POLITECNICO DI MILANO

Facoltà di Ingegneria dell'informazione

Corso di Laurea Specialistica in Ingegneria informatica

DESIGN AND IMPLEMENTATION OF AN OPEN SOURCE XPDL COMPLIANT WORKFLOW ENGINE

Supervisor:
Prof. Giuseppe Pozzi

Master thesis by:
Vinnichenko Alexey
Matr. 734189
Hossain Khondoker Zahid
Matr: 722488

2010

Abstract

Workflow Management Systems (WfMS) are getting important day by day as they allow one to automate, speed up and increase efficiency of work in enterprise. This paper presents key aspects of design and implementation of a Workflow Management System engine (the core of WfMS in short). Brief history and future trends of WfMS development have been shown and reasonable consderations towards XPDL as a standard to choose have been made basing on BPM industry surveys. Also the anaylsis of BPM standards shows the place of XPDL among other noticeable standards like BPMN and BPEL, that are rather complements then competitors with respect to XPDL.

We base our architecture of WfMS on the Open Business Engine(OBE) architecture that complies to our needs as it takes into accont Workflow Management Coalition (WfMC) standards (Workflow Reference Model or XPDL specification, for instance) and allows us to use modern thechnologies, like Java EE, that gives application security, transaction control, resource management and other features. Finally we present our WfMS engine that supports XPDL 1.0 and runs on Java EE 5 certified application server JBoss 5.

Sommario

Workflow Management Systems (WfMS) stanno diventando giorno dopo giorno molto importante in quanto permettene di automatizzare, accelerare e aumentare l'efficienza del lavoro nelle imprese.

Questo documento presenta gli aspetti chiave della progettazione e della realizzazione di un motore di Workflow Management System (in breve: il cuore del WfMS). Un breve racconto e le tendenze future dello sviluppo WfMS sono stati indicati e considerati ragionevoli verso XPDL come standard, quindi le scelte sono state fatte basandosi su indagini di settore BPM. Inoltre l'analisi degli standard BPM mostra che l'area di XPDL tra gli altri standard: BPMN e BPEL. Questi ultimi, si presentano conplementari rispetto a XPDL.

Basiamo la nostra architettura di WfMS sulla Open Business Engine (OBE), architettura che è conforme alle nostre esigenze in quanto prende in considerazione Workflow Management Coalition (WfMC) come standard (esempio: Workflow modello di riferimento le specifiche XPDL) e ci permette di usare tecnologie moderne, come ad esempio Java EE, la quale dà sicurezza nelle applicazioni. Inoltre da la possibilità di controllare le transazioni, gestione le risorse e altre caratteristiche. Infine presentiamo il nostro motore di WfMS che supporta XPDL 1.0 e gira su Java EE 5 application server JBoss 5.

Table of Contents

Table of Figures

1. Introduction

This first chapter presents an overview of the workflow and business process model world, introducing some preliminary definitions and general concepts which shall be deeply detailed in the remainder. For the structure of the work, please, see section 1.4

1.1. Workflow Process History

One of the first people to describe processes was Adam Smith in his famous (1776) example of a pin factory. Inspired by an article in Diderot's Encyclopedia Smith described the production of a pin in the following way:

"*One man draws out the wire, another straights it, a third cuts it, a fourth points it, a fifth grinds it at the top for receiving the head: to make the head requires two or three distinct operations: to put it on is a particular business, to whiten the pins is another ... and the important business of making a pin is, in this manner, divided into about eighteen distinct operations, which in some manufactories are all performed by distinct hands, though in others the same man will sometime perform two or three of them.*"

Smith also first recognized how the output could be increased through the use of labor divisions. Previously, in a society where production was dominated by handcrafted goods one man would perform all the activities required during the production process, while Smith described how the work was divided into a set of simple tasks, which would be performed by specialized workers. The result of labor division in Smith's example resulted in productivity increasing by 24,000 percent (e.g. that the same number of workers made 240 times as many pins as they have been producing before the introduction of labor divisions).

Smith did not advocate labor division at any price and per se. The appropriate level of task division was defined through experimental design of the production

process. In contrast to Smith's view which was limited to the same functional domain and comprised activities that are in direct sequence in the manufacturing process, today's process concept includes cross-functionality as an important characteristic. Following his ideas the division of labor was adopted widely, while the integration of tasks into functional, or cross-functional, process was not considered as an alternative option until much later.

The above text empathizes one of the first modern concept of “process” and “workflow”, since we can’t clearly identify an exact period of the history when those concepts were born.

If we limited the concept of process and schema at what our brain suggests us … we could say that those ideas born when the first man was born (or maybe early before). At the primitive time it can be imaged as a simple list of steps to follow in order to hunt prey, nowadays it could be seen inside the sequence of actions needing to assemble a car. For sure the attention to the processes and workflow theory has given a great boost in the last decades: the strength of informatics and computer evolution has made, what was an unconscious mental schema or rough manual process, a new real science.

Today Workflow Process Models can be considered as a consolidated matter which takes efforts from several different fields (database theory, programming languages flexibility, graph theory, system interoperability …).

Anyway, Workflow and Business Process Model continue growing inside a society like ours where technologies and innovation endlessly introduces new concepts and topics which need to be controlled, ordered and coded under common and efficacy modeling schemas.

A more formal classification traces the workflow definition as:

“A techniques to model business process such as the flow chart, functional flow block diagram, control flow diagram, Gantt chart, PERT diagram, and IDEF *have emerged since the beginning of the 20th century. The Gantt chart were among*

the first to arrive around 1900, the flow charts in the 1920s, Functional Flow Block Diagram and PERT in the 1950s, Data Flow Diagrams and IDEF in the 1970s. Among the modern methods are Unified Modeling Language and Business Process Modeling Notation. Still these represent just a fraction of the methodologies used over the years to document business processes. The term "business process modeling" itself was coined in the 1960s in the field of systems engineering by S. Williams in his 1967 article "Business Process Modeling Improves Administrative Control". His idea was that techniques for obtaining a better understanding of physical control systems could be used in a similar way for business processes. It took until the *1990s before the term became popular.*

In the 1990s the term "process" became *a new productivity paradigm. Companies were encouraged to think in processes instead of functions and procedures. Process thinking looks at the chain of events in the company from purchase to supply, from order retrieval to sales etc. The traditional modeling tools were developed to picture time and costs, while modern methods focus on cross-function activities. These cross-functional activities have increased severely in number and importance due to the growth of complexity and dependencies.*

New methodologies such as business process redesign, *business process* innovation, business process management, integrated business planning *among others all "aiming at improving processes across the traditional functions that comprise a company".*

In the field of software engineering the term "business process modeling" opposed the common software process modeling, aiming at focusing more on the state of the practice during software development. In *that time early 1990s all existing and new modeling techniques to picture business processes were considered and called "business process modeling languages."*

In the Object Oriented approach, *it was considered to be an essential step in the specification of Business Application Systems. Business process modeling became*

the base of new methodologies, that for example also supported data collection, data flow analysis, process flow diagrams and reporting facilities. Around 1995 the first visually oriented tools for business process modeling and implementation were being presented".

1.2. Workflow and Business Process Modeling Concepts

"Workflows describe business processes as the coordinated execution of simple activities (tasks) by human or automatic executors (agents) [1]"

Workflow helps us in every automation and standardization phases, either for an easy sequence of daily mechanical activities (e.g. the wake up once alarm clock rings, washing the face, having breakfast, wearing …), or a complex sequence of steps (which could be the start-up procedure for launching the Discovery shuttle). Based on this, with a more formalism, we can give the definition of **Workflow** as follows:

A workflow is concerned with the automation of a business process, in whole or partly, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [2].

Once clarified the concept of workflow, we can introduce some other definitions which are constantly used in all the rest of the paper.

A **Workflow Management Systems** (WfMSs) is a software system which supports and coordinates the execution of different activities inside a common organization.

- Nowadays the Workflow Management Systems are used within a very wide set of scenario, commonly we can find them in:
- Bank loans
- Purchase Orders
- Personnel recruitment
- Hospital admission

- Provisioning orders
- Generic tasks and activities assignment
- …

In the following chapters we see some of these examples, some of them taken from real scenarios and some other generated in order to stress the notion of temporality, which is the main topic of this paper.

Process definition: it is the formal representation of business processes in such a way that their simulation or automatic manipulation, as well as their enactment, is possible by a WfMS. The process definition is made of a set of activities, with relationship among activities, along with criteria to start and to complete a process, and with information about single activities, participant, document and related data, required software application etc.

Business Process Model definition: it is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. The BPM lifecycle (see Fig.1.1) begins by modeling the business process (taking an existing process or starting from scratch), followed by testing it, deploying it, and finally monitoring its execution in a production environment.

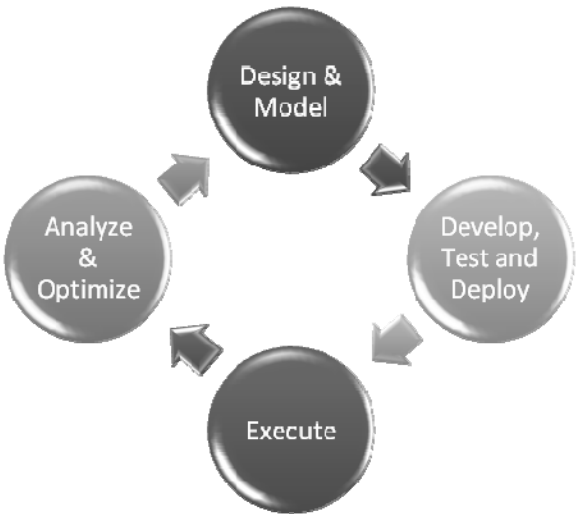

Figure 1.1: Business Process Model rule

The first step is **modeling**. It includes a phase of analysis if the modeling is being done for an existing process and if the modeling is being done for a new process, then it includes a design phase. Modeling is usually done by a business analyst who identifies and connects the basic building blocks of the process such as activities, roles, data, and so on.

The **testing** phase consists of two steps, namely, validation and verification. Validation asserts correctness of the solution while verification checks conformity to the requirements. These activities include several debug iterations and test runs in a development or testing environment. Once the process model is known to be correct, the analyst or programmer deploys the model in a process engine. This engine parses and identifies the model, and then **executes** the instructions and actions associated with it. The next step is monitoring the process in place. During monitoring, we spot the bottlenecks, superfluous steps, and possible automation activities. The **optimization** can be done manually from the feedback obtained in production or can be simulated from artificial or historical load. The optimization must be modeled

again, restarting the BPM lifecycle. The BPM cycle can be repeated until the business process performs like a fine-tuned machine.

1.3. The Relevance of Workflow and Business Process Modeling

Today's marketplace is more global and competitive than ever. Big players are trying to attract new clients along with maintaining the existing ones (thereby making more profit for shareholders), and smaller players are entering the markets with innovative products. To add to this, customers are demanding better prices and service and governments are continuously changing regulations.

Businesses redefine themselves in order to adapt to this environment or to create new market niches for exploitation.

Technology plays an important role in realizing these changes and adaptations. With the appropriate technology and tools, businesses can reduce costs, raise margins, and make the most of the existing information to understand the clients better and thus create new markets based on this data. Businesses are focusing on internal efficiency, which aims at realizing the goals of making profit, reducing costs, creating and maintaining customers, and negotiating with suppliers in faster, cheaper, and improved ways.

With the appropriate use of technology, businesses can benefit in many ways such as greater revenue, bigger margins, automated processes, improved decision making, and so on. Every business has a set of connected activities and functions driven by business rules such as sales, account receivables, R & D, and so on.

The daily operation of such essential processes costs a big chunk of revenue from shareholders. Business Process Management (BPM) technology promises continuous enhancement of the business processes. To diminish these costs and maximize the profit generated, each business process must be efficient and smooth.

The more visible these processes are, the more we can modify them to align with the organizational strategy.

Every corporate strategy and company mission wants to treat the customer in the best possible way it can, but are the customer service processes aligned with the strategy?

Is the bureaucracy of my service as minimal as possible?

Are my business processes impeding the efforts of employees to excel at customer support?

Am I generating long-lasting bonds between my customers and my company?

These questions can only be answered by knowing the processes, the actors, the context, etc.

Today's business executives are held accountable for the performance of the processes they manage. Accountability needs objective measures, and how can we measure what we don't know?

In this scenario one of main key point to lead in the Market is to quickly adapt our production to the Customer requirements as much as fast as possible and optimizing the costs.

Workflow and Business Process Modeling support these occurrences providing flexibility, integration of applications and a reduction in development costs.

In practice the main concept of "**standardization**" together with the increase of "**automation**" are the answers to success in our current state of affairs: especially looking at the recent years when the diffusion of common processes and procedures has exited from the legacy small firms, involving more wide scenarios, going towards a global situation which including different people in different countries.

Workflow and BPM are instruments to meet the needs of an increasingly growing globalization.

1.4. State of the Art

Nowadays the importance of the Workflow and BPM in general has acquired such a vast relevance to attract many Companies which try to impose their own

standards and models (today, few hundreds of different Workflow Management Systems are available on the market).

Despite of this “war” only few of them have already the characteristic to survive this morass of specifications; more in details in **Ошибка! Источник ссылки не найден.** we introduce the most significative companies with their own consolidated standard applicative.

Depending by the different structures and degree of complexity the different Business process models can find several different fields of application which can range from simple administrative task like the management of travel requests, to more compounded structures like telecommunication processes or more complex military procedures.

The figure below gives an overview how this application depending by the complexity of the structures where they are applied.

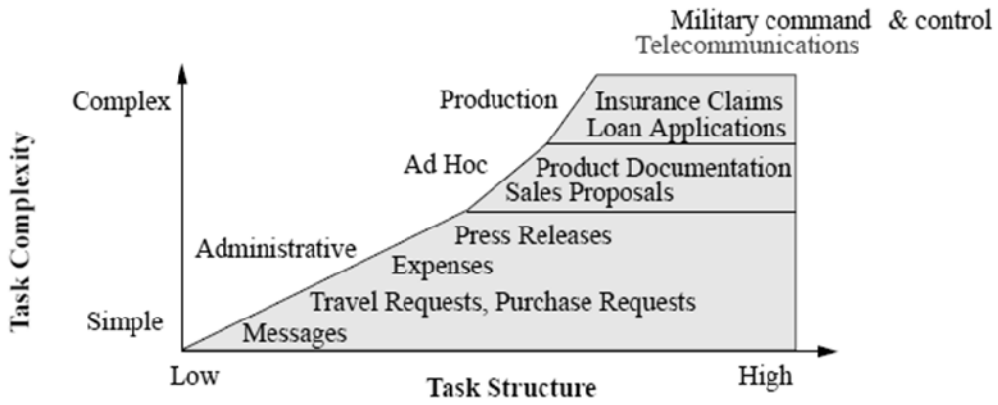

Figure 1.2: Workflow utilization

1.5. General workflow requirements

In order to satisfy this requirement a Workflow schema has to fulfill needs like:

- considering **exceptions**: asynchronous events (change the environments variables, change in the organizations, start or completion of a task) which need to be managed suitably;
- adapting at the **change of process models**;
- managing **temporal aspects**: it takes in consideration all the timing issues experiencing both at the design time and at the run-time, issues like time constraints and deadlines.

1.6. Main goals of the thesis work

Primary goal of the thesis work follows from its title and consists in design and implementation of an Open Source, XPDL compliant engine of Workflow Management System (WfMS). We are giving a temporal name for our Workflow Engine: XPDL Complaint Engine (XCE).

Our design should be based on proven or at least leading standards; therefore we will try to show the place of chosen standards in the world of BPM.

Standards support design, technologies support implementation and that's why another goal of the thesis work is finding technologies that will suit the requirements of a workflow and WfMS.

At the end, the future work should be outlined and formed according to achieved results.

1.7. Tasks

In order to achieve the main goals the following tasks must be performed:

- learn workflow systems and their origins;
- learn technologies and standards related to WfMS;

- choose appropriate technology(including programming language), design WfMS based on selected; standards or developed ideas;
- learn specific libraries and frameworks of a particular language in order to use them as components for a workable system;
- implement engine of WfMS while performing tests on completed parts while going back to design problems or component selection if needed.

1.8. Structure of the work

As the notion of workflow and BPM are given, and goals of the work are outlined, we would like to show the structure of the rest of the paper in a nutshell.

Section 2 provides the reader with general ideas around WfMS (its history, typical architecture, description of its components etc.)

Section 3 is dedicated to a problem of WfMS design (WfMS engine design in particular) from standards overview to design choices.

Section 4 describes developed system and demonstrates its work.

Section 5 shows the achievements of the work

Section 6 sums up the work being done and lists the main features of developed system.

Section 7 points out further possible work basing on limitations of the system at hand.

2. Background

2.1. Brief history of WfMS

Two major developments in information technology may be considered as the triggers of the advent of workflow management. The first one is the enormous technological progress in hardware and software technologies in the last two decades. With the rapid improvement in hardware technology a revolution took place in office work. Personal computers and workstations with continuously increasing performance were produced for lower prices. As a result, computer technology has started to dominate the office work. Besides this, network technology enabled the interconnection of computers to build up networks. Furthermore, the connectivity of hardware systems enabled the development of integrated software systems.

The evolution in database systems enabled the integration of office work opening new dimensions for software development. As a consequence of these improvements in hardware technology and database systems, more sophisticated software such as sophisticated word processing, spreadsheet and imaging applications were developed.

However, the overall productivity in application areas did not increase proportionally to these improvements due to the lack of integration of the software products. In order to achieve better results, a change in the concept of application system development was necessary.

This necessity became the second trigger for the advent of workflow management. The existing goal of automating the pieces of application systems had been replaced by the broader new approach of the comprehensive development of integrated application systems. This new goal required a shift from task- or data-centric view to process- or work-centric view, in the application system development process.

The task- or data-centric view focuses on the integration of data to achieve the efficient implementation of distinct tasks with respect to the dependencies among them. In contrast, the process- or work-centric view focuses on the work as a whole. This approach concentrates on the work from a broader perspective by integrating all related programs, processes, functions, data, documents, persons, organizations etc. Hence, this enables the optimization of processes to minimize the costs and increase the efficiency [3].

Workflow management systems are considered as a kind of Information Systems, which development went through the following steps [4]:

1965-1975: decompose applications.

During this period, information systems comprised decomposed applications, each with its own databases and definitions. The applications ran directly on the operating system and had just a simple command line interface or custom-made graphical interface.

1975-1985: database management – "take data management out of the applications".

This period is characterized by the rise of the database management system (DBMS). A database is a permanently available, integrated collection of data files, which can be used by many applications. The use of databases has the advantages that data managed by different applications can be combined, that data structures only need to be defined once, that the organization of data can be handed over to a database management system, and that the same data item only needs to be stored once.

1985-1995: user-interface management - "take the user interface out of the applications".

During this period, the user interface was extracted from the application program. Originally user interfaces were designed by the developers screen by screen,

field by field. Not only did this take up a lot of time, but also each designer had her own style, which meant that every system had to operate in a different way.

1995-2010: workflow management - "take the business processes out of the applications".

