

An evaluation method for Business Process Management products

Stefan R. Koster

Business Information Technology Master of Science graduation thesis from the University of Twente, Enschede, the Netherlands



An evaluation method for Business Process Management products

May 2009

Graduation thesis of:
Stefan R. Koster
Student number 0050156
Business Information Technology
University of Twente
s.r.koster@alumnus.utwente.nl

On behalf of:
Capgemini Nederland B.V.
Papendorpseweg 100
3528 BJ Utrecht

Under supervision of:
Dr. L. Ferreira Pires (University of Twente, faculty of EEMCS)
Dr. M.E. Iacob (University of Twente, faculty of MB)
Ir. L. v.d. Meulen (Capgemini Nederland B.V., practice P66)

Abstract

Business Process Management (BPM) is a management discipline aimed at describing and managing the business processes in an organization. The goal of Business Process Management is to achieve the organization's objectives by aligning the business processes with these objectives and to continually improve these processes.

Even though there are many vendors that supply BPM products, there is no scientific method available today to evaluate these products and to compare them.

This thesis proposes an evaluation method for Business Process Management products. This method was developed by creating a framework on BPM by analyzing the state of the literature regarding BPM. This framework contains literature and criteria that have been extracted from this literature.

The evaluation method itself consists of a selection of the framework's criteria, a case, information on how to use the criteria when evaluating the BPM products and a rating method which allows quantification of the evaluation.

The evaluation method was tested by applying it to three different BPM suites: Oracle BPM Suite, Cordys BPMS and IBM WebSphere BPM. Next to valuable feedback for our evaluation method, these evaluations formed the basis of a comparison between the three products.

The evaluations of the three BPM suites show that our evaluation method is a useful method for comparing BPM products, since it is capable of showing a significant distinction between different BPM suites. We believe that our evaluation method is best used when comparing the complete BPM related product stack of different vendors. Furthermore we believe that the evaluations should be done in a proof of concept where the vendors can walk through our evaluation method, while independent people observe and rate the products based on our rating method that is included in the evaluation method. This will result in an objective evaluation in which all the relevant points are evaluated.

Acknowledgements

First of all, thank you for your interest in this research. This thesis describes the results of my graduation research, carried out from September 2008 to May 2009 at Capgemini Netherlands. With this thesis I complete my Master Business Information Technology at the University of Twente.

I have a lot of people to thank for helping me completing this thesis. First and foremost I would like to express my gratitude to my parents who made it possible for me to attend University of Twente and always stood behind me and my decisions.

Also, I would like to thank all my supervisors, Arjan, Leo, Luis and Maria, who all provided me with guidance and feedback during my research.

My sincere appreciation goes to Capgemini for providing a very pleasant work environment and all the great colleagues I have had the opportunity to meet and who were more than willing to help me, in particular Chris, Frank, Günther, Lee, Patrick, René, Richard, Rick, Roeland, Soumya, Tobi, Tom, Wil and all the fellow students and Young Professionals I have met there.

I hope you will appreciate the contents of this thesis. For questions or feedback on this research project, always feel free to contact me.

Kind regards,

Stefan Koster
May 2009

Table of Contents

<u>1</u>	<u>Introduction</u>	<u>1</u>
1.1	Motivation	1
1.2	Research Objectives	1
1.3	Research Approach	2
1.4	Report Structure	3
<u>2</u>	<u>Business Process Management</u>	<u>4</u>
2.1	Definition on Business Process Management	4
2.2	Explicit Process Management	5
2.3	Importance of Business Process Management	8
<u>3</u>	<u>BPM Framework</u>	<u>9</u>
3.1	Strategy Development	9
3.2	Discovery	10
3.3	Modelling	13
3.4	Design	18
3.5	Deployment	21
3.6	Operation	22
3.7	Analysis	30
<u>4</u>	<u>Definition of the Method</u>	<u>32</u>
4.1	Criteria Selection	32
4.2	Case Description	32
4.3	Using the Selected Criteria	33
4.4	Rating the Products	35
4.5	Selected Products for Testing the Method	38
<u>5</u>	<u>Oracle BPM Suite</u>	<u>40</u>
5.1	Suite Description	40
5.2	Business Process Modelling	41
5.3	Key-Performance Indicators	44
5.4	Executable Business Process Models	45
5.5	Programming Languages	45
5.6	Designing User Interfaces	46
5.7	User Management	46
5.8	Informing the User	47
5.9	Technical Monitoring and Control	47
5.10	Business Monitoring	47
5.11	Business Control	48
5.12	Evaluation Summary	49
<u>6</u>	<u>Cordys BPMS</u>	<u>52</u>
6.1	Suite Description	52
6.2	Business Process Modelling	53

6.3	Key-Performance Indicators	55
6.4	Executable Business Process Models	56
6.5	Programming Languages	56
6.6	Designing User Interfaces	56
6.7	User Management	56
6.8	Informing the User	57
6.9	Technical Monitoring and Control	58
6.10	Business Monitoring	59
6.11	Business Control	59
6.12	Evaluation Summary	60
<u>7</u>	<u>IBM WebSphere BPM</u>	<u>63</u>
7.1	Suite Description	63
7.2	Business Process Modelling	64
7.3	Key-Performance Indicators	66
7.4	Executable Business Process Models	68
7.5	Programming Languages	70
7.6	Designing User Interfaces	70
7.7	User Management	71
7.8	Informing the User	73
7.9	Technical Monitoring and Control	73
7.10	Business Monitoring	74
7.11	Business Control	75
7.12	Evaluation Summary	76
<u>8</u>	<u>Products Comparison</u>	<u>80</u>
<u>9</u>	<u>Conclusions</u>	<u>83</u>
9.1	Contributions	83
9.2	Suggested Improvements to our Evaluation Method	83
9.3	General conclusions	84
9.4	Further Research	84
	List of Definitions	86
	List of References	87
	List of Figures	90
	List of Tables	92
	Appendix A: Criteria used by our method	93
	Appendix B: Oracle BPM Suite rating	97
	Appendix C: Cordys BPMS rating	99
	Appendix D: IBM WebSphere BPM rating	101

1 Introduction

This chapter presents the motivation, the objectives, and the structure of this thesis. It identifies the relevance of an evaluation method for Business Process Management products.

This chapter is further structured as follows: Section 1.1 presents the motivation of this work, Section 1.2 states the objectives of this thesis, Section 1.3 presents the approach adopted in the development of this thesis and Section 1.4 outlines the structure of this thesis by presenting an overview of the chapters.

1.1 Motivation

Every organization has business processes. They range from advertising and customer acquisition to manufacturing and shipping. How well the business processes are organized determines the success of an organization (Elzinga, Horak et al. 1995; Smith and Fingar 2003). For example, a suitable advertising process provides the organization with enough potential customers and a suitable manufacturing process makes sure that the product is of high quality and at a reasonable price which will increase customer satisfaction.

Business processes can be seen as a set of linked and related activities (organizational tasks) which are designed to take an input and to transform it into a specific output (Davenport 1993; Hammer and Champy 1993; Johansson, McHugh et al. 1993; Kalle, Lars et al. 1998). For example, take the business process of an insurance agency in which a customer applies for insurance. This business process involves activities related to checking the credibility of the customer, getting his information, calculating his monthly fee, and many more. All these activities are linked to each other, so that they occur in the right time and order. The input to this business process is the customer's information and the output is a signed insurance contract.

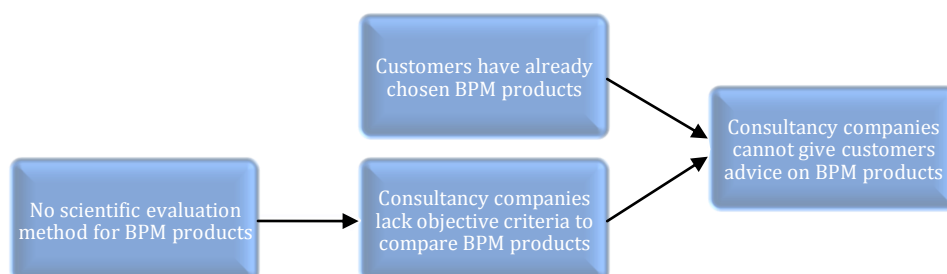
Business Process Management (BPM) is a management discipline aimed at describing and managing the business processes in an organization. The goal of Business Process Management is to achieve the organization's objectives by aligning the business processes with these objectives and to continually improve these processes.

Many vendors have released products that support BPM, among which are Oracle's BPEL Process Manager, BEA's Aqualogic BPM and Tibco iProcess Suite (BEA 2008; Oracle 2008; Tibco 2008). These BPM products either support a few steps of BPM (BPM tools) or the whole lifecycle, in which case they are referred to as BPM suites (Gillot 2008).

At the moment, most consultancy companies are approached by customers who have already chosen to start a BPM project and have selected a vendor for the supporting BPM products. These customers ask the consultancy companies to help them implement the BPM project. However, the consultancy companies want to be a step ahead of their customers and want to give their customers advice on selecting the most suitable vendor.

Since consultancy companies do not have objective criteria on which they can base their advice for a certain BPM product, they are reluctant to give their customers this advice. Scientific literature cannot help these consultancy companies, because at the moment no scientific method is available to evaluate different BPM products. These problems are depicted in Figure 1.

Figure 1 – Problem bundle



1.2 Research Objectives

The goal of this research is to develop an evaluation method for Business Process Management products by building a framework on Business Process Management based on relevant research literature from this field.

From this literature, criteria are derived that are ordered according to the phases of the BPM lifecycle. These criteria are used to evaluate a BPM product in order to assess which phases of the cycle are supported by this product and how suitable that support is.

The proposed evaluation method for BPM products is the most valuable addition of this research to current literature. As described in the problem definition, there is currently no scientific method to evaluate different BPM products. This research fills this gap by defining a method based on a framework. This research solves the problem that consultancy companies currently have. The evaluation method provides them with a way to evaluate different BPM products and provide their customers with an objective advice for a certain BPM product.

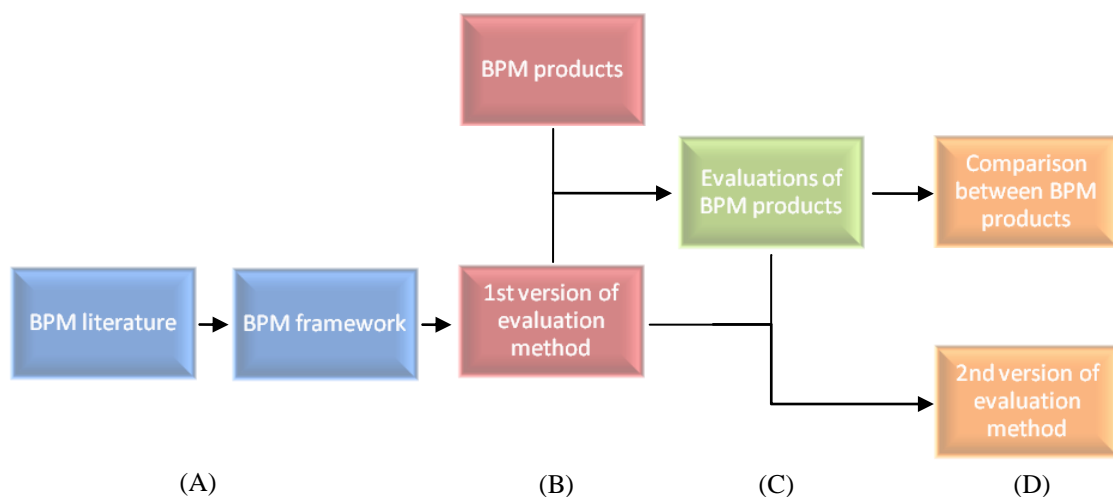
There seems to be a lot of misunderstanding in the definition of the term Business Process Management, especially between vendors and their customers. Some workflow products are labelled as BPM products, while they lack certain (necessary) capabilities for that (Smith and Fingar 2004). Some people talk about Business Process Management when they seem to mean Business Process Modelling (as the author observed at the BPM Congress 2008 in the Netherlands). Defining a framework for BPM helps create a common understanding of the term BPM.

Furthermore, we have tested the method on different BPM products, which resulted in a comparison between them. This comparison can directly be used by a consultancy company to justify choices when proposing one of these products to their customers.

1.3 Research Approach

The approach followed in this research (Figure 2) starts with a literature study (A) in the field of Business Process Management. A BPM lifecycle has been identified and used to order the BPM literature that has been found during this literature study. This literature study was used to define a framework for BPM, which provided the foundation for the first version of an evaluation method of BPM products (B). This first version has been applied to different BPM products, which resulted in an evaluation of these products (C). These evaluations have been used to improve the method and they have been compared (D).

Figure 2 – Research approach



In order to reach the objective of the research, the following research questions are answered in this thesis:

1. What is Business Process Management?
2. How does the BPM lifecycle look like?
3. Which literature is relevant for each lifecycle phase?
4. What criteria can be derived from the literature?
5. How can these criteria be used to evaluate BPM products?
6. What can we learn from the comparison of the analysis results of the three BPM products, when improving the evaluation method?
 - a. What improvements are necessary to the chosen phases?
 - b. What improvements are necessary to the chosen criteria of the phases?

- c. What improvements are necessary to the way these criteria are used?

1.4 Report Structure

The rest of structure of this report has been defined based on the research questions stated above.

- Chapter 2 (Business Process Management) answers the first and second research question by giving our own definition on BPM and the BPM lifecycle.
- Chapter 3 (BPM Framework) answers the third and fourth research questions. This chapter presents the framework that consists of the relevant research literature of each BPM lifecycle phase and the criteria that have been derived from the literature.
- Chapter 4 (Definition of the Method) describes our method which consists of criteria, a case and a way to test and rate the criteria. Also the selected BPM suites for our evaluation are described in this chapter. This chapter gives answer to research question five.
- Chapter 5 (Oracle BPM Suite), 6 (Cordys BPMS) and 7 (IBM WebSphere BPM) report on our evaluation of three different BPM products with help of the application of the method.
- Chapter 8 (Products Comparison) compares the results of the three evaluations and presents a comparison of the products in the form of a strengths and weaknesses analysis.
- Chapter 9 (Conclusions) answers the sixth research question by evaluating the results of the evaluation of the products. This chapter also gives suggestions for further research.

2 Business Process Management

This chapter presents our definition of Business Process Management, discusses how BPM uses explicit business processes, how this relates to the BPM lifecycle and indicates why Business Process Management is an important topic in research literature.

This chapter is further structured as follows: Section 2.1 presents our definition on Business Process Management, Section 2.2 discusses how BPM uses explicit business processes, which models are used and the steps of the BPM lifecycle and Section 2.3 indicates the importance of the field.

2.1 Definition on Business Process Management

Business Process Management (BPM) is the result of the convergence of many trends, among which are business process modelling, quality management, change management, distributed computing, workflow management and business reengineering (Smith and Fingar 2003).

For this research, we define Business Process Management as following:

“Business Process Management is a management discipline that takes a systematic, structured approach to support explicit process management using methods, techniques and tools, and involving humans, organizations, applications, documents and other sources of information, with the aim of reaching the organization’s objectives by aligning the business processes with these objectives.”

To justify our definition, we have compared ours to those of (Elzinga, Horak et al. 1995; Weske, van der Aalst et al. 2004; Gartner Inc. 2008).

(Elzinga, Horak et al. 1995) give the following definition of Business Process Management:

“Business Process Management (BPM) is a systematic, structured approach to analyze, improve, control, and manage processes with the aim of improving the quality of products and services.”

(Elzinga, Horak et al. 1995) argue that BPM uses a systematic structured approach. They define process management as analyzing, improving, controlling and managing. We include more functions (strategy development, discovery, modelling, design, deployment, execution and interaction) in process management, as we discuss in Section 2.2. We also state that BPM supports process management using methods, techniques and tools, involving all sorts of information, whereas (Elzinga, Horak et al. 1995) do not mention this. Finally we argue that the aim of BPM should be not only the improvement of quality, but also cost reduction, attracting more customers or whatever the objectives of an organization are.

(Weske, van der Aalst et al. 2004) give the following definition of BPM:

“Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information.”

When comparing our definition to that of (Weske, van der Aalst et al. 2004), we conclude that they also mention the use of methods, techniques and tools (although they only include software, whereas we also include non-software tools) and the involvement of different sources of information. Like (Elzinga, Horak et al. 1995), their definition of explicit process management does not match ours and they do not mention the alignment with the organization’s objectives.

(Gartner Inc. 2008) defines BPM as following:

“Services and tools that support explicit process management (analysis, definition, execution, monitoring, administration).”

We have compared our definition with the definition put forward by (Gartner Inc. 2008). This definition of BPM is very limited and does not mention that BPM is a systematic and structured approach, that it uses methods and techniques, involves different information sources and aligns the processes with the organization’s objectives.

When comparing our definition to the three mentioned above, we can conclude that none of these definitions mentions that BPM is not a technical method, but a management discipline and that it tries to reach

organization's objectives by aligning the business processes with the organizational objectives. We argue that these two points are essential in the definition of BPM.

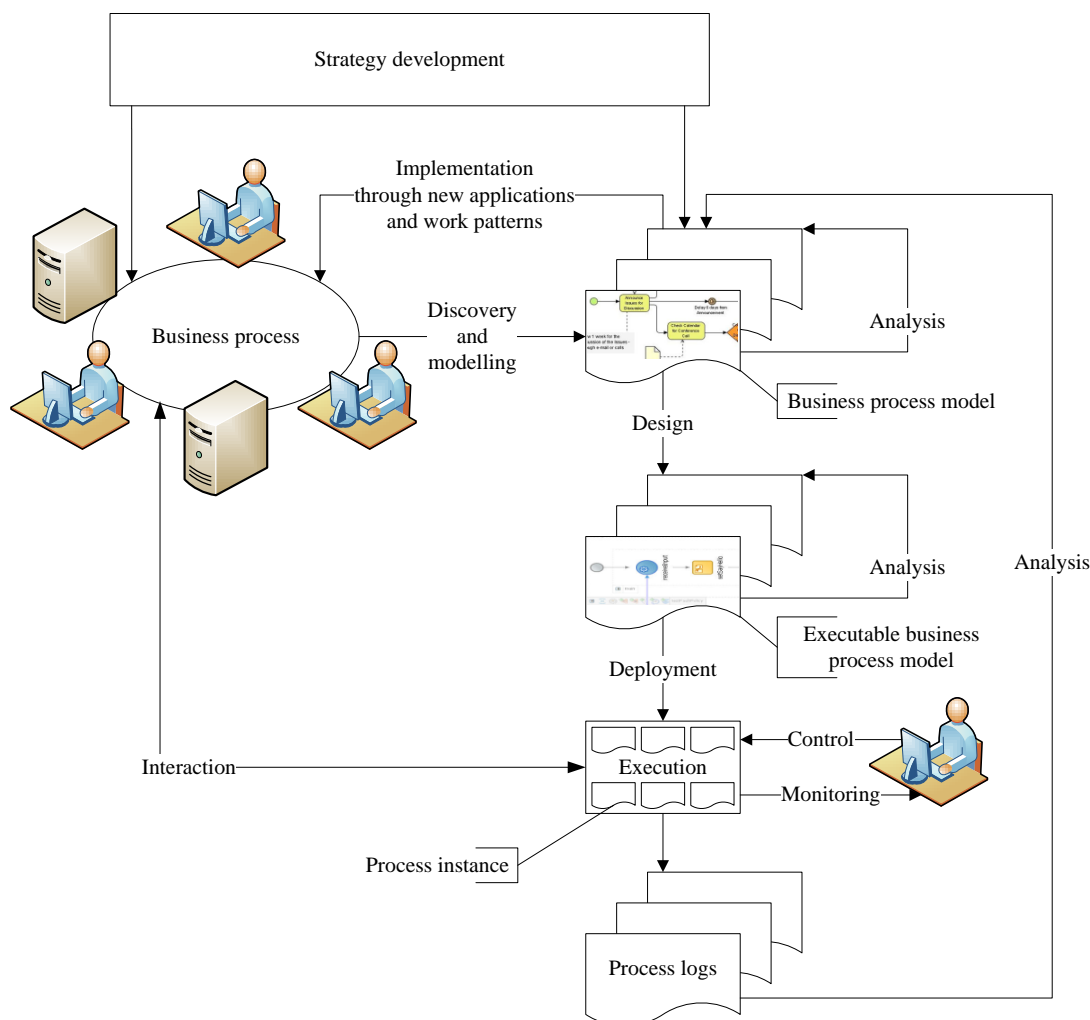
BPM is a management discipline as shown by the 'B' and the 'M' in BPM which stand for 'business' and 'management'. BPM is not only performed by the IT department but also requires the business departments to participate, for example, by defining and changing business rules. Our defining of BPM as a management discipline is backed up by other authors like (Chang 2005) and (Smith and Fingar 2003).

Aligning the business processes with the organization's objective is necessary since the business processes determine the success of an organization. Aligning them with the objectives facilitates the organization's ability to achieve these objectives (Smith and Fingar 2003).

2.2 Explicit Process Management

Our definition states that Business Process Management supports explicit process management. This means that the business processes have to be made explicit. Implicit business processes are embedded in the working patterns of employees and in the application logic of applications. Making a business process explicit implies that a business process model has to be produced in which the business processes are precisely defined.

Figure 3 – BPM lifecycle



This model should be analyzed and improved if necessary. A decision can be made to implement the model with or without IT support, or even to outsource the business process. When implementing a business process without IT support new policies and work patterns are created, to which the employees have to comply. In case IT support is available, the business process model is made executable and a business process execution environment is designed to support it. This business process execution environment consist of a business process

execution engine, which is able to execute the executable business process models, ways of interaction, which users can use to interact with the executable business process models, and management functionality.

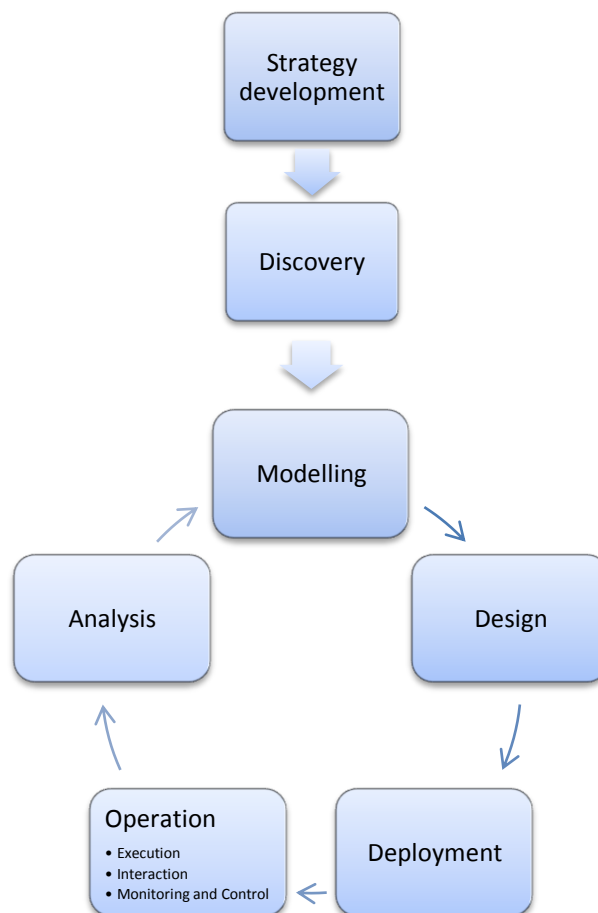
An executable business process model can be translated into code and executed by a business process execution engine. Employees can interact with running business processes and managers can monitor and control them. Running and finished business processes can be analyzed to find room for improvement, creating a continuously improvement loop.

On top of all this, strategy development is performed, which is the process in which the organization's management tries to define the strategic objectives that their organization is trying to reach.

The steps mentioned above can be captured in a BPM lifecycle, which is presented in Figure 3. This figure does not show the option of outsourcing (part of) the business processes. Analysis is represented as a loop, which indicates that after the analysis the model is modified based on the results of the analysis.

To justify our BPM lifecycle, we have compared it to the process management lifecycle by (Petzmann, Puncochar et al. 2007) and the process lifecycle by (Smith and Fingar 2003). The former lifecycle can be found in Figure 5, the latter in Figure 6. For the sake of comparison, we have provided our own model in the same notation as the other two, in Figure 4.

Figure 4 – BPM lifecycle in other notation

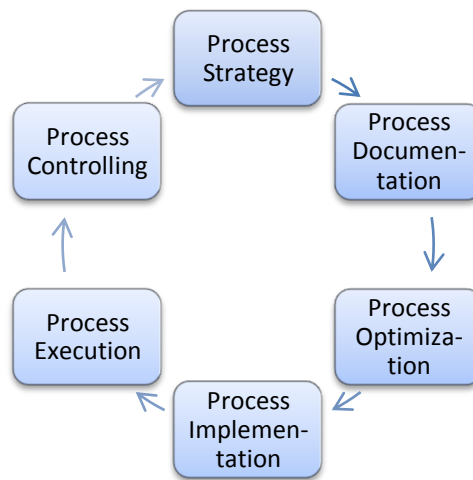


All steps from the lifecycle by (Petzmann, Puncochar et al. 2007), shown in Figure 5, can be found in our lifecycle:

- Process strategy: strategy development in our lifecycle.
- Process documentation: we have split the process documentation into discovery (finding out which business processes are available) and modelling (the production of a business process model) in our lifecycle.

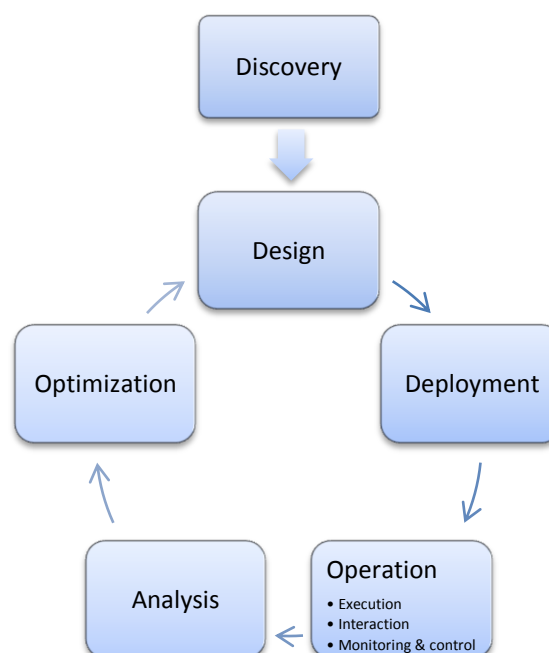
- Process optimization: the analysis cycle at the business process model in our lifecycle indicates that by analyzing the business process model, room for improvement can be found and is the input for improvement of the business process model.
- Process implementation: this can be either unrelated to IT, in which case it is modelled by the arrow from the business process model back to the business process, or IT-related, in which case the design arrow is followed.
- Process execution: we have split their process execution into the execution by the business process execution engine (execution) and the interaction between the users and the execution engine (the interaction arrow).
- Process controlling: we have added monitoring to the controlling phase.

Figure 5 – Process management lifecycle by (Petzmann, Puncochar et al. 2007)



We have also included some extra steps in the BPM lifecycle compared to (Petzmann, Puncochar et al. 2007). These are deployment (rolling out the executable business process to the infrastructure), analysis of the executable business process model and analysis of process logs.

Figure 6 – Process lifecycle by (Smith and Fingar 2003)



When comparing our cycle with the process lifecycle by (Smith and Fingar 2003), shown in Figure 6, we conclude that most steps can be found in both cycles (discovery, deployment, execution, interaction, monitoring, control). There are some differences though:

- We have added strategy development to stress the need of identifying the objectives of the organization in order to be able to align the business processes to these objectives.
- We have split the design step into modelling and design in order to make the distinction between the modelling of the business process into a business process model, and the designing of an executable business process model together with the business process execution environment that surrounds it and allows users to interact with it.
- We see analysis as the start of an improvement cycle that can use three different types of input: the business process model (to improve this model); the executable business process model (this model); historical data such as the process logs (to improve the business process model). (Smith and Fingar 2003) denotes the last type as analysis and optimization

2.3 Importance of Business Process Management

Factors like globalization, the opportunities of e-businesses, deregulation and political instability lead to a turbulent market in which an organization has to constantly adapt. If an organization does not change and adapt to its environment, it faces the risk of being put out of the market (Warner and Trahan 2000). Therefore organizational change is important for an organization's survival.

Traditionally, a distinction is made between revolutionary and evolutionary change. The former is a radical type of change which takes place in a relatively short period of time and is facilitated by people high in the hierarchy of the organisation. Business Process Reengineering projects are examples of this type of change. The latter type of change is more gradual. With this type of change, small steps in a long period are taken by people closer to the operational level. A good example of this type of change is Total Quality Management.

Business Process Management (BPM) is a method that facilitates both types of organizational change (Elzinga, Horak et al. 1995). It gives organizations the opportunity to manage (and thus change) their business processes, by making the business processes explicit and flexible, as opposed to the fixed process logic supported in most information systems. Moreover, it gives the organization the opportunity to change their business processes from a business expert's process viewpoint rather than from a technical viewpoint. It also allows organizations to align their business processes more directly with their organizational objectives (Smith and Fingar 2003). Among organizations there is also a clear demand to be able to change their business processes, making Business Process Management a relevant topic in research literature (Pritchard and Armistead 1999; Smith and Fingar 2003; Wang and Wang 2006; Chen, Zhang et al. 2007; Petzmann, Puncochar et al. 2007; Weske 2007).

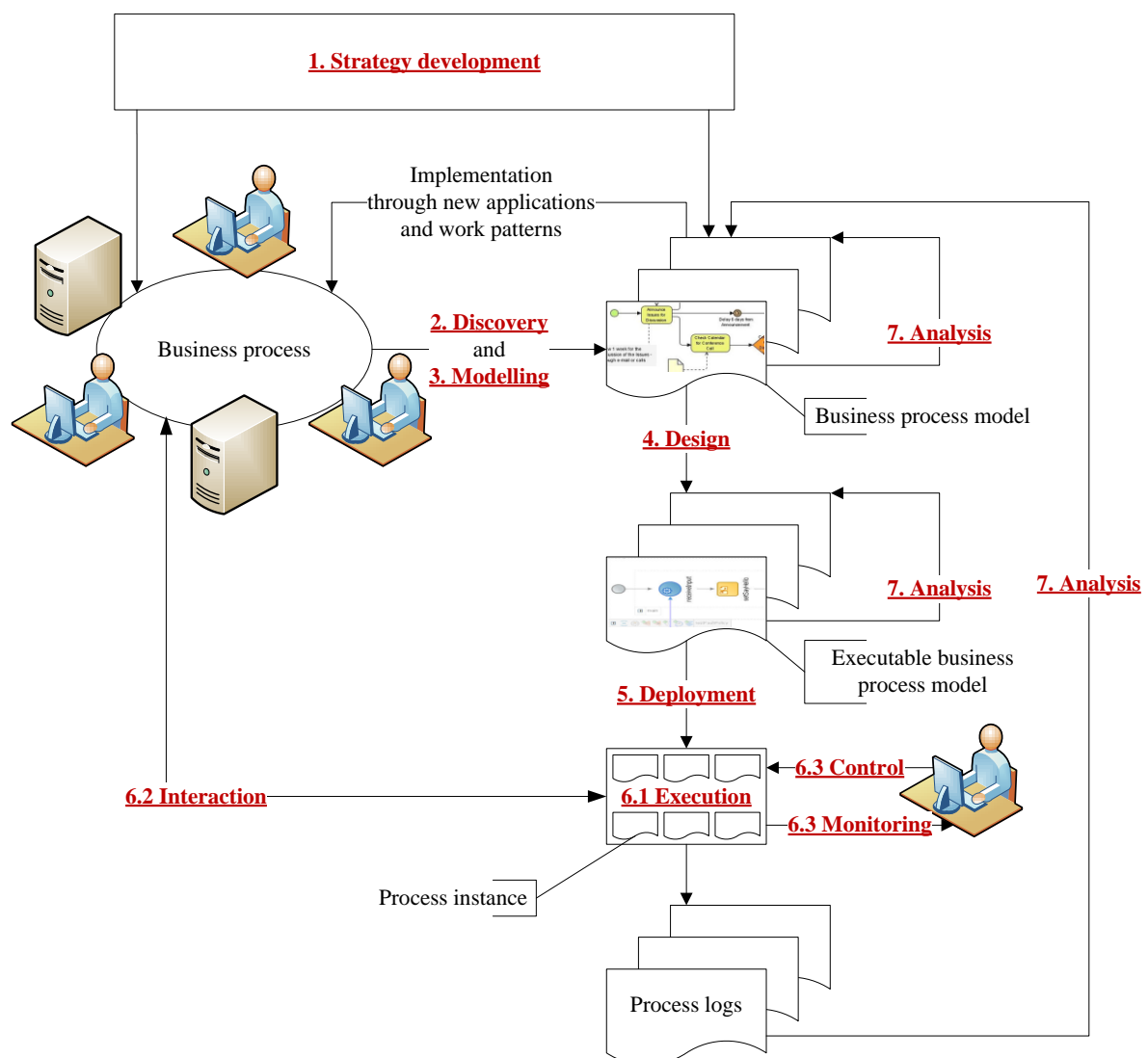
3 BPM Framework

This chapter describes the Business Process Management framework which is the result of our literature study on Business Process Management. The framework is ordered by the phases of our BPM lifecycle (Figure 7). A list of criteria has been derived from the literature discussed at each phase. These criteria are placed in tables at the end of the discussion of each phase. All the framework's criteria can be found in Appendix A: Criteria used by our method.

