbani, V.K., A. Garvert, and J.D. Herbsleb, *A case study of open source tools and practices in a commercial setting*. SIGSOFT Softw. Eng. Notes, 2005. **30**(4): p. 1-6.
21. Murray, M. *Strategies for the Successful Implementation of Workflow Systems within Healthcare: A Cross Case Comparison*. in *36th Annual Hawaii International Conference on System Sciences (HICSS'03)*. 2003: IEEE Computer Society Washington, DC, USA.

22. Stoilova, K. and T. Stoilo. *Comparison of Workflow Software Products*. in *International Conference on Computer Systems and Technologies - CompSysTech'2006*. 2006. Veliko Tarnovo, Bulgaria.