Now that data management and user interfacing have largely disappeared from applications themselves, it seems that much of the software is devoted to business processes (procedures) and the handling of cases. Therefore, it has become attractive to isolate this component and find a separate solution for it. Not only can this accelerate the development of information systems, but it also offers the added advantage that the business processes become easier to maintain. A workflow management system (WfMS) manages the workflows and organizes the routing of case data amongst the human resources and through application programs, and WfMS by itself is a distributed systems that cooperates with other WfMS'es of other organizations.

Until the last above mentioned period WfMS didn't actually exist in a form defined above, but as the following software technologies:

Office Automation

Workflow management can be regarded as an outcome of the office automation concept that started in the 1970s. Both workflow management systems and office automation aim at the automation of the execution of work.

The difference is that office automation aims at automating individual tasks of the work, whereas workflow management aims at automating the control of tasks during the execution of a business process. Thus, workflow management systems assist human workers in performing work processes.

Despite the different concepts of reaching automation of work, some fundamental requirements of office information systems such as activity scheduling, function integration, personal assistance and task management can be applied to workflow management systems [5].

Image-Processing

Several business processes involve interactions with paper-based information, which may need to be captured as image data as part of an automation process.

Usually the captured image data is required to be passed to persons involved in a particular process. The persons that work interact with different software programs for different purposes. As a consequence, image-processing systems are required to have some workflow capability either built-in or supplied in conjunction with a workflow management product [6].

Document Management

The wide-spread of computer technology in office work resulted in the replacement of paper documents with electronic documents. The first generation of document management systems were called *passive document management systems* because they only reacted on direct user requests. For instance, a user could query for a certain document or he could lock a class of documents. Context-based retrieval and document management functions (e.g. lock, release, combine, destroy) were the major features of passive document management systems.

Active document management systems enhanced passive document management by incorporating service functions which are based on time management. For example, triggers caused a document to be presented for review after a certain period of time. Oversimplified, this was a first step towards workflow management. Active document management systems are considered as the ancestors of document–centric workflow management systems [3].

Electronic Mail

Enhanced electronic mail systems are used to distribute information among individuals considering their attributes such as organization roles. Thus, electronic mail systems and workflow management systems share some common characteristics.

Groupware Applications

Groupware is defined as computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to shared environment [7].

Groupware is the technology designed to facilitate the work of groups. This technology may be used for communication, cooperation, coordination, problem solving, competition, or negotiation. While traditional communication technologies like the telephone also qualify as groupware, the term is ordinarily used to refer to a specific class of technologies relying on modern computer networks, such as email, newsgroups, videophones, or chat. Groupware technologies are typically categorized along two primary dimensions:

- whether users of the group work/act at the same time (synchronously) or distributed over time (asynchronously)
- whether users work/act in the same place or in different locations

CSCW (Computer-Supported Cooperative Work) is closely related and sometimes confused with groupware. CSCW refers to the field of study which examines the design, adoption, and use of groupware. Despite the name, this field of study is not restricted to issues of "cooperation" or "work" but also examines competition, socialization, and play.

Groupware is regarded as one possible and suitable superordinated application area for workflow management [8].

However, groupware and workflow management systems differ in their rationales. Groupware provides support for unstructured, ad-hoc processes whereas workflow management systems primarily support structured, recurring processes.

Database Management

Conventional database management systems are passive, so that they only response to explicit requests from applications or users. Active database management systems enrich passive database management systems by implementing event-

condition-action (ECA) rules [9]: When an event occurs and if a condition holds, an action will be executed

A simple workflow can be considered as a multi-step activity consisting of miscellaneous activities. Active database management systems can support multi-step activities by implementing each step of a multi-step activity as a transaction. In detail, this is achieved by implementing the control flow of transactions by ECA rules, which are embedded into the database.

Software Development Management

Software development management coordinates the development of software processes. The management of a software development process comprises three major phases: modeling of the software process, analysis of the model and execution of the model [3].

Often a software process has to provide some workflow functionality to transfer development tasks and necessary information among related persons.

Business Process Reengineering

With the change of conditions in the business world and the resulting change in the structure of business, redesigning business processes, also called business process reengineering (BPR), became obligatory for an increasing number of companies. As a consequence, business processes had and have to be remodeled, reanalyzed and redefined to achieve better performance.

Business Process Reengineering handles this problem with tools that support activities in areas such as analyzing, modeling and defining business processes. Here, the focus of interest is on the execution of whole business processes, not on the execution of single tasks.

Moreover, business processes consist of a number of different views on a business: technological view, economical view, information-oriented view etc. Thus, Business Process Reengineering takes different aspects of a business into account.

The conceptual ideas of business process modeling, especially its broad, multi-aspect approach, triggered the development of workflow management systems [3].

2.2. Definitions

The basic terms of workflow terminology and the relationships among them are illustrated in Figure 2.2 according to the Workflow Management Coalition [2]. Next, these basic terms will be defined:

Workflow

A workflow is concerned with the automation of a business process, in whole or partly, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [2].

The automation of a business process is done within the process definition with respect to several process activities, procedural rules and associated control data used to manage the workflow execution.

Workflow Management System

A Workflow Management System (WFMS) is 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 [2].

A workflow management system consists of software components to store and interpret process definitions, create and manage workflow instances as they are executed, and control their interaction with workflow participants and applications [2].

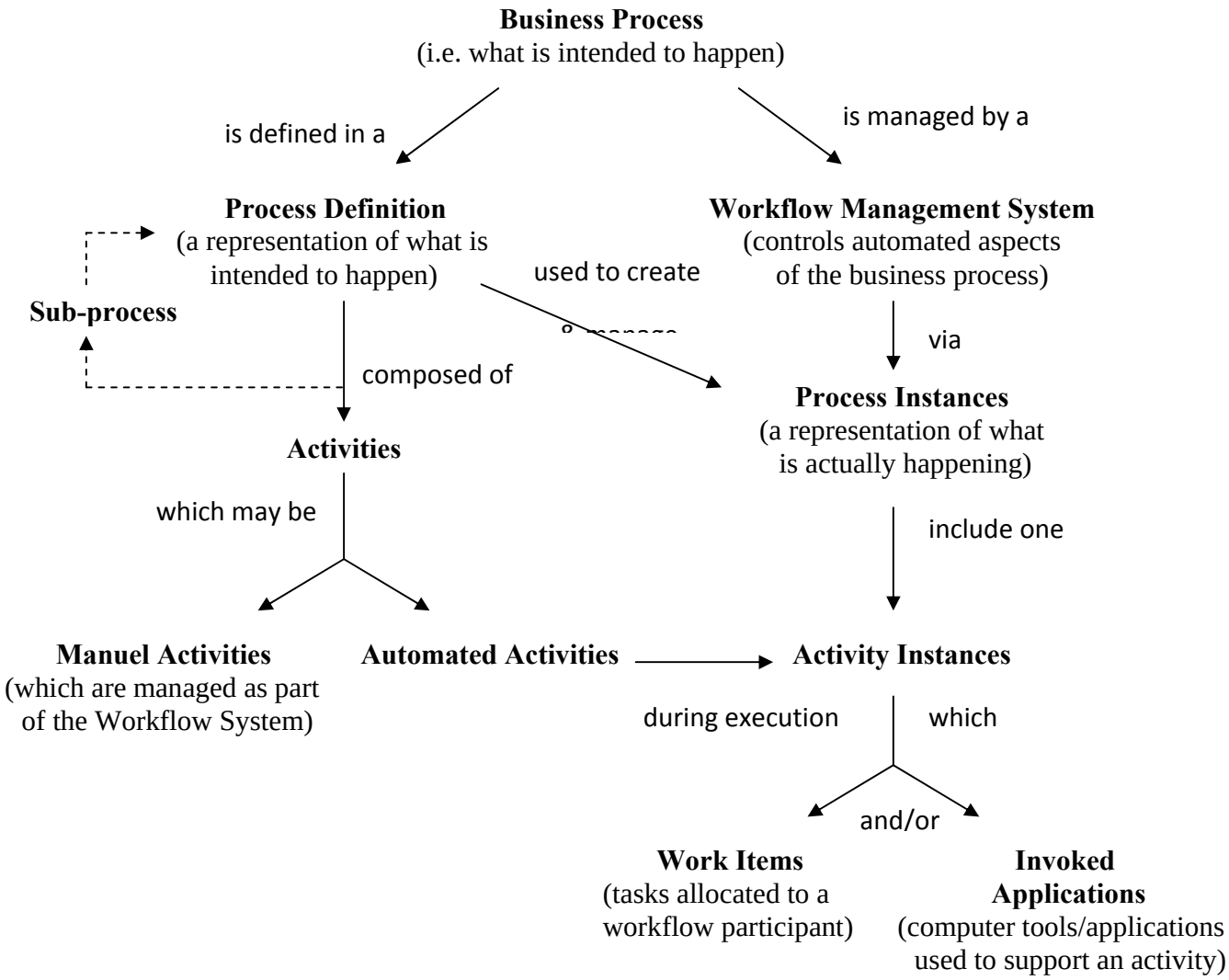

Figure 2.1: Basic Terms and their Relationships

Business Process

A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships [2].

Process Definition

Workflow process is a representation of a business process in a form, which supports automated manipulation, such as modeling, or enactment by a workflow management system. The process definition consists of a network of activities and

their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc [2].

Activity

A description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity. A workflow activity requires human and/or machine resource(s) to support process execution: where human resource is required, an activity is allocated to a workflow participant [2].

Automated Activity

An activity which is capable of computer automation using a workflow management system to manage the activity during execution of the business process of which it forms a part [2].

Manual Activity

An activity within a business process which is not capable of automation and hence lies outside the scope of a workflow management system. Such activities may be included within a process definition, for example to support the modelling of the process, but do not form part of a resulting workflow [2].

Process Instance

A process instance is the representation of a single enactment of a process including its associated data [2].

It therefore represents an instance of a process definition that includes manual and automated aspects.

Activity Instance

The representation of an activity within a (single) enactment of a process, i.e., within a process instance including its associated data [2].

Work Item

The representation of the work to be processed (by a workflow participant) in the context of an activity within a process instance [2]. Usually each activity generates one or more work items, which may be presented to a user via a work list.

Invoked Application

An invoked application is a workflow application that is invoked by the workflow management system to automate an activity, fully or partly, or to support a workflow participant during the processing a work item.

2.3. Architecture of Workflow Management Systems

Workflow management systems are typically used in areas such as administration, banking and insurance. However, they have many other application areas including industrial or manufacturing applications.

As a result of multiple application areas, workflow management products vary in their architecture, implementation techniques and specification areas. Despite this variety, all workflow systems show some common characteristics.

Therefore, the Workflow Management Coalition defined the workflow reference model to define a common model for the architecture including components and interfaces.

At the highest level, all workflow management systems may be characterized as providing support in three functional areas:

The build-time functions: these are used for the definition, the modeling and the analysis of workflow processes and related activities,

The run-time process control functions: these are concerned with managing the execution of workflow processes,

The run-time activity interactions: these enable the cooperation with human users and IT applications during the execution of various steps of workflow processes.

Figure 2.2 shows the basic characteristics of workflow management systems and the relationships between these main functions [2].

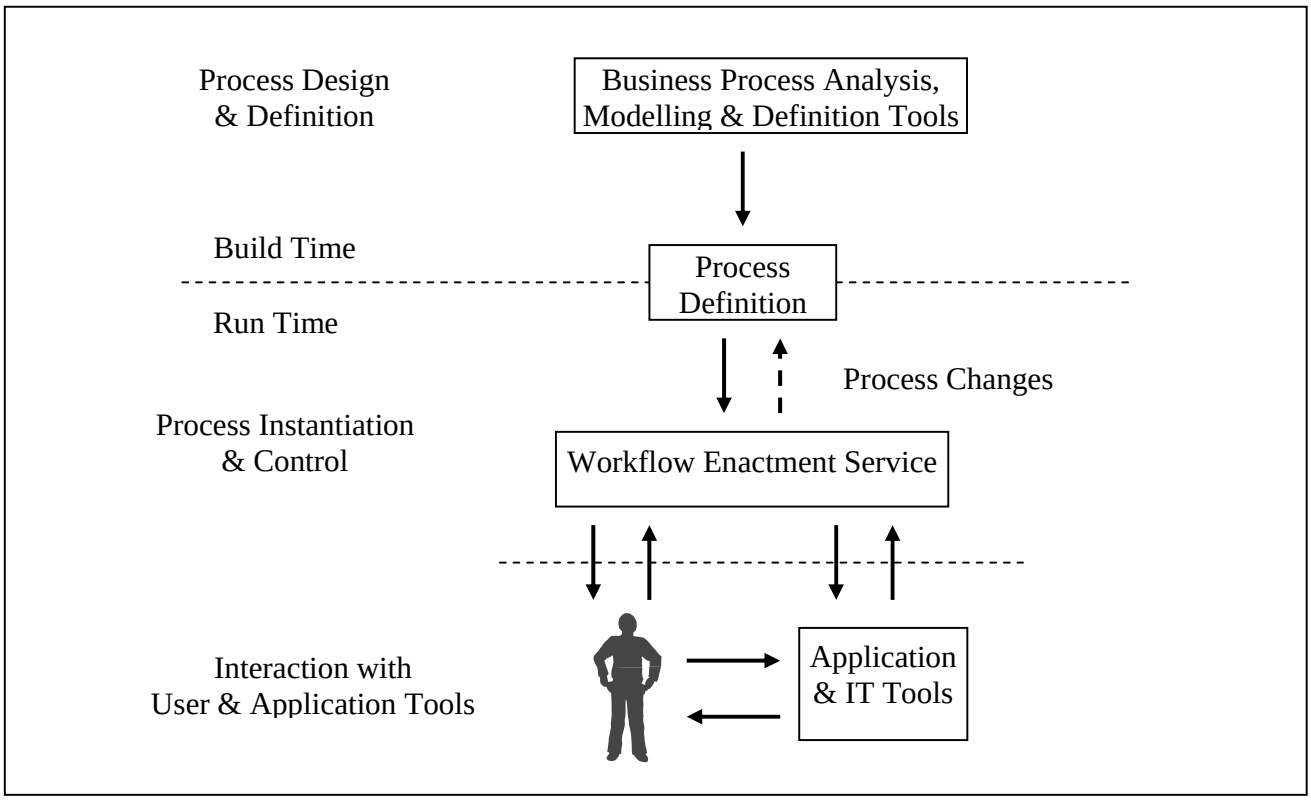

Figure 2.2: Workflow Overview

Build-time Functions

Build-time functions are used to define business processes in a way that can be interpreted and executed by the workflow management system later. Therefore a mapping of a real world process into a definition that can be managed by the workflow enactment service is necessary.

A *process definition* is also known as a process model, a process template, or a process metadata. The workflow management coalition prefers the term process definition. A process definition is defined as the computerized representation of a process that includes the manual definition and workflow definition [6].

The process definition has to comprise all required information for the execution of a process. This includes starting and finishing conditions, constituent activities, rules defining the execution, etc.

The process definition may be given in textual or graphical form or in a formal process definition language depending on the implementation of the definition tool of a specific product. Some workflow management systems have the ability to allow dynamic changes to process definitions at run-time, as shown in Figure 2.2 with the dashed arrow.

It is clear that the definition of a process is a prerequisite for the management of the execution of a process. Thus, build-time functions determine the expressiveness, comprehensiveness and functionality of a workflow management system.

Run-time process control functions

The process definition is interpreted at run-time by the enactment software, which instantiates a process instance and performs the enactment of the functional components in order to execute the process instance. The run-time process control functions are concerned with the mapping of the process as modelled in the process definition and the process in the real world. Therefore run-time control functions have to manage the interactions of users and IT application tools.

The core component of run-time control mechanism is the basic workflow management software, which is also called as workflow engine, responsible for process creation and deletion, control of the activity scheduling within an operational process and interaction with application tools or humans [6].

Run-time activity interactions

Individual activities, defined as steps of a process, are human- or computer-oriented and usually require the use of some IT tools or programs (e.g. word-processor). For the management of process execution, interactions with the process control software at run-time are unavoidable. These interactions are necessary to transfer control between activities, activate necessary application tools, change data etc.

Therefore, the workflow management coalition is concerned in standardizing interfaces that manage the necessary interactions. Detailed information about these interfaces will be given later in this work.

2.4. General Model of Workflow Management Systems

Despite the differences, workflow management products have some basic implementation structure in common. Thus, the Workflow Management Coalition defined an implementation model of a workflow system that suits the majority of the workflow management products.

The defined implementation model is an abstract model, which defines major functional components of a workflow system and interfaces between them. Consequently, there are several implementation variations of this model. Some vendors may prefer not to implement all the defined interfaces between the functional components. Therefore the Workflow Management Coalition defines different conformance levels to identify supported functions of the implementation model. The main functional components of a generic workflow system are illustrated in Figure 2.3.

The generic model has three types of components [6]:

Software components which provide support for various functions within the workflow system (shown in dark fill)

Various types of process definitions and control data (shown unfilled) which are used by one or more software components

Applications and application databases (shown in light fill), which are a part of the workflow product but may be invoked by it as a part of the total workflow system.