This chapter is organized as following: Section 3.1 discusses the strategy development phase, Section 3.2 discusses the discovery phase, Section 3.3 discusses the modelling phase, Section 3.4 discusses the design phase, Section 3.5 discusses the deployment phase, Section 3.6 discusses the operation phase, which consists of execution, interaction and monitoring and control and Section 3.7 describes the analysis phase.

For each phase, a graphical representation of the steps in the phase is given in the Business Process Modeling Notation (BPMN).

Figure 7 – BPM lifecycle with highlighted phases

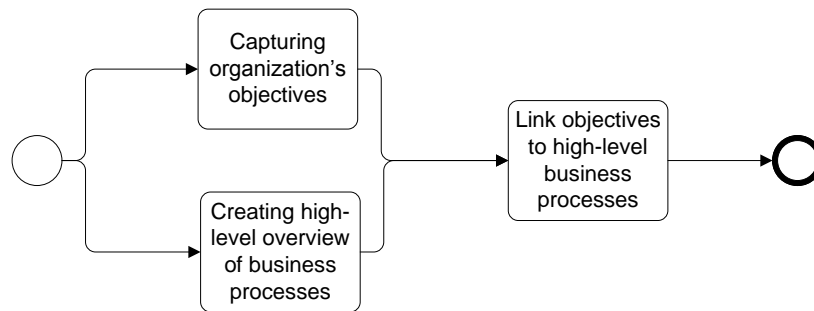


3.1 Strategy Development

During the strategy development phase the organization's objectives and a high-level overview of the business processes are developed (Petzmann, Puncchar et al. 2007; Ricken 2007). The objectives can then be linked to

the high-level business processes. This overview consists of a coarse-grain aggregation of the business processes and shows these business processes without details. Figure 8 shows a graphical representation of this phase.

Figure 8 – Strategy development phase



In a later phase these objectives and high-level business processes form the basis for the goals to be achieved by individual business processes, and for the Key Performance Indicators (KPIs) that measure if these goals are reached. This ensures that the business processes are aligned with the objectives of the organization and that the organization can be successful.

The objectives of the organization can be captured by using, for example, a Balanced Scorecard, and the high-level business processes can be shown by a value-added chain diagram (Ricken 2007). Figure 9 shows an example in which objectives are linked to high-level business processes.

Figure 9 – Linking objectives to high-level business processes, picture adapted from (Ricken 2007)

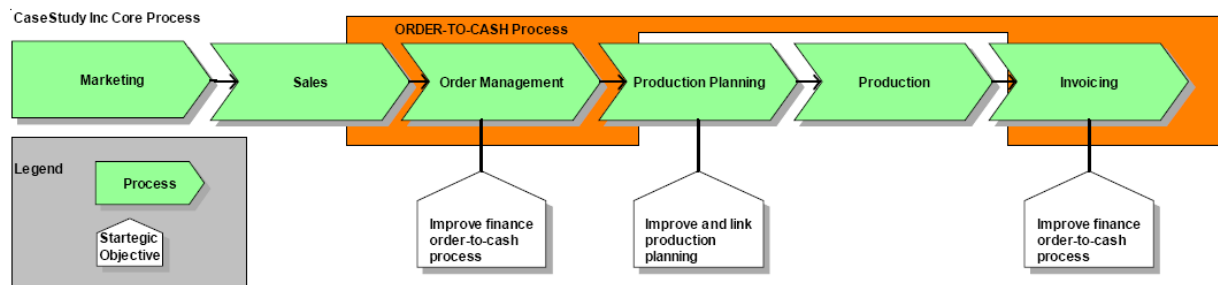


Table 1 – Strategy development criteria

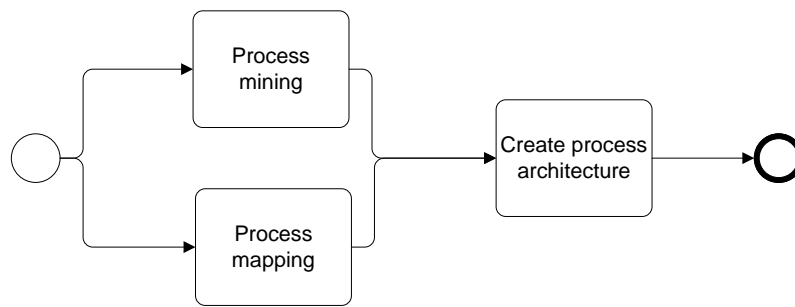
	Strategy development criteria
1.	Support for capturing the organization's objectives (for example, using a Balanced scorecard).
2.	Support for creating a high-level overview of the business processes (for example, using a value-added chain diagram).
3.	Support for linking the organization's objectives with the high-level business processes.

3.2 Discovery

The discovery phase details the high-level business processes into finer grained business processes. This is done by observing the way the organization is currently working and creating (informal or incomplete) business process models based on this information. These business process models are an approximation of the business processes. The implicit business processes are identified from work patterns and applications. These business processes can be made explicit either manually (process mapping) or automatically (process mining). Relations between the business processes can be shown in a process architecture.

Figure 10 shows a graphical representation of this phase.

Figure 10 – Discovery phase

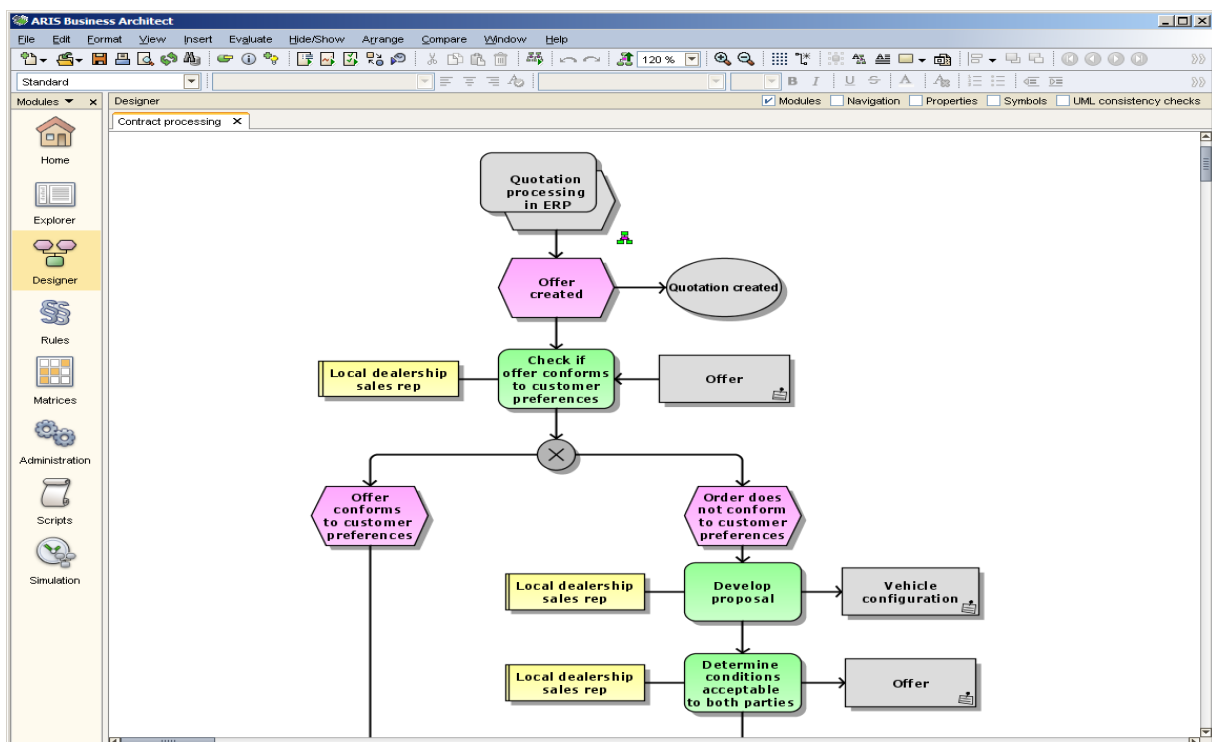


3.2.1 Process mapping

Manual discovery can be done by applying a technique called process mapping (Hunt 1996). This technique consists of drawing a model of the business process, which shows the involved activities, the participants who execute the activities and the flows between the activities. Process mapping can be done by an outsider (for example, a consultant) or by somebody from the organization who knows the business process (for example, the process owner).

Process mapping is a troublesome activity for an outsider, since it requires deep knowledge of the studied business process. If the outsider does not have enough knowledge of the business process, he can interview all the participants and draw a map of the business process based on these interviews, for example, during a workshop (Fülscher and Powell 1999). This takes a lot of time and can be costly (Davenport 1993; Hammer and Champy 1993; Hwang and Yang 2002; van der Aalst, van Dongen et al. 2003).

Figure 11 – ARIS business architect

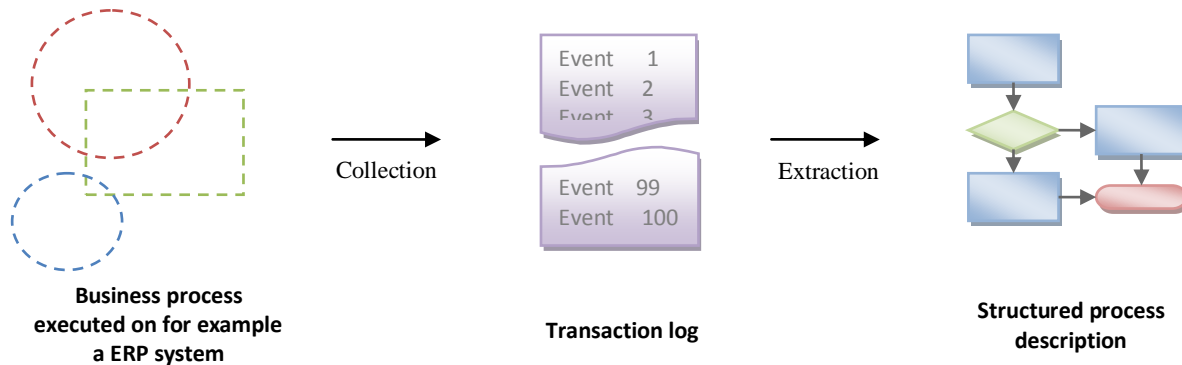


Most often process owners have a business background and no IT background. They can draw a map of the business processes by using popular techniques and notations, such as flowcharts and cause-and-effect (fishbone) diagrams. Other commonly used methods as IDS Sheer's ARIS methodology (Figure 11) and CSC's Catalyst can be used. In order to make BPM accessible for these people, a BPM product should support these techniques and commonly used methods by allowing process maps created with their help to be imported in the BPM product.

3.2.2 Process mining

Automatically inferring business processes from information systems can be done by using process mining (Figure 12), which extracts business process information from transaction logs (van der Aalst, van Dongen et al. 2003). The term process mining refers to methods for distilling a structured process description from a set of real executions (Cook and Wolf 1995; Mărușter, Weijters et al. 2006).

Figure 12 – Process mining



Possible sources for the automatic extraction of business process information are systems such as workflow management systems, ERP systems, CRM systems, and case handling systems. In principle, any information system that keeps a transaction log and registers the events concerning the execution of activities for supporting business processes is suitable for process mining (van der Aalst, van Dongen et al. 2003; Mărușter, Weijters et al. 2006).

The transaction log should at least contain the description of an activity (which can be only a name), the type of event (for example, activity starts, activity stops) that is recorded concerning an activity, the actor who executes the activity and a timestamp.

In the case of a BPM product, it is interesting to know whether process mining is available, and, if this is the case, how it handles some of the common aspects of process mining. These aspects are the following (van der Aalst, van Dongen et al. 2003):

- Incomplete logs: logs can be incomplete and not cover all possible process paths of the business process.
- Noise: sometimes parts of the logs are incorrect, incomplete or refer to exceptions.
- Duplicate activity names: this aspect refers to different activities with the same name. This makes it hard to distinguish them in the log.
- Structure: there are two major types of business process meta-models, namely graph-oriented meta models and block-oriented meta-models. This aspect refers to the type of meta-model that is produced as output of the process mining.
- Time: logs record the timestamps of the execution of activities. The product can use this information, for example, to calculate performance indicators such as waiting/synchronization times, flow times and utilization.
- Basic parallelism: this aspect refers to the ability of a product to mine simple business processes in case each AND-split corresponds to an AND-join. Some products impose requirements on the business process in order to extract a suitable model.
- Non-free choice: for some products, the non-free choice construct is a difficult pattern to mine, because it mixes synchronization and choice in one construct (Desel and Esparza 1995).
- Basic loops: basic loops are structured loops which have one entry and one exit point. Some products impose restrictions on the structure of these loops in order to guarantee the correctness of the discovered model.
- Arbitrary loops: arbitrary loops have multiple entry and/or exit points, which make them harder to mine, because business processes with these loops do not have a clear block structure.

Some products can still extract the business process correctly from incomplete logs or logs with noise, or at least indicate that some activities are not recorded, while other products require complete or noise free logs.

3.2.3 Process architecture

The final step in this phase is linking the business processes to each other in order to represent their relationship in a process architecture. Relations between business processes can be, for example, output/input (the output of one business process is the input of another business process) or managerial (one business process manages another business process).

Table 2 – Discovery criteria

	Discovery criteria
1.	Support for process mapping (manual discovery of business processes) <ol style="list-style-type: none"> I. Support for popular techniques and notations (for example, flowcharting and cause-and-effect diagrams). II. Support for popular methods like IDS Sheer's the ARIS methodology and CSC's Catalyst.
2.	Support for process mining <ol style="list-style-type: none"> I. Ability to handle incomplete logs. II. Ability to handle noise. III. Ability to handle duplicate activity names. IV. Types of structures supported (graph or block). V. Ability to handle timestamps. VI. Ability to handle basic parallelism. VII. Ability to handle non-free choice constructs. VIII. Ability to handle basic loops. IX. Ability to handle arbitrary loops.
3.	Support for defining a process architecture.

3.3 Modelling

The result of the discovery phase is typically a business process model modelled in an informal or incomplete way (for example, in a flowchart). To be able to analyse the business process model it has to be modelled in a formal business process modelling language. Different alternative models to the same business process are produced in the modelling phase, which can be compared in the analysis. Also, to keep a good distinction between these different versions, version management has to be in place. Version management is discussed in section 3.6.1 since it is especially important to distinguish between different running versions of the same business process.

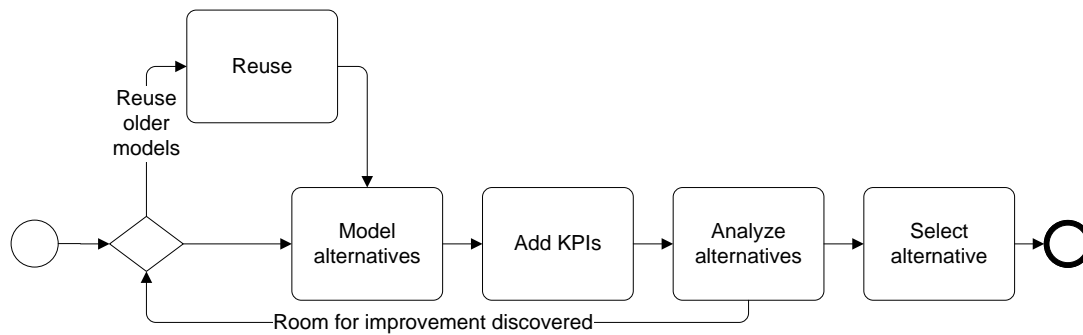
Furthermore, the objectives that were linked to the high-level business processes are to be expanded into goals and KPIs that apply to the individual business processes.

The first time the BPM lifecycle is executed, the input to the modelling phase are the informal or incomplete business process models from the discovery phase. When the BPM lifecycle has been completed the first time, the input to the modelling phase can also be complete (executable) business process models which have been already modelled. These business process models (or parts of them) can be reused in new business process models.

At the end of this phase, one of the modelled alternatives is selected. This alternative can be implemented without IT support, with (partial) IT support or outsourced. Only the second option is within our scope. In this case the design phase follows the modelling phase.

Figure 13 shows the graphical representation of this phase.

Figure 13 – Modelling phase



3.3.1 Reuse

The first step in this phase is to select the older business process models which are reused in the new business process model. Just as the reuse of software components is a way to improve the efficiency and quality of the design process in software engineering, business process model reuse improves the efficiency and quality of the business process models. In order to enable the reuse of business process models, they have to be stored and managed in a business process model repository (similar to a component repository). A business process model repository can retrieve processes in two ways (Zhuge 2002):

- Exact retrieval: the retrieval of business process models that exactly match the user requirements.
- Inexact retrieval: the retrieval of a business process model which approximately matches the user requirements.

If business process models (or parts of them) should be available for reuse, they also have to be harvested. Harvesting is the process of transforming one or more (executable) business process models into a business process model so that this model can be reused afterwards. There are four categories of mechanisms which can be used for harvesting and reuse and two strategies (bottom-up and top-down) (Franch and Ribó 2003). Table 3 shows the different harvesting and reuse mechanisms.

Problems can occur when using harvesting and reusing mechanisms. When looking at BPM products, it is interesting to see if and how these problems are handled correctly. The described problems are (Franch and Ribó 2003):

- Consistency problems: these occur when two elements of different business process models have the same identifier but do not represent the same concept.
- Redundancy problems: these occur when two elements represent the same concept but have different identifiers.
- Residue problems: these occur when an element becomes a part of a composite business process model because it was part of a component business process model, but is not required in the composite business process model.

Table 3 - Harvesting and reuse mechanisms (Franch and Ribó 2003)

Category	Subcategory	Mechanism
Abstraction		Generalization
		Parameterization
Adaptation	Addition	Specialization
		Instantiation
	Removal	Projection

	Substitution	Renaming
		Semantic substitution
Composition	Shallow composition	Grouping
	Deep composition	Combination
	Inclusion	Inclusion
Delayed binding		Refinement
		Inclusion

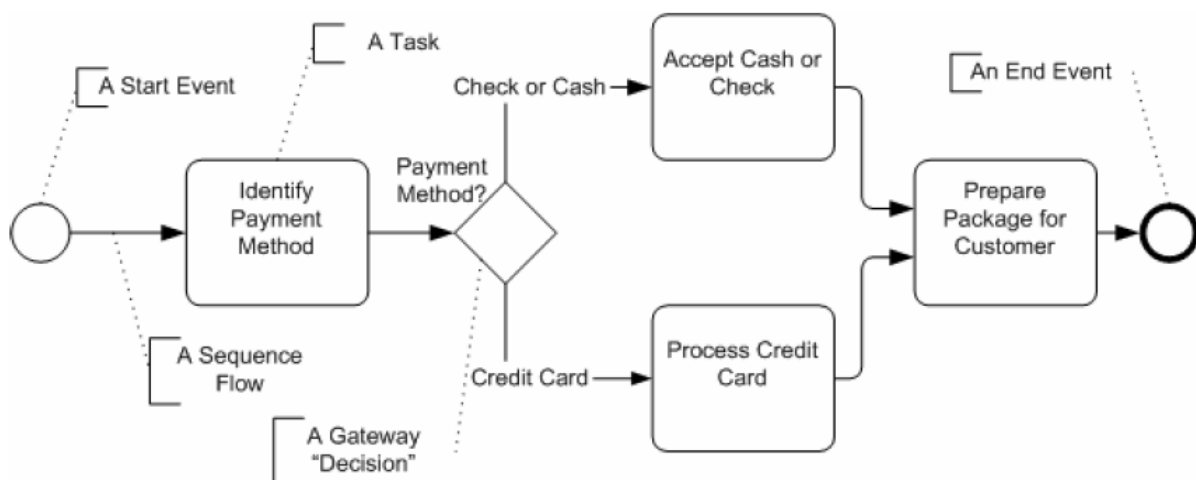
3.3.2 Business process modelling

After the optional reuse step, the business process is modelled in a formal business process modelling language. This step is called Business Process Modelling. This is needed to create an unambiguous definition of the business process model so alternative interpretations can be ruled out, and to be able to analyse the alternatives. However, there is a need to represent this formal model in different perspectives, so that different stakeholders can understand it (van der Aalst, ter Hofstede et al. 2003). This could mean that the business process model has a perspective designed for business oriented people and one for technical oriented people. These perspectives should assign the same meaning to the business process model, in example, there should not be any scope for alternative interpretations.

Business Process Modelling is not only done for the current way of working, but also for different alternatives of the business processes which are being considered.

There are different business process modelling languages available, which include UML 2 Activity Diagrams (AD), the Business Process Definition Metamodel (BPD), the Business Process Modeling Notation (BPMN) (Figure 14 shows an example of a business process modelled in BPMN), Event Driven Process Chains (EDPC), IDEF3, Petri Nets, Role Activity Diagrams (RAD) (List and Korherr 2006) and the more recently developed Business Process Execution Language for Web Services (WS-BPEL) which is also executable (Andrews, Curbera et al. 2003; Alves, Arkin et al. 2007). There is still a debate on what a standard business process modelling language should look like, and it seems unrealistic that a consensus will ever be reached (van der Aalst, Leymann et al. 2007). However interoperability among the different languages is necessary because this will allow the interchange of business process models between different products (Smith and Fingar 2003).

Figure 14 – Example of a simple business process modelled in BPMN (White 2004)



A discussion of the advantages and disadvantages of the different business process modelling languages is outside the scope of this research, but other authors have provided means to compare them (Russell, ter Hofstede et al. 2003; Russell, ter Hofstede et al. 2004; Gruhn and Laue 2006; List and Korherr 2006).

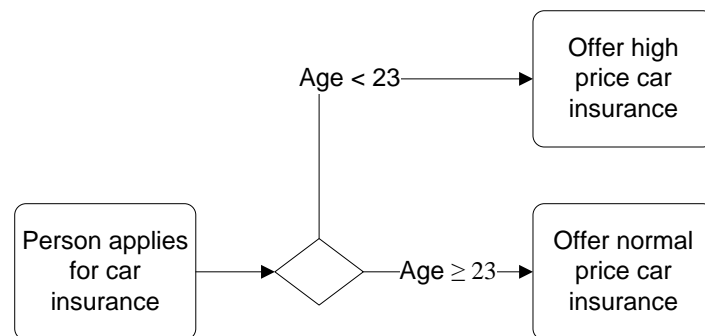
When modelling a business process, different views can be followed, but they should all be taken into account in the modelling phase. This has to be done because each view on business process is based on a set of assumptions and thus acts as a filter that allows one to see certain things but to omit others (Nuno Melão 2000). These views are (Curtis, Marc et al. 1992):

- Functional: representing the activities being performed with relevant flows;
- Behavioural: representing when and how the activities are performed;
- Organizational: representing where and by whom the activities are executed and what physical communication mechanisms and storage are used;
- Informational: representing information entities (like forms) flowing through the process, their structure and relationships.

Another aspect to take into account when modelling a business process is the modelling of business rules. These business rules can be modelled in three different ways:

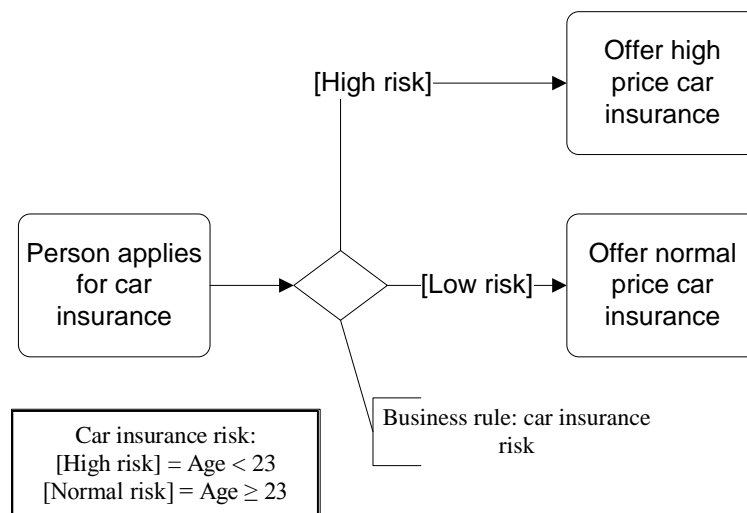
- Using implicit business rules: all the business rules are modelled into the business process model, typically as a decision point. For example, insurance companies have the business rule that people under the age of 23 who apply for car insurance pay a higher price than older people. Figure 15 shows this situation modelled with implicit business rules. Changing the business rule means changing the business process model.

Figure 15 – Implicit use of business rules



- Using explicit business rules: the business rules in the business process model are only modelled as a reference to the rule. The business rules themselves are stored in a repository and managed separately. Figure 16 shows the same situation with explicit use of business rules.

Figure 16 – Explicit use of business rules



- Model the complete business process using business rules: in this case the business process is not modelled in a business process model, but entirely made out of explicit business rules. These business rules will be executed by a business rules engine. Table 4 shows the same situation modelled using only business rules.

Table 4 – Business process modelling using only business rules

On	Person applies for car insurance
IF	Age < 23
Then Do	Offer high price car insurance
Else Do	Offer normal price car insurance

The awareness of the importance of making these business rules explicit is increasing (Ross 2003). Business rules are changed more frequently than the business processes. Making them explicit facilitates their reuse and simplifies their management. One of the main challenges when using explicit business rules is the way they are specified. Since the intended modelling audience of business rules are process owners, who most of the times have a business background, the way of specifying business rules in a BPM product should closely match to the way the process owners specify them, which is in natural language (Business Rule Group 2003).

3.3.3 Key-performance indicators

The third step in the modelling phase is the definition and addition of KPIs to the business process model. These KPIs should have a desired target value attached to them, so that during monitoring and post runtime analysis of the business process model the real value of the KPI can be compared to the desired target value.

KPIs can be defined to measure characteristics of process instances, processes, resources, or of the overall business operations. Examples of KPIs of interest can be the amount of successful executions of the business process compared to the total amount of business process executions, total amount of human labour needed or the amount of resources needed for a process (Castellanos, Casati et al. 2005).

Two challenges of KPI definition exist (Castellanos, Casati et al. 2005) and should be handled by BPM products:

- The identification of a KPI model that is sufficiently flexible, allowing the definition of different types of KPIs and of goals related to KPIs, and of different perspectives under which to consider a certain KPI.
- The definition of KPIs. This can require users to write code, but preferably KPIs should be made without requiring users to write code themselves. This can be done, for example, by using KPI templates or using graphical notations. User-defined KPIs should also be validated (Grigori, Casati et al. 2004; Castellanos, Casati et al. 2005).

Table 5 – Modelling criteria

	Modelling criteria
1.	Support for harvesting/reuse. <ol style="list-style-type: none"> Support for retrieval methods (exact/inexact). Support for harvesting/reuse mechanisms. Ability to handle consistency problems. Ability to handle redundancy problems. Ability to handle residue problems.
2.	Support for Business Process Modelling. <ol style="list-style-type: none"> Support for different business process modelling languages (for example, AD, BPDM, BPMN, EDPC, IDEF3, Petri Nets, RAD and BPEL4WS). Support for interoperability between different business process modelling languages. Support for interoperability between simple process modelling methods and business process modelling

	languages.
	IV. Support for different views when modelling a business process.
	V. Support for different perspectives of a business process.
	VI. Support for modelling with implicit business rules.
	VII. Support for modelling with explicit business rules.
	VIII. Support for modelling business processes using explicit business rules.
	IX. Ease of use of defining explicit business rules.
3.	Support for KPIs.
	I. Flexibility of KPI model.
	II. Ease of use of defining KPIs.

3.4 Design

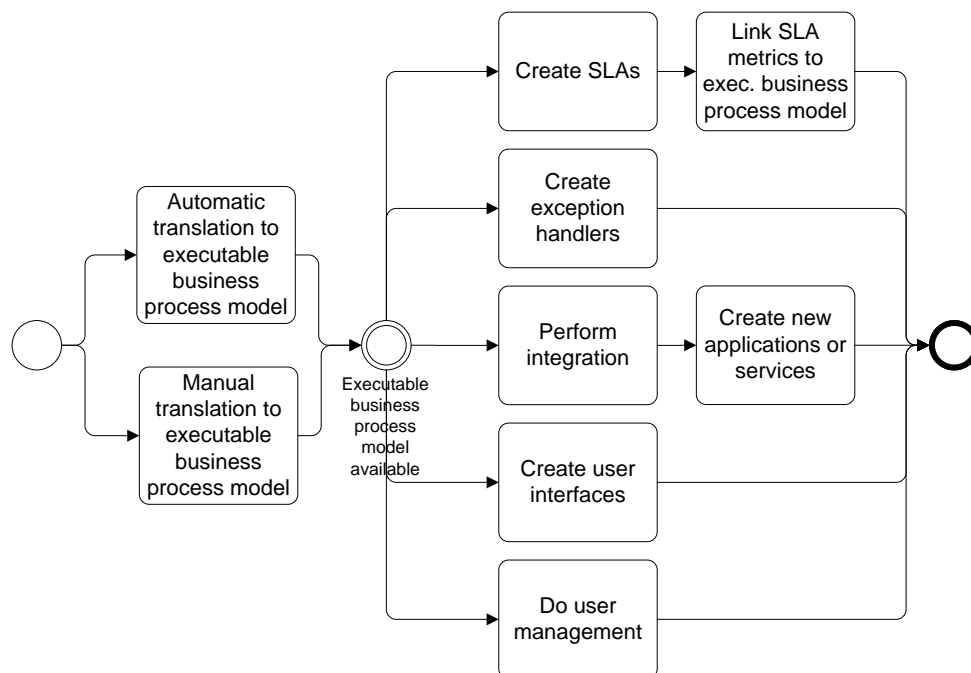
The design phase is entered when a business process model is implemented with the support of IT. In order to be able to execute the business process model, this model is translated into an executable business process model. As in the modelling phase, version management is also in place in this phase, but this is discussed in section 3.6.1, because version management is especially important when distinguishing between different running versions of the same business process.

When the executable business process model is created, Service Level Agreement (SLA) can be set up and the KPIs necessary to measure if the SLA is met can be linked to the executable business process model. Also exception handlers are added in order to be able to anticipate exceptions. Another step in this phase is the integration with other systems. This step is followed by the creation of new applications or services if necessary.

Since a business process requires interaction with users, user interfaces are created in the design phase and users are created and given roles and permissions. The last two steps are performed with the help of user management.

Figure 17 shows the graphical representation of the steps mentioned above.