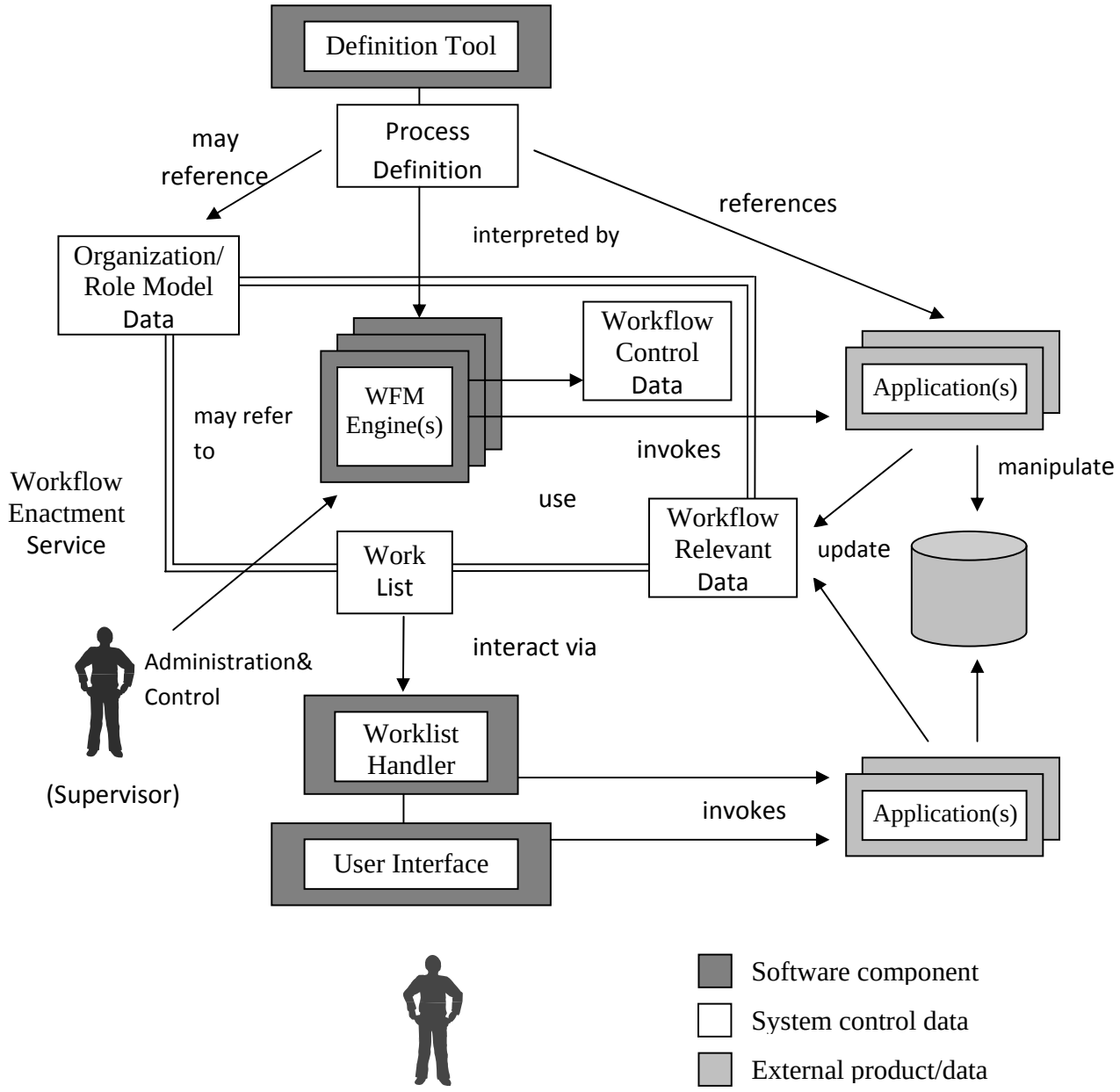

Figure 2.3: Generic Workflow Product Structure

Now some important functional components of the generic model will be discussed in more detail:

Process Definition Tool

The process definition tool enables the transformation of real world processes into a representation form that can be handled by the workflow engine. The representation can be in textual or graphical form.

The process definition tool may be part of a workflow management product, part of a business process analysis product or a standalone application from another vendor. If the definition tool is not part of the workflow management product it has to work with, a compatible interchange format must be implemented in both systems.

Workflow Enactment Service

The workflow enactment service consists of one or more workflow engines in order to create, manage and execute particular workflow instances [2]. Hereby, each workflow engine is responsible for the execution of a specific process instance.

The task of the workflow enactment service consists of interpreting the process definition, controlling the instantiation of processes, sequencing the related activities, adding work items to the related user work lists and invoking auxiliary application tools. The capability of invoking auxiliary applications is an important facility of workflow engines. Some products are limited to a number of tools whereas others can be extended to invoke a wider range of tools, which may be local or remote to a workflow engine.

The workflow enactment service maintains internal control data that includes state information about processes and activity instances. The workflow control data may also include further information, e.g., information about check pointing and recovery/restart that are used by the workflow engines to coordinate and recover from failure conditions [6].

The workflow enactment service gets all necessary information for the execution of a process instance from the process definition and other relevant data. For example, the process definition may refer to an organization-role model that contains information concerning the organizational structure and roles within this

structure. The workflow enactment service is responsible for the linking of this information to the participants included in the execution of a process.

Worklists

Where user interactions are necessary within the process execution, the workflow engine(s) places items on worklists for attention by the worklist handler, which manages the interactions with the workflow participants [6]. In some systems this process is invisible for users, so that users only get the next task provided by the worklist handler. In other systems users can see the whole worklist and choose individual items of work from the list.

Worklist Handler & User Interface (Workflow Client Application)

The workflow handler is a software component that is responsible for the interaction of the workflow enactment service with workflow participants. It provides support for the execution of processes by managing the activities requiring user attention.

The complexity of the worklist handler depends on the specific product. In some systems it may support sophisticated functions such as controlling the allocation of the work between a set of users to provide load balancing and work assignment.

In addition to these worklist handling functions, workflow engines typically support a wider range of client applications, including sign-on and –off of workflow participants, requesting the commencement of an instance of particular process types, requesting work items queued for particular participants, etc [6] This wide aspect is the reason why the term workflow client application is preferred and used instead of workflow handler in the reference model.

In Figure 2.4 user interfaces are shown as separated pieces of software. In some systems the worklist handler and the user interface may be integrated into a single software service. In several companies, especially in enterprises working with several software products, standardized interfaces for participants are required.

Therefore, it is an important facility for workflow management system products to have user interfaces as separated software components.

Both worklist handler and user interface may be able to invoke appropriate supporting applications. This facility is necessary to support a user in particular tasks to be undertaken. For example, it may be required that every time an employee finishes a required activity, a confirmation message should be sent automatically to his supervisor.

Besides this implementation model, as described by the Workflow Management Coalition, there are also other implementation scenarios for a workflow management system: The structural model of a generic workflow product identifies a series of software components and interfaces. In a concrete product implementation this structure may be realized in a variety of different ways. This is an important area of product differentiation [6].

Workflow Control Data, Workflow Relevant Data and Workflow Application Data

Information about the internal state of a workflow system with its processes and activity instances is termed as workflow control data. This information is workflow enactment service-specific and not accessible or interchangeable via workflow application programming interface (WAPI) commands. Nevertheless, some of this information may be provided by the workflow enactment service in response to specific requests.

The data, which is generated and updated by workflow application programs, are called workflow relevant data. Contrary to workflow control data, workflow relevant data may be manipulated by both workflow enactment services and workflow applications. Workflow enactment services can access these data in order to make navigation decisions or control operations.

Workflow relevant data is also defined as data that is used by a workflow enactment system to determine the state transitions of a workflow instance, for

example within pre- and post-conditions, transition conditions or workflow participant assignment [2]. Additionally, workflow relevant data may be transferred between activities by the workflow enactment service.

Workflow application data is application-specific and not accessible by the workflow enactment software. Thus, it can be accessed and manipulated only by the applications.

In case of heterogeneous workflow enactment services, workflow relevant data and workflow application data may be transferred or transformed between workflow enactment services.

Figure 2.5 illustrates the data types and their usage within a workflow management system

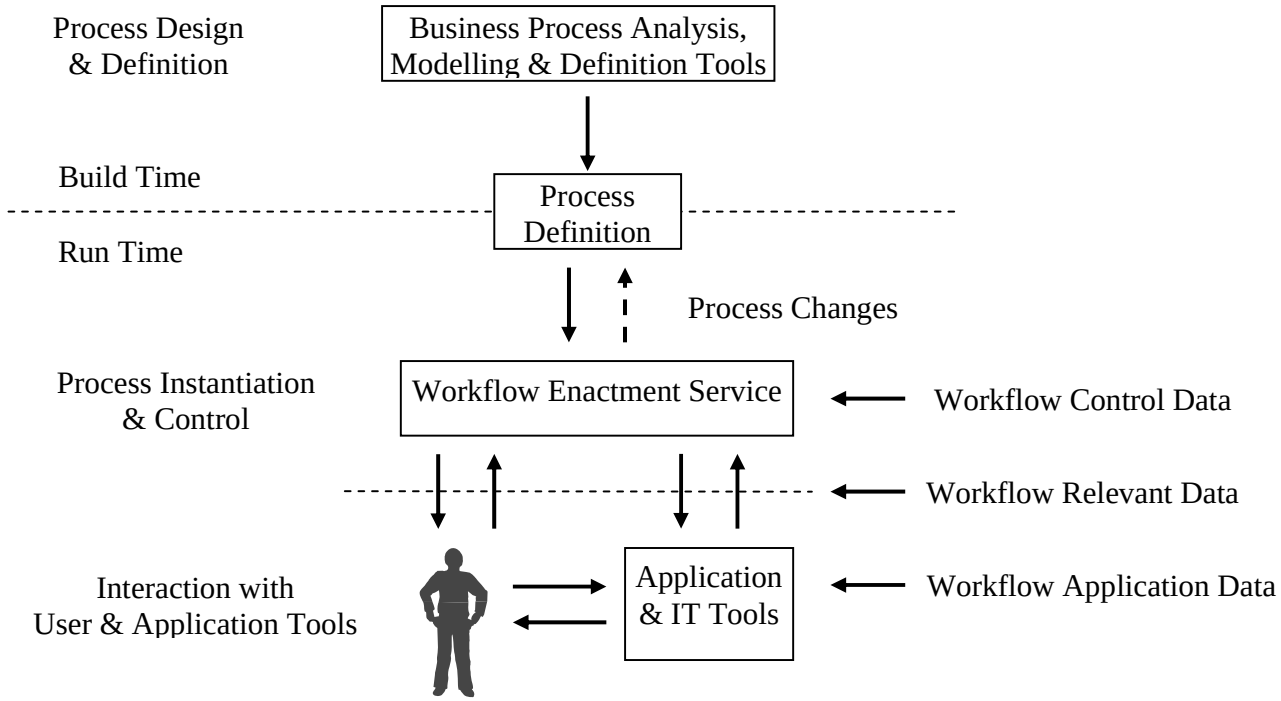

Figure 2.4: Types of Data in Workflow Management Systems

2.5. WFMC Reference Model

Workflow management (WFM) is a fast evolving technology that is used in a variety of industrial processes. The evolution of workflow management concentrates on the automation of business processes where interactions of human and machine-based activities play a major role. Within workflows, information or tasks are passed among participants in a way that is governed by rules or procedures.

The concept of workflow management has been researched and developed since the beginning of the 1980s with the efforts of several research groups. Thus, workflow software products evolved from several different origins. While some products have been developed as pure workflow software, many have evolved from image processing systems, document management systems, relational or object database systems, and electronic mail systems. As a consequence some vendors have invented new specific terminology and interfaces, whereas others adopted terminology and interfaces from other technologies.

This development resulted in the creation of a large number of different workflow products, which were focused on specific aspects of workflow management. This enabled the users to choose a product that met their specific application needs best. On the other hand, the definition of a common terminology to enable the standardization of workflow products became necessary. The definition of a common terminology is the only way to define new standards that enable different workflow products to interoperate.

Only a successful standardization could enable workflow management system users to choose the product that suits their requirements best in a specific application area and combine the strength of different products in one infrastructure.

Due to these requirements, several vendors developing workflow management systems founded the Workflow Management Coalition (WfMC) in 1993 as a nonprofit international organization. Although there had been a large number of products specialized on different aspects, all products had some common

characteristics. This helped defining standards for various functions so that some level of interoperability among different products could be achieved.

The Workflow Management Coalition (WfMC) has been established to identify these functional areas and to develop appropriate specifications for implementation in workflow products. It is intended that such specifications will enable interoperability between heterogeneous workflow products and improved integration of workflow applications with other IT services such as electronic mail and document management, thereby improving the opportunities for the effective use of workflow technology, to the benefit of both vendors and users of such technology [6].

The WfMC states its missions as:

- to increase the value of customers investments with workflow technology;
- to decrease the risk of using workflow products;
- to expand the workflow market through increasing the awareness of workflow.

After the definition of a common terminology of workflow management, the WfMC focused on the standardization of interfaces between modules of a workflow management system but also between workflow management systems and their clients. Several working groups were created to discuss and propose these interfaces in order to standardize them. Figure 2.5 illustrates briefly the components and interfaces standardized by WfMC:

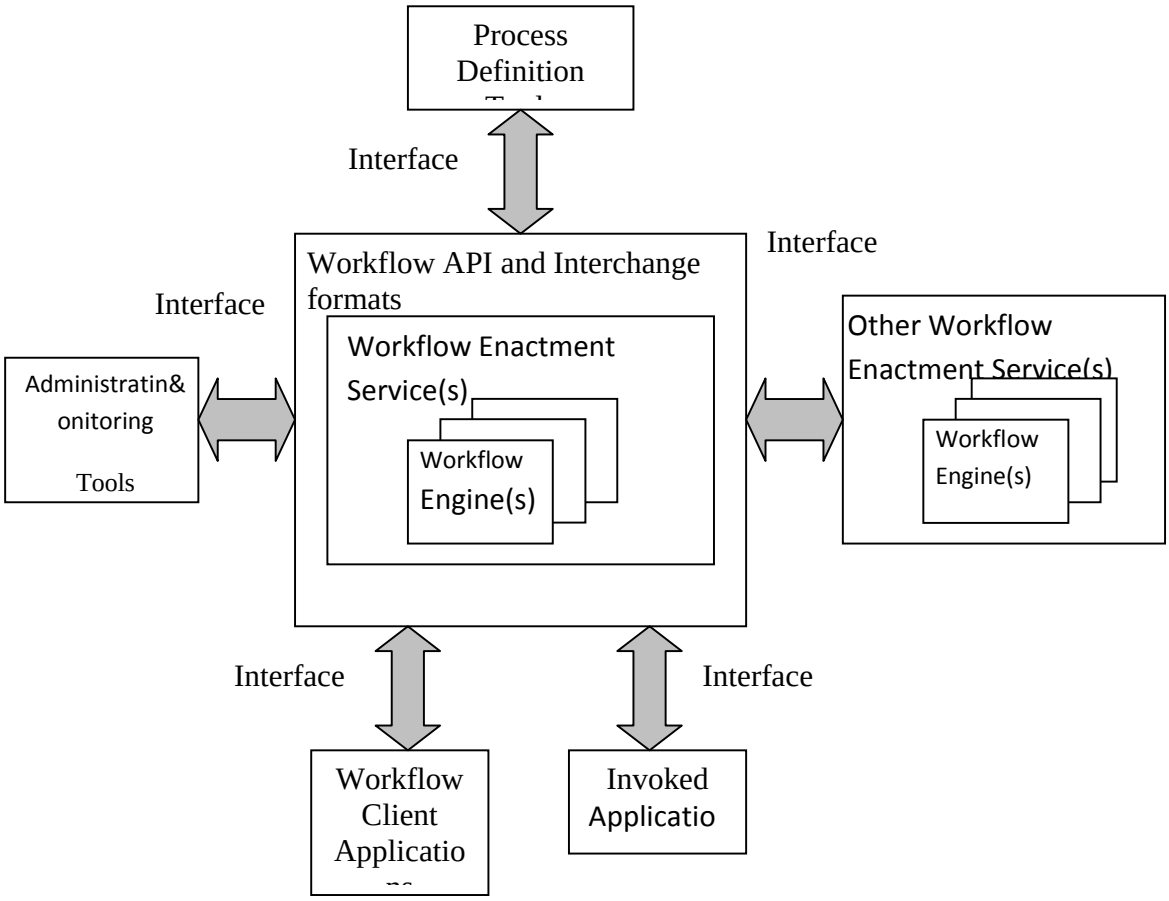

Figure 2.5: The WfMC Workflow Reference Model – Components & Interfaces

The Workflow Management Coalition has defined a common reference model for workflow management products by identifying their characteristics, terminology and concepts. The goal of this chapter is to study this workflow reference model. Here, the standards and publications of the Workflow Management Coalition will be used as the main reference.

Workflow Enactment Service

A workflow enactment service is a software service that consists of one or more workflow engines in order to create, manage and execute particular workflow instances. Applications may interface to this service via the workflow application programming interface (WAPI) [2].

In the reference model, as illustrated in Figure 2.5, there is a separation between the process and activity control service and the application tools and end user

tasks, which are required for the execution of the activities. As a consequence interactions between the workflow enactment service and external resources occur with the help of two interfaces:

- the Client Application Interface, also termed Interface 2, which is responsible for the cooperation of workflow enactment service with the worklist handler;
- the Invoked Applications Interface, also termed Interface 3, which supports the workflow enactment service to activate necessary tools to execute an activity;

The workflow management service may be centralized or functionally distributed. In a distributed workflow enactment service several engines are involved in the enactment of one process. Each workflow engine interacts only with users and application tools managed by it.

A homogenous workflow enactment service comprises one or more compatible workflow engines, which provide the run-time execution environment for workflow processes with a defined set of (product-specific) process definition attributes [2].

A heterogeneous workflow enactment service comprises two or more homogenous services, which follow common standards for interoperability at a defined conformance level [2].

When heterogeneous products are involved, a standardized interchange is necessary between workflow engines. Via interface 4, the enactment service may transfer activities or sub-processes to other enactment services for execution [6]. Interface 4 enables the delegation of a particular work, typically a sub-process, to other workflow engines. Transfer of process definition is not part of interface 4's responsibility. The activities and sub-processes, which have been delegated to a remote workflow enactment service for the execution, are pre-defined on that workflow enactment service.

Workflow Engine

A workflow engine is a software service or “engine” that provides the run-time execution environment for a process instance [2].

A workflow engine typically executes the following tasks [6]:

- Interpretation of the process definition
- Control of process instances: creation, activation, suspension, termination etc.
- Navigation between process activities, which may involve sequential or parallel operations, deadline scheduling, interpretation of workflow relevant data etc.
- Sign-in and sign-off of specific participants
- Identification of work items for user attention and an interface to support user interactions
- Maintenance of workflow control data and workflow relevant data, passing workflow relevant data to/from applications to users
- An interface to invoke external applications and link any workflow relevant data
- Supervisory actions for control, administration and audit purposes

A workflow engine may be responsible for the whole run-time environment but also for only a part of it. In the latter case several workflow engines constitute the workflow enactment service, where a workflow engine may be responsible only for a specific type of processes.

Workflow Application Programming Interface (WAPI) & Interchange Formats

WAPI is an abbreviation for Workflow API’s and Interchange Formats, published by the Workflow Management Coalition, and incorporating specifications

to enable interoperability between different components of workflow management systems and applications [2].

WAPI may be regarded as a set of API calls and interchange functions supported by a workflow enactment service at its boundary for interaction with other resources and applications. Although this architecture refers to the five interfaces within WAPI, a number of functions within each of these interfaces are common (for example process status calls may be issued from the client application interface or the administration interface) [6].

Most of the WAPI are APIs with defined parameter and result sets. Furthermore WAPI is usually required to define data interchange formats, e.g., for the interchange of workflow relevant and application data among the interfaces, for the exchange of process definitions, etc…

Now the five interfaces within the WAPI will be described (see Figure 2.5):

Process Definition Tools & Workflow Definition Interchange (Interface 1)

Process definition tools are software tools that are used by process designers to create a representation of a business process, including all process related data, which can be interpreted by a workflow enactment service later.

Process definition tools have different levels of sophistication. Besides modeling they may include services for the definition and analysis of process models. Another important aspect is the ability to handle organizational data. Organizational data enables the representation of the organization model within workflows. As an example this data may include information about the roles of participants in an organization, hierarchical relationships among them, etc.

Most of the products in the market provide definition tools that can represent the real world processes in a form that can be interpreted only by the specific workflow management product they belong to.

The process definition import/export interface, Interface 1, enables the interchange between a process definition tool and a run-time workflow management

software. This interface helps to separate process modeling and process execution responsibilities.

Consequently, implementation of the process definition import/export interface brings two important advantages. First of all, separating of the build-time and run-time environments a process definition can be executed by an arbitrary workflow product implementing this interface at run-time. Thus, it provides the independence of modeling tools and workflow run-time products.