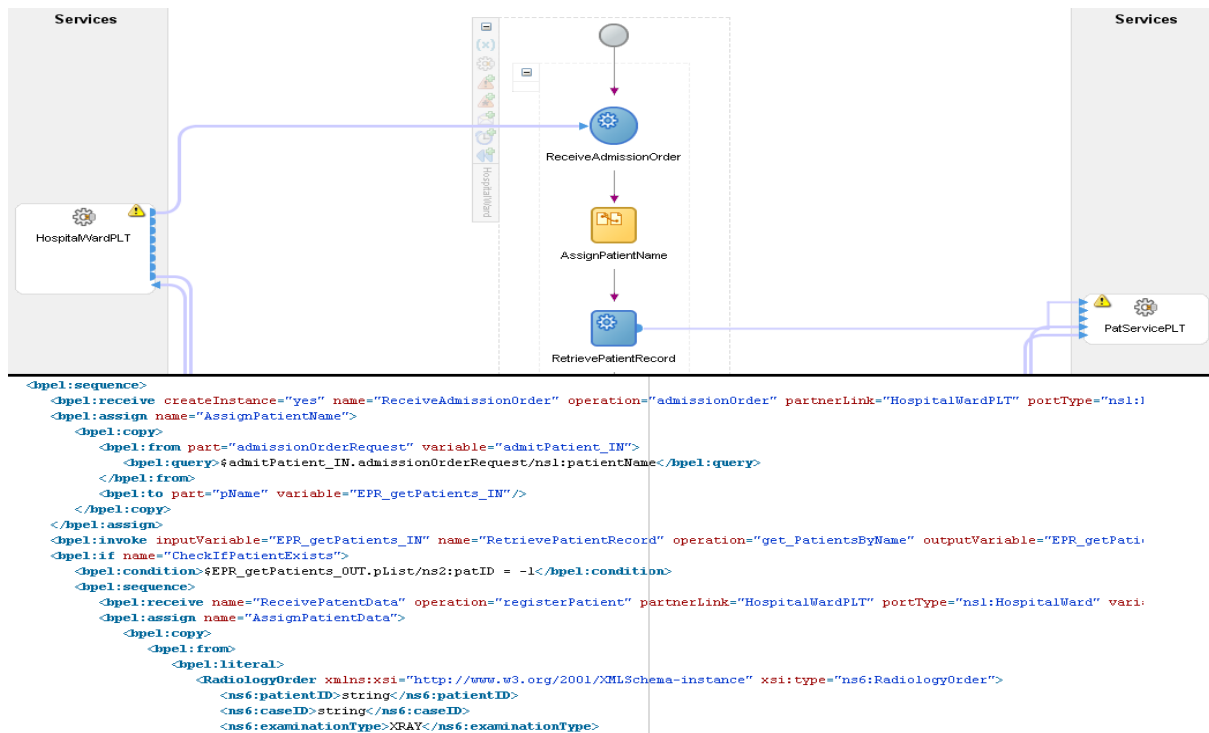
Figure 17 – Design phase



3.4.1 Translation into the executable business process model

The first step in the design phase is the translation of the business process model into the executable business process model. The executable business process model is modelled in a business process execution language. Currently the Business Process Execution Language for Web Services (WS-BPEL or BPEL for short) (Andrews, Curbera et al. 2003; Alves, Arkin et al. 2007) is the standard business process modelling and executing language for designing and implementing business processes based on web services (van der Aalst, Leymann et al. 2007). Figure 18 shows an example of a business process modelled in BPEL.

Figure 18 – Example of a part of a business process modelled in BPEL (graphical representation and the underlying BPEL code)



Some business process modelling languages can be easily translated into a business process execution language. In this case, the translation can be done automatically (Recker and Mendling 2006; Ouyang, Dumas et al. 2007). However, while automatic translation is preferred over manual translation, BPM products should offer a modelling environment in which the business process model can be modelled in a business process execution language. This is necessary since not all business process modelling languages map completely onto a business process execution language, in which case the translation has to be done manually (List and Korherr 2006).

After the translation of the business process model into the executable business process model, the two models have to be linked so that changes in the executable business process model are also reflected in the business process model. This is the reverse of the design arrow in our BPM lifecycle in Figure 7 and is called traceability. This makes sure that the process owners, who typically view the business process as business process models, are looking at the most recent version of the business process and not at a model that does not represent the running business process anymore.

3.4.2 Service level agreements

Once the business process model is translated into an executable business process model, the Service Level Agreements (SLAs) can be added to the model. In order to know if an SLA is met, SLA KPIs have to be defined and linked to the different executable business process models.

3.4.3 Designing exception handling

Another step in the design phase is the design of exceptions handling. Exceptions in the business process lead to a negative (but not abnormal) termination of an activity. Examples are trying to book a hotel room while the hotel is overbooked or a user that fails to pay his periodic instalments. These can be discarded if the activity is not obligatory, but are to be taken seriously when it is. Exceptions are recovered by re-executing the activity,

letting a user handle the exception or by rolling back the process (and the completed activities) until a consistent process state is reached. From here, the process can continue by following an alternative path or can be terminated.

In order to make sure that the rollback of a business process can be achieved, a failure and cancellation mechanism has to be in place. The ACID principle was borrowed from databases and applied to business processes since they contain activities that access shared persistent data resources, and so are subject to transactional semantics. In particular, the properties of atomicity and isolation can be usefully applied to business processes (or at least to parts of a business process) (Dayal, Hsu et al. 2001; Greenfield, Fekete et al. 2003).

The next step was the development of the Saga approach. A Saga is a long running process, consisting of smaller activities or transactions that can be performed as an ACID transaction. If an activity fails, the previous activities roll back in the reverse order by executing their compensation transactions. Therefore, this approach is also referred to as compensation handling (Dayal, Hsu et al. 2001; Greenfield, Fekete et al. 2003).

Later on, the idea of exception handlers was borrowed from the programming world. When a certain situation arises (an exception) the runtime engine terminates the normal process and executes the exception handler (also called compensation handler) (Greenfield, Fekete et al. 2003).

While the ideas from Sagas and exception handlers are used in many BPM products, they do not solve all failure. Some activities cannot be rolled back (for example, the physical movement of a good) and most of the times the compensation does not take the state of the process into account (Greenfield, Fekete et al. 2003).

3.4.4 Integration

The integration of the business process with different systems is also a subject in the design phase. BPM aims to integrate different systems (for example, databases, customer relationship management systems and payroll systems) from both inside and outside an organization. Traditionally enterprise application integration (EAI) has been used for integration inside the organization and Business to Business integration (B2Bi) for integrating with applications outside the organization (Smith and Fingar 2003).

The underlying middleware technology plays an important role in the integration. BPM products can rely on different technologies like message brokers, transactional queue managers, publish/subscribe middleware or an enterprise service bus (Dayal, Hsu et al. 2001; Pasley 2005).

When working together with other organizations, two approaches are available (Papazoglou and Ribbers 2006; van der Aalst, Leymann et al. 2007):

- **Orchestration:** this is a method to describe how different organizations work together from the perspective and under control of a single endpoint.
- **Choreography:** this is a method where each party describes the role that they play in the interaction, but no party owns the conversation.

After the existing systems have been integrated, new applications or services that supply missing functionality to the business processes can be designed. While these can be designed with the help of external programs, they may also be designed within the BPM product. An important aspect when evaluating this functionality is the supported programming languages.

3.4.5 Designing user interfaces

The final two steps of the design phase focus on the users who interact with the business processes. Users typically send information to the business process through forms. Separate applications can be created for this purpose, but there is a trend to do this through a (web) portal. These user interfaces can either be created or generated automatically, in which case some BPM products offer many customization options and others generate only a standard interface. An activity list with all the activities the users has to complete for the BPM product can sometimes be added to the interface. From this list the user can launch the associated tool (Leymann 2003; Smith and Fingar 2004).

3.4.6 User management

Finally, user management is necessary to determine who has access to the BPM product with which privileges. This is done by defining roles. To establish which role users should have, a way to define capabilities of users has to be employed in the BPM product. This can be done based on the roles users have in the organisation

(preferably by importing the organisation structure from an existing system like LDAP) and/or based on tokens, which embody certain capabilities of users to execute certain functions, procedures or activities. Based on their capabilities, users should be assigned the appropriate role, which can be either done manually or automatically (Chiu, Li et al. 1999). The permissions for the activities in the executable business process model are set, based on roles. These permissions describe which role can execute, skip or retry a certain activity.

Table 6 – Design criteria

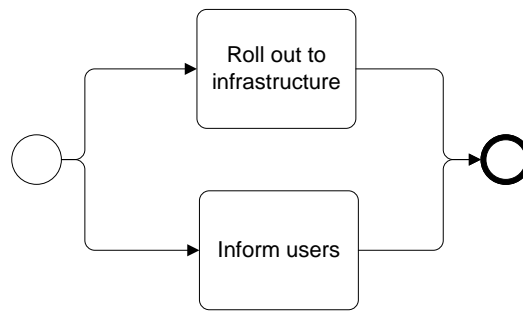
	Design criteria
1.	Support for creating executable business process models. <ul style="list-style-type: none"> I. Supported business process execution languages. II. Support for manual translation between business process modelling languages and business process execution languages. III. Support for automatic translation between business process modelling languages and business process execution languages. IV. Support for synchronizing the business process execution model with the business process model.
2.	Support for adding SLA and linking SLA KPIs to business processes.
3.	Supported exception handling methods (for example, ACID, Saga or compensation handlers).
4.	Support for integration. <ul style="list-style-type: none"> I. Underlying middleware technology. II. Support for orchestration. III. Support for choreography.
5.	Supported programming languages for implementing services/applications.
6.	Support for designing user interfaces. <ul style="list-style-type: none"> I. Support for separate applications for data entry. II. Support for portal technology. III. Support for an activity list.
7.	Support for user management. <ul style="list-style-type: none"> I. Way of describing roles (by names/description or by capabilities). II. Support for determining user capabilities based on organizational role. III. Support for importing organizational structure from other systems. IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities). V. Support for manually assigning a user to a role. VI. Support for automatically assigning a user to a role, based on his capabilities. VII. Support for activity permissions based on roles.

3.5 Deployment

The deployment phase makes sure that the new business processes are rolled out to all participants. The users are informed that the new business process is in place and the BPM environment (new applications/services, portals and the executable business process models) are rolled out to the technical infrastructure (for example, application servers, web servers, business process execution engines).

Figure 19 shows the graphical representation of the deployment phase.

Figure 19 – Deployment phase



One of the two steps in this phase is the rolling out of the BPM environment to the technical infrastructure. Applications and services are mapped onto servers and the executable business process models are sent to the business process execution engine.

BPM can be deployed in a small environment (for example, in a single department) in which case the infrastructure is homogeneous (for example, a specific business process execution engine and local resources), but in most cases it is deployed in a larger environment (in the entire organization or between different organizations). In the latter case, the processes are executed in a federated process management infrastructure (Leymann, Roller et al. 2002). Two types of distributed execution environments can alternatively be supported by BPM (Smith and Fingar 2003):

- 1) Distributed execution of the process, where a business process runs on multiple business process execution engines. This happens, for example, when the total business process spans multiple organizations, which all play a part in the business process and all use their own business process execution engine for their part of the business process.
- 2) The second is a distributed business process execution engine where a business process execution engine runs on multiple computers. This occurs when the load on the business process execution engine is too high to let one server take care of it. Therefore the business process execution engine is running on a cluster of servers.

A way to compare BPM products based on their capability to represent and handle resources is the compliance with the workflow resource patterns (Russell, ter Hofstede et al. 2004).

The other step in this phase is informing the users that a new business process is in place and that they have to participate in this process. Section 3.6.2 discusses the informing of users in more detail, since most of the informing of users takes places in the interaction sub-phase.

Table 7 – Deployment criteria

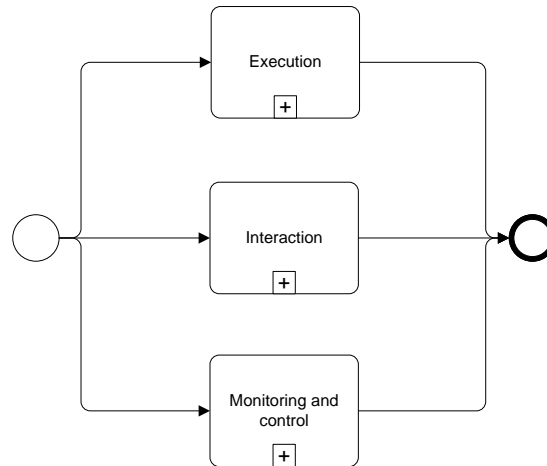
	Deployment criteria
1.	Support for distributed execution <ol style="list-style-type: none">I. Support for multiple business process execution engines.II. Support for distributed business process execution engines.
2.	Support for workflow resource patterns.
3.	Support for informing the user of a new executable business process model.

3.6 Operation

The operation phase consists of three sub-phases, namely, the execution of the business processes, the interaction between users and the business processes, and the monitoring and control of the business processes and their execution environment.

Figure 20 shows the graphical representation of this phase.

Figure 20 – Operation phase



3.6.1 Execution

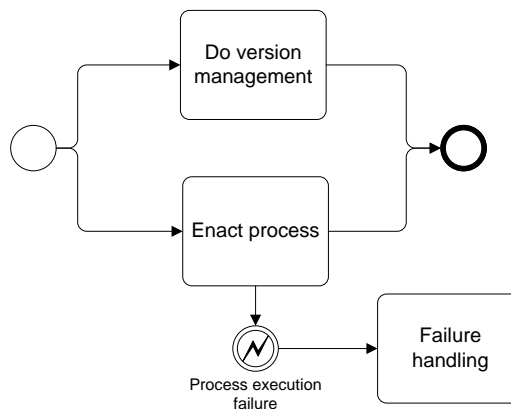
In the execution sub-phase of the operation phase, the executable business process models are executed. Every time an executable business process model is executed a process instance is created. The execution sub-phase manages this process instance and makes sure that no data is lost and that the process instance's activities are executed by the correct user, application or services.

When the execution of the process instances generates a fault, this fault is automatically solved in this phase. However, sometimes human interaction is needed to handle a fault, in which case the failure handling is transferred to the interaction sub-phase.

Finally, since multiple versions of the same business processes can be in place, version management is required.

Figure 21 shows the graphical representation of this sub-phase.

Figure 21 – Execution sub-phase



Process enactment

The BPM product's business process execution engine is responsible for the execution of executable business process models. If business processes have to be executed efficiently, then their execution has to be optimized according to some measurable criteria, such as cost, process execution time, or customer satisfaction (Leymann, Roller et al. 2002). This can be done through achieving an optimal configuration of the system, assignment of resources and dynamic adaptation of the process. (Dayal, Hsu et al. 2001) Furthermore, the structure of the business process can also be changed. This will be done when the execution of a business process proves inefficient and when analysis shows room for improvement. This however is discussed in the analysis phase in

section 3.7. Finally the computation of KPIs is done by the execution engine, which has to make sure that this also happens efficiently (Castellanos, Casati et al. 2005).

When a business process is executed, a process instance is created, possibly using initial values. This instance is executed using both data passed to the process and data generated by the process's activities. When the process instance is completed or after some user-specified time period, the process instance is deleted. During execution, log files are created that store process related information, such as values of KPIs, user and activity information, and time of execution. Processing of an activity includes selecting the activity, determining the required resources and tools, activation, and termination (Dayal, Hsu et al. 2001; Leymann, Roller et al. 2002).

During the execution, the business process execution engine has to determine which actor (human or system) has to execute a certain activity. This routing can be done in several ways (Gillot 2008):

- Based on business rules.
- Based on priorities (certain activities can have a higher priority and have to be executed before others with a lower priority).
- Based on roles (a certain activity has to be performed by a certain role).
- Based on events (when a certain event occurs, the activity has to be performed by somebody else, instead of the actor normally responsible for the activity).

When routing work to users, the possibility can arise that the intended actor is unavailable for a certain time (for example, because he is busy doing other work or on vacation). In order to avoid a long delay in the execution of an activity, a BPM product can use calendar management, allowing the actor to indicate when he is available. The BPM product can then use this information when deciding to which actor the execution of the activity is routed (Gillot 2008). Some commercial tools are available and are widely used for this purpose (for example, Microsoft Office Outlook). Integration with these tools is more likely to increase the effectiveness of calendar management.

Process-associated state should be stored in a database and the BPM product should ensure that all database access occurs within the appropriate transaction and that system failures will never cause inconsistency in a process's internal state. Should a failure occur, then the state can be loaded from the database, either manually or automatically (Pasley 2005).

A way to compare BPM products in the execution phase is to check their compliance with the workflow data patterns (Russell, ter Hofstede et al. 2004) and the workflow control-flow patterns (Russell, ter Hofstede et al. 2003).

Failure handling

If every activity of a business process is always completed successfully and would never be cancelled, modelling and executing a business process would be a simple and straightforward exercise. However, not every business process will execute as intended, therefore about 80% of the time is spent on handling exceptions and failures to ensure a reliable execution of the business process (Greenfield, Fekete et al. 2003).

Ways to handle all exceptions and failures can be modelled into the process model, however, this would cause the model to become too complex (Eder and Liebhart 1996). Therefore different techniques have been developed to master this complexity.

There are three types of failures in a workflow environment which are still relevant in a BPM environment (Eder and Liebhart 1996):

- 1) Business process execution engine failure: this leads to an abnormal termination of the business process execution. After an engine failure, the goal is to restore the latest consistent state of the system, after which the process execution can continue. A process database which stores the state of each process is useful to recover from these types of failures.
- 2) Activity failure: this failure happens inside an activity. There are three type of activity failures:
 - a) System failure: this failure is caused by an IT or application failure and causes an abnormal termination of an activity. In case of a system failure, crash recovery and forward execution has to be applied. Crash

recovery consists of removing all inconsistent execution results of activities. This can be done using a rollback mechanism. Forward execution consists of starting the process execution from the closest consistent point (Eder and Liebhart 1996). This can be done by restarting the same activity, starting an alternative activity, which should remove the effects of the interrupted activity and tries to achieve the original goal, or by manual intervention.

- b) Semantic failure (exception): designing ways to handle exceptions is already discussed in section 3.4.3.
 - c) Behavioural exception: this is the result from the improper execution order of process activities and is typical for web application, where a user can navigate through the web pages freely. An approach that features triggers, exception-aware links and recovery policies can solve these problems (Brambilla, Ceri et al. 2005). A user who tries an inappropriate step creates a trigger. When the activity is completed, the exception-aware link is executed, which sees the trigger and applies the correct recovery policy.
- 3) Communication failure: this occurs during communication between the BPM product and the activities. Middleware components can ensure good communication between components (Eder and Liebhart 1996).

Another way to compare BPM products based on their capability to handle exceptions is the compliance with the workflow exception patterns (Russell, van der Aalst et al. 2006).

Version management

Finally, BPM makes it possible to have different versions of the same business process and thus requires a version management to be in place. Three functions of version management can be identified (Gillot 2008):

- Keeping track of all versions of the business processes.
- Supporting the execution of different version of the same business process at the same time.
- Upgrading a business process and altering the affected process instances accordingly.

Table 8 – Execution criteria

	Execution criteria
1.	Support for optimized execution according to some measurable criteria.
2.	Efficiency of KPI computation.
3.	Support for process state persistency.
4.	Supported modes of routing.
5.	Support for calendar management.
6.	Ability to cope with failures. <ul style="list-style-type: none"> I. Ability to cope with business process engine failures. II. Ability to cope with activity failures. III. Ability to cope with system failures. IV. Ability to cope with semantic failures. V. Ability to cope with behavioural failures. VI. Ability to cope with communication failures.
7.	Compliance with workflow exception patterns.
8.	Compliance with workflow control-flow patterns.
9.	Compliance with workflow data patterns.
10.	Support for version management.

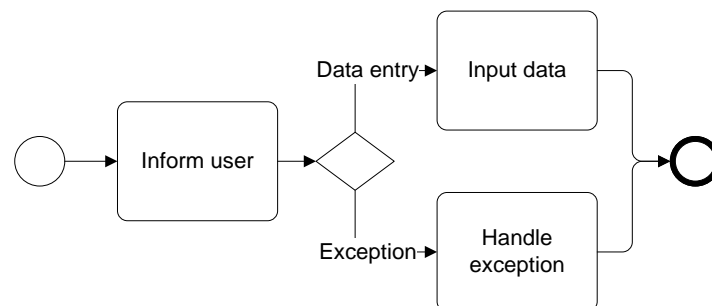
3.6.2 Interaction

Some activities cannot be done automatically, because they require human input. The interaction sub-phase is the phase where the users interact with the running business processes. First of all a user has to be informed that his interaction with the business process is required.

Depending on the situation, the user either inputs data into the business processes (for example, validating a request for a loan or enter the amount of computer he wants to order) or tries to resolve an exception that has occurred (for example, by letting somebody else execute the activity because the person who was assigned to it is on holidays).

Figure 22 shows a graphical representation of this sub-phase.

Figure 22 – Interaction phase



Informing the user

The first step in involving a user in the business processes is to inform him that his intervention is needed. Informing a user that work has to be done can be either done by push or pull technologies. In the former case the user will be informed that he needs to do work, for example, through e-mail. In the latter case the user can look up his activity list, for example, through a portal. Figure 23 shows an example of a BPM portal.

Figure 23 – Example of a BPM portal (Metastorm 2008)

The screenshot shows the Metastorm BPM portal interface. At the top, there is a navigation bar with links: Home, Documents and Lists, Create, Site Settings, and Help. Below this is the Metastorm BPM logo and the text 'Enterprise Process Advantage'. The user 'Neil Hudspeth' is logged in at 'Saturday, April 22, 2006 2:40:18 PM'. The main content area is divided into several sections:

- Events:** A list of upcoming events, including 'Marketing Team Meeting', 'Company Update', 'Echo Plus Product Launch', and 'Major World Conference'.
- Metastorm To Do List:** A table showing tasks with columns for Folder, Subject, and Priority. Tasks include 'Purchase order from Neil H...', 'Purchase order from Lucy...', and several 'Enquiry from' tasks.
- Key Performance Indicators:** A dashboard showing SLA, Delayed, and Priority 1 metrics with corresponding bar charts and values (1, 5, and 11 respectively).
- Start Process:** A section for starting new processes, with a dropdown for Group and a list of process descriptions.
- Metastorm Watch List:** A table showing tasks with columns for Folder, Subject, and Deadline. Tasks include 'Flight 004', 'Flight 005', 'Flight 006', and 'Flight 007'.
- Contacts:** A table listing contacts with columns for Last Name, First Name, Business Phone, and E-mail Address. Contacts include Blauer, Eddy, Jackson, Joiner, Krammer, Linkup, Pascal, and Smith.

Input data and handle exceptions

When the user has been informed, he can interact with the business processes through, for example, a separate application, a portal or email. Apart from normal interaction with business processes (which is nothing more than entering information through forms), human intervention may also be needed to resolve exceptions, since not all exceptions can be handled automatically. Users have to be notified that exceptions need their intervention. Synchronous and asynchronous exceptions require different mechanisms. If an exception occurs synchronously during the execution of an workflow activity, the user can be notified immediately while he is executing the affected activity; otherwise, the notification should take place outside the process flow and through the same methods that are used for informing the user of his normal work (Brambilla, Ceri et al. 2005).

Some BPM products can assist users when handling exception, by providing suggestions. The chosen solution can sometimes be saved by the users, so that the chosen solution is applied automatically when the same exception occurs (Chiu, Li et al. 1999).

When solving an exception, the execution path (the order in which the different activities are executed) of the process instance may be maintained or modified (Chiu, Li et al. 1999). Handling an exception and maintaining the execution path includes actions like:

- Changing the role or resource required for an activity.
- Changing the constraints (for example, deadlines) for an activity.
- Adding capabilities to one or more users (for example, through training).
- Adding users or resources (for example, by hiring new people or buying a new server).

Handling exceptions and modifying the execution path of the process instance include actions like:

- Putting the progress of the process instance on hold.
- Repeating an activity.
- Switching users assigned for an activity (work allocation).
- Choosing an existing alternate branch for execution.
- Skipping an activity.
- Aborting an activity or an entire process instance.

Table 9 – Interaction criteria

	Interaction criteria
1.	Support for informing the user. <ul style="list-style-type: none">I. Support for active informing.II. Support for passive informing.
2.	Support for exception handling. <ul style="list-style-type: none">I. Support for synchronous notification of exceptions.II. Support for asynchronous notification of exceptions.III. Support for providing suggestions when handling exceptions.IV. Support for saving exception handling solution.V. Support for exception handling while maintaining execution behaviour.VI. Support for exception handling while modifying execution behaviour.

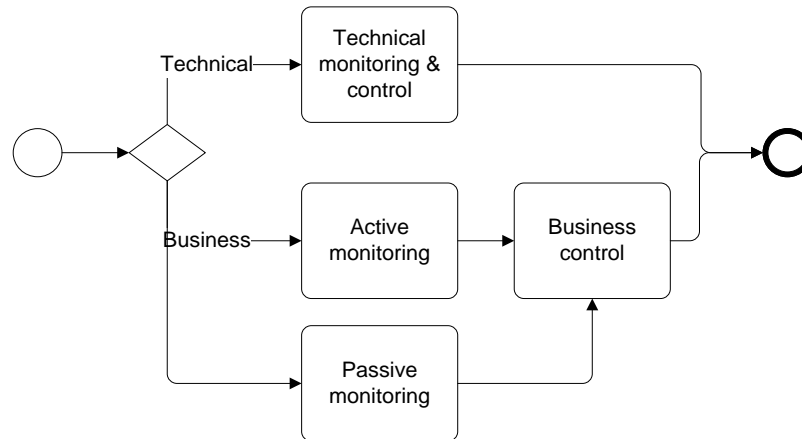
3.6.3 Monitoring and Control

The monitoring and control sub-phase is needed to maintain the health of the processes and the technical infrastructure on which execute. Monitoring and control can be either done from a technical perspective or from a business perspective. The former is necessary to identify and solve the technical causes of abnormal operations of the business processes. The latter deals with the business processes themselves and the business problems that

occur when executing them (zur Muehlen 2001). Business-related monitoring can be either done using push (active monitoring) or pull (passive monitoring) technology.

Figure 24 shows the graphical representation of this sub-phase.

Figure 24 – Monitoring and Control phase



Technical monitoring and control

Technical monitoring and control deals with failures like servers that stop working, problems with network connections or inaccessible data (Gillot 2008). Furthermore, it deals with parameters like response times and system load. Technical monitoring in a BPM product does not differ from monitoring complex application systems (zur Muehlen 2001). Therefore, we have focussed on monitoring and control from a business perspective.

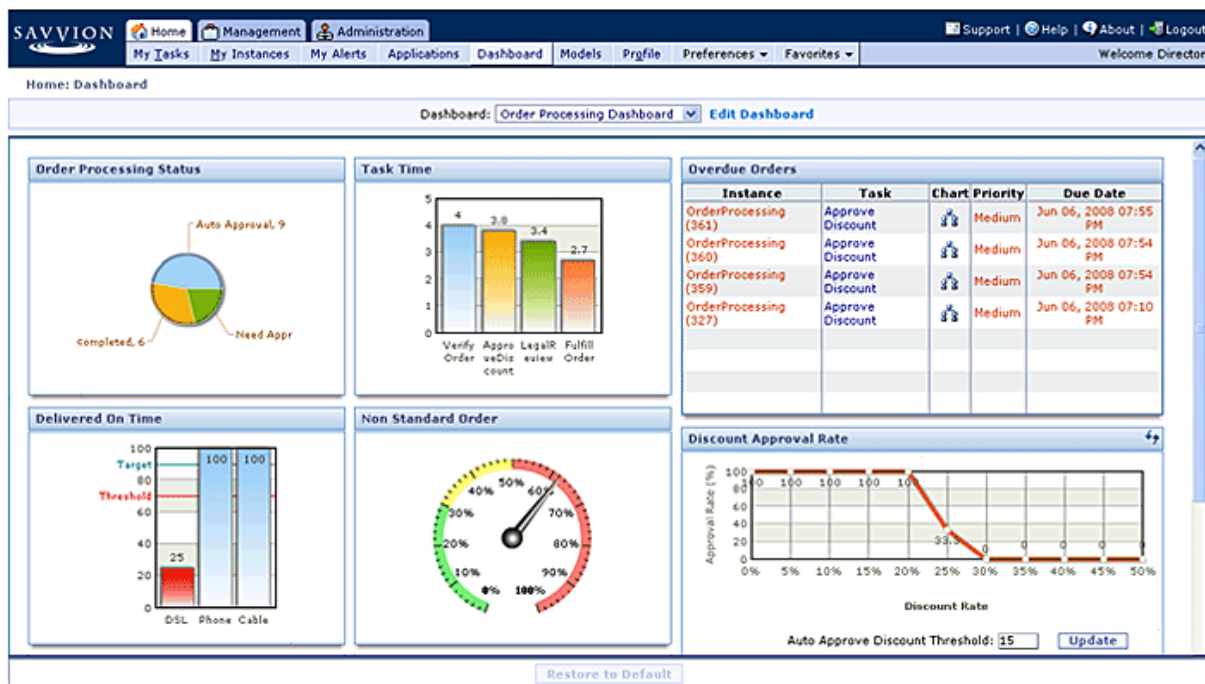
Business monitoring

Monitoring running process instances from a business perspective can be done actively by using push technology or passively using pull technology (zur Muehlen and Rosemann 2000). Active monitoring is done using Business Activity Monitoring, which allows the (near) real-time monitoring of process instances (Golfarelli, Rizzi et al. 2004). This is done by monitoring the log files of the BPM product, gathering performance metrics like their processing time and calculating the KPIs of the business processes. It applies techniques like statistical process control, which requires the definition of an upper and lower bound for the KPIs. These bounds are either manually entered into the system or derived from historical runtime information (Gillot 2008). When these bounds are crossed, the BPM product notifies users or solves the problem itself. This technique can either be used for management-by-exception (trying to solve the problem when it occurs) or to predict and avoid undesirable situations. When predicting situations, the process instance is evaluated and compared to historical data. This allows the BPM product to predict if certain unwanted behaviour is likely to occur (Grigori, Casati et al. 2004).

Passive monitoring, on the other hand, includes requesting information about running process instances, for example, to check the status of an order to answer an inquiry by a customer.

Monitoring information is better consumed when it is provided in various levels of detail, depending on the interest of the target information recipient and the quality/quantity of information provided. Some users only want high-level information, but others might want to drill-down to more detailed information (Leymann, Roller et al. 2002). Furthermore, different views help to hide the complexity (zur Muehlen and Rosemann 2000). Relevant views include the overall view of the system, a process view with information on a specific process instance or a group of process instances, a resource view with information of individual or groups of resources (users or automated resources) and a service view with information related to services invoked within the execution of the business processes (zur Muehlen 2001; Grigori, Casati et al. 2004). Monitoring information is typically presented in a monitoring dashboard such as the one shown in Figure 25.

Figure 25 – Example of a monitoring dashboard (Savvion 2008)



Business control

Based on the monitoring information provided, users may want to control the process instances by adjusting them. These changes can be applied to a single process instance (in which case the techniques used resemble those of exception handling) or multiple instances. While business control may resemble remodelling or redesign, it differs from those because the changes only affect the currently running process instances and not the (executable) business process model. Different methods can be applied (Chiu, Li et al. 1999; zur Muehlen 2001):

- 1) Process instance evolution, which includes:
 - a) Changing the role or resource required for an activity.
 - b) Changing capabilities or roles of users.
 - c) Changing the organizational structure.
 - d) Changing pre-conditions by adding an extra check or adding pre-requisites.
 - e) Eliminating problematic branches and adding new branches.
 - f) Adding or removing activities.
- 2) Adding explicit procedural exception handlers that perform automatic exception handling, which includes:
 - a) Procedural exception handlers before an activity. These exception handlers consist of added automatic activities which are executed when a pre-condition is met.
 - b) Re-execution patterns. These patterns automatically skip or repeat an activity, choose another existing branch for execution, or abort the current activity.
 - c) Compensation handlers.
- 3) Changing business rules.
- 4) Changing the structure of activities.
- 5) Changing the workload balance between users.

Table 10 – Monitoring and Control criteria

	Monitoring and Control criteria
1.	Support for technical monitoring and control.
2.	Support for business monitoring. <ol style="list-style-type: none"> I. Support for active monitoring. II. Support for passive monitoring. III. Support for various level of detail. IV. Support for different views of monitoring information.
3.	Support for business control. <ol style="list-style-type: none"> I. Support for prediction of unwanted situations. II. Support for process instance evolution. III. Support for explicit procedural exception handlers. IV. Support for changing business rules. V. Support for changing activities. VI. Support for changing the workload balance between users.

3.7 Analysis

Analysis techniques can be used in different phases: during the modelling phase on business process models, during the design phase on executable business process models or after the execution on process logs.

The techniques available in the analysis phase differ slightly depending on the phase in which they are used, but still most of the techniques can be applied on both models and the process logs. These techniques include verification, simulation and quantification techniques.

Verification is used to check the correctness of the modelling of the (executable) business process model against the semantics of the language used. This can only be done if the (executable) business process model is modelled in a business process modelling/execution language with a clear semantics (Grigori, Casati et al. 2004; Weske, van der Aalst et al. 2004).

Simulation can be used to assert if a (executable) business process model behaves as expected and to evaluate the ability to meet certain requirements with respect to throughput times, service levels and resource utilization. For example, when different versions of the same process have been modelled, they can be simulated before taken into production to choose between them. Simulation tools can help determine the costs of individual activities or even of the total process itself. Further they can be used to calculate the minimum and maximum processing times (Leymann, Roller et al. 2002; van der Aalst, Leymann et al. 2007). Simulation is useful since it is difficult to predict the actual workload of the business process model, and thus difficult to define business process aspects (such as assignment to resources) that are affected by it (Jin, Casati et al. 2001).