The second advantage of implementing this interface is that it provides the potential to export a process definition to several different workflow products that could cooperate to provide a distributed run-time enactment service.

The specification for this interface [10] defines a common meta-model for describing the process definition and also a textual grammar for the interchange of process definitions (Workflow Process Definition Language – WPDL) and APIs for the manipulation of process definition data. WPDL is a first implementation of Interface 1. And current work considers XPDL that is defined in specification [14]. XPDL will be considered in section 2.7.

Workflow Client Applications & Workflow Client Application Interface (Interface 2)

A workflow client application is an application, which interacts with a workflow engine, requesting facilities and services from the engine. Client applications may interact with a workflow engine for a variety of reasons. Client applications may perform some common functions:

- Worklist handling
- Process instance initiation and process state control functions (e.g. suspend, resume, etc.)
- Retrieval and manipulation of process definition data
- Various system administration functions (e.g., suspending the use of certain process definitions [2])

The worklist handler is a software service that enables interactions for human-involved activities. It may be implemented as part of the workflow management product or as a separate application. Furthermore, it supports the adaptation of user applications to the specific requirements of an enterprise. This enables the enterprise to use different workflow management products with the same enterprise-specific end-user applications.

The workflow client application interface, which is also called Interface 2 in the workflow reference model, enables the access from a workflow client application to the workflow engine and worklist in a product independent manner.

The intended APIs for workflow client application use are grouped into various functional areas as follows [11]:

- session establishment;
- workflow definition operations;
- process control functions;
- process status functions;
- worklist/work-item handling functions;
- process supervisory functions;
- data handling functions;
- administration functions;
- application invocation.

Implementation of these APIs supports workflow client applications to operate with different workflow engines.

Invoked Applications & Invoked Applications Interface (Interface 3)

An invoked application is a workflow application that is invoked by the workflow management system to carry out an activity by an application, fully or partly, or to support a workflow participant in processing a work-item [2].

Application invocation is not a workflow specific functionality but it is of great importance for a workflow management system. Many workflow management systems have to deal with limited types of applications such as word processors or spreadsheets. For other types of applications the required operations may be executed using standard interchange mechanisms such as OSI TP protocol.

Some workflow products use so-called “Tool Agents” that can handle the application control and information exchange. These tool agents represent at least one specific invocation technology: for instance, some tool agents support MS Windows DDE commands, others can communicate based on protocols like MS OLE or CORBA [11].

Another possibility of application invocation is the implementation of workflow-enabled applications. Workflow-enabled applications are applications that can interact with the workflow enactment service by using standardized APIs.

The invoked applications interface is defined as a set of APIs to enable the cooperation of workflow enactment service and tool agents or workflow enabled applications. Figure 2.6 illustrates the approach for this interface:

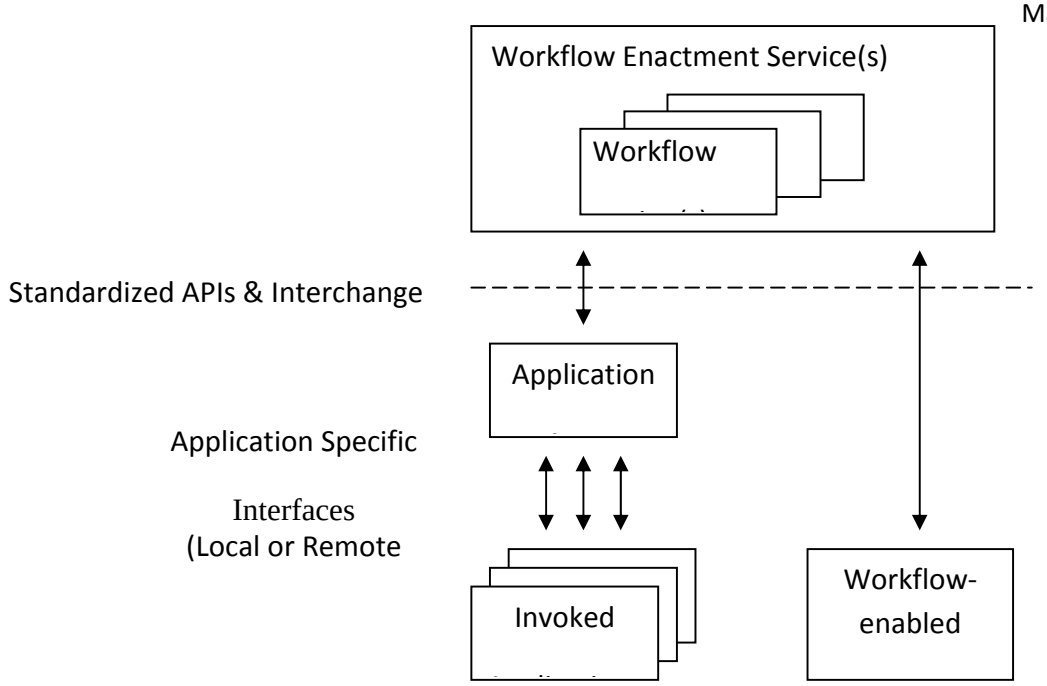

Figure 2.6: Invoked Application Interface

Application invocation is not limited to local applications as depicted in Figure 2.6. The application to be invoked may be located on the same system as the workflow engine or even on a remote, network accessible system.

As defined in the Workflow Management Coalition's specification on WAPI [11], the API operates as "calls" at run-time. These API calls should enable the usage of enterprise-specific single end-user interfaces regardless of the number of workflow management products. WAPI calls may be implemented in several languages.

They may be used by workflow applications, such as worklist handlers or other applications, or workflow engines that require interaction with another workflow management product within the context of API functions. The interoperability of different workflow management products will be discussed next.

Workflow Interoperability & Interoperability Interface (Interface 4)

Workflow interoperability is defined as the ability of two or more workflow engines to communicate and interoperate in order to coordinate and execute workflow process instances across those engines [12].

There is a wide range of workflow management products specialized on different aspects ranging from ad-hoc routing of tasks to regularized production processes in the market. Thus, users of workflow management technology, companies or other organizations planning to introduce this technology, have the chance to choose the product that suits their requirements best.

However, this product specialization can only become an advantage for the user if the workflow management products are able to work together. Therefore, the Workflow Management Coalition defined an interoperability interface supporting simple interoperability scenarios such as the instantiation of a process on a remote workflow engine.

More complex interoperability scenarios, such as the cooperation of different vendor's workflow engines to provide a single workflow enactment service, may be realized in the future.

Different levels of interoperability are defined by workflow management coalition in the interoperability abstract specification [12]. After the definition of interoperability abstract specification a specification defining an XML-based language, which is designed to model the data transfer, is released [13].

Administration and Monitoring Tools & Interface (Interface 5)

Persons responsible for the management of workflow are usually called workflow administrators or system administrators. They use administration and monitoring tools for workflow administration and monitoring purposes.

An administration and monitoring tool may exist as an independent management application interacting with different workflow engines. Besides, they can also be implemented as an integral part of the workflow enactment service with the additional functionality to manage other workflow engines.

The Administration and Monitoring Interface, Interface 5, enables several workflow services to share a range of common administration and monitoring functions. Thus, it includes specific commands within the WAPI set to manipulate

designated administration and monitoring functions. This interface is proposed to allow a complete view on the status of the work flowing through the organization, regardless of which system it is in [6].

The Administration and Monitoring Interface may support the following types of operations (some of which are common to other interface areas) [6]:

- user management operations: establish / delete / suspend / amend privileges of users of workgroups;
- role management operations: define / delete / amend role-participant relationships, set or unset role attributes
- audit management operations: query / print / start new / delete audit trail or event log, etc.
- resource management operations: set / unset / modify process or activity concurrency levels, interrogate resource control data (counts, thresholds, usage parameters, etc.)
- process supervisory functions: change the operational status of a workflow process definition and/or its existent process instances, enable or disable particular versions of a process definition, change the state of all process or activity instances of a specified type, assign attribute(s) to all process or activity instances of a specified type, terminate all process instances
- process status functions: open / close a process or activity instances, query, set optional filter criteria, fetch details of process instances or activity instances, filtered as specified, fetch details of a specific (individual) process or activity instance

At present, most of the workflow management products in the market do not implement all interfaces of the workflow reference model. Usually, they implement

some of these interfaces and only some part of the functionality that is defined in the specification.

2.6. BPM standards

Figure 2.7 shows a scope of different standards for business and technical solutions in BPM (Business Process Modeling) area.

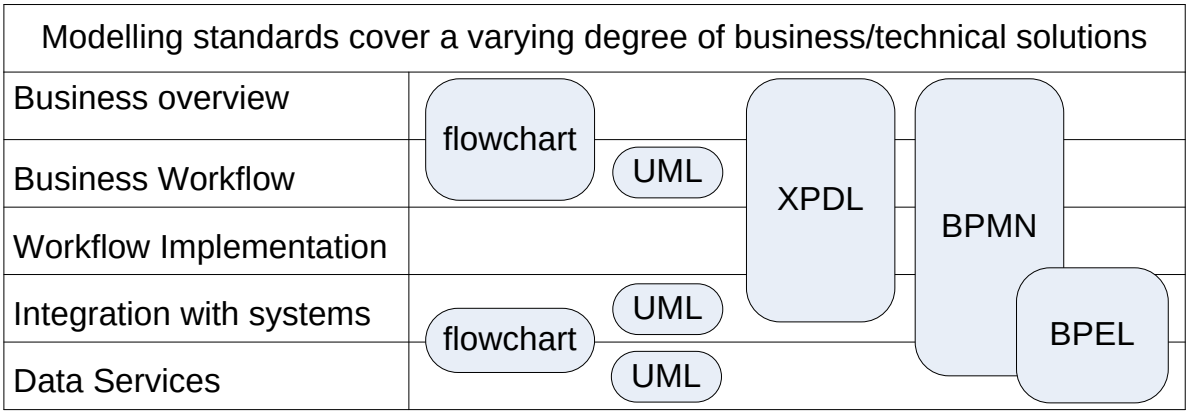

Figure 2.7: Modeling standards and their scope [15]

Flowchart

Flowchart or a control flow diagram is a type of diagram in which suitably annotated geometrical figures are used to represent operations, data, or equipment, and arrows are used to indicate the sequential flow from one to another [16]

UML

UML (Unified Modeling Language) is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology[17]

XPDL

(XML Process Definition Language) An XML-based language from the Workflow Management Coalition (WfMC) for defining business processes. Whereas BPML and other business process languages are geared to Web services, the foundation of XPDL was based around a common set of functions for work distribution found in most workflow products.[18] For more information, see section 2.7

BPMN

Business Process Modeling Notation (BPMN) is a method of illustrating business processes in the form of a diagram similar to a flowchart. BPMN was originally conceived and developed by the Business Process Management Initiative (BPMI). It is currently maintained by the Object Management Group (OMG).[19]

BPEL

BPEL (Business Process Execution Language) for Web services is an XML-based language designed to enable task-sharing for a distributed computing or grid computing environment - even across multiple organizations - using a combination of Web services. Written by developers from BEA Systems, IBM, and Microsoft, BPEL combines and replaces IBM's Web Services Flow Language (WSFL) and Microsoft's XLANG specification. (BPEL is also sometimes identified as BPELWS or BPEL4WS.)[20]

2.7. XPDL

XPDL stands for XML process definition language. The XML Process Definition Language XPDL is a format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products, i.e. between different modeling tools and management suites. XPDL defines an XML schema for specifying the declarative part of workflow / business process.

XPDL is designed to exchange the process definition, both the graphics and the semantics of a workflow business process. XPDL is currently the best file format for exchange of BPMN diagrams; it has been designed specifically to store all aspects of a BPMN diagram. XPDL contains elements to hold graphical information, such as the X and Y position of the nodes, as well as executable aspects which would be used to run a process. This distinguishes XPDL from BPEL which focuses exclusively on the executable aspects of the process. BPEL does not contain elements to represent the graphical aspects of a process diagram [14].

The process definition interface (Interface 1) defines a common interchange format, which supports the transfer of process definitions between separate products. The principles of process definition interchange are illustrated in Figure 2.8.

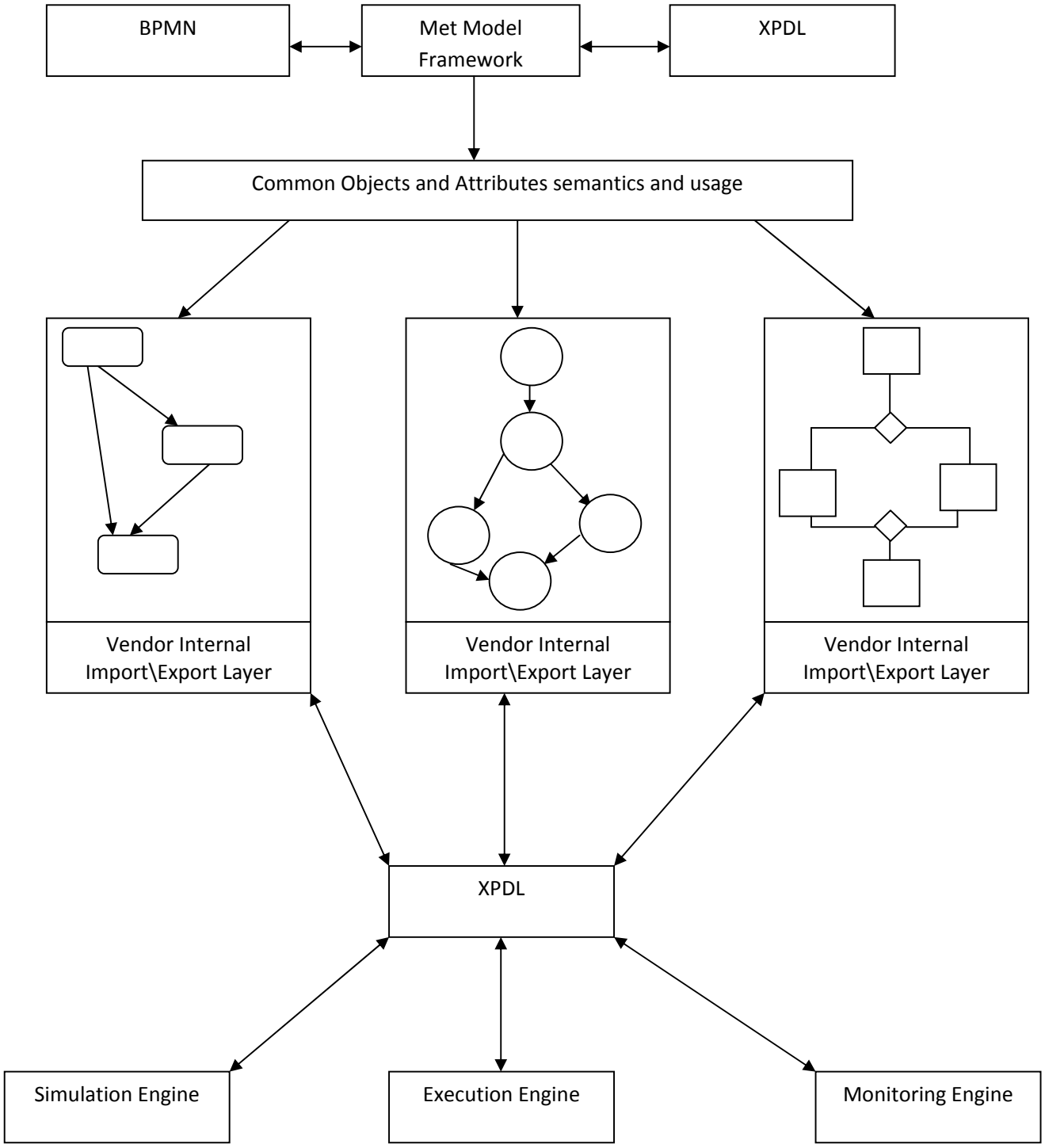

Figure 2.8: The Concept of the Process Definition Interchange [14]

XPDL provides means for serialization of BPMN diagrams, however it's was not always true, for, actually XPDL 1.0 doesn't specify graphical patterns (their size or position) of BPMN diagrams. And the XPDL 2.2 schema effort is focused on only covering the Process Modeling Conformance class of the BPMN 2.0 specification [21]. Figure 2.9 shows concurrent development of these two standards (XPDL and BPMN).

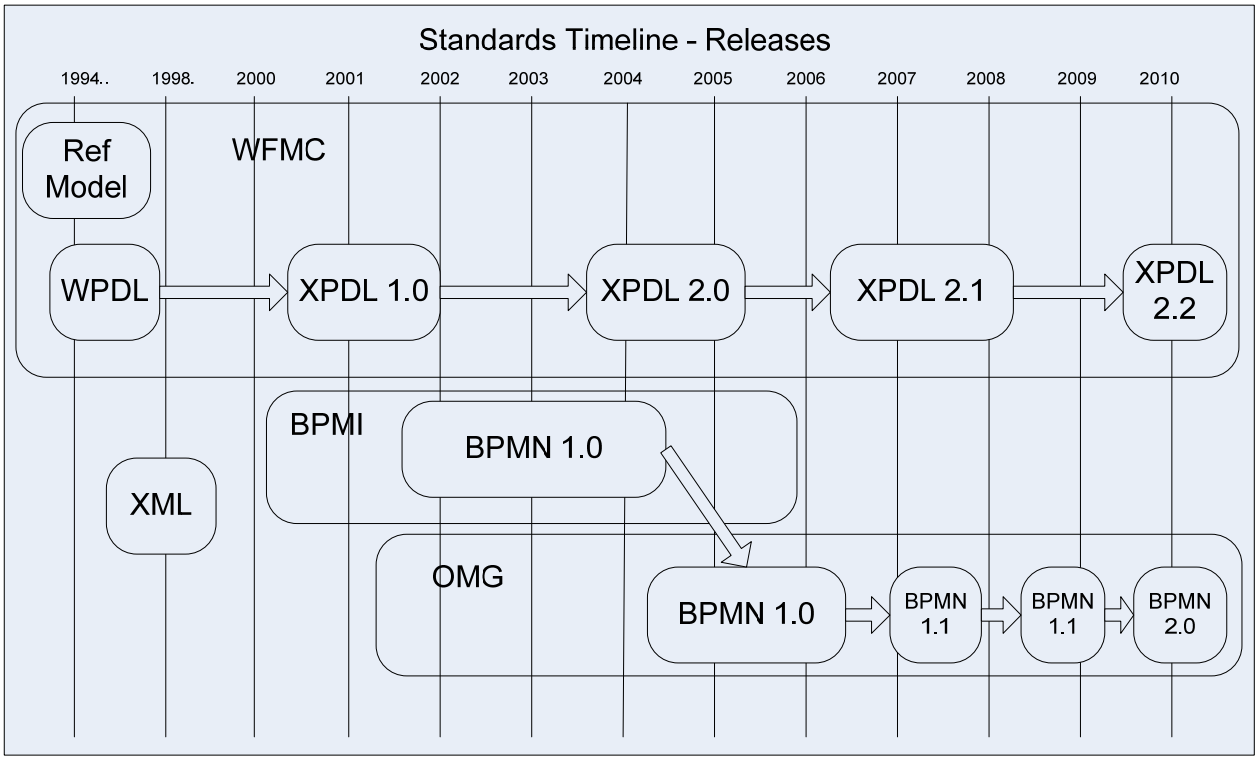

Figure 2.9: Standards Timeline – Releases [21]

XPDL language specified in XML Schema (XSD), which is an XML-based alternative to DTD (for more information on this, please refer to [22]). Any XPDL file that defines a process or processes must be compliant with xsd.