When simulating business processes, a number of cases are fed to the (executable) business process model and the environment is also simulated to match the real environment. The simulated environment, which includes resources, services and their loads, can be set based on the experience of the user or based on historical execution information. Traditional business process simulation tools only allow for user-defined parameters, while more innovative tools can also simulate based on historical data. Innovative simulation tools are preferred above traditional simulation tools because they take into account (Jin, Casati et al. 2001):

- the load on resources and services (possibly caused by executions of different processes);
- the load on the execution environment of the BPM product;
- the impact changes in the process have on other business processes;
- historical data, therefore not relying on the user's guess;
- deviation from the average of execution times, the behaviour of the execution environment and resources.

This last step in analysis is the analysis of historical data, stored as a result of the execution of the business processes. This analysis is also referred to as Business Intelligence (BI). It aims to detect long-term developments and identify opportunities for optimization in the business process models and in the assignment of resources (Grigori, Casati et al. 2004). This historical data contains information from the process instance log files which contain data about the execution of the process instance (for example, the start time of an activity, the completion times of activities, input and output data, messages received and the resource used) (Sayal, Casati et al. 2002). Other historical data can include computed KPI values, application data (managed and stored by the applications that are invoked during the execution of the business processes, for example, a letter to a customer that is managed by the word processing system) and the status of the IT infrastructure (for example, the load on a certain server) (zur Muehlen 2001; Castellanos, Casati et al. 2005). This historical data is typically stored in a data warehouse in order to preserve information as long as necessary and to support analysis and data mining techniques like OLAP (Dayal, Hsu et al. 2001; List, Schiefer et al. 2002).

When analyzing a process, a quantification of the process is necessary to identify extraneous resources or possible targets for improvement (Elzinga, Horak et al. 1995). Information on resource requirements, cost, time and value should be obtained. This can be represented using simple visual data representation tools such as cost-time profiles, pie charts, histograms, Pareto charts, and bar graphs which can be used to facilitate quantified understanding of the process (Elzinga, Horak et al. 1995).

Complex methods for quantification, like traditional accounting or activity-based costing (ABC) can also be used. Especially ABC has been used to identify processes that can be optimized with BPM (Johnson 1988). When using ABC, the cost of each activity has to be added to the business process model in the modelling phase.

Computed KPIs can be analyzed to check for deviations. The models provided to users for analyzing the critical factors affecting KPI behaviours should be easily interpretable (Castellanos, Casati et al. 2005). Classifying process instances in a user-defined way, depending on their characteristics facilitates the analysis of the historical data (Sayal, Casati et al. 2002). This can be done by using taxonomies with different categories (for example, the taxonomy *negotiation outcome*, with the categories *success* and *failure*).

Other analysis techniques include the identification of bottlenecks in a business process model (van der Aalst, Leymann et al. 2007). This can be done by analyzing the resources used by extremely fast or slow process instances of a certain business process (zur Muehlen 2001).

All these analysis techniques can be used to discover room for improvement. Most of these techniques need a human to analyse the result, but some BPM products might offer suggestions concerning possible improvements.

Table 11 – Analysis criteria

	Analysis criteria
1.	Support for modelling-time testing. I. Support for verification. II. Support for verifying compliance with rules. III. Support for traditional simulation. IV. Support for innovative simulation.
2.	Supported types of historical data available for analysis (execution logs, computed KPIs, application data, status of the IT infrastructure).
3.	Support for simple visual data representation tools (cost-time profiles, pie charts, histograms, Pareto charts, and bar graphs).
4.	Support for complex methods of quantification (traditional accounting or activity-based costing).
5.	Support for defining processes in taxonomies.
6.	Support for suggestions on improvement.

4 Definition of the Method

The Business Process Management framework from chapter 3 provided the basis for our evaluation method. Due to time constraints, our evaluation method is not based on all the criteria from the framework, but on a selection of these criteria. Next to the criteria, our evaluation method also includes a case that is complex enough to test all our selected criteria, information on how to use the selected criteria to evaluate BPM products, information on how to rate each criteria and how to add up all points to get an overall rating of the BPM products.

This chapter starts with section 4.1 in which the criteria selection is described. In section 4.2 presents the case that is part of our evaluation method. Section 4.3 describes how to use the selected criteria when evaluating BPM products. The rating method is described in section 4.4 and finally section 4.5 presents the BPM products that we have used to test the method.

4.1 Criteria Selection

Due to time constraints we had to limit the evaluation method to four phases. We have chosen for the modelling, design, interaction and monitoring and control phases. The choice for these phases is based on the fact that they can be tested in a relatively short amount of time and without having a comprehensive infrastructure in place. However, we have dropped some criteria of these phases from our testing, because they did require an infrastructure to be in place or more background knowledge. The dropped criteria are:

- Modelling criterion #1: support for harvesting and reuse. Dropped because this criterion requires reusable business process models to be available in a business process repository.
- Modelling criterion #3.I: flexibility of KPI model. Not tested, since this criterion requires a complex business process in which many different kinds of KPIs can be defined.
- Design criterion #2: support for adding SLA and linking SLA KPIs to business processes. Dropped because this criterion requires deeper understanding of SLAs.
- Design criterion #3: supported exception handling methods. Dropped because this criterion requires extensive knowledge on exception handling.
- Design criterion #4: support for integration. Dropped because this criterion requires an infrastructure to be in place and knowledge on how to integrate this infrastructure with the BPM suite.
- Interaction criterion #2: support for exception handling. Dropped because this criterion requires extensive knowledge on exception handling.
- Monitoring and Control criterion #3.I: support for prediction of unwanted situations. Dropped because this criterion requires that many instances have been executed before the capability can work.
- Monitoring and Control criterion #3.IV: support for explicit procedural exception handlers. Dropped because this criterion requires more extensive knowledge of procedural exception handlers.

4.2 Case Description

Based on the selected criteria we have defined a case which can be implemented using a BPM product. This case is based on the business process in which a customer applies for a mobile phone subscription. This business process is described below.

“A customer applies for a mobile phone subscription through an application form. In this form he enters his name (first and family name), address (street, number, city and postal code), date of birth (day, month and year) and the type of mobile phone subscription he wants (price per month and length of subscription).

Based on the information the customer enters in his application form, the risk is determined automatically by a risk service. This service classifies the application either as low risk or as high risk.

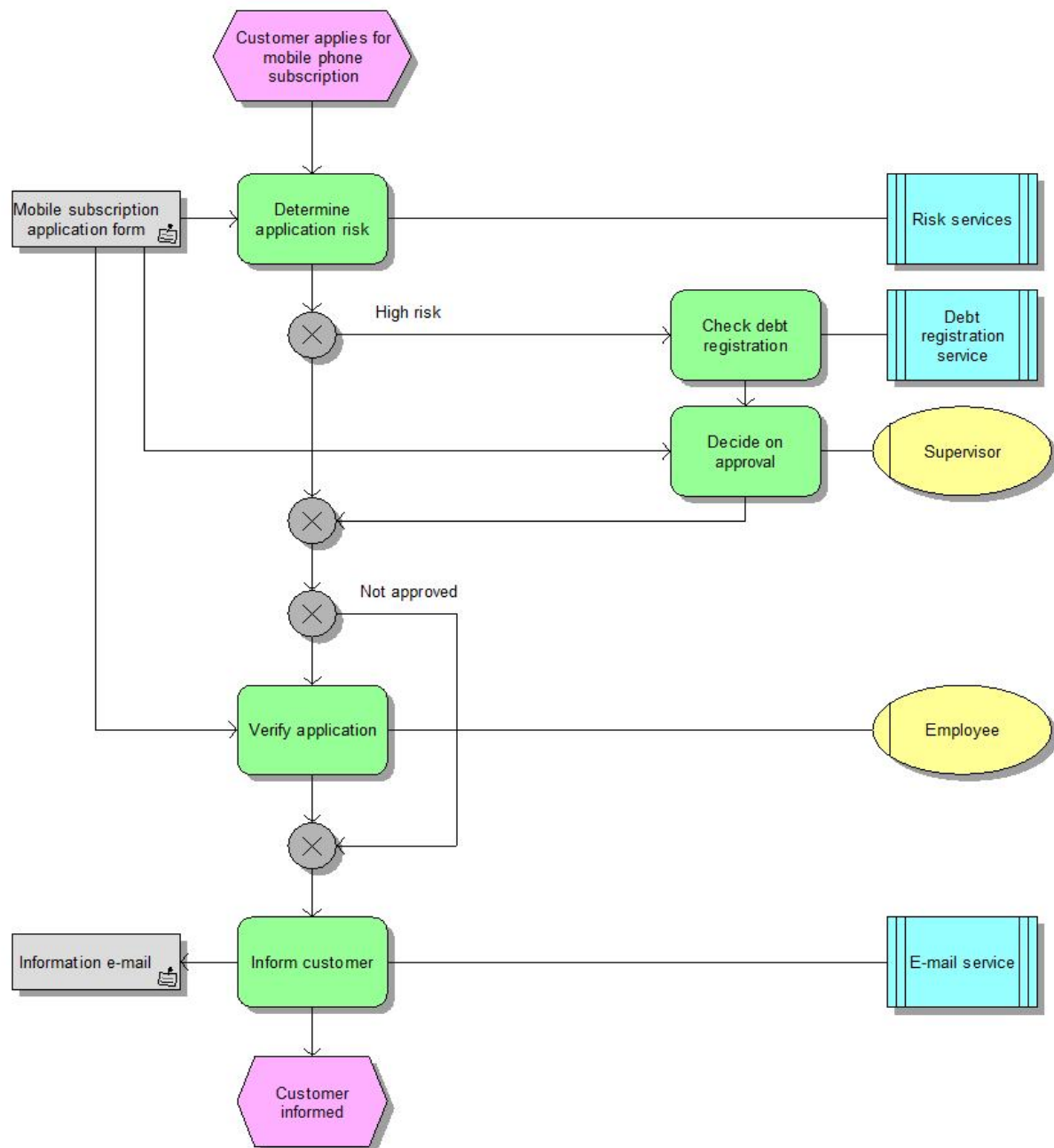
If an application is classified as high risk, the customer is checked for any debts, and based on the result of this check, a supervisor decides to approve the application or not. If the application is of low risk it is approved automatically.

Approved applications are verified by an employee. If the data in the application form is correct, the application is verified.

The customer is informed by e-mail of the acceptance or rejection of his application. The contents of this e-mail message are determined by the acceptance or rejection of his application. When an application is rejected the reason for this rejection is given in the e-mail (for example, no person with the given name lives at the provided address or the applicant has too much debt)."

Figure 26 shows an event-driven process chain (EPC) of the business process described above.

Figure 26 – EPC diagram of the mobile subscription application business process



4.3 Using the Selected Criteria

The sections below describe how the selected criteria have to be applied to a BPM product using the presented case.

4.3.1 Business process modelling

The business process described by our case has to be modelled in BPM products with all the business process modelling language available (modelling criterion #2.I). To check the interoperability between the provided business process modelling languages, the models have to be translated from one language to another (modelling criterion #2.II).

The business process not only has to be modelled in the BPM products, but a model of this process also has to be imported. The EPC diagram in Figure 26 is modelled into IDS Scheer's ARIS and, if possible, this model has been imported into the BPM products (modelling criterion #2.III).

To evaluate the expressiveness of the modelling tools, the BPM product has to be checked for the four modelling views: functional, behavioural, organizational and informational (modelling criterion #2.IV). Furthermore, the BPM products have to be checked for providing different perspectives of the business process (modelling criterion #2.V).

Our business process contains a business rule (determining if the risk of the application is low or high). This business rule first has to be modelled implicitly by categorizing all applications from a customer below the age of 18 as high risk (modelling criterion #2.VI). Later on, this business rule has to be modelled explicitly (modelling criterion #2.VII) and even the entire business process has been represented by business rules, if possible (modelling criterion #2.VIII). Also, the ease of use of defining business rules has to be evaluated by looking how they have been defined: in (almost) natural language, graphical notations, code with the help of templates or (programming) code (modelling criterion #2.IX).

4.3.2 Key-performance indicators

Two KPIs have been defined for our business process: the amount of accepted application in comparison to the total amount of applications; the amount of high risk applications in comparison to the total amount of applications. In order to evaluate the flexibility of the KPI model these two KPIs have to be modelled. The ease of use for defining KPIs has to be evaluated by looking at the way they have been defined: in (almost) natural language, graphical notations, code with the help of templates or (programming) code (modelling criterion #3.II).

4.3.3 Executable business process models

The business process model has to be translated into an executable business process model. To evaluate this capability, all the supported business process execution languages have to be indexed (design criterion #1.II). Furthermore, the way this translation is supported has to be evaluated, which can be either manual or automatic translation, or by using one model in both the modelling and design phases (design criteria #1.II and #1.III).

Finally, to check the traceability, an extra activity in the executable business process model has to be added: after the customer has been informed of his confirmed application, this step sends the application data to a third role, the second-line employee. This capability has to be evaluated by checking if the business process model is (automatically) changed (design criterion #1.IV).

4.3.4 Programming languages

In order to evaluate the support to programming languages for implementing services and applications, a simple debt service (which, for example returns a random number ranging for -1000 to 0) has to be implemented and the available programming languages have been indexed (design criterion #5).

4.3.5 Designing user interfaces

To evaluate the support for designing user interfaces, forms for a customer to fill in the application form, for a supervisor to approve the application and for an employee to verify the application have to be designed, and attention has to be paid how this is done (manually or automatically generated). Also, the functionality to use a separate application for data entry has to be considered (design criterion #6.I) and a portal has to be implemented (design criterion #6.II) with an activity list (design criterion #6.III), if available.

4.3.6 User management

Three types of users have to be managed in our case: customers, supervisors and employees. The way these roles are described in the BPM products has to be evaluated (design criterion #7.I). Also, the ability to make a distinction between an organizational role (based on the organizational structure) and other (user-defined) roles has to be analyzed (design criterion #7.II).

Since many organizations already have models of their organizational structure, it has to be checked if the functionality to import this structure from another system is available in the BPM products (design criterion #7.III).

The support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities) has to be evaluated by, if possible, modelling an employee who speaks Dutch, with the intention of letting him handle the applications that come from the Netherlands (design criterion #7.IV). Assigning a role to a user can be either done manually or automatically. Both ways have to be evaluated (design criteria #7.V and #7.VI).

To evaluate the support for activity permissions based on roles, the supervisor role has to receive the permission to skip the verification of the application (design criterion #7.VII).

4.3.7 Informing the user

The support for active informing the user has to be evaluated by having the supervisor be informed by e-mail if an application has to be approved (interaction criterion #1.I). The support for passive informing the user has to be evaluated by adding an activity list to the portal, in which the user can look which work he has to do (interaction criterion #1.II).

4.3.8 Technical monitoring and control

The BPM products have to be checked for their technical monitoring and control capabilities (Monitoring and Control criterion #1).

4.3.9 Business monitoring

The support for active monitoring has to be evaluated by looking for Business Activity Monitoring capabilities (Monitoring and Control criterion #2.I).

Customers want to know the status of their application and contact the company. A supervisor then has to be able to retrieve the status of the application (passive monitoring) (Monitoring and Control criterion #2.II).

To evaluate the effectiveness of business monitoring, the BPM products have to be checked for support of various level of detail by providing the possibility to drill-down into, for example, the applications which were high risk (Monitoring and Control criterion #2.III). Also it has to be checked which of the four views of monitoring have been supported: the overall view of the system, the process view, the resource view and the service view (Monitoring and Control criterion #2.IV).

4.3.10 Business control

To evaluate the support for process instance evolution, the process instances have to be edited to let the supervisor verify the application, instead of the employee. Furthermore, a new activity has to be added to the process instances (Monitoring and Control criterion #3.II).

The support for changing business rules has to be evaluated by changing the age below which an application is marked as high risk from 18 to 16 (Monitoring and Control criterion #3.IV).

The “inform customer” activity has to be changed to evaluate the support for changing activities (Monitoring and Control criterion #3.V).

To evaluate the support for changing the workload balance between users, one employee has to be assigned to do all the verification (Monitoring and Control criterion #3.VI).

4.4 Rating the Products

In order to make a quick comparison between the different BPM products, we have devised a rating method. Because giving one single mark for the total BPM product is difficult, we have decided to assign a rating to each of the evaluated phases of the BPM lifecycle. This allows us to quickly compare the capabilities of the products.

The rating of each phase is based on a scale from zero to ten, with ten being the best. Each criterion is awarded a certain amount of points depending on the evaluation. All the points from each criterion belonging to a certain phase are added up, and then divided by the maximum amount of points that can be scored for the phase and multiplied by ten to get the score for that phase. Table 12 shows how the selected criteria are rated.

Table 12 – Selected criteria and the method of rating

	Modelling criteria (max points: 95)	Way of rating
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	Support for only one business process modelling language will result in six points (eight if this language is BPMN since this is a standard language). Each extra supported language will earn one extra point (up to a max of a ten total) if the interoperability between the languages works always.
	II. Support for interoperability between different business process modelling languages.	
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	Import of the functional view of the ARIS model will be awarded with six points (two extra points if the other information is imported too). Importing from Microsoft Visio earns another two points.
	IV. Support for different views when modelling a business process.	Support of each single view will be rewarded with two and a half points.
	V. Support for different perspectives of a business process.	Each different perspective gets the suite one point (four more if the perspective is useful) with a max of ten points total.
	VI. Support for modelling with implicit business rules.	Support available earns ten points, no support earns zero points.
	VII. Support for modelling with explicit business rules.	Support available earns ten points, no support earns zero points.
	VIII. Support for modelling business processes using explicit business rules.	Support available earns five points, no support earns zero points.
2.	IX. Ease of use of defining explicit business rules.	Definition of explicit business rules using (almost) natural language will be awarded seven points, using graphical notation five points and using code three points. When templates are available an extra three points are added.
	Support for KPIs.	For each of the KPIs we have tried to define, five points are rewarded if they were defined successfully. Defining KPIs using (almost) natural language will be awarded seven points, using graphical notation five points and using code three points. When templates are available an extra three points are added.
	Design criteria (max points: 140)	
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	Support for only one business process execution language will result in six points (eight if this language is BPEL since this is a standard language). Each extra supported language will earn one extra point (up to a max of a ten total) if the interoperability between the languages works always.
	II. Support for manual translation between business process modelling languages and business process execution languages.	For each combination of the supported business process modelling languages and the business process execution languages, zero points are given if the translation is only manual and a ten is automatic translation is available. All these ratings

	III.	Support for automatic translation between business process modelling languages and business process execution languages.	are added up and divided by the total amount of combinations.
	IV.	Support for linking the business process execution model with the business process model.	<p>If all synchronization between the models is manual, the product gets zero points.</p> <p>If changes from business process model to executable business process model happen automatically, the product gets five points. The product gets eight points if changes both ways happen automatically.</p> <p>When the business process model is also executable (and no linking is needed) the product gets ten points.</p>
4.		Supported programming languages for implementing services/applications.	Each supported programming language from the top ten of the TIOBE Programming Community Index (TIOBE Software BV 2009) earns one point.
5.		Support for designing user interfaces.	Automatic generation of a form to input data earns a product ten points, manually defining a form only seven. No support earns zero points.
	I.	Support for separate applications for data entry.	Support available earns ten points, no support earns zero points.
	II.	Support for portal technology.	Support available earns ten points, no support earns zero points.
	III.	Support for an activity list.	Support for an activity list inside the portal gets seven points. Three extra points are added if the activity list can be reused in a separate application or a third-party portal. No support earns zero points.
6.		Support for user management.	
	I.	Way of describing roles (by names/description or by capabilities).	Description of roles by only names or only capabilities earns five points. Ten points are awarded if both ways of description can be used.
	II.	Support for determining user capabilities based on organizational role.	Supporting only roles based on the organizational structure earns seven points. Three extra points are added if user-defined roles can be added as well.
	III.	Support for importing organizational structure from other systems.	Support available earns ten points, no support earns zero points.
	IV.	Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	Support available earns ten points, no support earns zero points.
	V.	Support for manually assigning a user to a role.	Automatically assigning a user to a role earns a product ten points, manually assigning a user only seven. No roles earn zero points.
	VI.	Support for automatically assigning a user to a role, based on his capabilities.	
	VII.	Support for activity permissions based on roles.	Support available earns ten points, no support earns zero points.

Interaction criteria (max points: 20)		
7.	Support for informing the user.	
	I. Support for active informing.	If a user can be informed by email, the product gets eight points. An extra point is added for each other way of active informing (with a max of ten points total).
	II. Support for passive informing.	Support available earns ten points, no support earns zero points.
Monitoring and Control criteria (max points: 90)		
8.	Support for technical monitoring and control.	Technical monitoring and control inside the suite earns the product ten points. Five points are awarded if the technical monitoring and control is done by a component that is not part of the BPM suite, but can be provided by the vendor. No support earns zero points.
9.	Support for business monitoring.	
	I. Support for active monitoring.	Eight points will be awarded for a fully integrated BAM solution in the BPM suite. An extra point is added for each extra way to inform the user (up to a total of ten).
	II. Support for passive monitoring.	If it is possible to retrieve the status of a process instance six points are given. Four extra points are added if the process instance can be retrieved using user-defined variable (for example, the name of the customer).
	III. Support for various level of detail.	Support available earns ten points, no support earns zero points.
	IV. Support for different views of monitoring information.	Availability of each of the four views will receive two and a half points.
10.	Support for business control.	
	I. Support for process instance evolution.	If the running process instance can be changed, but the change only applies to new process instances, four points are given. If the change is forced on all running process instances, four points are given. Two points extra are added if the designer can choose between the two options.
	II. Support for changing business rules.	Support for changing business rules during runtime receives seven points. Three extra points are added if there is an option to re-evaluate running process instances based on the changed business rule.
	III. Support for changing activities.	If the running process instance can be changed, but the change only applies to new process instances, four points are given. If the change is forced on all running process instances, four points are given. Two points extra are added if the designer can choose between the two options.
	IV. Support for changing the workload balance between users.	Ten points are awarded if the workload balance can be changed without changing the business process model. Only four points are awarded if the business process model has to be changed.

4.5 Selected Products for Testing the Method

Especially the relationship between the modelling and design phase is an interesting topic, since different vendors use different approaches to connect these phases. Some offer automatic translation, while others only offer manual translation. Even others may choose to integrate the business process model with the executable

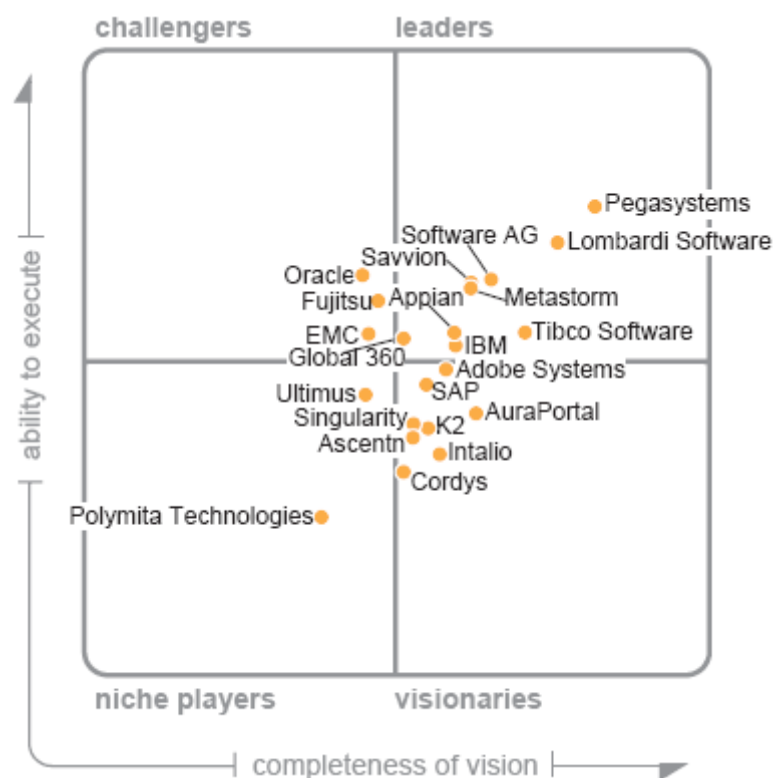
business process model into one model. This relationship between the two phases is a topic for BPM suites which cover the entire BPM lifecycle and not for BPM tools which only support either modelling or design. Therefore we have chosen to use BPM suites in the testing of the method, as opposed to using BPM tools.

The BPM suites we have chosen for testing our method are the following:

- Oracle's BPM Suite
- Cordys BPMS
- IBM WebSphere BPM

We have chosen Oracle since this was a requirement put forward by the company where our research has been performed. Since Oracle is viewed as a challenger in the BPM suites market, as can be seen in Figure 27, we have selected the two other vendors based on their position in one of the other quadrants: Cordys is seen as a visionary and IBM as a leader.

Figure 27 – Gartner's Magic Quadrant for BPM suites (Hill, Cantara et al. 2009)



5 Oracle BPM Suite

This chapter discusses the evaluation of the Oracle BPM Suite. Section 5.1 gives a short description of the products of the Oracle BPM Suite. Section 5.2 till section 5.11 present the evaluation of the Oracle BPM Suite divided by our ten selected criteria. Section 5.12 gives the summary of this evaluation and presents the rating of the suite.

5.1 Suite Description

Oracle traditionally is a big player in the database market, but since the acquisition of BEA in 2008, they are aiming to acquire a bigger market share in the BPM market. They want to achieve this with their Oracle BPM Suite, which features some of the former BEA products. The Oracle BPM Suite consists of five Oracle products (Oracle 2008):

1. Oracle BPM: a set of tools for creating, executing, and optimizing business processes, aimed at business users. These tools consist of BPM Studio (used for modelling, design and some analysis), BPM Enterprise (used for execution) and BPM Workspace (used for interaction, monitoring and control and analysis). These tools aim at business people who do the modelling, interaction, monitoring and control and analysis, and at the technical users who take care of the design, execution and analysis.
2. Oracle BPEL Process Manager: enables the creation of a service orchestration using BPEL. It aims at the technical user.
3. Oracle Business Activity Monitoring: a solution for building interactive, real-time dashboards and proactive alerts for monitoring business processes and services. This product aims at business executives and operation managers.
4. Oracle Business Rules: Oracle's solution for explicitly using business rules and is aimed at business analysts.
5. Oracle WebCenter Suite: a user interaction and portal platform. It is aimed at the users who interact with the business processes.

Figure 28 maps the tools from the Oracle BPM Suite to the phases of our BPM lifecycle, based on the product documentation (Oracle 2008).

Figure 28 – Tools from the Oracle BPM Suite mapped to our BPM lifecycle

	Strategy Development	Discovery	Modelling	Design	Deployment	Execution	Interaction	Monitoring & control	Analysis
BPM Studio									
BPM Enterprise									
BPM Workspace									
BPEL Process Manager									
BAM									
Business Rules									
Webcenter Suite									

In the Oracle BPM Suite the Oracle BPM products are leading and the others can be seen as extensions. Most of our evaluation focuses on Oracle BPM, since it covers all phases of our evaluation.

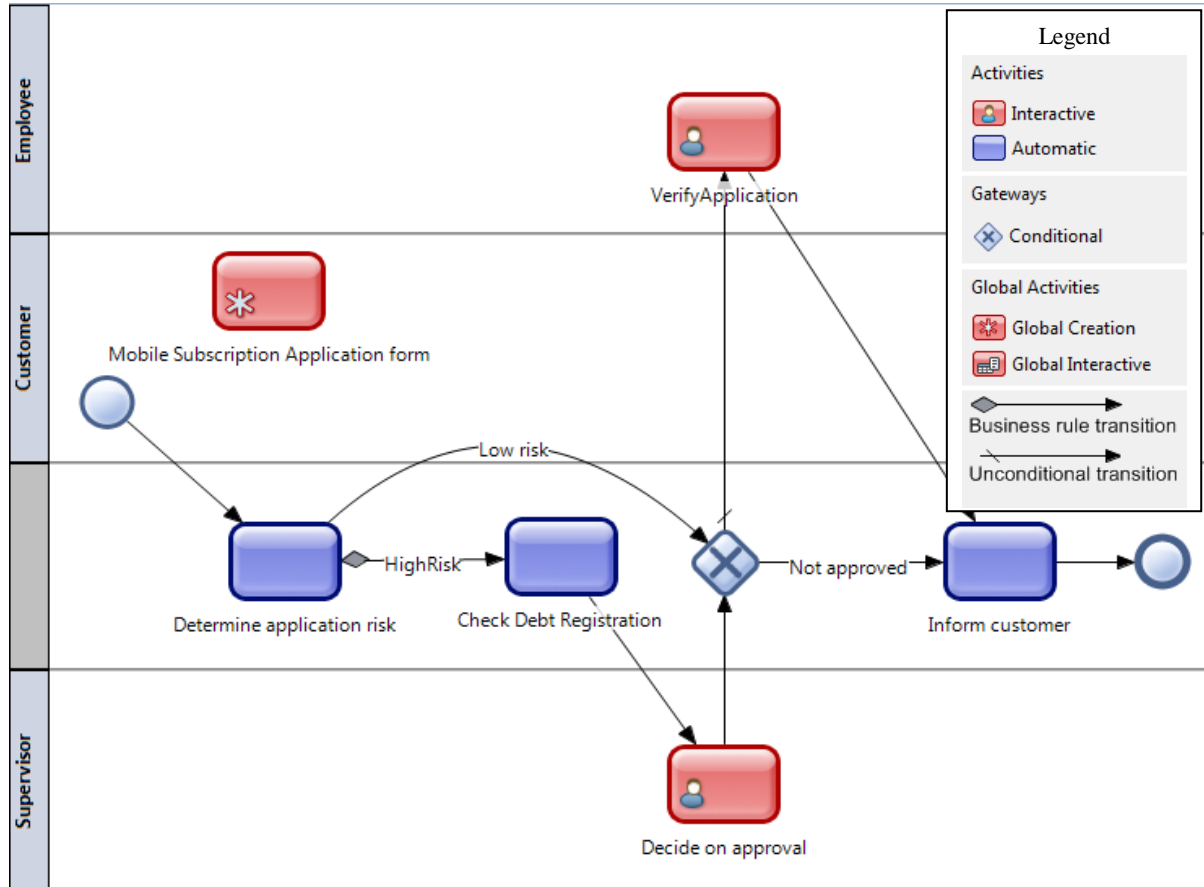
For our evaluation, we have used the following versions of the products:

- Oracle BPM Studio 10gR3.
- Oracle BPM Enterprise 10gR3.
- Oracle BPM Workspace 10gR3.
- Oracle BPEL Process Manager 10.1.3.1.

5.2 Business Process Modelling

We have used Oracle BPM Studio for the modelling phase. Figure 29 shows the business process modelled in Oracle BPM Studio. The “Mobile Subscription Application form” global creation activity initiates this business process model, which means that the business process is started when a customer fills in this form.

Figure 29 – Business process modelled in Oracle BPM Studio



The only supported business process modelling language is the Business Process Modeling Notation.

The EPC diagram which we had modelled in ARIS could be exported into a XML Process Definition Language (XPDL) model. This model can be imported into Oracle BPM Studio, which allows import to different models: XPDL, Microsoft Visio XML Drawings (vdx) and Oracle Procedure/Screenflow files (xsl). However, only the functional view of the business process was imported into the BPM Studio, so we had to model the different actors and the “mobile subscription application form” again. Figure 30 shows the result of importing to the XPDL model. The names are all mixed up because Oracle BPM Studio uses the identifier code of the activity, used in the internal identification schema of the ARIS model, as their names, instead of the activities’ names which were defined in the ARIS model.

Figure 30 – Imported business process

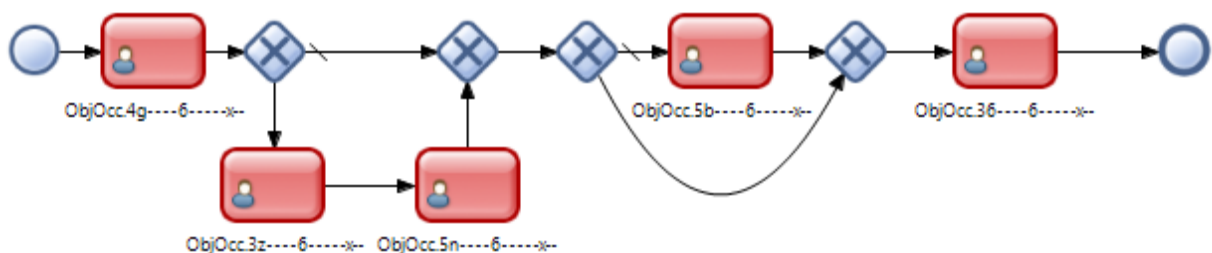


Figure 29 shows the functional view of the business process, with the activities that are performed and the relevant flows. It also shows the roles that execute the activities, which is a part of the organizational view. Figure 31 and Figure 32 respectively show the behavioural and informational view of the business process. However, the behavioural view lacks the timing of the execution of the activities, and the informational view lacks the relationship and the flow of information entities, such as the application form, and only shows their structure.

Figure 31 – Behavioural view of the business process

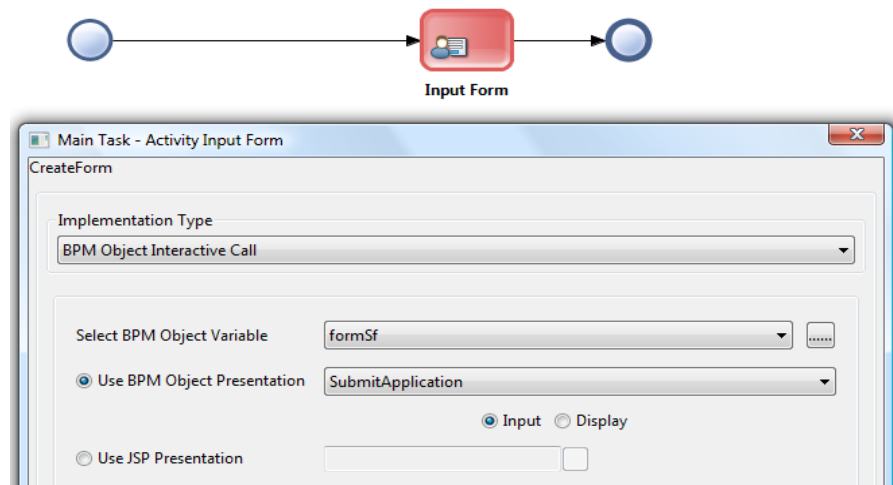


Figure 32 – Informational view of the business process

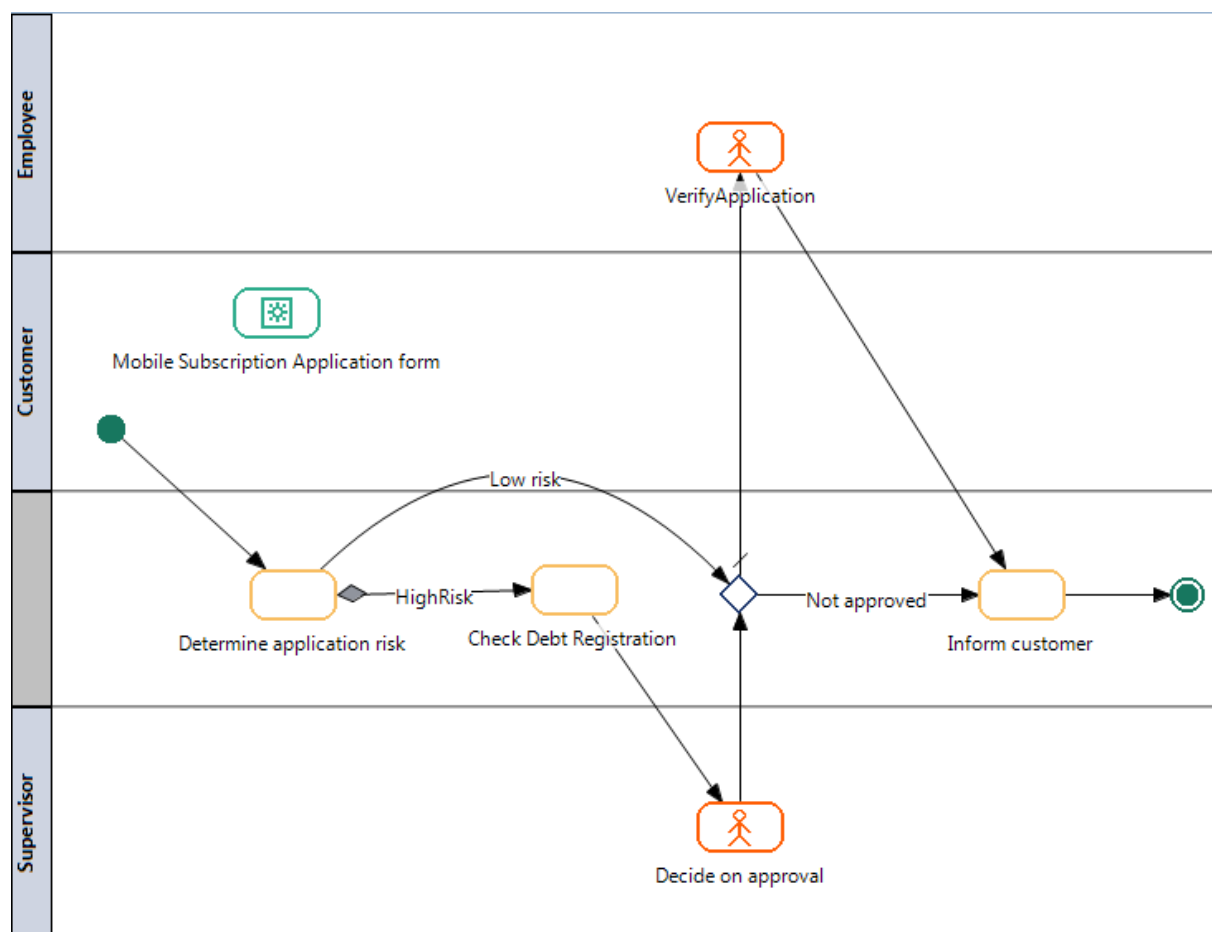
Description: ☐ Runs on Client

Default Presentation Name:

Name	?	Virtual	Type	Not Null	Default Value
firstname	No	No	String	No	null
lastname	No	No	String	No	null
age	No	No	Int	No	1
subscriptiontime	No	No	Int	No	6
applicationtype	No	No	Decimal	No	10.0
isApproved	No	No	Bool	No	null
isVerified	No	No	Bool	No	null
email	No	No	String	No	null

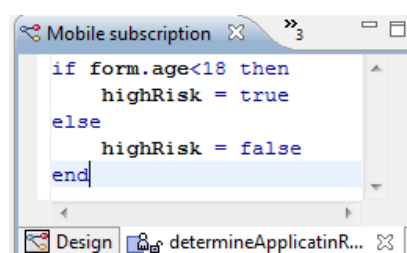
Oracle BPM Studio can show the BPMN business process in an UML notation (Figure 33). However, this notation does not differ very much from the BPMN notation. Probably this graphical representation has the same target audience as the BPMN notation.

Figure 33 – Business process in UML notation



We have modelled the business rule which determines if an application is either low or high risk implicitly using Oracle BPM Studio. This was done by adding some simple code to the “Determine application risk” activity (Figure 34).

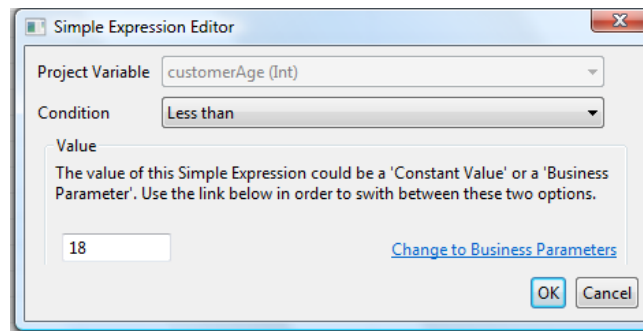
Figure 34 – Implicit business rules in Oracle BPM Studio



We modelled the previous business rule, which was hard-coded into the business process model (Figure 34), explicitly using Oracle BPM Studio. This was done using a simple expression editor (Figure 35) in which we were able to define the condition which had to be met in order to classify an application as high risk. Adding an editing activity to a role allows that role to modify the business rule at runtime. In Oracle’s BPM Suite the business rules can be designed with the help of templates (Figure 35), but can also be defined by writing code (Figure 34).

We tried to model the entire business process using business rules, but found this to be impossible, since business rules cannot be linked to each other in the Oracle BPM Suite.

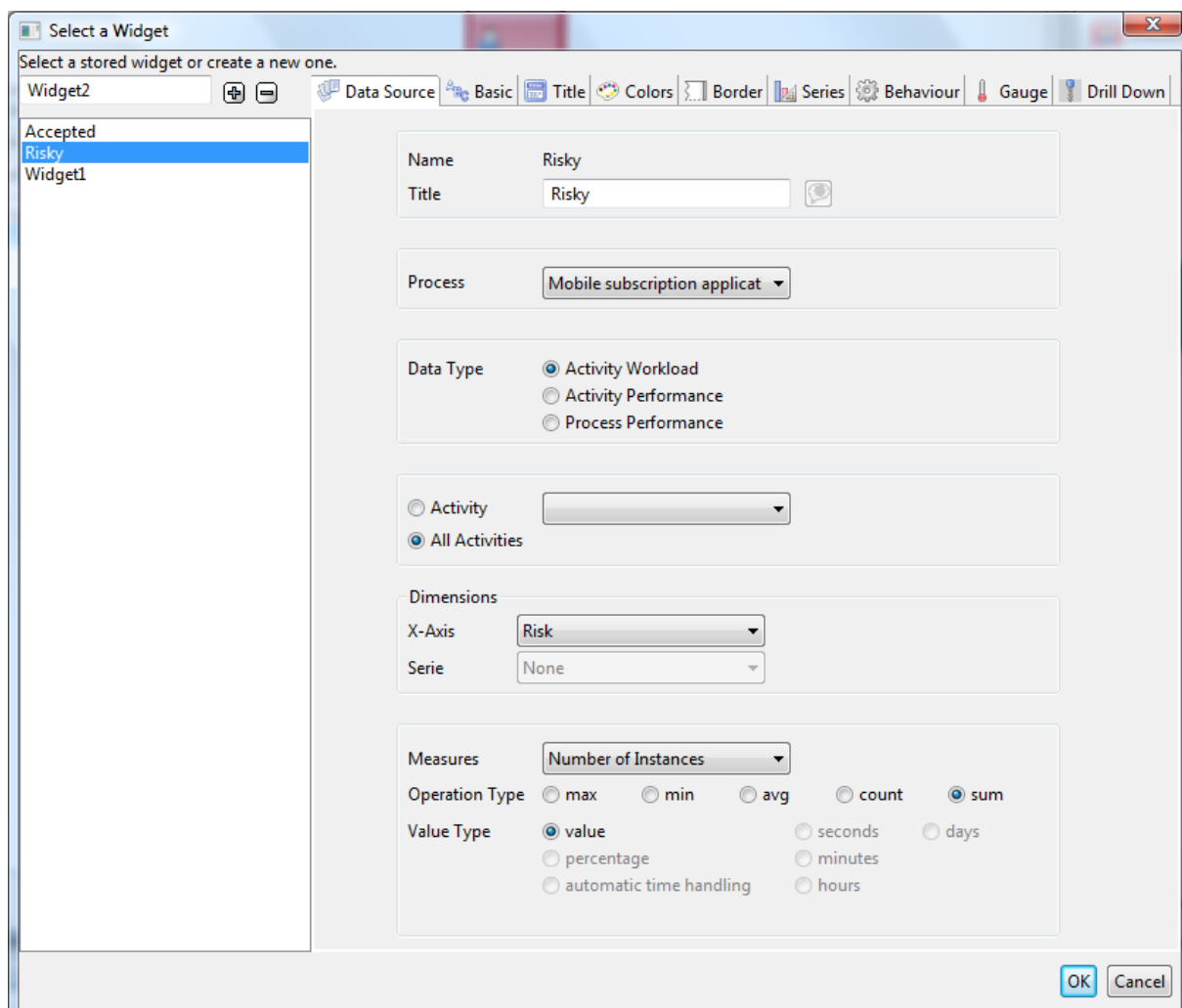
Figure 35 – Explicit business rules in Oracle BPM Studio



5.3 Key-Performance Indicators

Oracle BPM Studio allows the use of widgets (Figure 36) to define graphs on activity workload, activity performance and process performance. These widgets can be considered as templates with limited options and underlying code. They can be used on a BAM dashboard that is showed in the Oracle BPM Workspace.

Figure 36 – Explicit business rules in Oracle BPM Studio

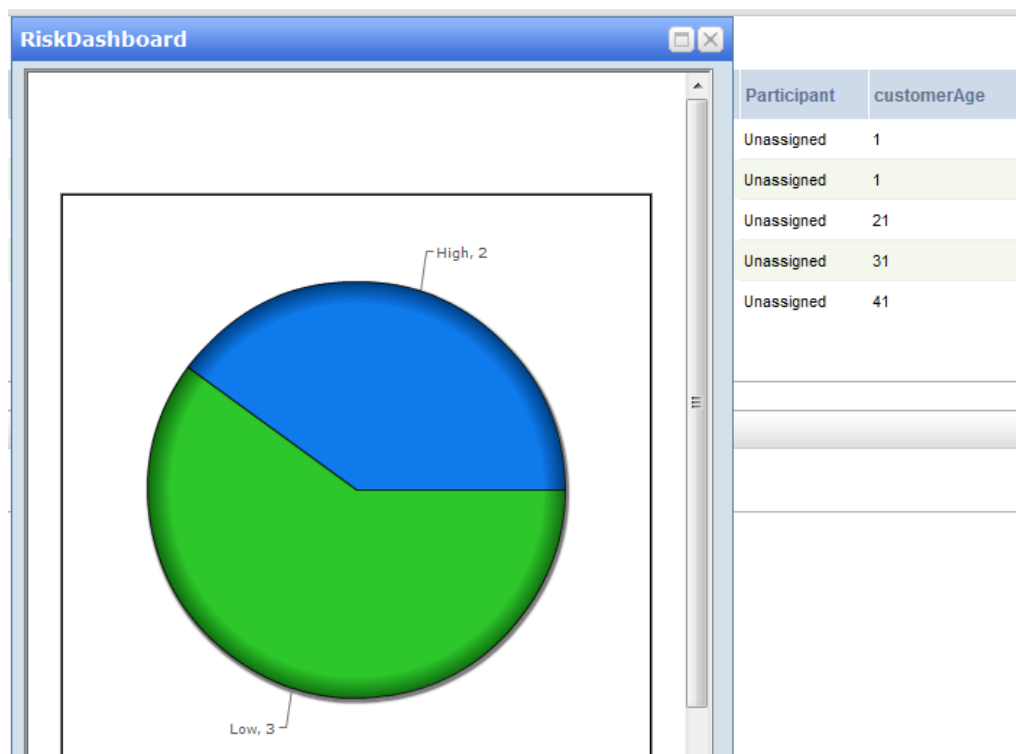


Oracle BAM can present its user with KPIs based on information from many different systems. However, in the studied version of Oracle BPM Suite Oracle BPM is not yet integrated with Oracle BAM. Oracle indicates that this is planned for future release. In the evaluated version, KPIs of the business processes are defined in Oracle

BPM Studio and showed using the BAM dashboards in the Oracle BPM Workspace and not with all other KPIs in the Oracle BAM product.

We have defined the proposed KPIs using Oracle BPM Studio. However, we have had to alter the definition of our KPIs slightly since it was not able to compare the amount of high risk application with the total amount of applications. Instead, we have defined a KPI that compares the amount of high risk applications to the amount of low risk applications (see Figure 37). Similarly we made a KPI that compared the amount of accepted applications to the amount of applications that were not accepted.

Figure 37 – Risk KPI in Oracle BPM Workspace



5.4 Executable Business Process Models

The business process models created in Oracle BPM Studio can be executed by Oracle BPM Enterprise. The execution language used by this product is proprietary and Oracle is not willing to share this information. The business process models created in the Oracle BPM Studio can be exported and imported into the Oracle BPM Enterprise which transforms it into process instances. Since Oracle BPM Studio is not only capable of modelling a business process, but also adding the information to make it executable, a single model is used and no translation between business process model and executable business process model is necessary.

Some of the activities from the business process model, which is made in Oracle BPM Studio, consist of complex composite services. These composite services can be designed in the Oracle BPEL Process Manager, which can also execute them.

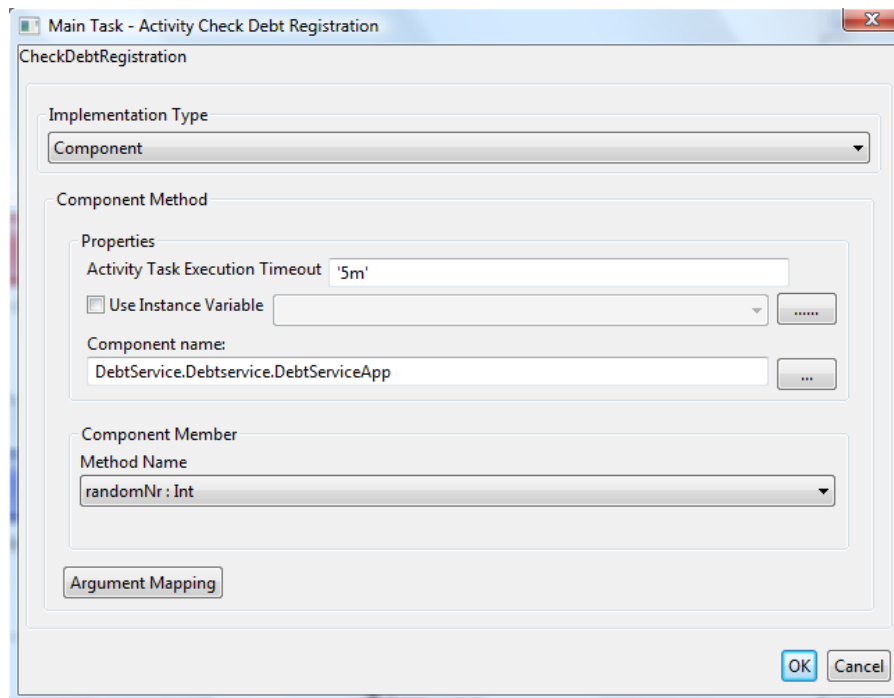
5.5 Programming Languages

Oracle BPM Studio does not support service or application design (except through the simple code also used for defining business rules, Figure 34), but is capable of importing externally made services or application code written in different languages like .NET or Java.

Oracle BPEL Process Manager only supports BPEL. However, the BPEL Process Manager is used as a plug-in for Oracle JDeveloper, which supports Java, XML, SQL and PL/SQL, HTML, JavaScript and PHP. These languages can be used to create a service which can be used in Oracle BPM Studio.

For example, in JDeveloper we have made a Java class with a method that returned a random number between - 1000 and 0, and used this class in Oracle BPM Studio to represent the debt service (Figure 38).

Figure 38 – Using a Java class in Oracle BPM Studio

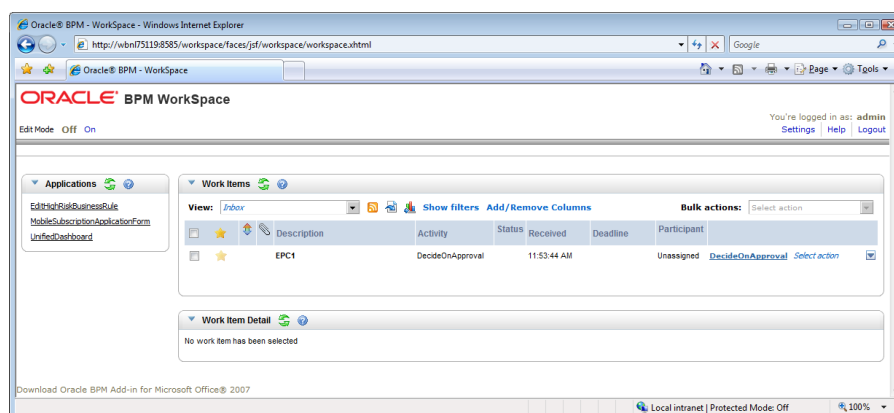


5.6 Designing User Interfaces

A separate application can be implemented, presented as a web service and linked to the executable business process model. However, Oracle BPM Studio also allows to define forms which can be presented in a portal (Oracle BPM Workspace). Oracle BPM Workspace can be used to let the users interact with the business process models made in Oracle BPM Studio. Workspace also features a work list for each of the users (Figure 39).

The forms which are made in Oracle BPM Studio are generated based on the business object (in our case the application for a mobile phone subscription) and the selected attributes of this business object. For example, the form to verify the application information includes the attributes which should be verified (name, age, etc.) and the Boolean value “verified” (which can be selected if the information is correct or not selected when it is not).

Figure 39 – Using the Oracle BPM Workspace



5.7 User Management

User management is done in Oracle BPM. Users (called participants) are described using standard information like name, e-mail, efficiency, and cost per hour and using organizational information like organizational unit,

roles and groups. Organizational units are only used to document which working hours the participants from that unit have. Roles determine which activities the participants can execute and are described using a name and a description (Figure 40). These roles can be either organizational or user-defined role. Finally, groups are collection of roles.

Figure 40 – Defining roles in Oracle BPM

Mapping roles to participants can only be done manually. The user management of Oracle BPM can be connected to a LDAP service. During runtime participants can be given general permission. These permissions indicate whether they can, for example, execute or suspend activities that belong to a certain role. Participants can have the same role, but different permissions. For example, one participant could execute a certain activity, while another can execute and reassign the activity, both having the same role. These permissions cannot be granted during design-time.

5.8 Informing the User

Users of the Oracle Workspace can be notified by e-mail upon the arrival of new work items in their activity list. Alternatively they can log in into the Workspace portal and check the items that need attention in their activity list (Figure 39).

5.9 Technical Monitoring and Control

Oracle places the technical monitoring and control of the BPM suite and its environment in its service bus. This service bus not only monitors and controls the services that are used by the Oracle BPM Suite, but also all other services and the business processes that are presented to the service bus as a service. Oracle's service bus also has capabilities to monitor and control the service level agreements which are influenced by the services.

5.10 Business Monitoring

In contrast with the other criteria, we evaluated the business monitoring criteria based on product documentation (Oracle 2008) and not on the implementation of our case, because the testing of Oracle BAM required an extensive infrastructure to be in place, which was not available at the time of our testing.

Business monitoring is done using Oracle BAM (Business Activity Monitoring). Oracle BAM gives users the ability to define their own proactive alerts and allows users to drill-down from the monitoring dashboard into more detailed information (for example, by using a dashboard which shows the amount of high risk applications and allows the user to drill-down to the list of all high risk applications). The proactive alerts can be sent to the user by email, SMS or instance message (for example, using Microsoft Office Communicator).

Since Oracle BAM is not yet integrated with Oracle BPM, business monitoring of the process instances can only be done passively through Oracle BPM Workspace. No active alerts can be sent until Oracle BAM is integrated.

The Oracle BPM Workspace shows the running process instances. In this portal, the status of the application can be retrieved. When having sufficient rights, a user can retrieve all the instances and at which step of the business process the process instance currently is. Searching an instance can be done using some standard variables like the status of the instance, which step (activity) the instance currently is at or which participant is working on it.

More useful variables can also be defined manually by creating a process variable in the Oracle BPM Studio. These variables can be any information, like the age of the customer or his name.

Figure 41 – Searching the status of an application

Description	Activity	Status	Received	Participant	customerAge	customerLastName
EPC1	VerifyApplication	9:59:42 AM	Unassigned	21	Koster	VerifyApplication Select action

We have added the last name of the customer as a project variable to our BPM project, which made it possible to search for the status of an application based on the last name of the customer. Figure 41 shows the result of this.

The Oracle BPM Workspace only offers the process view to its user (see Figure 39). The service, resource and overall view of the system are not supported.

5.11 Business Control

We have changed the executable business process model by letting the supervisor and not the employee verify the application. This change only affected new process instances. Process instances which were already running still followed the old business process model. The same situation occurred when the control-flow was changed by adding a new activity or when the implementation of an activity was changed (which resulted in a change in the business process model). There was no option to force the new business process model on the running process instances.

Changing business rules during runtime is supported by the Oracle BPM Suite (see Figure 42). We have changed the high risk business rule during runtime and lowered the age below which an application is considered high risk to 16 years instead of 18 years. This change only affects new instances of the business process. Running instances of the business process are not affected.

Figure 42 – Changing business rules during runtime in the BPM Workspace

Changing the workload balance is possible with the Oracle BPM Suite. Workload distribution is done based on roles. If the workload balance has to be changed, the roles have to be changed. In our case we tried to let one employee do all the verification. To accomplish this, we defined a new role, which we only assigned to one employee. The business process model was changed so that the verification task was performed by this new role.

5.12 Evaluation Summary

Table 13 summarizes our evaluation of Oracle BPM Suite.

Table 13 – Oracle BPM Suite evaluation summary

	Modelling criteria	Result
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	BPMN
	II. Support for interoperability between different business process modelling languages.	-
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	Only functional view from ARIS' EPC diagram can be imported but this import does not work flawless. Microsoft Visio import functionality is available.
	IV. Support for different views when modelling a business process.	Functional and organizational view available, behavioural and informational view lack information.
	V. Support for different perspectives of a business process.	UML activity diagram supported, but does not add much information to the BPMN notation.
	VI. Support for modelling with implicit business rules.	√
	VII. Support for modelling with explicit business rules.	√
	VIII. Support for modelling business processes using explicit business rules.	-
	IX. Ease of use of defining explicit business rules.	With the help of templates or by writing code.
2.	Support for KPIs.	KPI model is not flexible enough to model own KPIs, but is allowed the definition of a slightly modified version. KPIs are defined using a limited template with underlying code.
	Design criteria	
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	Proprietary
	II. Support for manual translation between business process modelling languages and business process execution languages.	Since one single model is used for both the business process model and the executable business process model, no translation and linking are needed.

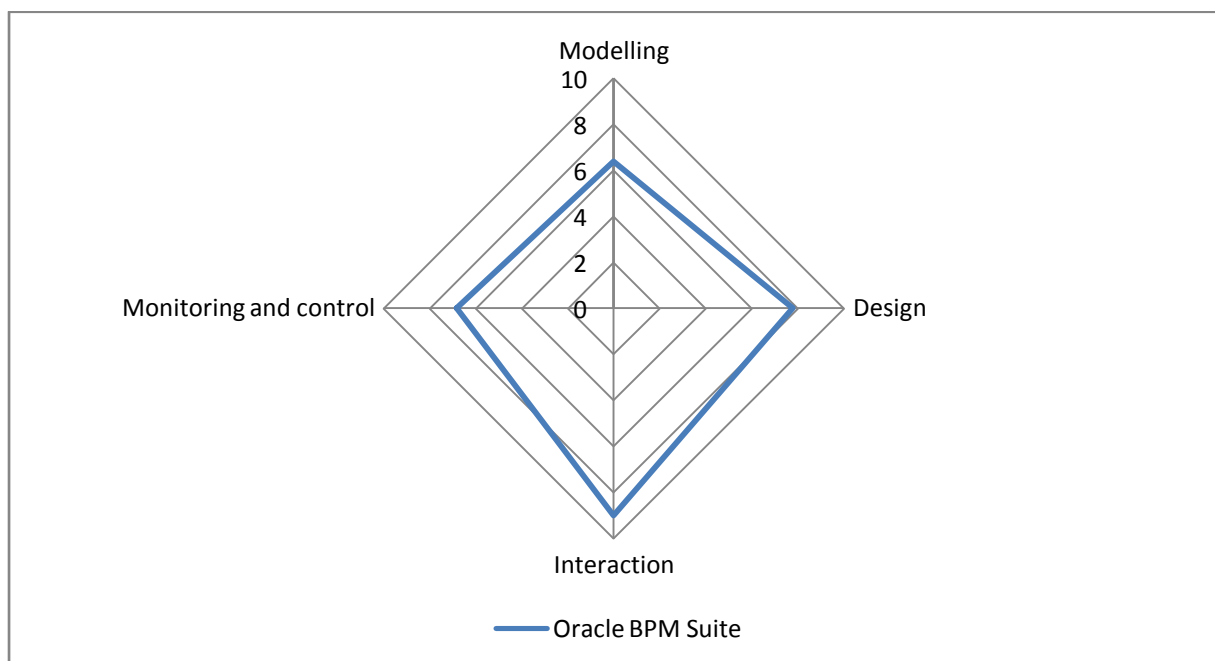
	III.	Support for automatic translation between business process modelling languages and business process execution languages.	
	IV.	Support for linking the business process execution model with the business process model.	
4.	Supported programming languages for implementing services/applications.		Java, XML, SQL and PL/SQL, HTML, JavaScript and PHP.
5.	Support for designing user interfaces.		
	I.	Support for separate applications for data entry.	√
	II.	Support for portal technology.	√
	III.	Support for an activity list.	√
6.	Support for user management.		
	I.	Way of describing roles (by names/description or by capabilities).	Names/description
	II.	Support for determining user capabilities based on organizational role.	Role can be organizational or user-defined.
	III.	Support for importing organizational structure from other systems.	LDAP
	IV.	Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	-
	V.	Support for manually assigning a user to a role.	√
	VI.	Support for automatically assigning a user to a role, based on his capabilities.	-
	VII.	Support for activity permissions based on roles.	√ (runtime, not design-time)
Interaction criteria			
7.	Support for informing the user.		
	I.	Support for active informing.	√
	II.	Support for passive informing.	√
Monitoring and Control criteria			
8.	Support for technical monitoring and control.		- (part of the Oracle service bus)
9.	Support for business monitoring.		
	I.	Support for active monitoring.	√ (alerting through email, SMS or instance message, but only in Oracle BAM, Oracle BPM does not support active monitoring)
	II.	Support for passive monitoring.	√

	III. Support for various level of detail.	√
	IV. Support for different views of monitoring information.	Only the process view is available
10.	Support for business control.	
	I. Support for process instance evolution.	√ (change only affects new process instances, running process instances still follow old business process model)
	II. Support for changing business rules.	√ (a changed business rule affect all new process instances, running process instances are not re-evaluated)
	III. Support for changing activities.	√ (change only affects new process instances, running process instances still follow old business process model)
	IV. Support for changing the workload balance between users.	√ (changing role that executes the activity)

Figure 43 shows a radar chart with the Oracle BPM Suite's rating for each of the evaluated phases. This figure shows that the Oracle BPM Suite scores medium at the modelling, and monitoring and control phases and high at the design and the interaction phases.

Appendix B explains how we have come up with the rating for each of the evaluated phases, by giving a rating based on the method described in section 4.4, to each of the selected (sub-) criteria.

Figure 43 – Radar chart with rating of the Oracle BPM Suite



6 Cordys BPMS

This chapter discusses the evaluation of the Cordys BPMS. Section 6.1 gives a short description of the products of the Cordys BPMS. Section 6.2 till section 6.11 present the evaluation of the Cordys BPMS divided by our ten selected criteria. Section 6.12 gives the summary of this evaluation and presents the rating of the suite.

6.1 Suite Description

Cordys is a pure-play BPM vendor of Dutch origins. They have created their own product and have not put it together by acquisitions. The result is that Cordys Business Process Management Suite (Cordys BPMS) is a single toolset and not a range of different applications. Cordys BPMS consists of five components, which are shown in Figure 44 and are:

1. **CAF (Composite Application Framework):** this component is used to create Internet applications that act as a user interface, using Web 2.0 technologies.
2. **BPM:** this component is used for modelling, design, execution and analysis/improvement of the business process models.
3. **Service-Oriented Architecture grid:** this component enables the integration between the business process layer and the underlying IT systems.
4. **BAM:** this component is used to monitor performance by means of KPIs and can alert the user of violations.
5. **Non-Stop High Availability Framework:** this component provides stability and continuity by allowing on-demand reconfiguration of the IT platform, avoiding any downtime when adding or removing services.