Figure 2.10 shows the basic structure of any XPDL file.

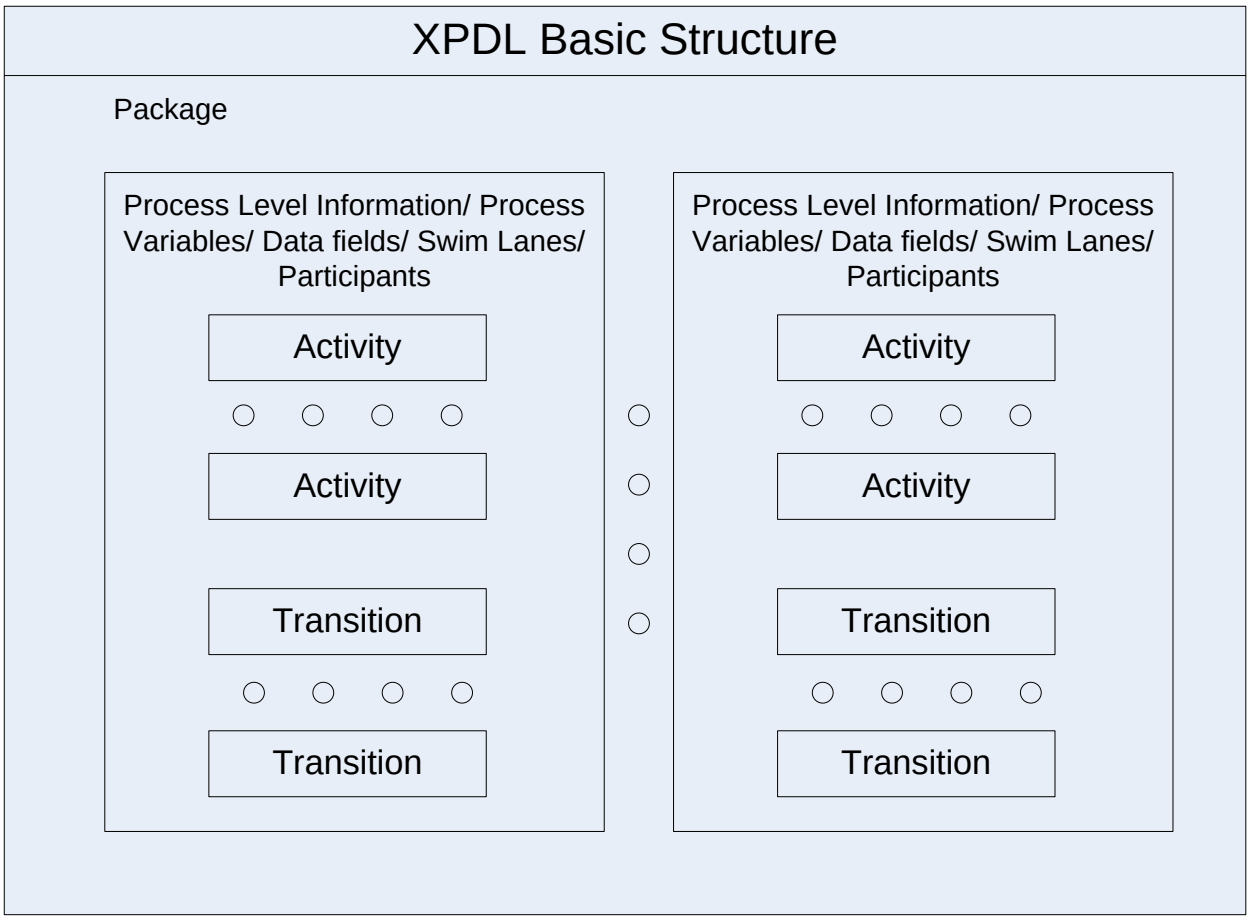

Figure 2.10: XPDL Basic Structure

As it is seen from 2.9 each XPDL file contains a package that consists of one or more processes that in turn has their activities and transitions between them.

Package contains information shared across multiple processes.

Information for a single process includes swim lanes, artifacts, data items, participants.

Activity box on figure 2.10 represents all the nodes of BPMN, including activities, gateways, and events. Transitions represent all the “connections” including sequence flow, message flow, and associations.

It is useful to picture in mind XPDL as a portion that process data related to BPM. Typical use of XPDL might be as it’s seen from the following figure.

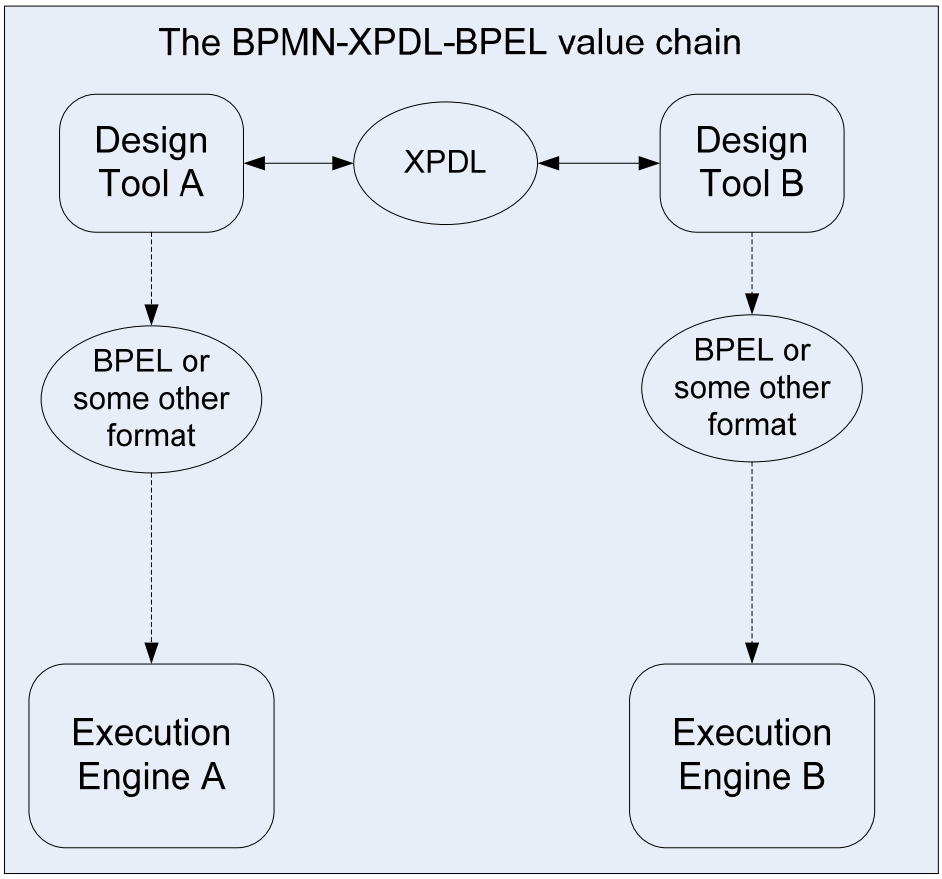

Figure 2.11: The BPMN-XPDL-BPEL value chain [23].

Design tool, which may have graphical user interface and be compliant with BPMN, is used to define a workflow or a part of it, let's us call it the result. This result might be given as input to workflow engine through translation into some other standard like BPEL. It's also possible to take an xpdl-file as an input into execution engine (it's actually the choice of this thesis work).

Here is the example of a workflow defined using “Together workflow editor” [24].

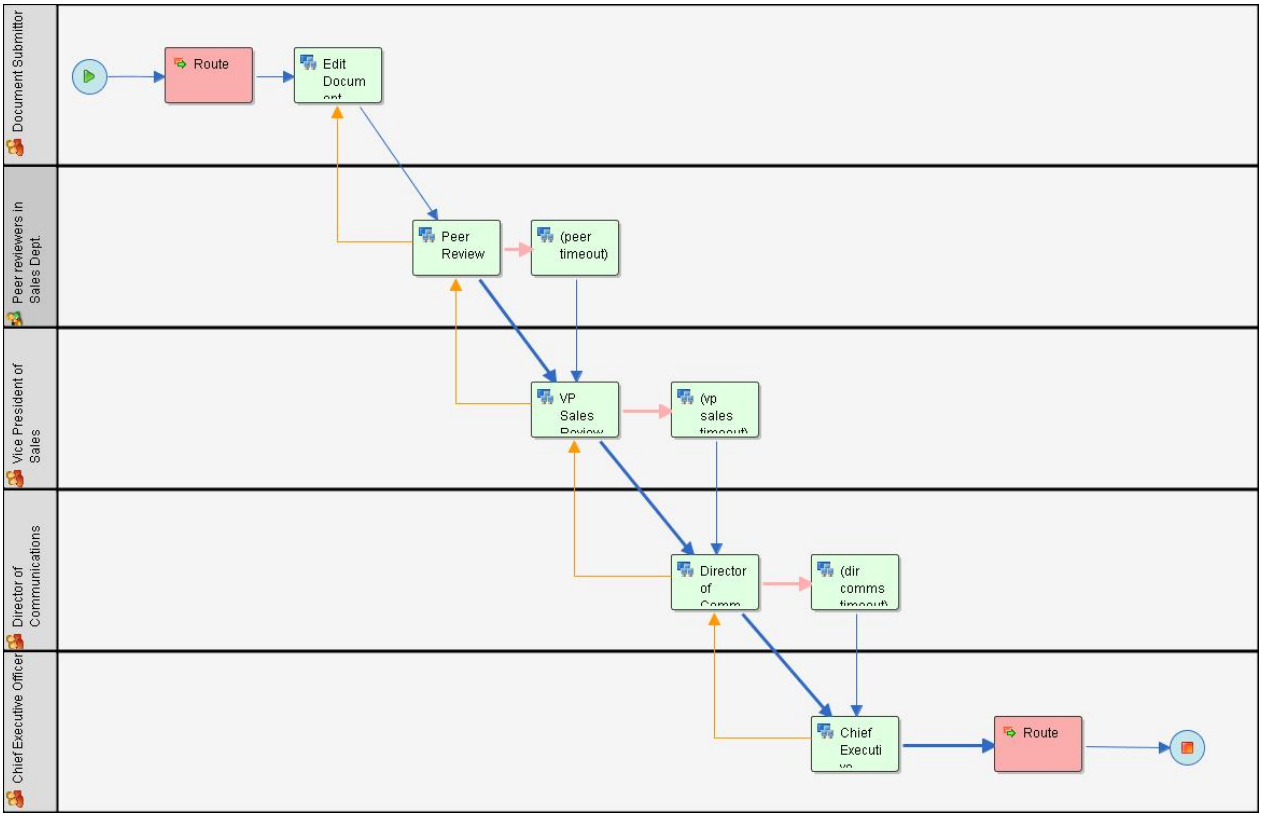

Figure 2.12: Workflow Example

As it can be seen from figure 2.12 typical workflow consists of participants, their tasks to perform (activities) and transition between tasks.

3. Design choices

During system design one needs to choose partitioning (components) of the system and purposes of each partition (functions). While choosing technologies (building blocks of the system) we had to take into account the following parameters:

- cost of the technology;
- knowledge and experience that we had at the start of the work;
- reusability of the technology;
- documentation availability;
- user support(forums, bug trackers, RSS feeds);
- portability of the technology;
- amount of time needed to get acquainted with the technology;

First things to choose:

- standards, specifications, design recommendations;
- programming language and libraries;
- framework, IDE

Next three sections are devoted to the choices made.

Note that if a technology (or library) is chosen and some of the above mentioned criteria are not described for the technology explicitly (like amount of time needed to get aquatinted with the technology), than one should consider such a selection as best implicitly as a tradeoff between the criteria.

3.1. Standards, specifications, design recommendations

We choose to design workflow engine that is XPDL complaint since XPDL is de facto a global standard for workflow and BMP[25].

As XPDL is developed by WFMC it is natural to choose standards and recommendations of WfMC as our guide during design of the system with a starting point in Workflow Reference Model described in [6]

Besides specifications, ready or almost ready solutions in BPM area can be helpful. It's a topic of the next section.

3.2. Known solutions

There are more than 70 different products that claim to support XPDL, most of them are proprietary [27]. Here is the list of known Open Source solutions that support XPDL:

- Bonita[28];
- Together XPDL Workflow Server[29];
- jawFlow[30];
- Joget[31];
- Open Business Engine(OBE)[32];

Among above-mentioned only Open Business Engine is both having its source code easily available to download and structure closely compliant with WfMC specifications on interfaces of Workflow Management System, that is why it's considered as basis for our thesis work.

OBE is a flexible, modular, standards-compliant Open Source Java workflow engine. It is fully J2EE compliant, and supports several J2EE application servers, operating systems and databases.

It is now reasonable to choose Java as our implementation language. However, next section presents some other strong points to support such decision.

3.3. Programming language

There are hundreds of available programming languages, but among them we choose Java, because it gives us the advantages listed below [26]:

- Java's higher level of abstraction allows for increased programmer productivity (although recognizing that the tradeoff is runtime efficiency)
- Java is relatively easier to master than C++

- Java is relatively secure, keeping software components (including the JVM itself) protected from one another
- Java supports dynamic loading of new classes
- Java is highly dynamic, supporting object and thread creation at runtime
- Java is designed to support component integration and reuse
- The Java technologies have been developed with careful consideration, erring on the conservative side using concepts and techniques that have been scrutinized by the community
- The Java programming language and Java platforms support application portability
- The Java technologies support distributed applications
- Java provides well-defined execution semantics

Java EE that's usually built upon Java SE (set of standard libraries) comprises of features that are very important for WfMS development, such as transaction, scalability, security, concurrency handling, components management and distribution.

Java EE compatibility particularly means possibility to integrate Java EE complaint software into Java EE compliant application server. There are basically two leading solutions in this area: Glassfish [33] developed by Sun Microsystems and JBoss[34] by Red Hat. Despite of the biased opinions that prefer Glassfish over JBoss[35], the latter is chosen on the strength of its popularity[36], maturity and licensing: JBoss is distributed under LGPL that has more freedom for a user than dual-license CDDL and the GPL with the classpath exception under which Glassfish is distributed[37].

3.4. Frameworks, IDE

Among appropriate (Java language, Java EE support) IDE's there are two prominent open source solutions: Eclipse [39] and NetBeans[40].

NetBeans

Netbeans is a free IDE backed by Oracle Corporation. It is the main competitor of Eclipse. Netbeans is built on plugin architecture, and it has third-party vendor support. The main advantage of Netbeans over Eclipse is Netbean's GUI designer. It includes syntax highlighting and language support for Java, JSP, XML/XHTML, visual design tools, code generators, ant and CVS support.

Eclipse

Eclipse is a free IDE, built on a plugin architecture. Eclipse is highly extensible and customizable. Third-party vendors have embraced Eclipse and are increasingly providing Eclipse integration. Eclipse is built on its own SWT GUI library. Eclipse excels at refactoring, J2EE support, and plugin support.

The result of several review articles that compare Eclipse and NetBeans are more or less even for both [40] [41][42]. But our choice is Eclipse and it is mainly dictated by its didactic and practical usefulness: constant growths Eclipse popularity is proven by statistics [43].

4. System description

Taking into account above considerations let us outline the design choices made so far:

- WfMS model: WfMC standards including XPDL;
- Programming language and technologies associated with it: Java, Java EE;
- Java EE application server: JBoss;
- Base: Open Business Engine
- IDE: Eclipse with its excellent refactoring capabilities with Java EE support;

4.1. Architecture of WfMS

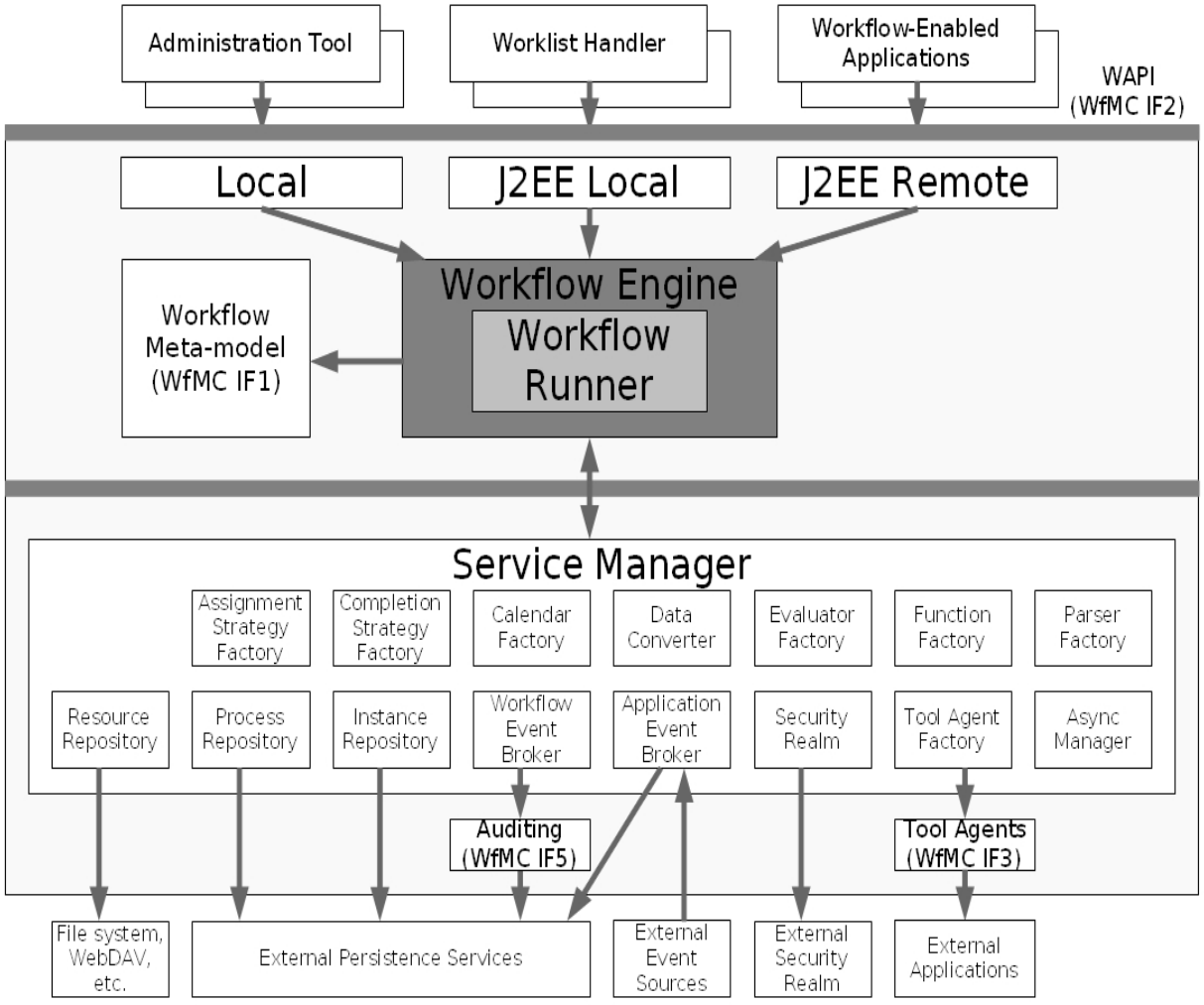

Figure 4.1: Architecture of WfMS

As can be seen from figure 4.1 the architecture of workable WfMS complies with 5 Interfaces defined in [6] and it is based on the architecture of Open Business Engine [32]. We focus primarily on J2EE part, as our goal is to make WfMS engine and WfMS itself scalable, secure, transaction save application, all these and other features are gained from using Java EE 5 technologies and in particular EJB. OBE uses EJB 2.1 that was released in 2003 since that time the technology has been changed dramatically in order to provide better services on one hand and simplify the

code on another. The following figure shows the major differences between EJB 2.1 and EJB 3.0