Figure 44 – Cordys BPMS components (Cordys 2008)

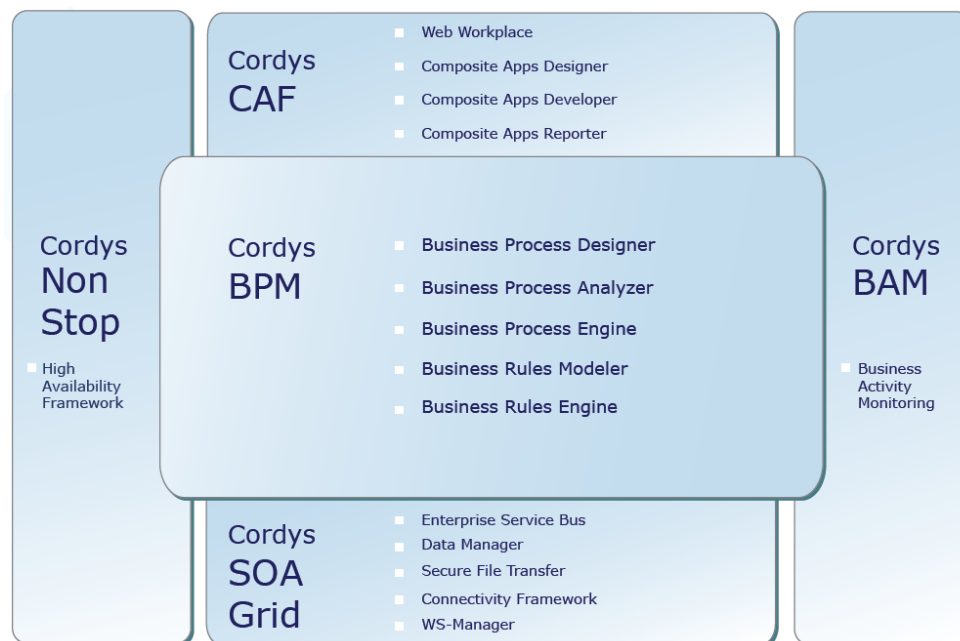


Figure 45 maps the components from the Cordys BPMS to the phases of our BPM lifecycle, based on the product documentation (Cordys 2008).

This evaluation of Cordys BPMS is based on Cordys BPMS C3 version 4.2.

Figure 45 – Components from Cordys BPMS mapped to our BPM lifecycle

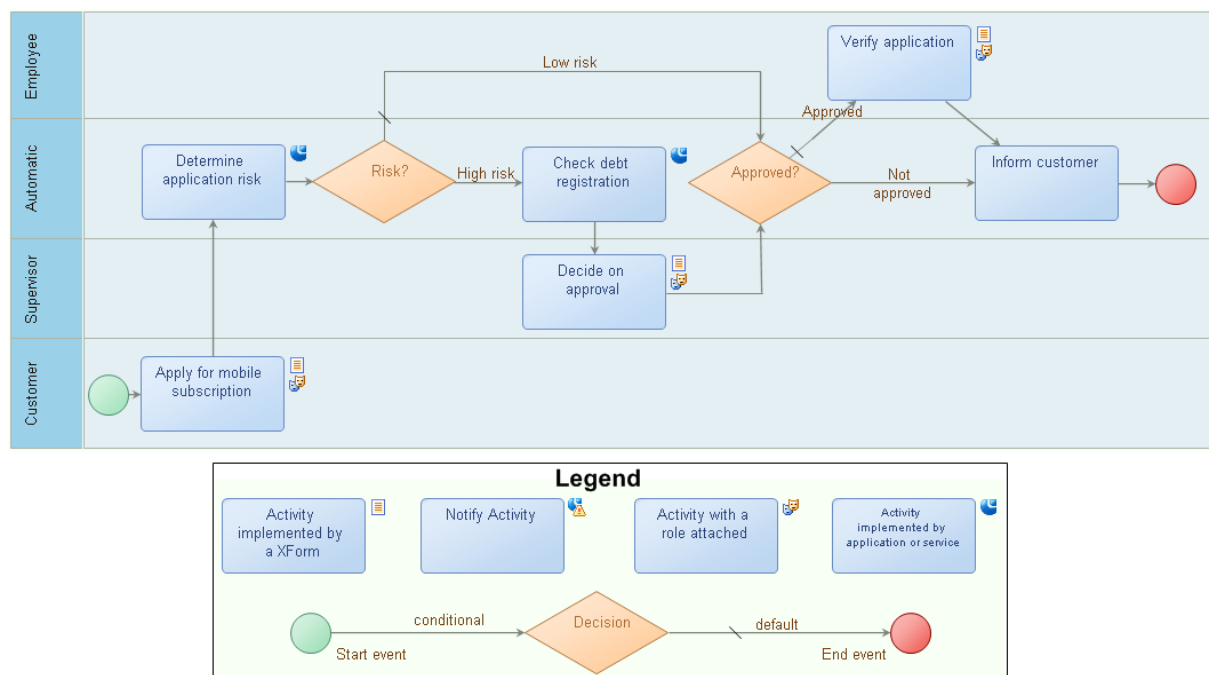
	Strategy Development	Discovery	Modelling	Design	Deployment	Execution	Interaction	Monitoring & control	Analysis
CAF									
Business Process Designer									
Business Process Analyzer									
Business Process Engine									
Business Rules Modeler									
Business Rules Engine									
SOA Grid									
BAM									
Non Stop									

6.2 Business Process Modelling

Modelling with Cordys BPMS is done in the Business Process Modeling Notation and no other business process modelling languages are available. We have modelled our business process in the Cordys Business Process Designer (also called Cordys Studio) (see Figure 46).

Figure 46 also shows the implicit modelling of a business rule in Cordys, which has been done using the “Determine application risk” activity, which determines the amount of risk, and the decision “Risk?”, which acts on the value delivered by the previous activity.

Figure 46 – Business process modelled in Cordys Business Process Designer



Importing the ARIS EPC model is possible using the XPD L format. Cordys can use different plug-ins to import and export business process models, and one of the standard plug-ins can be used to import an XPD L from ARIS. When importing from ARIS, only the functional view of the business process is imported (see Figure 47). Cordys currently does not have any plug-ins that allow import from Microsoft Visio.

Figure 47 – Imported business process from ARIS into Cordys

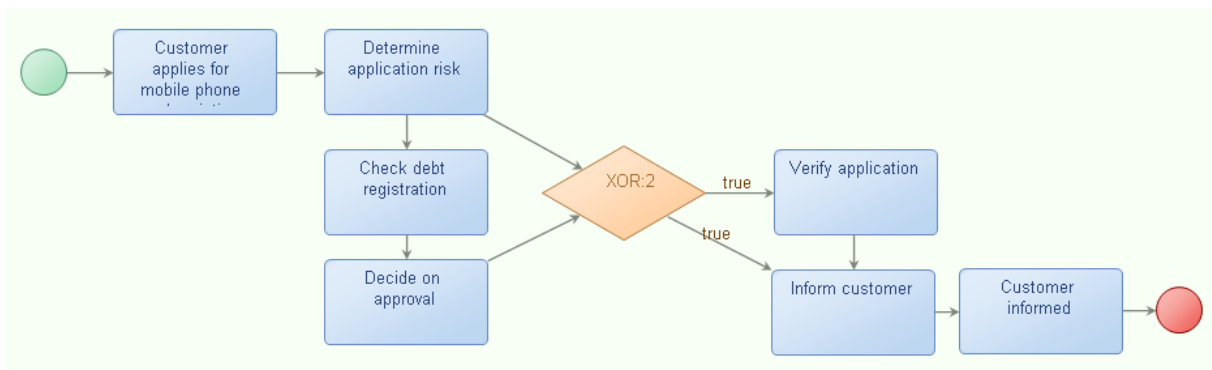


Figure 46 presents a functional view of the business process by showing the activities being performed with the relevant flows between them. It also shows the behavioural view by showing the order in which the activities are performed and how they are performed (by showing the type of activity). Figure 48 shows details of the behavioural view in Cordys.

Figure 49 shows part of the organizational view of the business process model. This figure shows who executes which activity. Information like which physical communication mechanism or which storage is used cannot be represented in the business process model, neither is the informational view of the business processes.

Figure 48 – Behavioural view of the business process in Cordys

The screenshot shows the 'Properties - Activity' window for the 'Apply for mobile subscription' activity. The 'General' tab is selected. The 'Type' is set to 'Application'. The 'Code' is 'XForms/XF_Apply.caf'. The 'Application Service Type' is 'xform'. The 'Description' is 'Apply for Mobile Subscription'. There is a 'Docked' checkbox which is currently unchecked.

Figure 49 – Organizational view of the business process in Cordys

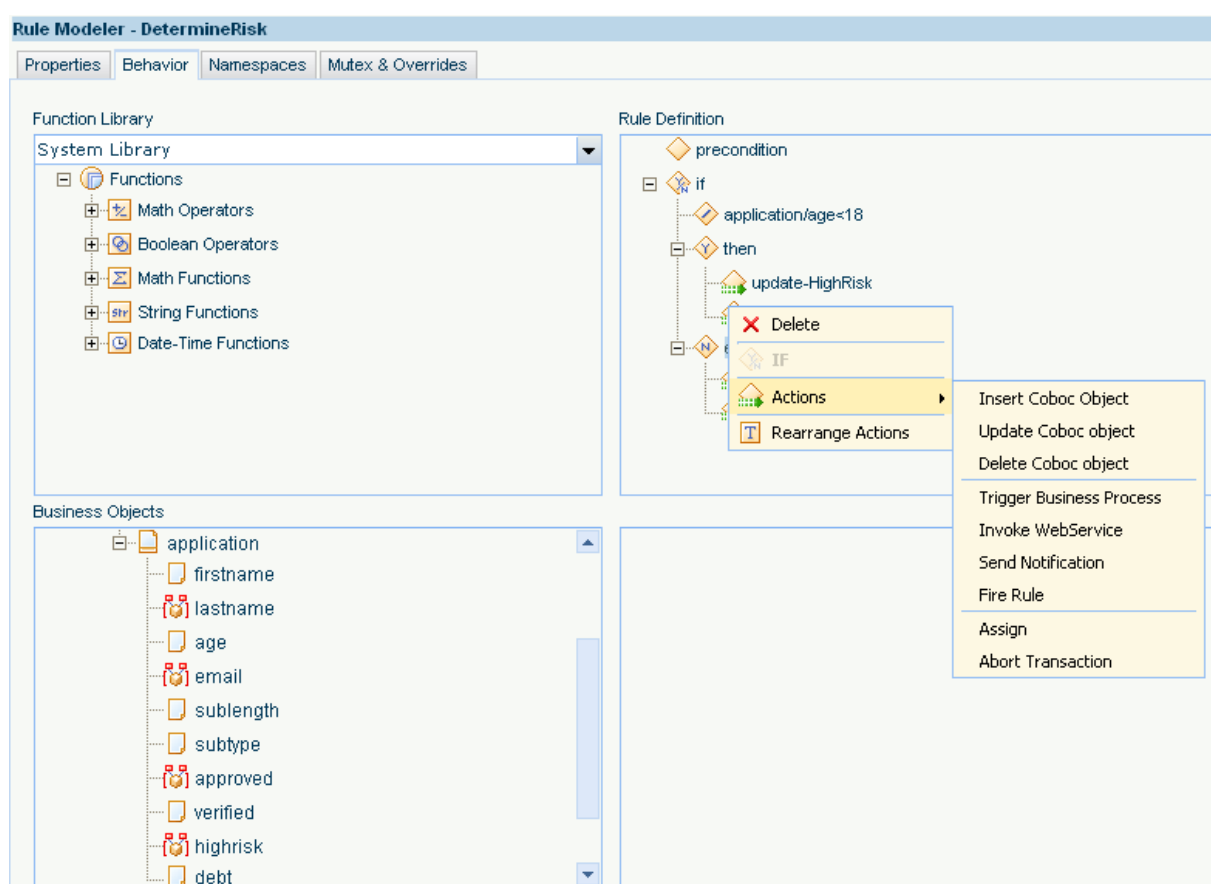
The screenshot shows the 'Properties - Activity' window for the 'Apply for mobile subscription' activity. The 'Roles to select' tab is selected. It displays a list of roles: 'Employee (System)', 'Everyone In System (System)', 'Supervisor (System)', 'System Administrator (System)', 'Audit Manager (Cordys Audit Service)', and 'everyoneInCordys Audit Service (Cordys Audit Service)'. To the right, the 'Linked Roles' tab is visible, showing 'Customer (System)' as the linked role.

Cordys uses business objects to store the data used in a business process. These business objects are defined in XML, and some examples are a sales order or an application for a mobile phone subscription. Explicit business rules can be defined on these business objects, to make sure that the right action is taken when an attribute of a business object reaches a certain value. These business rules can be used, for example, to make sure that every sales order is verified by a manager once the total amount is higher than five million euro's. Since these business rules are explicit, they can be changed without changing the business process models.

For our case, these business rules have been used to model our risk business rule. This allowed us to set the risk of an application as high in case the applicant was younger than a certain age.

Defining explicit business rules in Cordys is done using a rule modeller (see Figure 50). This rule modeller uses a standard template in which a precondition, condition and actions are entered. These actions can be defined using predefined templates (shown as actions in Figure 50) such as notifying someone or updating an attribute of the business object.

Figure 50 – Defining explicit business rules in Cordys

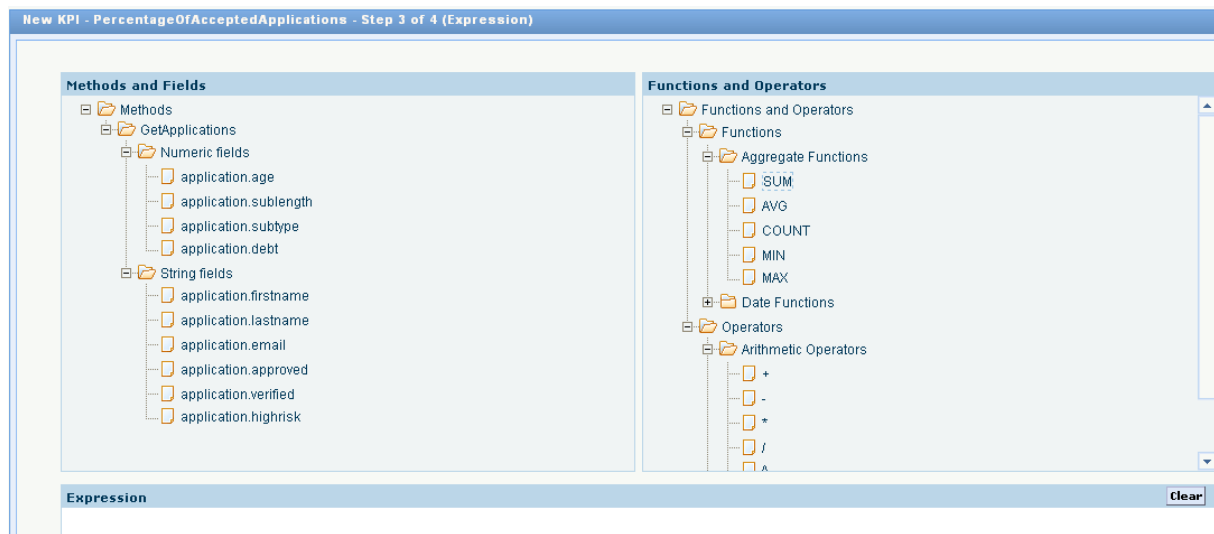


6.3 Key-Performance Indicators

KPIs are defined in Cordys BPMS using the KPIComposer. To calculate KPIs, the required information has to be stored in a database and retrieved from this database using user-defined methods. These methods are then called in the KPIComposer and the KPIs are built based on the information that these methods retrieve. This makes defining KPIs not really user friendly, since these methods first have to be written.

In our case, we defined the application information entity which has to be stored in the database and a method has to be written that retrieves the information from the database. We were able to write a method that retrieved all applications from the database, but were unable to compose a KPI that could calculate the required KPIs. This failed because the KPIComposer only allows to create a KPI based on some predefined elements, which focus mainly on arithmetic operations, and does support the writing of code by designers. Figure 51 shows the screen of the KPIComposer where the KPI is composed using predefined elements.

Figure 51 – Cordys KPIComposer



6.4 Executable Business Process Models

The business process model represented in BPMN can be automatically translated into an executable business process model, which is written in the Business Process Modeling Language (BPML). Since the BPML code is not changed after it has been generated, no linking between the executable business process model and the business process model is needed.

6.5 Programming Languages

Cordys BPMS allows the Cordys generated code to be exported into Eclipse (The Eclipse Foundation 2008) which is an open source integrated development environment. Within Eclipse one can edit the code or add new code in many different languages for example, Java, JavaScript, C#, C and C++.

Services and applications can be called by the business process through various SOAP nodes. Standard SOAP connectors for Java Enterprise Edition and .Net are included, but others can also be written.

For our case, we have written a Java class with a method that picked a random number between -1000 and 0 and added 400 to this number. If the result was positive, the method returned 0 (indicating that the applicant has no debts), otherwise the negative number was returned (showing that the applicant has debts). Using a SOAP connector, this class and method could be called by the business process, thereby implementing a debt service.

6.6 Designing User Interfaces

Although separate applications can be designed for inputting information, we have used the XForm editor in Cordys (see Figure 52). This XForm editor allows the creation of forms which can be used as front-end applications. These XForms can only be generated when they are used in conjunction with a method. These methods can be written in Cordys and, for example, be used to extract information from a database. In this case, an XForm can be generated based on the output of the method. In other cases, the XForm has to be created and mapped to the business process using a message map. This message map allows data from the business process to be sent to the XForm and back to the business process.

One can design his own applications to interact with the business processes or use some portal technology. This portal is the one provided by Cordys or another one, in which case the interaction with the business processes is performed using web service technology. In both cases, an activity list can be added to the portal to show the user which activities he has to execute.

6.7 User Management

Roles in Cordys BPMS are defined by a name and a description of the role, and different roles can be grouped. The specific kind of roles used (based on the organizational structure or user-defined roles) are irrelevant. User capabilities cannot be modelled in Cordys, unless a different role is created for people who have a certain capability. Figure 53 shows an example of the “MobileSubscriptionPowerUser” role, which also has the “Customer”, “Employee” and “Supervisor” roles.

Figure 52 – Cordys XForm editor with preview

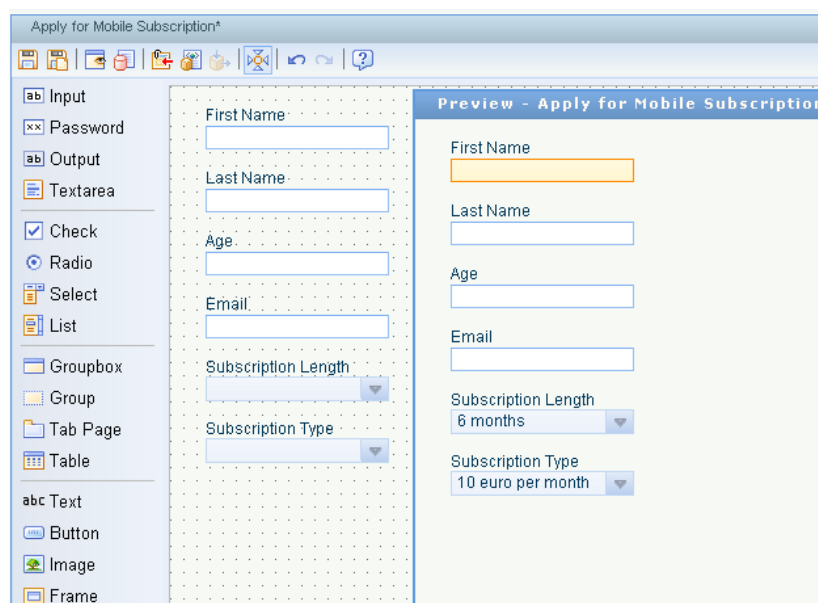
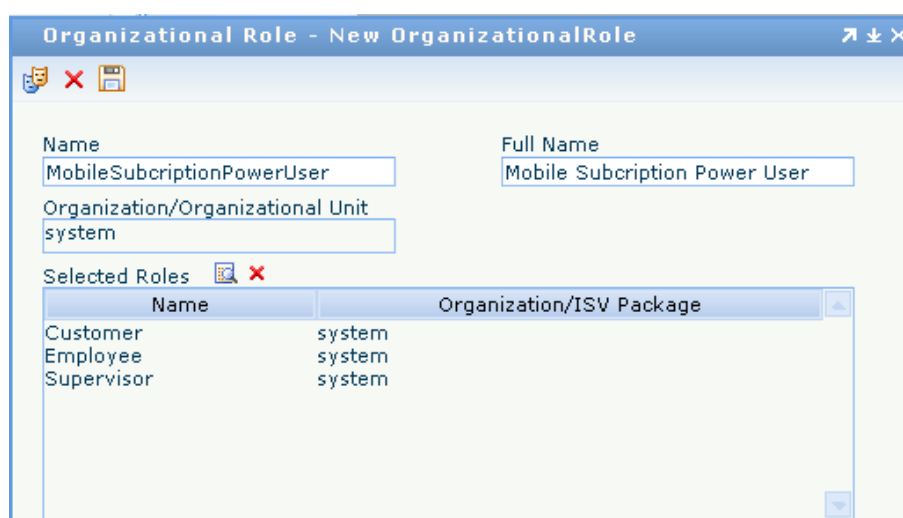


Figure 53 – Role management in Cordys



The organizational structure cannot be imported from another system (such as an Active Directory). Instead, Cordys uses its own LDAP server, thereby duplicating user information.

Users are assigned to a role by the system administrator and not automatically by the system itself. Users can have multiple roles at the same time. Permissions are attached to a role. For example, this allows someone of a certain role to skip activities.

6.8 Informing the User

Users can either logon to the Cordys BPMS using a portal to see whether they need to execute an activity (for example, by looking in their inbox), or can be informed using some communication means as, e.g., e-mail.

In our case, we have added an action to the risk business rule, which informs the supervisor when a new high risk application is submitted and is awaiting his approval (see Figure 54).

Figure 54 – My Inbox of the Cordys portal

My Inbox - 1 to 1 of 1 messages				
Basic Search				
<input type="checkbox"/>		Subject	Source	Sent By
<input type="checkbox"/>		New high risk application awaiting approval	RuleEngine	Stefan
			Received	Due By
			Friday, December 19,...	None

6.9 Technical Monitoring and Control

The Cordys product's components communicate with each other and with other services and applications through SOAP processors. These SOAP processors are essential for the operation of the product. Therefore, these SOAP processors can be monitored (see Figure 55). When a processor is malfunctioning, it can be restarted from this monitoring screen or additional actions can be taken.

Figure 55 – Monitoring SOAP processors

Monitor SOAP Processors - system				
List By	-All Computers-	Refresh Status Rate	15 sec	Show
Name	Organization	Computer	Startup Type	Status
Audit Service Processor	system	stefan-vm	Automatic	Started
Business Process Management	system	stefan-vm	Automatic	Started
Classic Studio Processor	system	stefan-vm	Automatic	Started
COBOC Processor	system	stefan-vm	Automatic	Started
Data Transformation Processor	system	stefan-vm	Automatic	Started
Email Processor	system	stefan-vm	Automatic	Configuration Er

Since Cordys runs as a web application, the web gateway to the application is also monitored (see Figure 56).

Figure 56 – Monitoring the web gateway

Monitor Web Gateway	
Gateway Statistics	
Number of requests	146
Total processing time	57.996 seconds
Average processing time per request	397.23288 milliseconds
Total back end wait time	35.932 seconds
Invalid messages	0
Unknown Web users	0
Invalid organization requests	0
Messages sent to Cordys	146
Replies from Cordys	144
Timed out requests	0
LDAP errors	1
Cordys errors	0

6.10 Business Monitoring

The Cordys BAM product takes care of the Business Activity Monitoring (Cordys 2008). In case the KPIs defined in section 6.3 are violated, users are notified using email or through a message in their inbox of the Cordys portal (Figure 54).

Passive monitoring is also available (Figure 57). We were able to track down a business process and determine the status of the application involved. Business processes can be tracked down based on their starting time or on predefined variables like the last name of the customer (which can also be seen in Figure 57). These predefined variables are named process identifiers.

Figure 57 – Passive monitoring of business processes

The screenshot displays the 'Instances by Process Definition' window. It shows a summary of process instances for '3.Business Process Models/ApplyForMobSub' (version vcmdemo10, description: Apply For Mobile Subscription). Below the summary is a table of process instances with columns: Start Time, Lastname of th..., Status, and Type. The instances are listed with their start times, last names, and statuses (Complete, Debug, Aborted). Below this is the 'Activities' section, which shows the main process 'Apply For Mobile Subscription' and a specific process instance 'Lastname of the applicant: Koster'. It contains a table of activities with columns: Activity Name, Activity Type, Status, and Start Time. The activities listed include 'Apply for mobile subscription', 'Determine application risk', 'Risk?', 'Check debt registration', 'Decide on approval', and 'Approved?'. All activities are marked as 'Complete'.

Start Time	Lastname of th...	Status	Type
1/14/2009 16:47	Koster	Complete	Proce...
1/14/2009 16:43	Koster	Debug	Proce...
1/14/2009 16:39	bpm:Applicatio...	Debug	Proce...
1/14/2009 16:35	bpm:Applicatio...	Debug	Proce...
1/14/2009 16:33	NA	Aborted	Proce...
1/14/2009 16:32		Aborted	Proce...

Activity Name	Activity Type	Status	Start Time
Apply for mobile subscription	Manual (Task)	Complete	1/14/2009 16:47
Determine application risk	Web Service	Complete	1/14/2009 16:47
Risk?	Decision	Complete	1/14/2009 16:47
Check debt registration	Web Service	Complete	1/14/2009 16:47
Decide on approval	Manual (Task)	Complete	1/14/2009 16:47
Approved?	Decision	Complete	1/14/2009 16:52

Drilling down from KPIs into process instances is possible with Cordys BAM. Due to the unavailability of the product for our evaluation, we had to draw this conclusion based on product documentation (Cordys 2008). Cordys BAM supports an overall system view and the process view. Furthermore, the service view is supported by Cordys BPM (see Figure 55 and Figure 56).

6.11 Business Control

Changes in the business process model are made in the Business Process Designer and then deployed onto the Business Process Engine. If an older version of the same business process is already running on the Business Process Engine, the running process instances are finished using the older version of the business process model and the new process instances are started using the new version.

Cordys allows changing business rules through the same interface used for defining the business rules (see Figure 50). The changed business rules affect all new and running process instances, but running process instances cannot be re-evaluated.

Cordys provides the flexibility to design and implement custom dispatch logic that can determine how tasks are delegated to users. By changing this dispatch logic for a certain activity workload balancing can be performed. In our case, we have changed this dispatch logic for the verification activity to include only one employee

6.12 Evaluation Summary

Table 14 summarizes our evaluation of Cordys BPMS.

Table 14 – Cordys BPMS evaluation summary

	Modelling criteria	Result
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	BPMN
	II. Support for interoperability between different business process modelling languages.	-
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	Importing models from ARIS is possible (but only the functional view is imported).
	IV. Support for different views when modelling a business process.	Functional and behavioural available. Organizational and informational view lack information.
	V. Support for different perspectives of a business process.	-
	VI. Support for modelling with implicit business rules.	√
	VII. Support for modelling with explicit business rules.	√
	VIII. Support for modelling business processes using explicit business rules.	-
	IX. Ease of use of defining explicit business rules.	Using a rule modeller with pre-defined templates.
2.	Support for KPIs.	KPI model is inflexible, preventing us from modelling our own KPIs. The KPIs are defined using code and templates.
	Design criteria	
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	BPML
	II. Support for manual translation between business process modelling languages and business process execution languages.	-
	III. Support for automatic translation between business process modelling languages and business process execution languages.	√
	IV. Support for linking the business process execution model with the business process model.	The generated BPML code does not change (only when it is generated again from the business process model) so no linking is necessary.

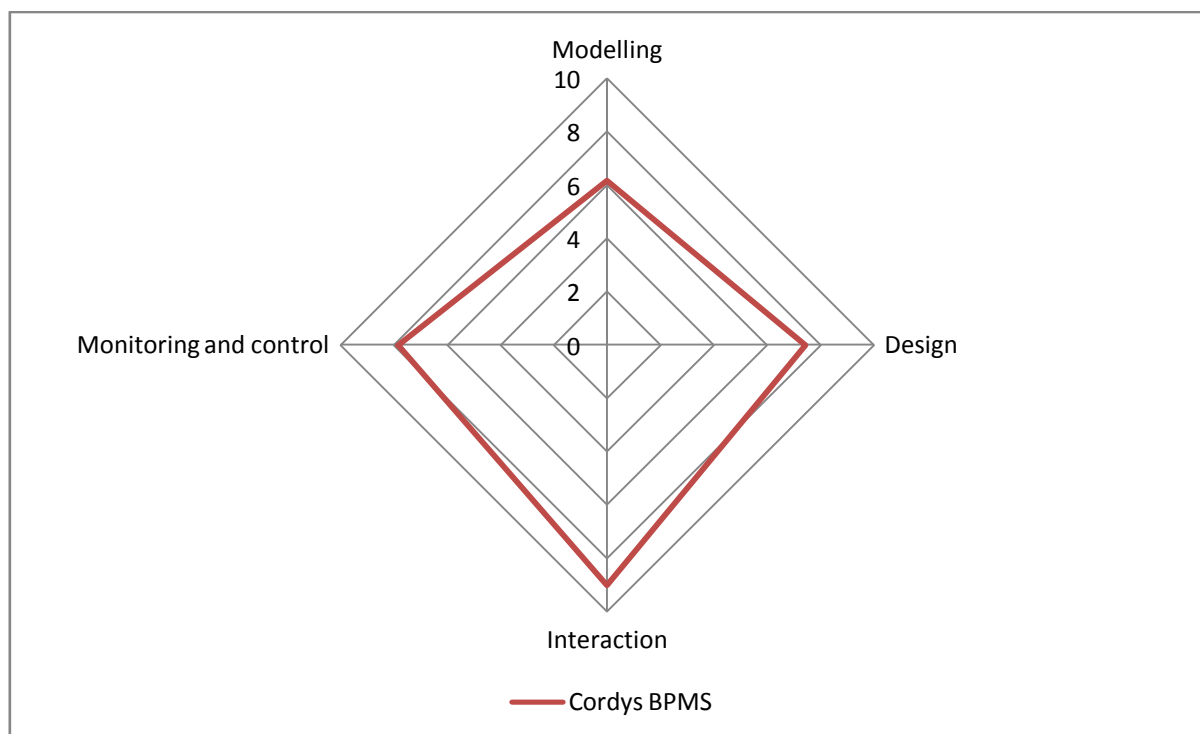
4.	Supported programming languages for implementing services/applications.	Java, C, C++, PHP, Python, C#, JavaScript, Perl and Delphi using Eclipse.
5.	Support for designing user interfaces.	Automatic generation available.
	I. Support for separate applications for data entry.	√
	II. Support for portal technology.	√ (own portal or integrate with other portal using web services)
	III. Support for an activity list.	√
6.	Support for user management.	
	I. Way of describing roles (by names/description or by capabilities).	Names/description
	II. Support for determining user capabilities based on organizational role.	Role can be organizational or user-defined.
	III. Support for importing organizational structure from other systems.	-
	IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	-
	V. Support for manually assigning a user to a role.	√
	VI. Support for automatically assigning a user to a role, based on his capabilities.	-
	VII. Support for activity permissions based on roles.	√
Interaction criteria		
7.	Support for informing the user.	
	I. Support for active informing.	√
	II. Support for passive informing.	√
Monitoring and Control criteria		
8.	Support for technical monitoring and control.	√
9.	Support for business monitoring.	
	I. Support for active monitoring.	√
	II. Support for passive monitoring.	√
	III. Support for various level of detail.	√
	IV. Support for different views of monitoring information.	Overall system view, process view and service view.
10.	Support for business control.	
	I. Support for process instance evolution.	√ (change only affects new process instances, running process instances still follow old business

		process model)
II.	Support for changing business rules.	√ (no re-evaluation of running process instances)
III.	Support for changing activities.	√ (change only affects new process instances, running process instances still follow old business process model)
IV.	Support for changing the workload balance between users.	√ (changing activity's dispatch logic)

Figure 58 shows a radar chart with the Cordys BPMS' rating for each of the evaluated phases. This figure shows that Cordys BPMS is scoring medium at the modelling, design phase, and high at the interaction, and monitoring and control phases.

Appendix C explains details how we have come up with the rating for each of the evaluated phases, by giving ratings, based on the method described in section 4.4, to each of the selected criteria.

Figure 58 – Radar chart with rating of Cordys BPMS



7 IBM WebSphere BPM

This chapter discusses the evaluation of the IBM WebSphere BPM suite. Section 7.1 gives a short description of the products of the IBM WebSphere BPM suite. Section 7.2 till section 7.11 present the evaluation of the IBM WebSphere BPM suite divided by our ten selected criteria. Section 7.12 gives the summary of this evaluation and presents the rating of the suite.

7.1 Suite Description

IBM is a vendor with a background in servers and also more recently in services, but they have been expanding their business into other areas for a long time by means of acquisitions. The most important products from the IBM WebSphere Business Process Management suite are (IBM 2009):

- WebSphere Business Events: this product is used to detect, evaluate and respond to business events.
- WebSphere Business Modeler: this product is aimed at business users and allows them to model, document, simulate and analyze their business processes. It can also be used to design simple forms when combined with the Lotus Form Designer.
- WebSphere Business Monitor: this product enables monitoring of business events in real-time by providing users with a visual display of business process status on customizable dashboards. The design of these monitoring tools is supported by the WebSphere Business Monitor development toolkit.
- WebSphere Business Services Fabric: this product is a SOA-based solution aimed at business users and lets them assemble and maintain composite business applications
- WebSphere Business Space: this product serves as the user interface which is used to interact with the business processes. Users can also access the monitoring information from the WebSphere Business Monitor through this product.
- WebSphere Integration Developer: this product is the design tool of the IBM WebSphere BPM suite and can be used to integrate the business processes in the IT infrastructure and to design user interfaces.
- WebSphere Process Server: this product is the business process execution engine of the WebSphere BPM suite.
- WebSphere Service Registry and Repository: this product stores the business process models for reuse and registers them for runtime execution.

In addition to these products there is also a partner gateway (for B2B transaction) and a premise server (for sensor information, like RFID).

Figure 59 maps the products from IBM WebSphere BPM to the phases of our BPM lifecycle, based on the product documentation (IBM 2009).

Figure 59 – IBM WebSphere BPM products mapped to our BPM lifecycle

	Strategy Development	Discovery	Modelling	Design	Deployment	Execution	Interaction	Monitoring & control	Analysis
Business Events									
Business Modeler									
Business Monitor									
Business Services Fabric									
Business Space									
Integration Developer									
Process Server									
Service Registry and Repository									

For our evaluation we have used the following products from IBM WebSphere BPM:

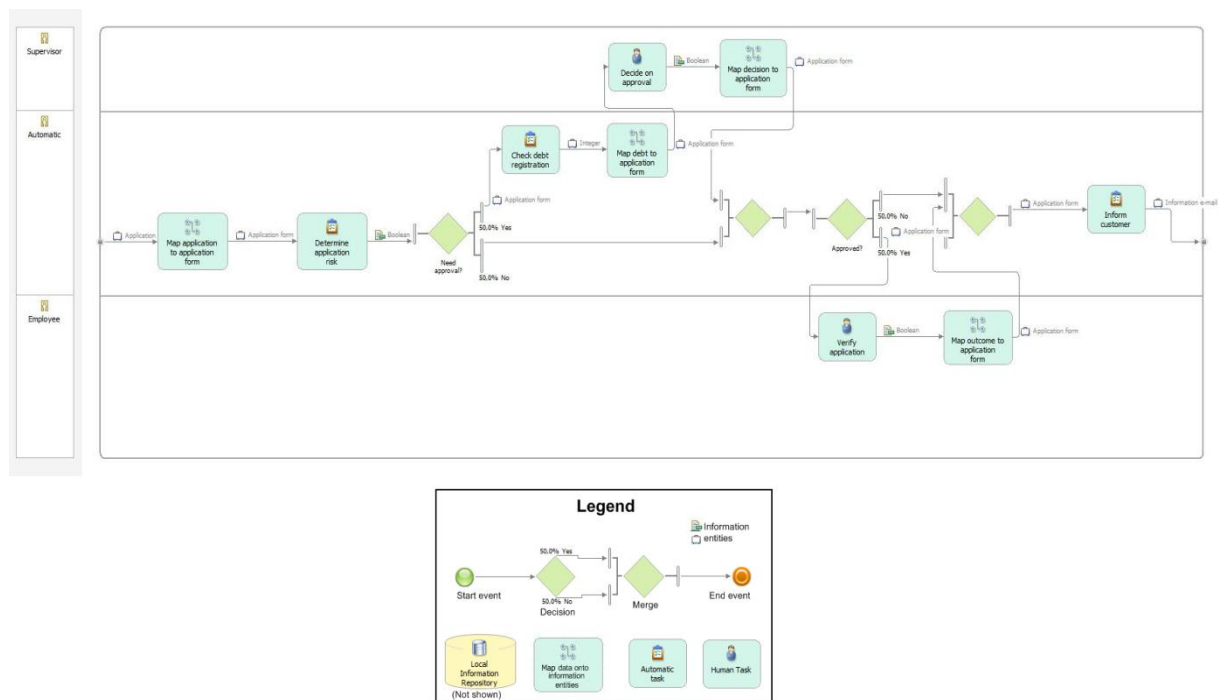
- WebSphere Business Modeler Advanced 6.2.
- WebSphere Business Monitor 6.2.

- WebSphere Business Space 6.2
- WebSphere Integration Developer 6.2.
- WebSphere Process Server 6.2.

7.2 Business Process Modelling

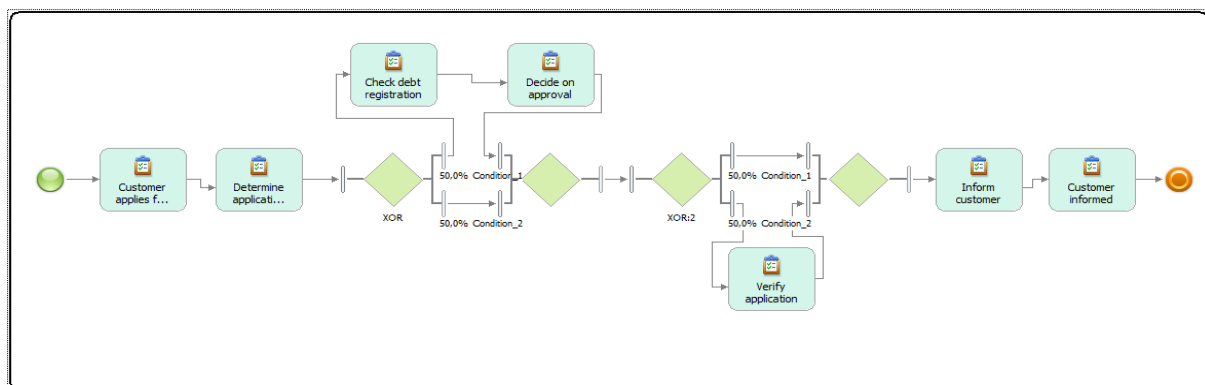
IBM WebSphere Business Modeler is the tool that should be used in the modelling phase. We have modelled our business process in this tool (see Figure 60). The only supported business process modelling language is the Business Process Modeling Notation.

Figure 60 – Business process modelled in IBM WebSphere Business Modeler



The ARIS EPC diagram could be exported into a XML file, which we have imported into IBM WebSphere Business Modeler. Figure 61 shows the business process that was imported from the ARIS XML into the program. Next to this business process also the data objects and the participants could be imported from ARIS. Microsoft Visio XML drawings can be imported into the Business Modeler, which has some predefined mappings for shapes of the Basic Flowchart stencil, IDEF0 stencil, and SDL Diagram stencil. Other mappings can be defined manually.

Figure 61 – ARIS information imported into IBM WebSphere Business Modeler

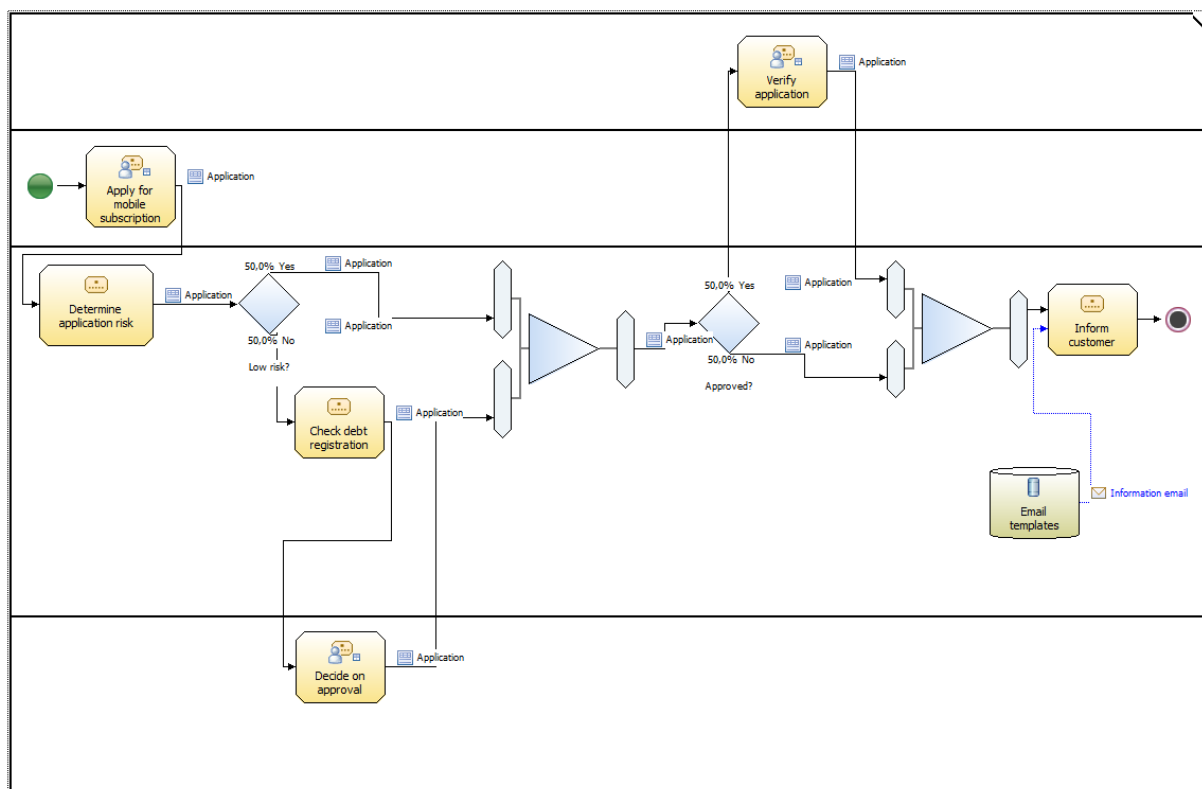


The business process model from Figure 60 shows the functional view, representing the activities that are performed with the relevant flows between them. The behavioural view is also present in the business process model, as indicated by the order in which the activities are performed and their implementation type (human task, system task, etc.). However, detailed information about their implementation (if the system task is executed by a Java class or a web service, for example) is not present in this business process model and is not added until the business process model is translated into an executable business process model, which can be done using IBM WebSphere Integrator Developer.

The organizational view is present in the business process model. The model shows by whom the activities are executed by adding swim lanes which represent the roles assigned to each activity. The model also shows the physical storages, which are represented by cylinders (for example, the “email templates” storage). The informational view is also added to the business process model and represents the information entities that flow through the process. In our case the application form is shown flowing through the process, which indicates that all activities use this entity, either by reading it or updating it. The structure of the information entities can also be viewed in IBM WebSphere Business Modeler.

IBM WebSphere Business Modeler supports an alternative notation in addition to the BPMN notation. This alternative notation is shown in Figure 62 and only differs from the BPM notation in respect to the icons it uses, so it does not have much added value. Next to the business perspective from the Business Modeler, IBM WebSphere Integration Developer provides the technical user with a perspective that shows more technical details. This perspective is shown in Figure 69.

Figure 62 – IBM WebSphere Business Modeler alternative notation



We have modelled the risk business rule, which was hard-coded into the business process model (see Figure 60), explicitly using IBM WebSphere Integration Developer (see Figure 63). This was done by replacing the Java class by a rule group with a business rule inside it. Business rules are defined using natural language in combination with templates.

We tried to model the entire business process using business rules, but found this to be impossible, since business rules cannot be linked to each other in Cordys BPMS.

Figure 63 – Defining business rules in IBM WebSphere Integration Developer

▼Rule Set				
Name	DetermineRisk		Display Name DetermineRisk	
▼Interface				
Interface	Determineapplicationrisk			
Operation	DetermineHighRisk			
Input	Age	int		
Output	HighRisk	boolean		
▼Variables				
Name	Type			
▼Rules				
Name	Rule1			
Template	Risktemplate			
Presentation	HighRisk if age<=18			
Name	Rule2			
Presentation				
Action	HighRisk = false			
▼Templates				
Name	Risktemplate			
Presentation	HighRisk if age<=NoRiskAge			
Description				
Parameters	Name	Type	Constraint Description	
	NoRiskAge	int	None	
If	Age <= NoRiskAge			
Then	HighRisk = true return			

7.3 Key-Performance Indicators

The functionality for Modelling KPIs with the IBM WebSphere BPM suite is spread over two BPM tools, namely the IBM WebSphere Business Modeler and the IBM WebSphere Integration Developer. IBM assumes that the business analyst, who is using the Business Modeler, only defines what should be measured. The integration developer, who uses the Integration Developer product, imports the monitor model and specifies the implementation of these business measures.

In the Business Modeler, the business analyst can describe the business measure textually and add information such as the type of KPI, the target value and the alerts that can be triggered. In our case, we have described a KPI as following: “Percentage of accepted application in comparison with the total amount of applications.” We have also added a target value of 75% and a range between 60% and 90%, in which the percentage is also acceptable. Finally we have described an alert which alerts the process owner when the acceptable is violated. Parts of these definitions are shown in Figure 64.

Figure 64 – Adding a business measure in IBM WebSphere Business Modeler

▼ Business Measure Information	
Name	Percentage of accepted application in comparison with the total amount of applications
Type	<input checked="" type="radio"/> KPI <input type="radio"/> Instance metric <input type="radio"/> Aggregate metric <input type="radio"/> Unspecified
Description	This KPI will measure the percentage of accepted application in comparison with the total amount of applications, using the "Approved application?" instance metric.
Business Measure Details Dashboard Samples	
▼ Target Type and Value	
The target is an exact value that the KPI should achieve.	
<input checked="" type="checkbox"/> Specify a target type and value	
Type:	Unspecified
Target value:	75% (we aim to accept 3 out of 4 applications)

Afterwards, we exported the model from the Business Modeler and imported it into the Integration Developer. The imported monitor model is only used as a skeleton and shows the documentation that is defined by the business analyst using the Business Modeler. Since the monitor model is not yet executable, it has to be enhanced using the IBM WebSphere Business Monitor development kit, which runs as a plug-in of the Integration Developer. Using this development kit, we prepared the monitor model for running on the IBM WebSphere Business Monitor.

The monitor model works in the following way: a certain event, for example, the start or end of an activity, fires a trigger, which updates a metric. This metrics is at a process instances level, while KPIs are calculated based on the metrics from multiple process instances.

For example, as shown in Figure 65, a trigger can be fired when activity “Verify application” ends and the condition “Verify_applicationEXIT/Verified = true()” holds. This condition checks if the output of the activity equals true, which indicates that the application is verified and therefore accepted. Figure 66 shows that this trigger is used by a metric “Accepted Application Counter”, which gets the value 1 when the trigger is fired. This metric is used to calculate the KPI “Accepted Applications” (see Figure 67), which sums all the values of the metric “Accepted Application Counter” and thus counts how many applications are accepted. Another KPI, “Accepted Application Percentage”, uses the “Accepted Applications” KPI and a “Total Applications” KPI to determine the percentage of accepted applications compared to the total amount of applications. The “Accepted Application Percentage” KPI is one of the KPIs that had to be defined for our case. Figure 68 shows these three KPIs in the IBM WebSphere Business Space.

Figure 65 – Defining a trigger in IBM WebSphere Integration Developer

Trigger Details
Edit the details of the trigger, which detects an occurrence and initiates an action in response.

ID: Edit...

Name:

Trigger Sources
Specify the source of this trigger.

Source Type	Source
Event	Verify applicationEXIT

Trigger Condition
Specify the condition that determines whether the trigger will fire.

Figure 66 – Defining a metric in IBM WebSphere Integration Developer

Metric Details
Edit the details of the metric, which is a holding spot for information used in other calculations.

ID: Edit...

Name:

Description:

Type:

☐ A value is required for this metric

Default Value: Edit...

☐ This metric can be used for sorting

Metric Value Expressions
Specify the expressions that set the value of the metric. If a trigger is specified, the expression is evaluated when the trigger fires.

Trigger	Expression
Accepted Application Trigger	1

Figure 67 – Defining a KPI in IBM WebSphere Integration Developer

KPI Details
Edit the details of the KPI, which is a performance measurement used to track business objectives.

ID: [Edit...](#)

Name:

Description:

Type:

Currency: Decimal Precision: ☐ Show as a percentage

☒ Keep track of historical values for this KPI

KPI Target and Ranges

KPI Definition
Specify how the value of the KPI is set.

KPI Value
Choose how the KPI will get its value:
☒ Base this KPI on a metric and an aggregation function.
☐ Write an expression to calculate this KPI based on existing KPIs

KPI Details

Monitoring context: [Browse...](#)

Metric: [Browse...](#)

Aggregation function:

Figure 68 – Accepted Applications Percentage KPI in the IBM WebSphere Business Space

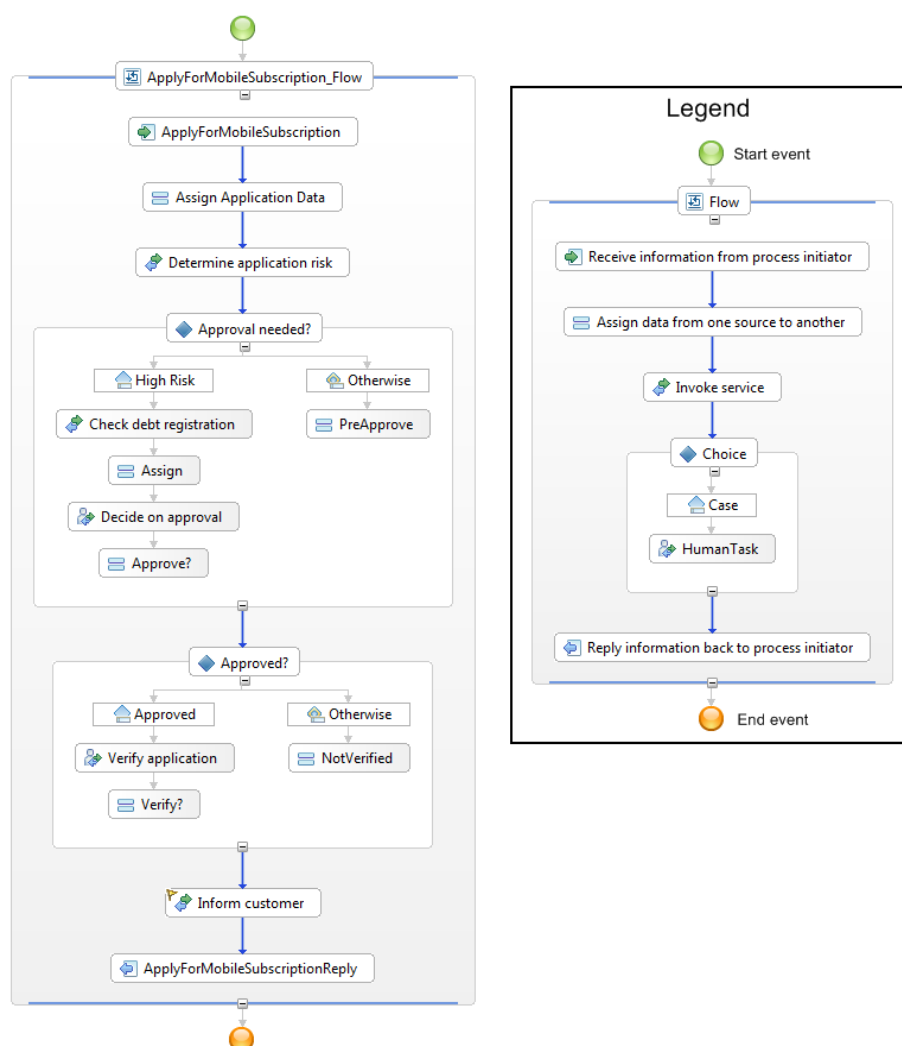


Both our KPIs can be defined in IBM WebSphere BPM in this way. As shown in the figures above, the KPIs are defined using standard forms. Information like an aggregation function between two KPIs or a trigger condition can be defined by writing code, but the available options are provided, making it only a matter of selecting the right elements in the right order.

7.4 Executable Business Process Models

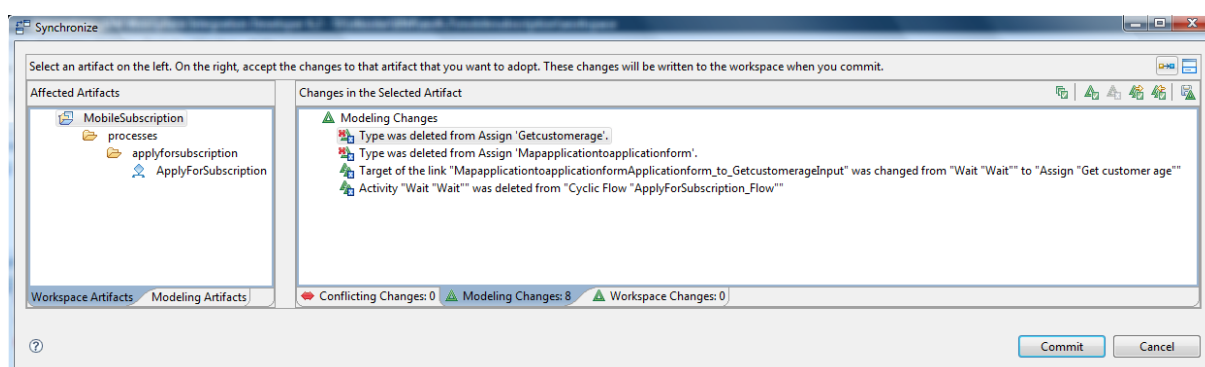
The executable business process models are defined using the IBM WebSphere Integration Developer. The business process model created in the Business Modeler is exported into a zip-archive and imported by the Integration Developer, which translates the BPMN model into an executable business process model written in the Business Process Execution Language (BPEL). The resulting executable business process model is shown in Figure 69.

Figure 69 – Executable business process model made in IBM WebSphere Integration Developer



Synchronization is a big issue in the IBM WebSphere BPM suite, since two different programs, each of them with their own model, are used for the modelling and design phases. When the business process model is imported into the Integration Developer, the business logic is placed in a separate project which should not be edited by the integration developer. He should only focus on the implementation details. These guidelines enable only one-way changes, from the business process model to the executable business process model. These changes can be synchronized by a synchronization function in the Integration Developer. This function allows the comparison between the two models and allows one to approve the desired changes. This synchronization function is shown in Figure 70.

Figure 70 – Synchronization function of the IBM WebSphere Integration Developer

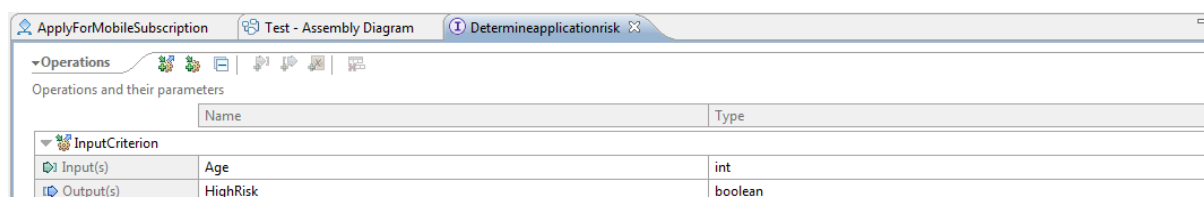


Changes in the opposite direction, from IBM WebSphere Integration Developer to the IBM WebSphere Business modeller are not supported. In a previous version (IBM WebSphere BPM 6.1) these changes could be synchronized by a similar function as described above. However, IBM decided to deprecate this function, since not all changes made in the Integration Developer in the business logic could be synchronized back to the Business Modeler.

7.5 Programming Languages

IBM WebSphere Integration Developer is used for implementing Java classes, which can be used as an application or service. The biggest part of these Java classes can be generated by the product itself, based on the input and output variables which have been defined in an interface. In our case, the DetermineApplicationRisk interface was generated, which takes the age of a customer as an input and returns the Boolean value “HighRisk” that indicates whether the application is of high risk or not. Figure 71 shows this interface.

Figure 71 – DetermineApplicationRisk interface



	Name	Type
InputCriteria		
Input(s)	Age	int
Output(s)	HighRisk	boolean

Based on the interface, a Java class is generated by the Integration Developer in which only the custom method, in our case InputCriterion, has to be implemented. Since the input and output variables are already defined, they are also added to this custom method. In our case, we implemented the method so that it picks a random number between -1000 and 0 and adds 400 to this number. If the result is positive, the method returns 0 (indicating that the applicant has no debts), otherwise the negative number is returned (showing that the applicant has debts).

7.6 Designing User Interfaces

User interfaces only have to be designed for human tasks. Based on the input and output that have been defined for these human tasks, forms can be generated. These forms can be JavaServer Faces (JSF) client applications, Portlets for the IBM WebSphere Portal server, Lotus Forms or HTML-Dojo pages (see Figure 72). Separate applications for data entry can also be used and called as web services.

Figure 72 – Generating human task user interfaces using IBM WebSphere Integration Developer

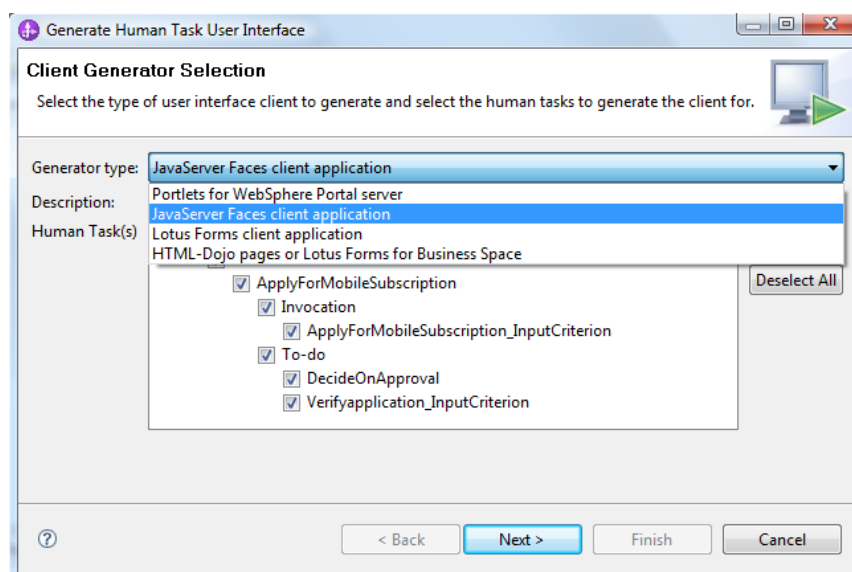


Figure 73 shows a JSF form that is generated to start the business process. It allows a customer to send an application for a mobile subscription. The JSP form is published as part of a portal, which also shows an activity list that links to other generated JSP forms. This activity list is shown in Figure 74.

Figure 73 – Generated JSF form

The screenshot shows the IBM WebSphere Process Server portal. The left sidebar contains navigation links: HOME, Business Case (with sub-links New and Status), My ToDo's (with sub-links Open and Claimed), and User: admin (with a Logout link). The main content area is titled 'Business Cases > New > ApplyForMobileSubscription > ApplyForMobileSubscription_InputCriterion'. Below the title is a subtitle: 'Enter the values for the input data and optionally provide additional information to create your task.' The form is titled 'Input Data' and contains several input fields: firstname, lastname, age, subscriptiontime, applicationtype, and email. A 'Create' button is located at the bottom of the form.

Figure 74 – Activity list as part of the generated portal

The screenshot shows the IBM WebSphere Process Server portal. The left sidebar is the same as in Figure 73. The main content area is titled 'My ToDo's > Open' with a subtitle: 'Select a task to see more details and finally claim the task.' Below the subtitle is a table with the following columns: Task Name, Description, First Activated, and Originator. The table contains five rows of tasks. Below the table is a summary bar showing 'Items found: 5', 'Page 1 of 1', and 'Items per page: 20'. A 'Refresh' button is located at the bottom of the table.

Task Name	Description	First Activated	Originator
→ ApplyForMobileSubscription\$ApplyForMobileSubscription_Decideonapproval		1/21/09 11:59:13 AM	admin
→ ApplyForMobileSubscription\$ApplyForMobileSubscription_Decideonapproval		1/21/09 11:59:39 AM	admin
→ ApplyForMobileSubscription\$ApplyForMobileSubscription_Decideonapproval		1/21/09 12:02:03 PM	admin
→ ApplyForMobileSubscription\$ApplyForMobileSubscription_Verifyapplication		1/21/09 12:00:09 PM	admin
→ ApplyForMobileSubscription\$ApplyForMobileSubscription_Verifyapplication		1/21/09 12:02:54 PM	admin

7.7 User Management

Users are defined in the IBM WebSphere Process Server. The Process Server does not support roles to order users, but users can be put together in groups. These groups have names and descriptions, and a group can contain other groups, forming a hierarchy. The groups can be used to represent the organizational structure, but can also be user-defined. Figure 75 shows the administrative console of the IBM WebSphere Process Server in which users and groups are defined.

Organizational structures defined in an LDAP directory can be used by the IBM WebSphere Process Server. As an alternative, the Process Server has its own user registry which uses the structure described above.

User capabilities cannot be defined in IBM WebSphere BPM, unless a group is created for people who have a certain capability.

Since roles are not supported, roles cannot be assigned to users. However, a task can be assigned to users or groups of users. Therefore, it is possible to let a specific employee execute all instances of a specific task (see Figure 76). However, it is not possible to let the system automatically select the employee that is most suitable for a specific human task.

Figure 75 – Defining a user in the administrative console of the IBM WebSphere Process Server

Integrated Solutions Console Welcome admin

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Resources
- Security
- Environment
- Services
- Integration Applications
- System administration
 - Users and Groups
 - Administrative User Roles
 - Administrative Group Roles
 - Manage Users
 - Manage Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Manage Users

Manage Users

User Properties

General Groups

*User ID
Employee

*First name *Last name
E Mployee

E-mail

Password Confirm password

OK Apply Cancel

Figure 76 – Assigning a task to a user using IBM WebSphere Integration Developer

ApplyForMobileSubscription *ApplyForMobileSubscription_Verifyapplication

▼ To-do Task

Name ApplyForMobileSubscription_Verifyapplication

► Service Interface

▼ People Assignment (Receiver) + X

Administrators	Everybody
Potential Owners	Users
Editors	Everybody
Readers	Everybody

Name * Stefan

▼ User Interface + X

User Interface

▼ Escalation + X

Ready Claimed Subtask started

Build Activities Properties Problems Server Logs Servers

Staff role: Potential Owners

Assign People

People assignment criteria: Users

Assigns users.
Define user name as a unique identifier

Users
Group Members
Group
Person Search
Group Search

☐ If only one person qualifies, claim task automatically.

Name	Value
Name *	Stefan
AlternativeName1	
AlternativeName2	

IBM WebSphere Integration Developer can give users four types of permissions for each human task: administrator, potential owner, editor and reader permissions (see Figure 76). Administrator permissions give the user, among others, the permission to skip the activities that are assigned to him. In our case we have given the supervisor the administrator permissions for the activity “verify application”, allowing him to skip this activity.

7.8 Informing the User

Users can either logon to a portal to see whether they need to execute an activity (by looking in their activity list, see Figure 74), or can be informed using, for example, e-mail.

In our case, we have added an escalation to the “decide on approval” activity, which informs the supervisor when a new high risk application is submitted, is awaiting approval and is not claimed for two hours (see Figure 77).