Differences between EJB 2.1 and EJB 3.0	
EJB 2.1 enterprise beans	**EJB 3.0 enterprise beans**
An EJB 2.1 session bean must implement the SessionBean interface.	An EJB 3.0 session bean does not implement the SessionBean interface.
An EJB 2.1 session bean includes one or more ejbCreate(), ejbPostCreate(), ejbActivate(), ejbPassivate(), ejbRemove(), and setSessionContext(), and the business methods defined in the local/remote interface.	An EJB 3.0 session bean class includes only business methods.
EJB 2.1 includes home interface that extends the EjbHome interface, remote interface that extends the javax.ejb.EJBObject interface, local home interface that extends the javax.ejb.EJBLocalHome interface, and local interface that extends the javax.ejb.EJBLocalObject interface.	EJB 3.0 interfaces are POJI (Plain Old Java Interface) business interfaces and do not require home and component interfaces.
EJB 2.1 must have the deployment descriptor.	EJB 3.0 may or may not contain deployment descriptors. Annotations are added to the language that can serve almost all the purposes that were previously served by the deployment descriptor.
An EJB 2.1 Entity EJB bean class must implement the EntityBean interface and must provide implementation to the ejbCreate() and ejbPostCreate() methods.	EJB 3.0 entities are persistent objects that are not uired to implement the EntityBean interface.
EJB 2.1 entity bean includes the home, component, local home and local component interfaces that extend the EJBHome, EJBObject, EJBLocalHome and EJBObject interfaces respectively.	EJB 3.0 entities are Plain Old Java Objects (POJO) and do not require home and component interfaces.
An EJB 2.1 bean must define resource-ref in ejb-jar.xml to lookup resources.	An EJB 3.0 bean can use either dependency injection or JNDI lookup.
An EJB 2.1 message-driven must implement the javax.ejb.MessgeDrivenBean interface.	An EJB 3.0 message-driven may or may not implement the javax.ejb.MessgeDrivenBean interface.

Figure 4.2: Differences between EJB 2.1 and EJB 3.0

EJB 3.0 is not only more efficient in terms of simplicity and code clarity, but there are evidences [44] [45] that it has better performance (up to 50% faster).

In order to understand where Java EE and EJB take their place in our system let us consider general package dependency of the system.

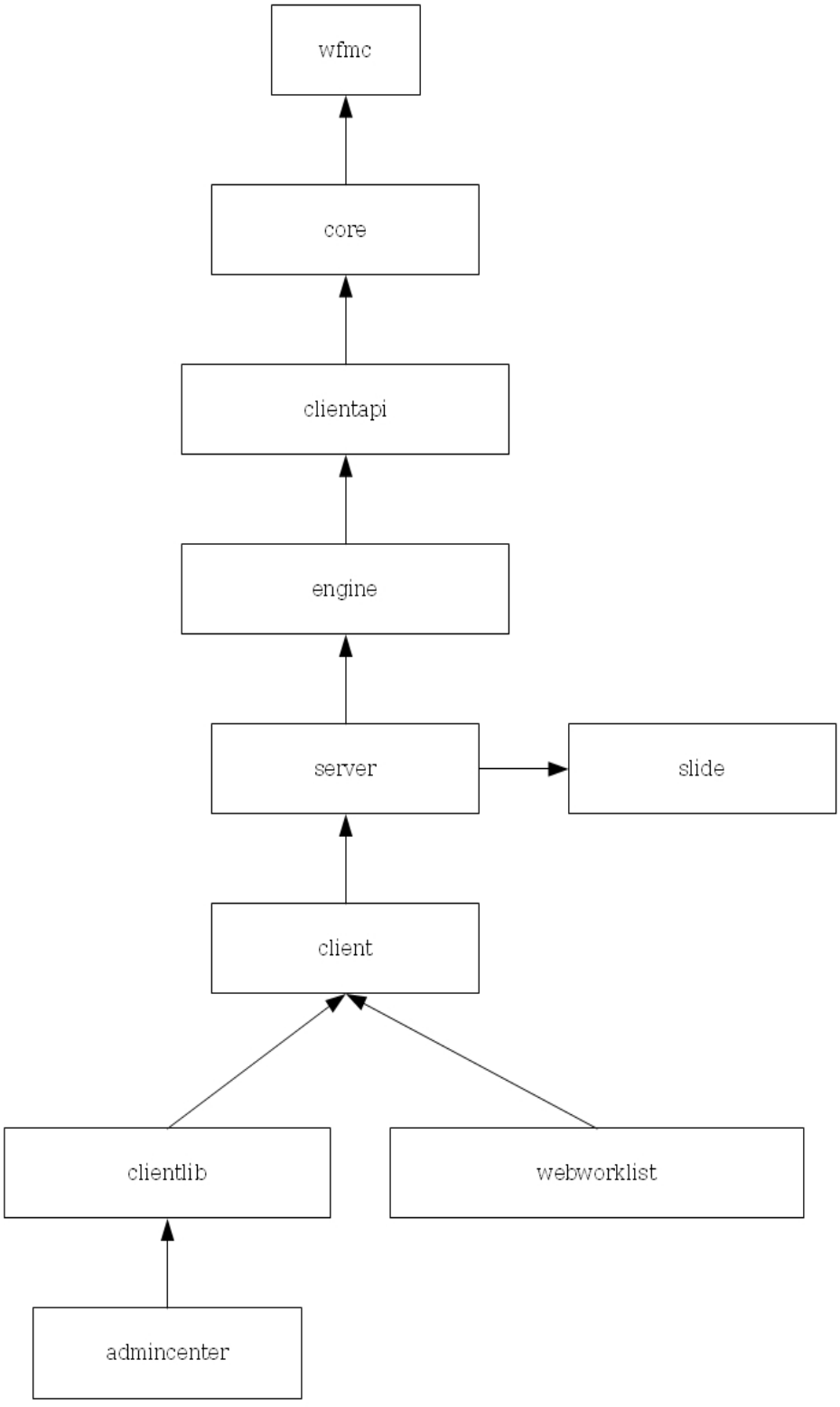

Figure 4.3: Package dependency

WfMC

WfMC package provides a Java binding of WfMC Interface 5 (v.1.1), binding of WfMC Interface 2/3 (WAPI v.1.1) and binding of WfMC Interface 2/3 (WAPI v.2.0).

Core

Core package object model classes and processing utilities: classes related to XPDL workflow activities, classes related to workflow applications as represented in XPDL, to XPDL conditions and expressions, to model XPDL data type declarations, miscellaneous classes and interfaces used by the XPDL object model, classes for modeling XPDL participants, classes for modeling XPDL transitions and other classes that compose XPDL 1.1 specification.

Clientapi

Clientapi provides client API along with process model.

Engine

Actual implementation of the engine

Server

Server package is where engine is plugged into Java EE container. This is actually where EJB 3.0 takes its place.

Slide

Slide provides Web-directory for storing files.

Client, Clientlib, WebWorkList, AdminCenter

Packages for working with workflow engine that overall form WfMS.

While applying Java EE 5 technologies (including EJB 3.0) one should always remember about specifics of Java EE application server (messaging systems, packaging etc.), we use JBoss 5 that is Java EE 5 certified, it brings all the benefits that come from JEE 5(including code simplification and performance boost).

4.2. Demo

This section demonstrates the work of the WfMS engine compliant with XPDL 1.0.

Firstly, JBoss server should be run with the workflow engine deployed and needed process packages, and then the user can access the worklist handler with his credentials using any available browser.

The figures below along with captions to them should be self explanatory. Overall they show, that the workflow engine can operate on the processes state, execute particular work items, invoke third party tools, provide all needed information (participants, activities etc.) for managing different process and their definitions, provide general information about the system.

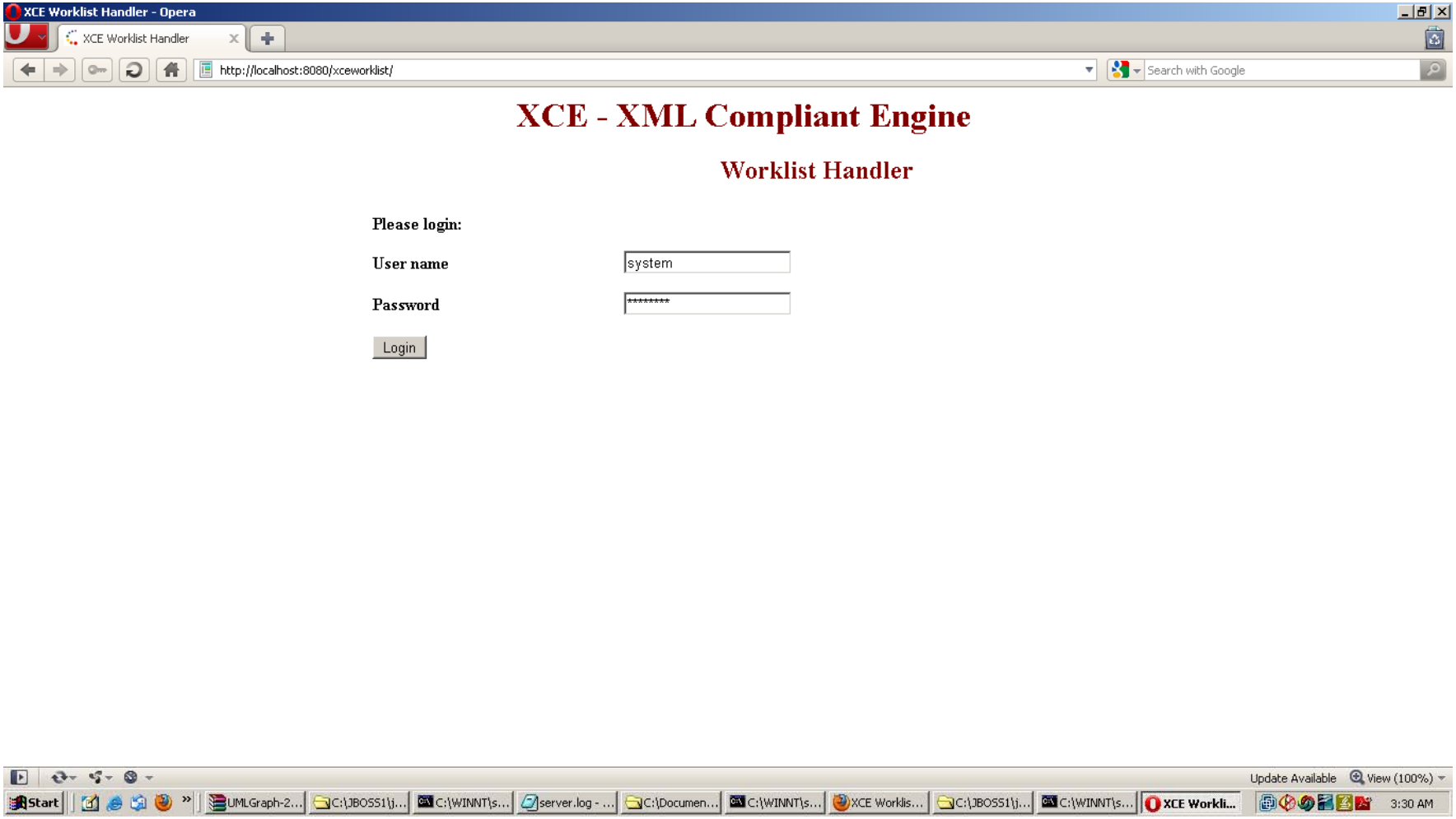

Figure 4.4: Worklist handler. Login window

Worklist Handler basically gives a user control over the process depending on his rights.

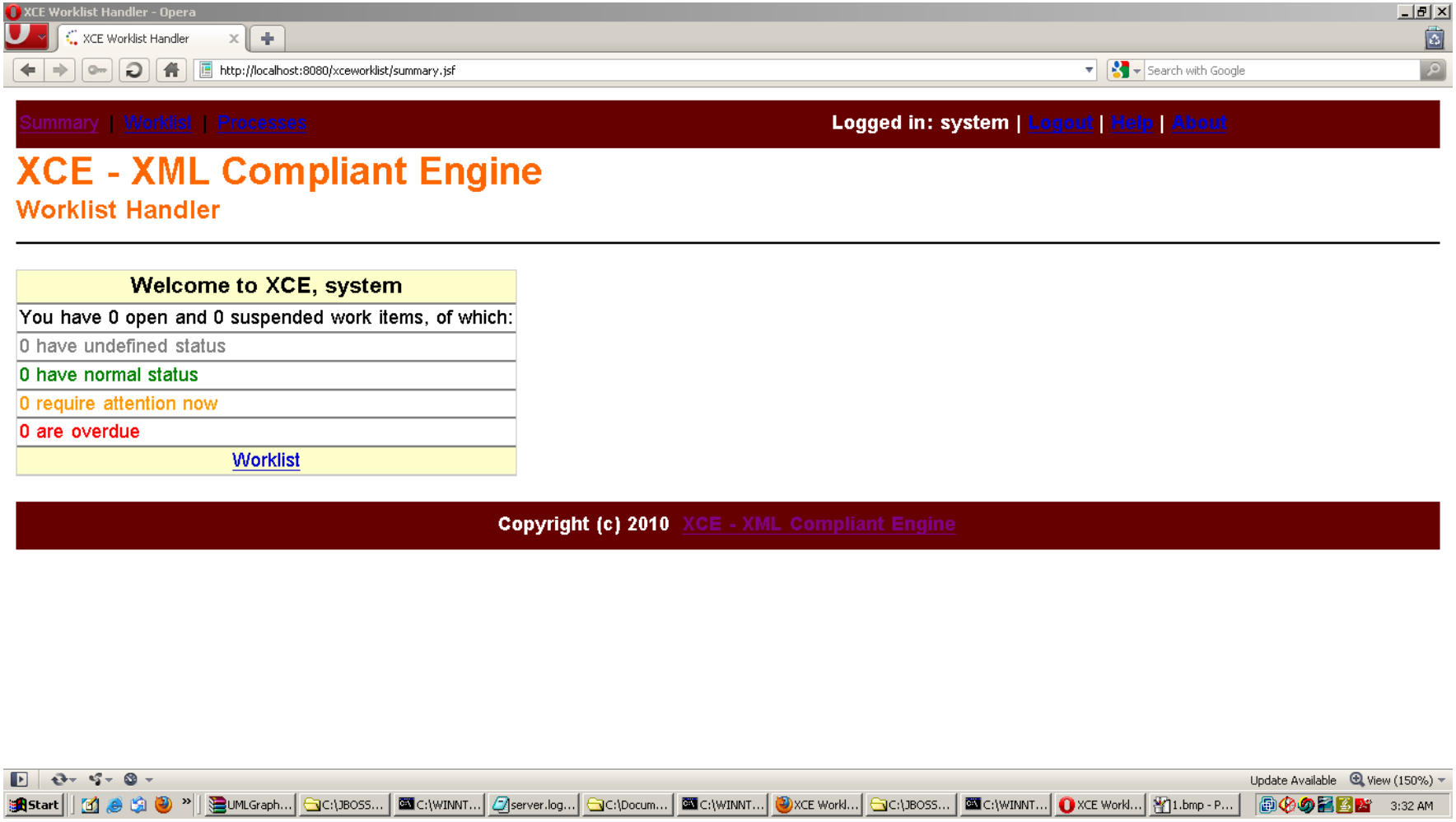

Figure 4.5: Summary.

Summary gives generic information about the processes and amount of work to be done.

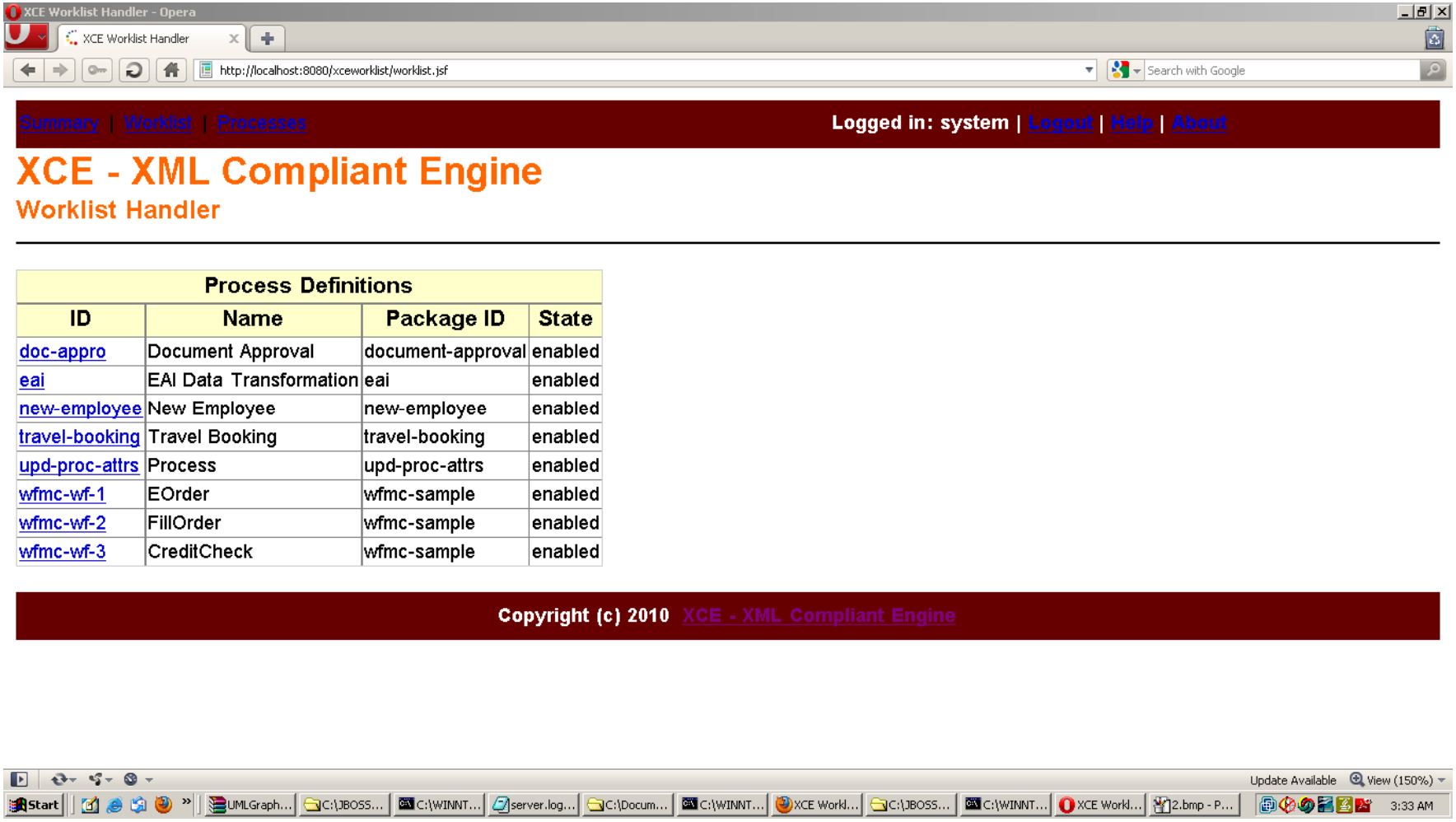

Figure 4.6: Process definitions

Process definitions table lists all defined packages (process definition here, actually, means package) available for the client with the possibility to choose a definition for further considerations.

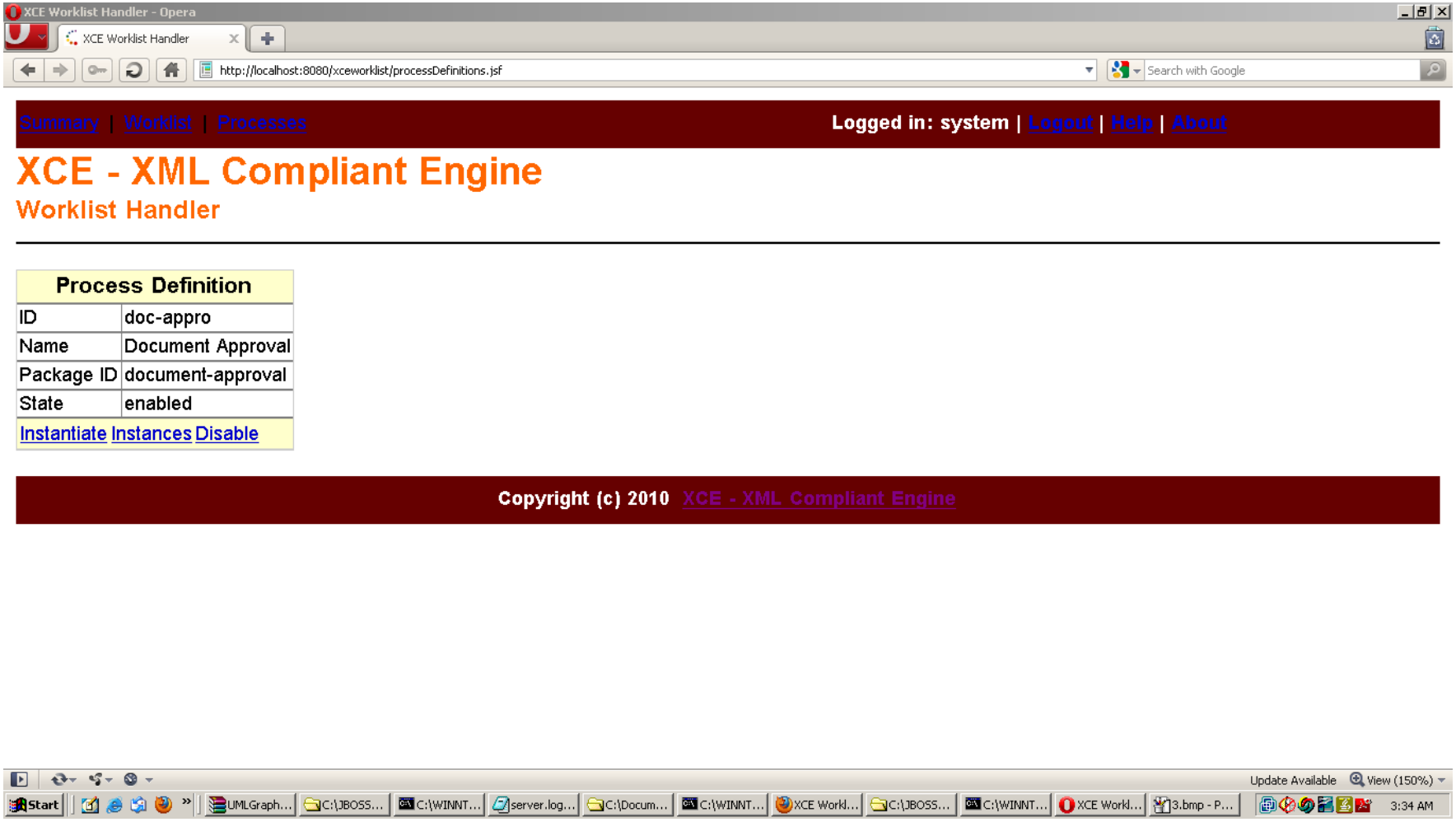

Figure 4.7: Particular process

Process definition table shows basic information about the selected definition and allows changing its state, list all its created instances, disable it or enable.

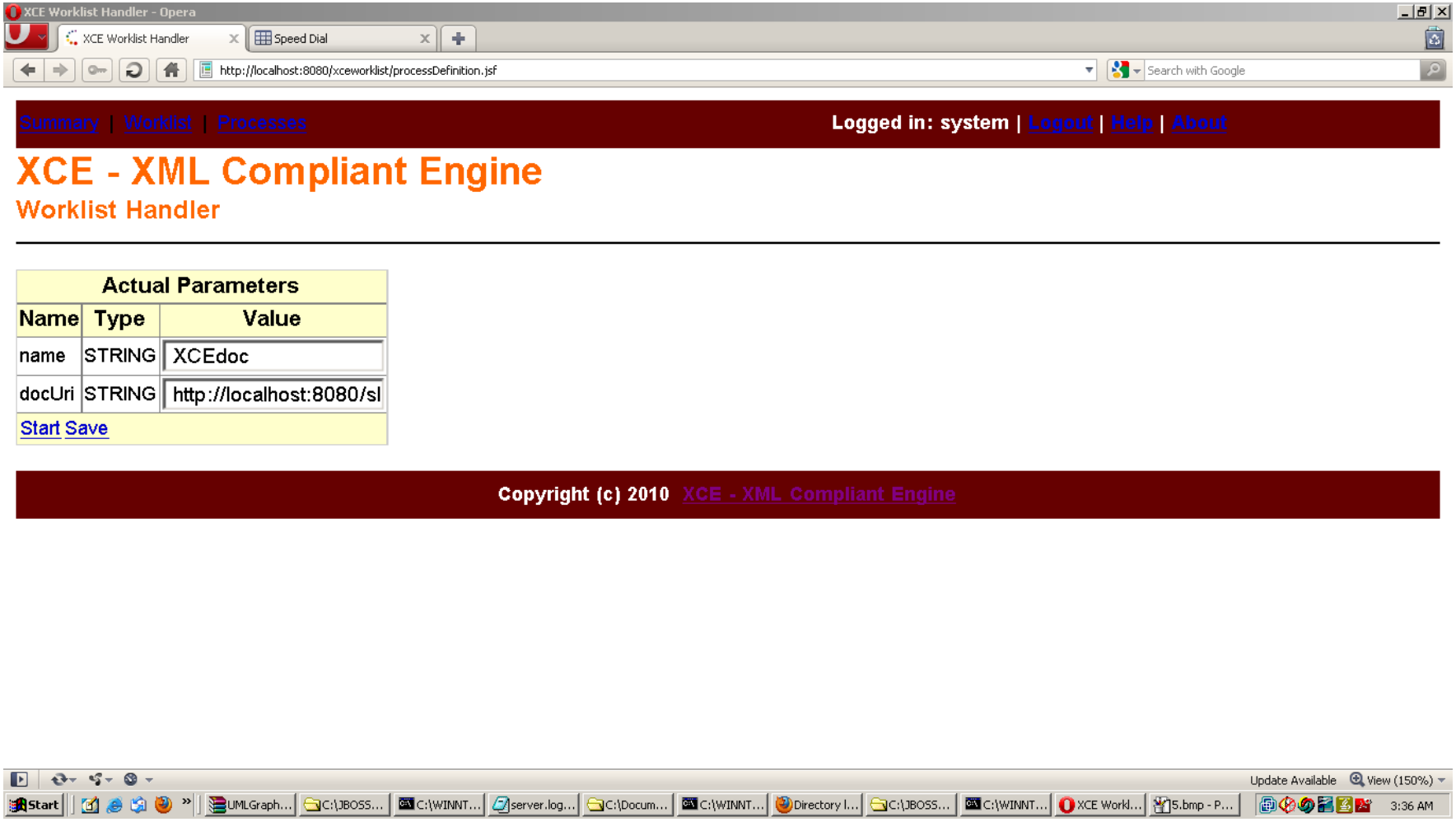

Figure 4.8: Instantiation parameters for a particular process.

When a user clicks instantiate a package he can either provide initial parameters or start a process(set of processes) defined in package.

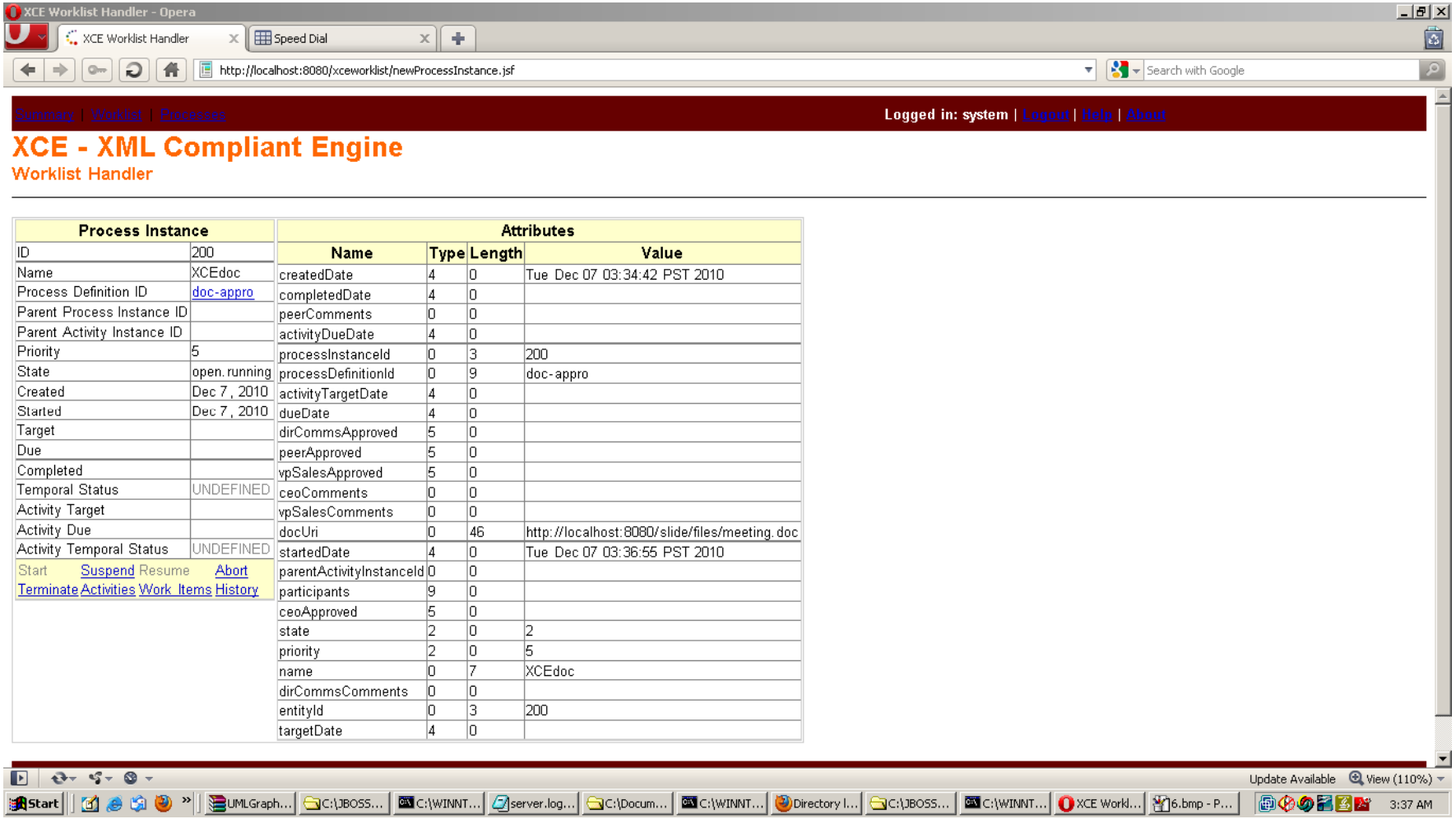

Figure 4.9: Process specific parameters

After the process instantiation the user can see process specific information including id of the process definition and id its instance, also super user is able to manipulate the state of the process (termination, suspension etc) and navigate to workitems related to this process instance.

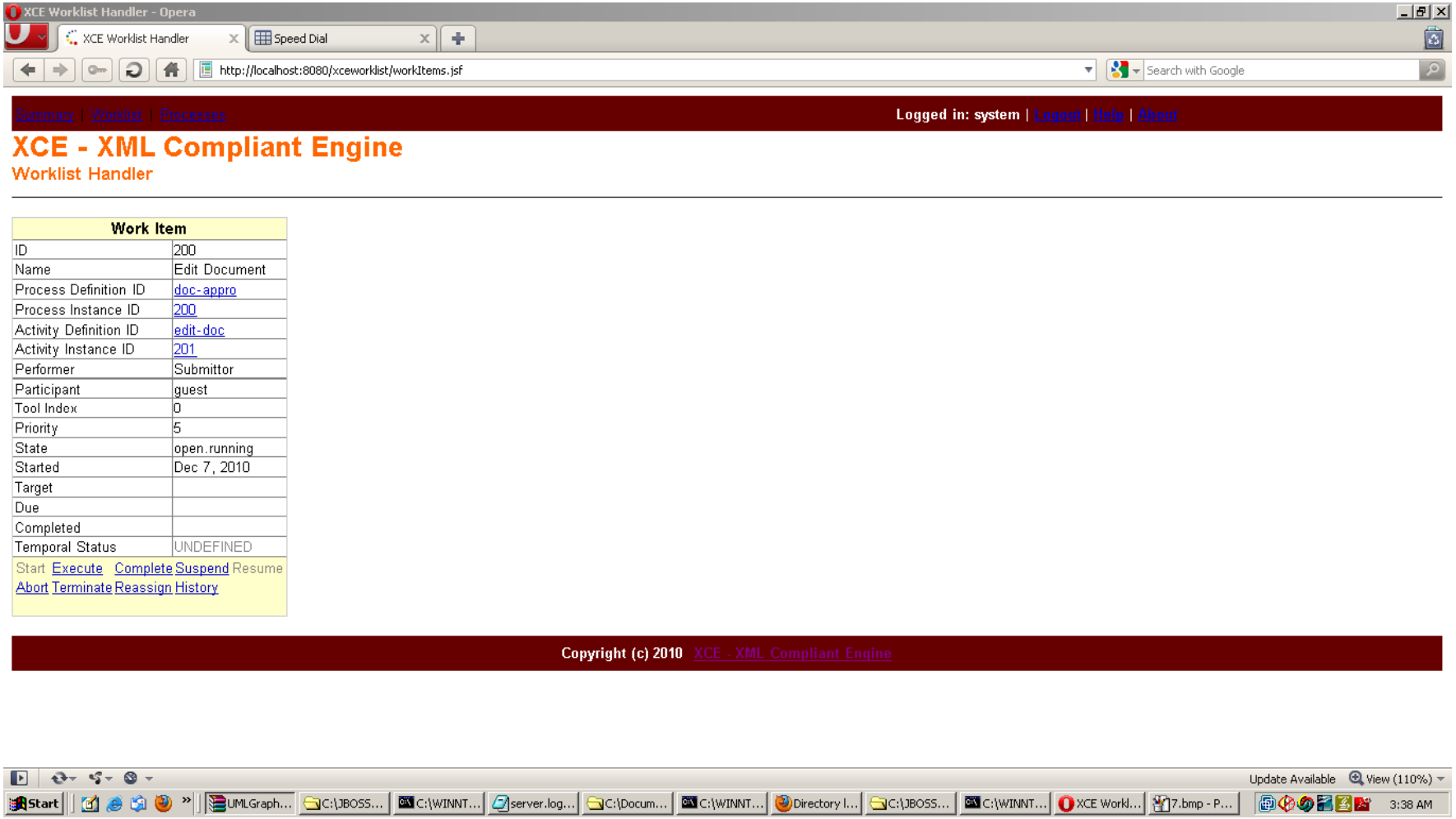

Figure 4.10: Particular work item

Super user can manually change the state of a work item and see history of its execution.

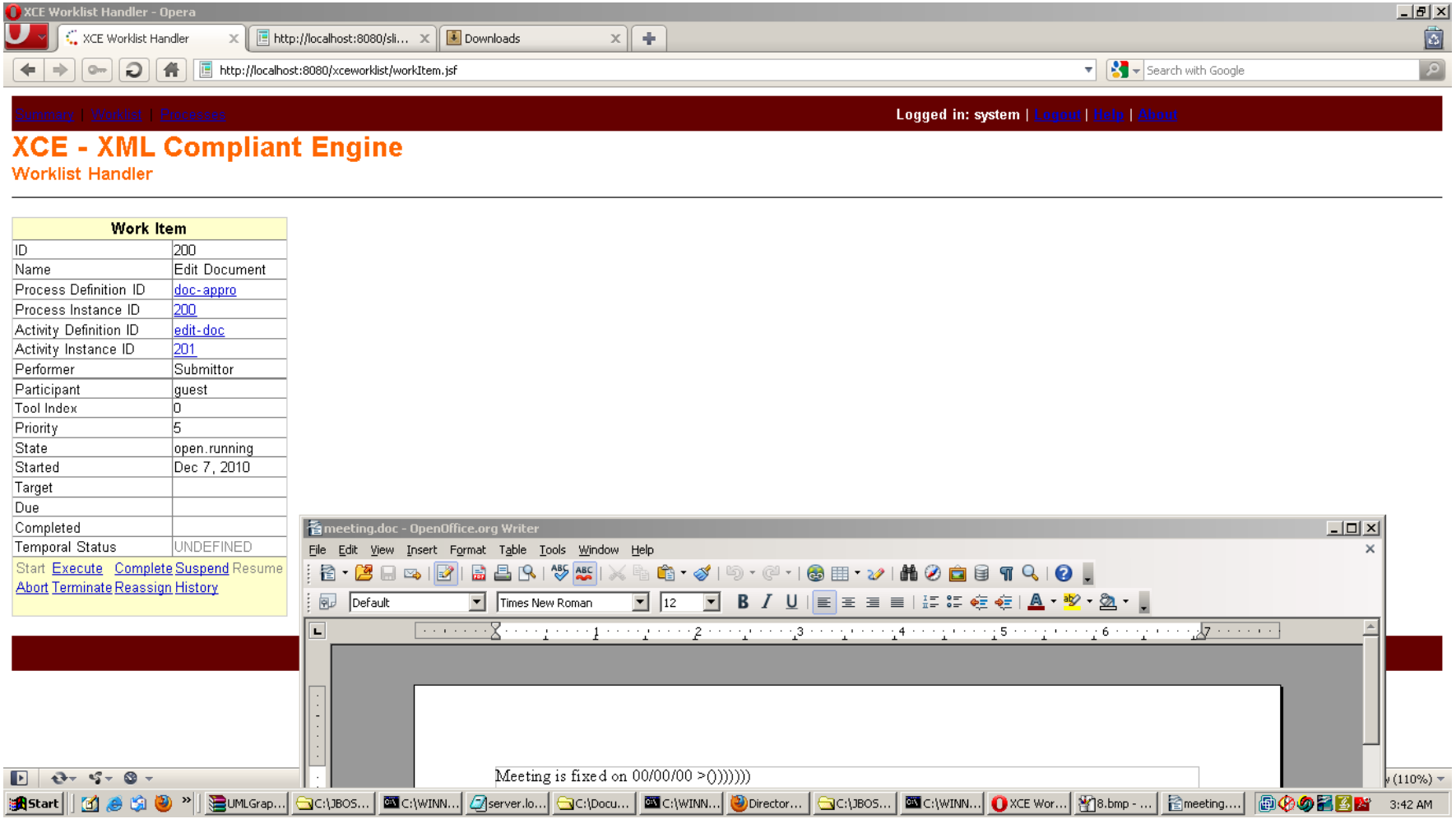

Figure 4.11: Third party tool invocation

When the workitem is executed, engine calls third party tools if needed according to the process definition.

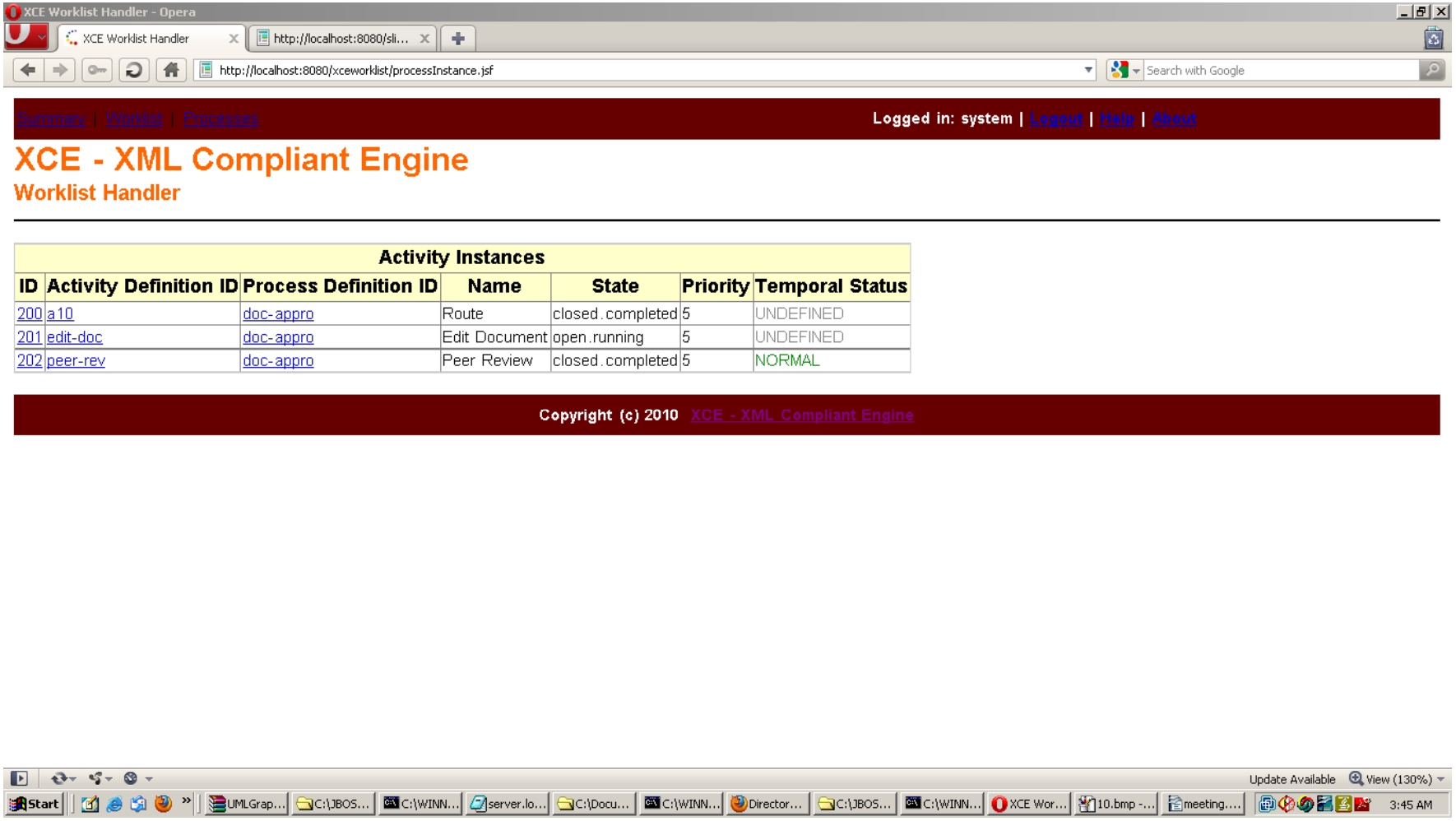

Figure 4.12: Activity instances for a particular process

Using id of a process instance it is also possible to retrieve the list of its activity instances along with related information about them.

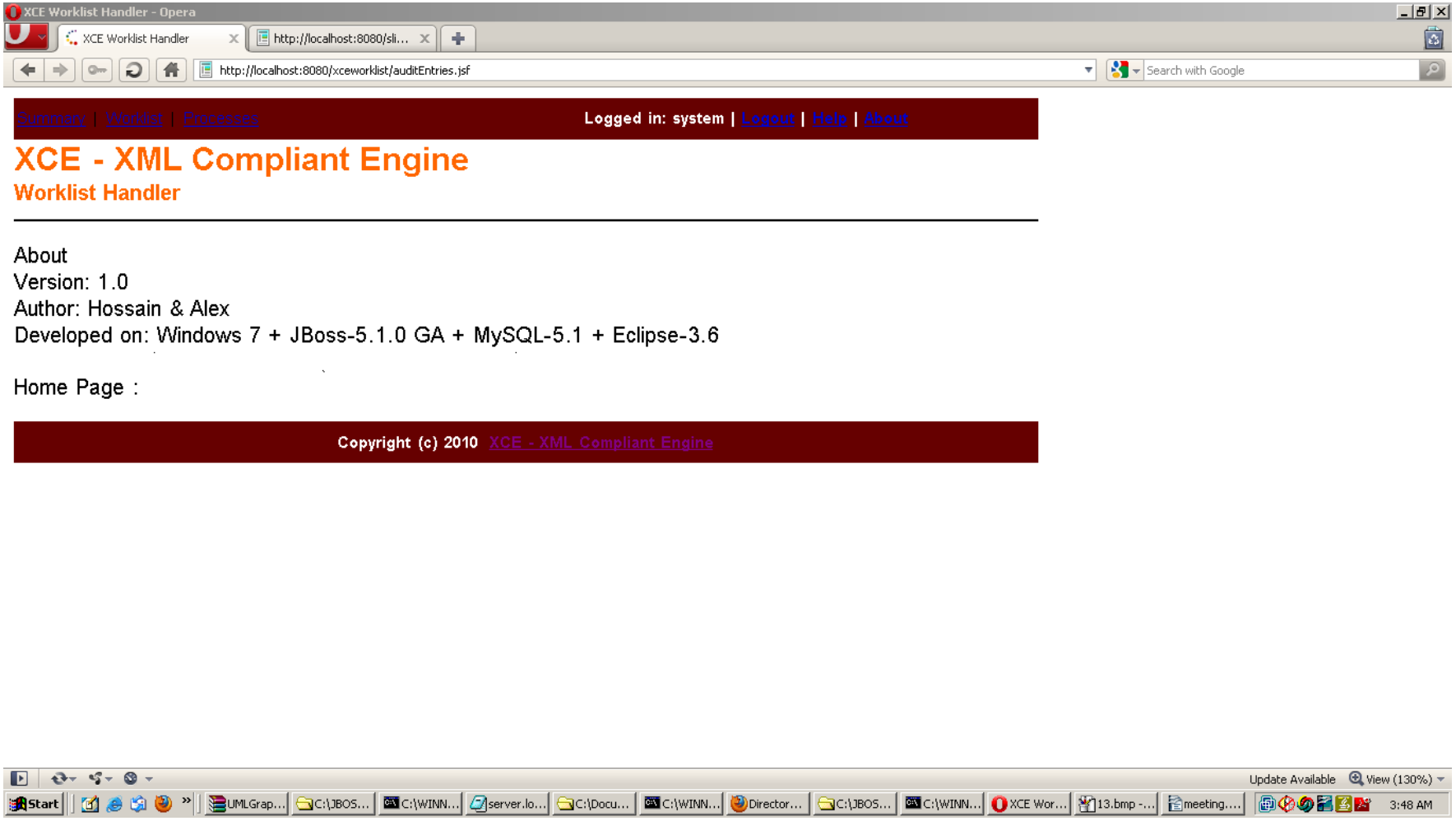

Figure 4.13: Information of the system

Information of the system is more or less equal to traditional “about” window\page\section of a program providing general data about author and system.

5. Results

Currently our Workflow Engine supports JBoss 5 and leverages the benefits of Java EE 5 technology with EJB 3.0, but it supports only XPDL 1.0, however, the current WfMS architecture and the nature of XPDL allows one to extend the specification.

Here is the general description of packages of the engine:

WfMC Application Program Interfaces	
org.wfmc.audit	Provides a Java binding of WfMC Interface 5 (v.1.1).
org.wfmc.wapi	Provides a Java binding of WfMC Interface 2/3 (WAPI v.1.1).
org.wfmc.wapi2	Provides a Java binding of WfMC Interface 2/3 (WAPI v.2.0).
XCE Client Application Program Interface	
org.xce.client.api	Provides the XCE client application program interface.
org.xce.client.api.base	Provides abstract base for J2EE client implementations
org.xce.client.api.model	Provides public details of the XCE process model api.
org.xce.client.api.repository	Provides resource repository meta-data objects.
org.xce.client.api.rmi	Provides communication means with a workflow engine via RMI-RPC
org.xce.client.api.tool	Provides classes for defining new types of application and procedure.
org.xce.client.api.xpdl	Provides validation for XPDL packages
XCE Core XPDL Object Model	
org.xce.xpdl	Provides interfaces for XPDL processing
org.xce.xpdl.model	Provides public details of the XCE process model
org.xce.xpdl.model.activity	Provides classes related to XPDL workflow activities.
org.xce.xpdl.model.application	Provides classes related to workflow applications as represented in XPDL.
org.xce.xpdl.model.condition	Provides classes related to XPDL conditions and expressions.
org.xce.xpdl.model.data	Provides classes to model XPDL data type declarations.
org.xce.xpdl.model.ext	Provides classes that model XCE XPDL extensions.
org.xce.xpdl.model.misc	Provides miscellaneous classes and interfaces used by the XPDL object model.
org.xce.xpdl.model.participant	Provides classes for modeling XPDL participants.
org.xce.xpdl.model.pkg	Provides classes to model an XPDL package.
org.xce.xpdl.model.transition	Provides classes for modeling XPDL transitions.
org.xce.xpdl.model.workflow	Provides classes to represent an XPDL workflow process definition.
org.xce.xpdl.parser	Provides an interface for parsing XPDL documents.
org.xce.xpdl.parser.dom4j	Provides an XPDL parser that uses DOM4J to read an XML document.
org.xce.xpdl.serializer	Provides an interface for serializing XPDL documents.
org.xce.xpdl.serializer.dom4j	Provides an XPDL serializer that uses DOM4J to write an XML document.
Other Packages	
org.xce	Provides core object model classes and processing utilities.
org.xce.spi.model	Provides access to run-time process entities.
org.xce.util	Provides various classes for modeling miscellaneous XPDL elements.

Figure 4.14: Main packages description

6. Conclusions

As we have seen Workflow Reference Model defined by WfMC in 1995 still suitable for implementing Workflow Management Systems, giving enough flexibility to the designer and developer in order to benefit from usage of modern technologies such as Java EE. However model is just a start point and additional standards are required.

We've chosen the most prominent standard for defining process (XPDL) that is not used simply as an exchange format between different workflow definition tools, but a as an input to our Workflow Engine, so there is no need for additional standards on the same level of abstraction (like BPEL).

We based our work on Open Business Engine that conformed to our needs in terms of architecture, extensibility and possibility to use modern technologies. As a result we have a new WfMS that is XPDL 1.0 complaint, supports JBoss 5 and leverages Java EE 5 technologies.

Here is the list of main features of the engine:

- highly configurable and extensible;
- pluggable;
- supports automated, manual and mixed workflow processes;
- can invoke third party tools.

7. Future Research Directions

We base future research directions in compliance with limitations of the system at hand.

First of all our WfMs engine supports only XPDL 1.0, hence one of the possible future work directions is adding XPDL 2.1 or XPDL 2.2 support. Anticipating things let us mention that XPDL 2.1 specification more than 4 times bigger than XPDL 1.0 specification.

Another limitation is lack of Interface 4(Interoperability between workflow engines) support. Implementation of such an interface is a significant improvement of capabilities of the engine (for instance, possibility to embed engine into distributed computing environment).

Also, as it was seen from the demo, current user interface is primarily intended for a technical user, and might not be suitable for a non-technical person. Analysis of workflow user interfaces and further application of the research results can help enhance the available interface.

References

[1] - Architectures for a Temporal Workflow Management System. Giuseppe Pozzi. 2004

[2] - [WfMC99a] Workflow Management Coalition: Terminology & Glossary. Document Number WFMC-TC-1011, February-1999

[3] - [JaBu96] Jablonski S.; Bussler C.: Workflow Management. Modeling Concepts, Architecture and Implementation, International Thomson Computer Press, 1996

[4] - [5] W.M.P. van der Aalst and K.M. van Hee. Workflow Management: Models, Methods, andSystems. MIT press, Cambridge, MA, 2002.

[5] - [BrPe84] Bracchi G.; Pernici B.: The Design Requirements Office Systems. ACM Transactions on Office Information Systems, Vol. 2 (1984), No. 2, 151-170

[6] - [WfMC95] Workflow Management Coalition: The Workflow Reference Model. Document Number TC00-1003, January-1995

[7] - [EGR91] Ellis C.A.; Gibbs S.J.; Rein G.L.: Groupware Some Issues and Experiences. Communications of the ACM, Vol. 43 (1991), No. 1

[8] - [Boc93] Bock, G: Workflow as Groupware. A Case for Group Language? Proc. GroupWare '93, San Jose, 1993

[9] - [McDa98] McCarthy D.R.; Dayal U.: The Architecture of Active Database Management System. Proc. ACM SIGMOD, Portland, Oregon, 1989

[10] - [WfMC99b] Workflow Management Coalition: Interface 1 Process Definition Interchange. Document Number WFMC-TC-1016-P, October-1999

[11] - [WfMC98] Workflow Management Coalition: Workflow Management Application Programming Interface (Interface 2&3) Specification. Document Number WFMC-TC-1009, Version 2.0, July-98

[12] - [WfMC96] Workflow Management Coalition: Workflow Standard – Interoperability Abstract Specification. Document Number WFMC-TC-1012, 26-October-1995

[13] - [WfMC00] Workflow Management Coalition Workflow Standard – Interoperability Wf-XML Binding. Document Number WFMC-TC-1023, 1-May-2000

[14] - Workflow Management Coalition Workflow Standard - Process Definition Interface -- XML Process Definition Language. Document Number WFMC-TC-1025, Version 2.1a, October 10, 2008

[15] - http://www.youtube.com/watch?v=77G9NbqJxpw - XPDL BPMN (table) 2008

[16] - ISO 5807:1985 Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts, 3.3

[17] - http://searchsoftwarequality.techtarget.com/definition/Unified-Modeling-Language

[18] - http://www.pcmag.com/encyclopedia_term/0,2542,t=XPDL&i=55083,00.asp

[19] - http://searchcio.techtarget.com/definition/Business-Process-Modeling-Notation

[20] - http://searchsoa.techtarget.com/definition/BPEL

[21] - Robert Shapiro. Update on BPMN Release 2.0. February 3, 2010

[22] - http://www.w3schools.com/schema/schema_intro.asp

[23] - Understanding the BPMN-XPDL-BPEL Value Chain. Nathaniel Palmer. 2006

[24] - http://www.together.at/index

[25] - XPDL – Bringing Business and Software Together – a Case Study. Orlin Genchev, John Galletly - 2010

[26] - M. Schoeberl, JOP: A Java Optimized Processor for Embedded Real-Time Systems, PhD thesis, Vienna University of Technology, 2005

[27] - http://www.wfmc.org/xpdl-implementations.html

[28] - http://www.bonitasoft.com/

[29] - http://www.together.at/prod/workflow/tws

[30] - http://jawflow.sourceforge.net/

[31] - http://www.joget.org/download

[32] - http://obe.sourceforge.net/

[33] - http://glassfish.java.net/

[34] - http://jboss.org/jbossas

[35] - Comparing Glassfish™ and JBoss which application server is right for you? White Paper. October 2009

[36] - http://java.blogs.webucator.com/2010/06/27/java-application-servers-which-one-fits-your-project-best/

[37] - The Official Online Journal of Chief Executive Prankster, http://blogs.sun.com/chandan/entry/copyrights_licenses_and_cddl_illustrated.

[38] - http://eclipse.org/

[39] - http://netbeans.org/

[40] - NetBeans vs. Eclipse. Simon Martinelli. 2010 http://simonmartinelli.blogspot.com/2010/03/netbeans-vs-eclipse-again.html

[41] - http://faq.programmerworld.net/programming/best-java-ide-review.html

[42] - NetBeans vs. Eclipse, this time with attitude. Ed Burnette. 2007

[43] - http://www.indeed.com/jobtrends?q=visual+studio,netbeans,+eclipse+or+%22flex+studio%22+or+aptana,+delphi+and+database,zend&l=

[44] - http://jaoo.dk/jaoo2006/articles/ejb3perf.jsp

[45] - http://jroller.com/raghukodali/entry/comparing_ejb_3_0_and