Figure 77 – Informing the supervisor through email

To-do Task

Name
ApplyForMobileSubscriptionTask1

Service Interface

People Assignment (Receiver)

Potential Owners	Group Members				
	<table border="1"> <thead> <tr> <th>GroupName *</th> <th>cn=Supervisors,o=defaultWIMFileBasedRealm</th> </tr> </thead> <tbody> <tr> <td>IncludeSubgroups *</td> <td>☒</td> </tr> </tbody> </table>	GroupName *	cn=Supervisors,o=defaultWIMFileBasedRealm	IncludeSubgroups *	☒
GroupName *	cn=Supervisors,o=defaultWIMFileBasedRealm				
IncludeSubgroups *	☒				

User Interface

Escalation

Ready → Claimed Subtask started

Inform supervisor

Build Activities | **Properties** | **Problems** | **Server Logs** | **Servers**

Escalation

Description	Expected task state: *
Details	Claimed
Assign People	Escalate after: *
Event Monitor	<input checked="" type="radio"/> 0 Days 2 Hours 0 Minutes 0 Seconds <input type="radio"/>
Global Event Settings	Notification type: E-mail
	E-mail message: DefaultMessage
	Repeat notification every: <input checked="" type="radio"/> 0 Days 0 Hours 0 Minutes 0 Seconds <input type="radio"/>
	Increase task priority: No

7.9 Technical Monitoring and Control

The administrative console of the IBM WebSphere Process Server has some technical monitoring features. It mainly allows the monitoring of the performance of the technical infrastructure. It can measure the load on the CPUs and offer advice on how to reduce it (see Figure 78). However, it can only offer advice but not execute it; this advice then has to be executed manually.

The amount of servlets that is loaded onto the server is another technical feature that can be monitored. Figure 79 shows the amount of JavaServer Faces servlets of our user interfaces loaded on the IBM WebSphere Process Server at some point in time.

Figure 78 – The administrative console providing advice on reducing the CPU load

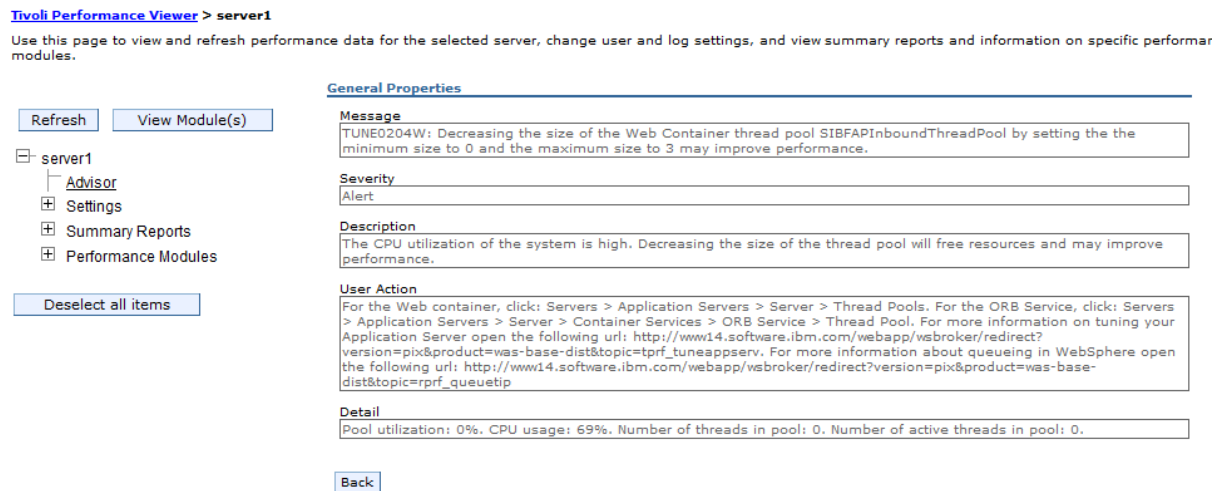
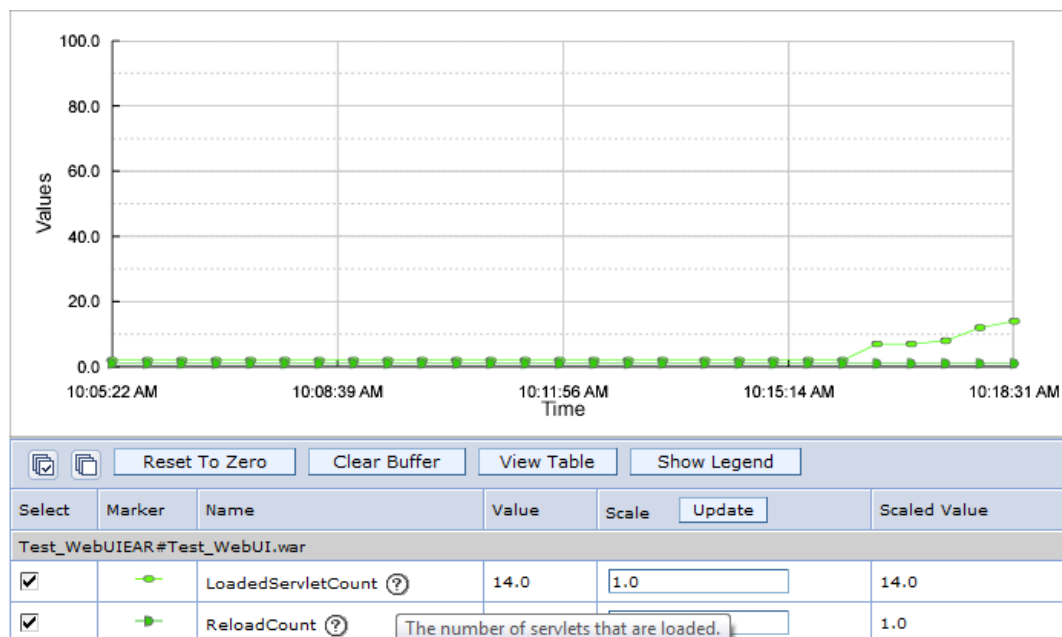


Figure 79 – The administrative console showing servlets load

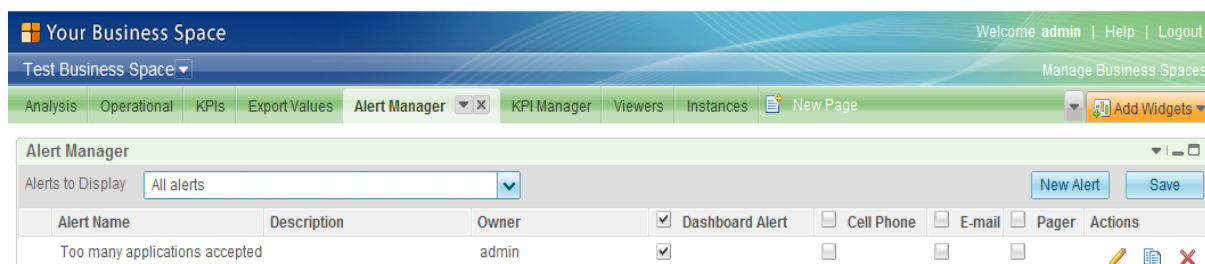


7.10 Business Monitoring

The IBM WebSphere BPM suite supports Business Activity Monitoring using the IBM WebSphere Business Space. Business Space is among others used to present the current values of the KPIs, but is also capable of predicting future values of the KPIs and sending alerts.

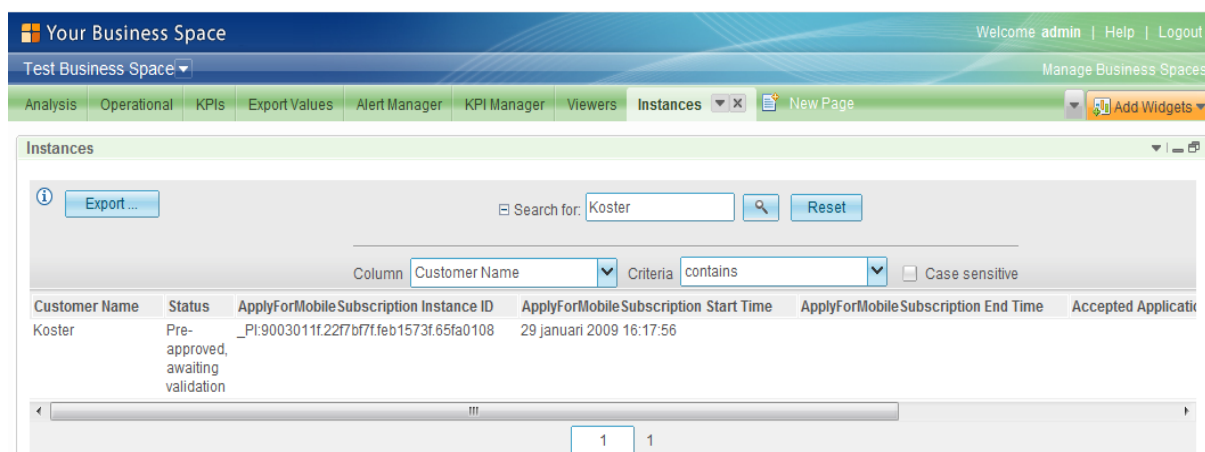
Figure 80 shows the alert manager from the IBM WebSphere Business Space, in which we have created an alert that informs someone when too many applications have been accepted. In this case, the alert is only put on the dashboard of the user, but this alert could also have been sent to a cell phone, pager or an e-mail address.

Figure 80 – Alert manager from IBM WebSphere Business Space



The status of a process instance can be retrieved using the IBM Business Process Choreography Explorer, which is a tool included in the IBM WebSphere Process Server. However, this tool does not support the use of user-defined variables to search a process instance. This is supported by the IBM WebSphere Business Space, which allows one to search the process instances using the metrics from the monitor model. If these metrics are defined correctly, they can be used to show, for example, the name of the customer and the status of his application, allowing one to passively monitor the applications. Figure 81 shows this functionality in Business Space.

Figure 81 – Passive monitoring in the IBM WebSphere Business Space



Drilling down into a higher level of detail is possible. The KPI widgets can be linked to instances widgets. For example, a KPI showing the amount of accepted applications can be linked to a list of instances, which shows these instances. The IBM WebSphere BPM offers the process view (see Figure 81) and some information that relates to the services view (see Figure 78 and Figure 79).

7.11 Business Control

Since a versioning system is in place, it is possible to have different versions of the executable business process model running on the server. Evolving the business process by changing the user that is responsible for the execution of a certain activity or adding a new activity, will only affect new process instances. These changes cannot be forced upon already running process instances.

The business rules that have been defined using the IBM WebSphere Integration Developer can be changed during runtime using the IBM Business Rules Manager running on the IBM WebSphere Process Server. The templates made during the development (see Figure 63) define the implementation of the business rules. These templates cannot be changed during runtime, but the predefined parameters used by the template can be changed anytime during execution. The user is presented with the interface shown in Figure 82, which allows him to change these parameters.

Using the Business Rules Manager we have changed the high risk business rule and lowered the age below which an application is considered high risk to 16 years instead of 18 years. This change affects new process instances and process instances which have not evaluated the business rule yet. The change cannot be forced upon process instances that have already evaluated the business rule.

Figure 82 – Changing business rules during runtime using the Business Rules Manager

Edit Mode: DetermineRisk - Rule Set

Messages:

General Information

Display Name	DetermineRisk	<input checked="" type="checkbox"/> Synchronize with the name
Last Published	Jan 21, 2009 12:05 (Local Time)	Status: Original
Description	<input type="text"/>	

Rules

Name	Display Name	Rule	Description	Action
Rule1	Rule1	HighRisk if age <= 18	<input type="text"/>	<input type="button" value="Delete"/> <input checked="" type="checkbox"/> Synchronize Name

In IBM WebSphere BPM, the implementation of an activity is called through an interface. When the implementation changes, but the interface remains the same, no changes have to be made to the executable business process model. This allows the implementation of an activity to be changed and this change to be applied to running process instances. However, changes to activities for which the interfaces also changes, result in process evolution and thus cannot be applied to running process instances.

There is no way to dynamically change the workload balance between users without changing the executable business process model or changing the groups to which the users belong.

7.12 Evaluation Summary

Table 15 summarizes our evaluation of IBM WebSphere BPM.

Table 15 – IBM WebSphere BPM evaluation summary

	Modelling criteria	Result
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	BPMN
	II. Support for interoperability between different business process modelling languages.	-
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	The functional view, as well as part of the organizational and informational view can be imported from the ARIS EPC diagram. Importing from Microsoft Visio is available.
	IV. Support for different views when modelling a business process.	The business process model shows all four views, but does not show the physical communication mechanisms.
	V. Support for different perspectives of a business process.	Two graphical notations of the business perspective in Business Modeler. IBM WebSphere BPM is capable of showing a business perspective (in the Business Modeler) and technical perspective (Integration Developer) of the process.
	VI. Support for modelling with implicit business rules.	√

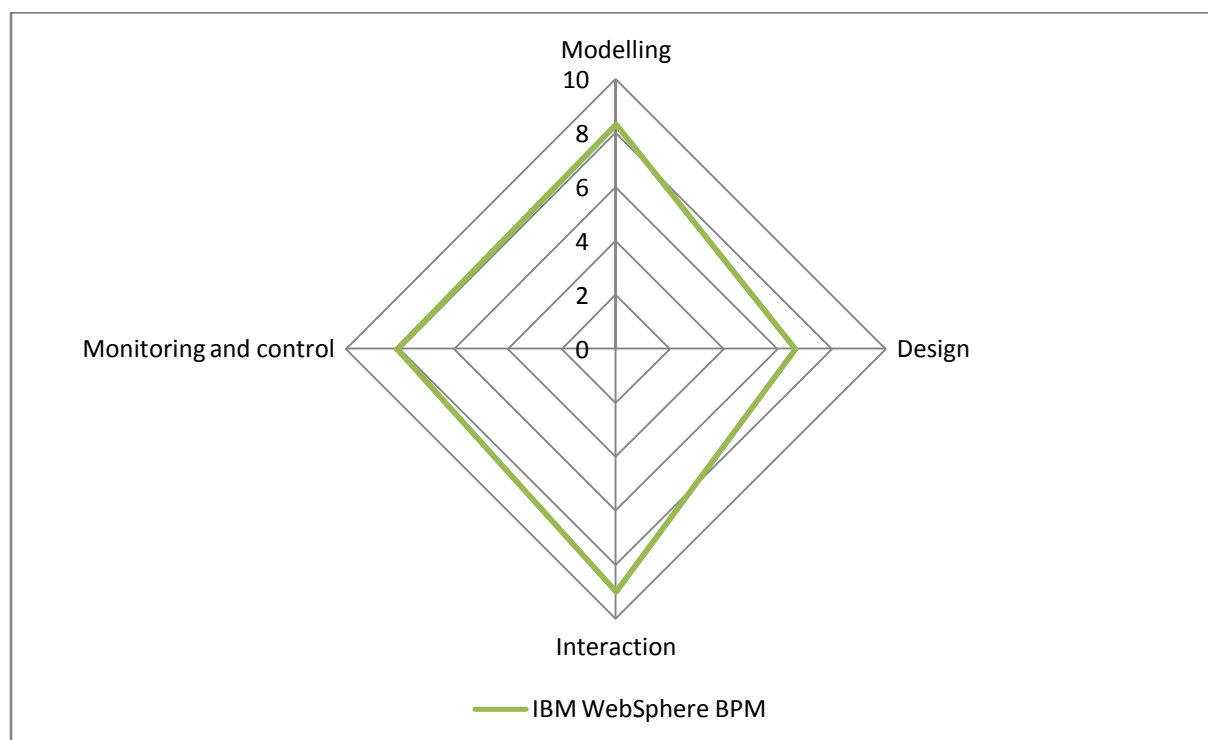
	VII.	Support for modelling with explicit business rules.	√
	VIII.	Support for modelling business processes using explicit business rules.	-
	IX.	Ease of use of defining explicit business rules.	Business rules are defined using natural language and with the help of templates.
2.	Support for KPIs.		KPI model is flexible; both our KPIs can be modelled. The KPIs are defined using code and templates.
	Design criteria		
3.	Support for creating executable business process models.		
	I.	Supported business process execution languages.	BPEL
	II.	Support for manual translation between business process modelling languages and business process execution languages.	Automatic translation is available, although it does not work always as intended, thus requiring partly manual rework.
	III.	Support for automatic translation between business process modelling languages and business process execution languages.	
	IV.	Support for linking the business process execution model with the business process model.	Only changes made in the business process model can be synchronized to the executable business process model, not the other way around.
4.	Supported programming languages for implementing services/applications.		Only Java is supported for implementing your own services and applications.
5.	Support for designing user interfaces.		Automatic generation available.
	I.	Support for separate applications for data entry.	√
	II.	Support for portal technology.	√
	III.	Support for an activity list.	√
6.	Support for user management.		
	I.	Way of describing roles (by names/description or by capabilities).	Users are described by names and grouped into groups.
	II.	Support for determining user capabilities based on organizational role.	Both organizational and user-defined groups are possible.
	III.	Support for importing organizational structure from other systems.	Import from an existing LDAP directory is possible.
	IV.	Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	-
	V.	Support for manually assigning a user to a role.	-

	VI. Support for automatically assigning a user to a role, based on his capabilities.	-
	VII. Support for activity permissions based on roles.	√
	Interaction criteria	
7.	Support for informing the user.	
	I. Support for active informing.	√
	II. Support for passive informing.	√
	Monitoring and Control criteria	
8.	Support for technical monitoring and control.	√
9.	Support for business monitoring.	
	I. Support for active monitoring.	√ (alerting through Email, dashboard, cell phone or pager)
	II. Support for passive monitoring.	√
	III. Support for various level of detail.	√
	IV. Support for different views of monitoring information.	Process view and some information from the services views.
10.	Support for business control.	
	I. Support for process instance evolution.	√ (change only affects new process instances, running process instances still follow old business process model)
	II. Support for changing business rules.	√ (a changed business rule affect all new process instances, running process instances are not re-evaluated)
	III. Support for changing activities.	√ (some changes can affect running process instances)
	IV. Support for changing the workload balance between users.	√ (executable business process model had to change)

Figure 83 shows a radar chart with IBM WebSphere BPM's rating for each of the evaluated phases. This figure shows that IBM WebSphere BPM scores medium at the design phase, and scores high at the other three phases.

Appendix D justifies the rating for each of the evaluated phases, based on the method described in section 4.4.

Figure 83 – Radar chart with rating of IBM WebSphere BPM



8 Products Comparison

This chapter compares the evaluation results of Oracle BPM Suite, Cordys BPMS and IBM WebSphere BPM. Figure 84 shows a radar chart which combines the three radar charts of the individual BPM suites from Figure 43, Figure 58 and Figure 83. Since the BPM suites in Figure 84 do not appear to differ that much we have also included a radar chart that compares BPM suites on the ten main criteria we have used (see Figure 85). This radar chart shows that the BPM suites do differ significantly when looking at a lower level. This chapter compares these BPM suites in more detail.

Figure 84 – Radar chart comparing the phases of the evaluated BPM suites

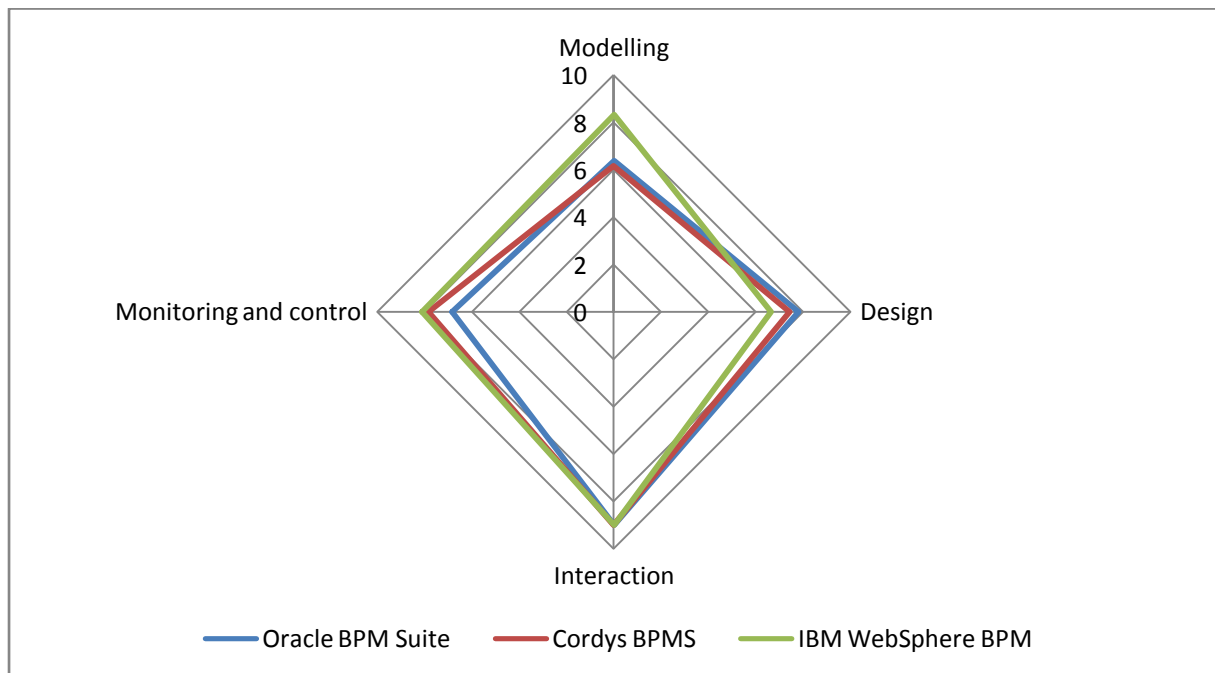
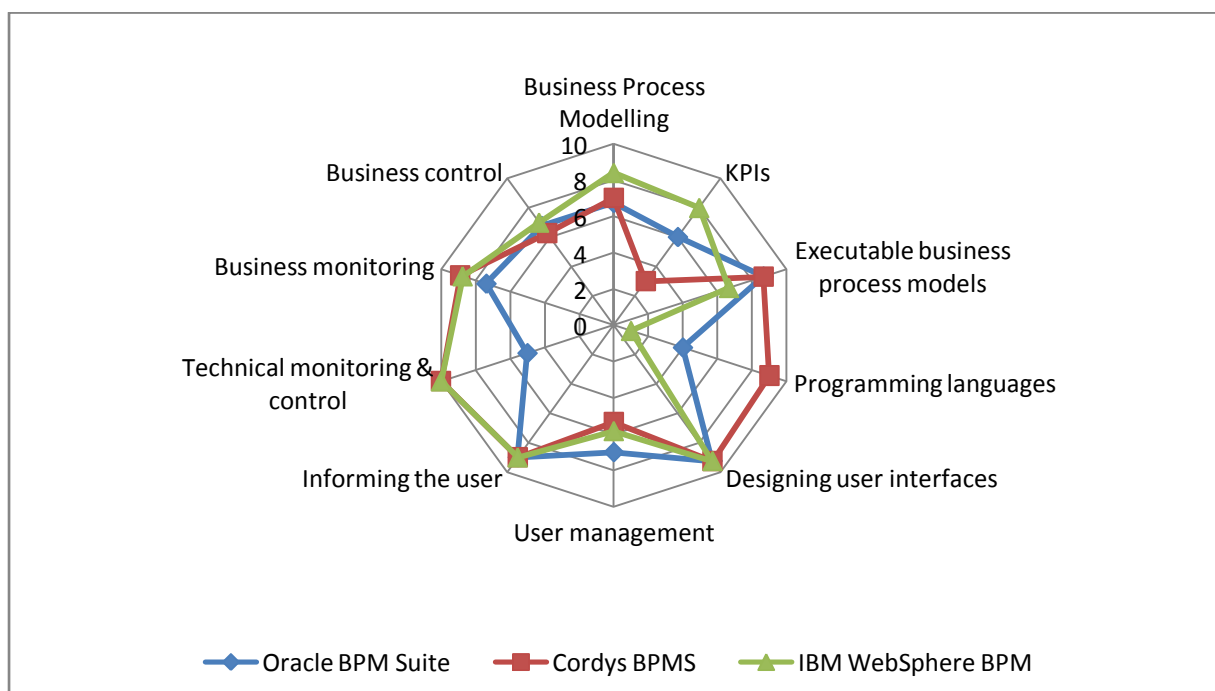


Figure 85 – Radar chart comparing the ten criteria of the evaluated BPM suites



At first sight the three BPM suites have comparable scores in the design and interaction phases. The biggest difference is in the modelling phase, where the IBM WebSphere BPM suite scores significantly higher than the other two suites. This is the result of the score of four criteria. First, IBM WebSphere BPM is the only BPM suite that imports all the information from the ARIS EPC diagram. While the other two BPM suites only import the business process, IBM WebSphere BPM can also import the data objects and the participants. Second, IBM WebSphere BPM shows more information in the business process model. Compared to the other suites, IBM WebSphere BPM also shows the physical storages and the information entities that flow through the business process. Third, IBM WebSphere BPM is the only BPM suite that has another perspective of the business process. IBM WebSphere Integration Developer shows a technical perspective of the business process model, while IBM WebSphere Business Modeler shows a business perspective. The final point that makes a difference is the support for KPIs. The IBM WebSphere BPM suite is the only one that has a KPI model that is flexible enough to let us define the KPIs we have proposed. Oracle BPM Suite came close to this functionality, but the Oracle BPM Suite KPI model was more complex to use than the one from IBM WebSphere BPM.

The Oracle BPM Suite and Cordys BPMS have about the same score for the design phase, with IBM WebSphere BPM trailing them. Oracle BPM Suite is capable of generating user interfaces and of importing the organizational structure from an existing LDAP directory, while Cordys BPMS is not. On the other hand, Cordys BPMS has a more flexible activity list, allows the use of this list in custom portals and support the most programming languages through the Eclipse platform. When comparing Oracle BPM Suite to IBM WebSphere BPM, the Oracle BPM Suite supports more programming languages and has support for user roles. IBM WebSphere BPM uses BPEL as a business process execution language, while the language used by Oracle BPM Suite is proprietary. The biggest weakness of IBM WebSphere BPM in this phase is the automatic translation between the business process model and the executable business process model, and the synchronization between these two. Both these features do not work flawlessly. Manual rework may be necessary after the translation, and synchronization only works in one way.

In the integration phase, all three BPM suites have the same total score, since they score the same on each criterion.

IBM WebSphere BPM and Cordys BPMS score close in the monitoring and control phase. Oracle BPM Suite is the only one that does not support technical monitoring and control; Oracle provides this capability in their ESB. Oracle BAM offers BAM through many channels but is not yet fully integrated with all the other products in the Oracle BPM Suite. Cordys BPMS is less flexible for changing the executable business process model, since changes only affect new process instances and cannot be forced on running process instances. On the other hand, Cordys BPMS is the only one that offers the support for external workload balancing logic, allowing the workload balance to be changed without altering the executable business process model. Further, Cordys BPMS provides the most different views of monitoring information. The active monitoring of IBM WebSphere BPM scores higher than the other two, because it is fully integrated and has the most channels through which alerts can be sent.

Table 16 summarizes the strengths and weaknesses of the evaluated BPM suites.

Table 16 – Strengths and weaknesses of the evaluated BPM suites

	Strengths	Weaknesses
Oracle BPM Suite	<ul style="list-style-type: none"> Most programming languages for implementing services and applications supported. KPI model is flexible (although not as flexible as the KPI model from IBM WebSphere BPM). Business process model and executable business process model are integrated into one model, making translation and synchronization between the models unnecessary. User interfaces can be generated automatically. 	<ul style="list-style-type: none"> Business rules can be defined using code instead of (almost) natural language, which makes it more complex than with the others. Ease of use of KPI model is low. Technical monitoring and control is done using a product outside the BPM suite. Oracle BAM not fully integrated with the other products in the BPM suite.

Cordys BPMS	<ul style="list-style-type: none"> • Ease of use of defining business rules is high. • Activity list can be used in custom portals. • Allows changing of the workload balance without changing the executable business process model. • Executable business process model is translated automatically and does not permit making changes to it, making synchronization unnecessary. • Support the most programming languages through the Eclipse platform. • Provides the most different views of monitoring information. 	<ul style="list-style-type: none"> • No import from Microsoft Visio. • Flexibility of KPI model is low. • Ease of use of KPI model is low. • User interfaces cannot always be generated automatically. • Import of organizational structure from other systems is not possible. • Changes in the executable business process model only affect new process instances.
IBM WebSphere BPM	<ul style="list-style-type: none"> • Can import all information from an ARIS EPC diagram. • Shows physical storages and information entities flowing through the business process. • Has a business and technical perspective of the business process. • Ease of use of defining business rules is high. • Most flexible KPI model. • BPEL as business process execution language. • User interfaces can be generated automatically. • Best active business monitoring. 	<ul style="list-style-type: none"> • Automatic translation does not work flawlessly, manual rework is sometimes needed. • Synchronization only works from modelling to design and not the other way around. • No support for user roles.
All suites	<ul style="list-style-type: none"> • All suites support both implicit and explicit use of business rules. • All suites have support for separate applications for data entry and provide a portal. • All suites allow describing users based on their role in the organization. • All suites allow setting activity permissions. • All suites support passive informing of the user. • All suites have support for passive monitoring. • All suites have support for drilling down from KPIs into process instances. 	<ul style="list-style-type: none"> • None of the suites allow the modelling of a business process using only business rules. • None of the suites give a choice for either forcing a new executable business process model on all process instances or only affecting new process instances.

9 Conclusions

This chapter presents the main contributions of this thesis, suggests improvements to our evaluation method, draws some relevant conclusions and identifies points where further investigation is necessary.

This chapter is structured as following: Section 9.1 summarizes the main contributions of this thesis. Section 9.2 suggests improvements to the evaluation method. Section 9.3 present our general conclusions and section 9.4 identifies some future work.

9.1 Contributions

We have proposed an evaluation method for Business Process Management products based on scientific literature. Our efforts towards this evaluation method included: a literature study on Business Process Management and a Business Process Management framework. The evaluation method itself consists of selected criteria from our framework, information on how to use these criteria, a case, which is needed to evaluate Business Process Management products based on the criteria and a rating method to quantify the evaluation of the Business Process Management products. Finally, for demonstration purposes, we have used our evaluation method to evaluate three Business Process Management suites: the Oracle BPM Suite, the Cordys BPMS and the IBM WebSphere BPM suite. We also made compared these products based on their evaluation.

The proposed evaluation method addresses the lack of no scientific evaluation methods for Business Process Management products, which hinders consultancy companies in providing objective advice for a certain Business Process Management product. The evaluations of the BPM suites and their comparison discussed in this work can be used to decide between the three Business Process Management suites.

9.2 Suggested Improvements to our Evaluation Method

When evaluating the different Business Process Management suites, no new BPM lifecycle phases came to our attention. Furthermore, no BPM lifecycle phases proved to be useless. However, the modelling phase and the design phase were not separated clearly in the suites, which gives rise to the question whether we should keep them separated in our BPM lifecycle or merge them together, as proposed by Smith and Fingar (Smith and Fingar 2003).

There are some criteria for which all three Business Process Management suites score maximum points. These criteria seem to be trivial for a Business Process Management product and can be omitted from the evaluation method, since do not differentiate these products. The criteria we suggest to omit are the following:

- Modelling criterion #2.VI: Support for modelling with implicit business rules.
- Modelling criterion #2.VII: Support for modelling with explicit business rules.
- Design criterion #6.I: Support for separate applications for data entry.
- Design criterion #6.II: Support for portal technology.
- Design criterion #7.II: Support for determining user capabilities based on organizational role.
- Design criterion #7.VII: Support for activity permissions based on roles.
- Interaction criterion #1.II: Support for passive informing.
- Monitoring and Control criterion #2.II: Support for passive monitoring.
- Monitoring and Control criterion #2.III: Support for various level of detail.

While implementing the case, some missing criteria came to our attention. These criteria are suggested below:

- Modelling criterion: Support for modelling business objects. This criterion should evaluate how business objects are modelled and the ease of use of modelling business objects.
- Design criterion: User interface generation. This criterion should be concerned with the ease of use of generating user interfaces. Many Business Process Management products are capable of generating user interfaces, but the ease of use of doing this differs significantly between them. Also the options of customizing the generated user interface should be taken into account.

- **Monitoring and Control criterion:** Timeliness of business monitoring information. Business monitoring information can be either (near) real-time (seconds to minutes, often referred to as Business Activity Monitoring), right-time (days to weeks, often referred to as Business Performance Management) or monthly (often referred to as Business Intelligence). This criterion should identify which of these three types of business monitoring is available.
- **Overall criterion:** Integration of the Business Process Management suite. Not all Business Process Management suites are equally well integrated. Some suites consist of clearly independent products, which may have been acquired from different companies. The result can be poor integration between the products. This criterion should evaluate how well the products inside a Business Process Management suite are integrated.
- **Overall criterion:** Amount of support. The evaluated Business Process Management suites differed in the amount of support that was offered. Some had many (online) documents and tutorials that allow a new user to get started with the product via self-study, while one was provided quite limited supporting information. This criterion also influences the learning curve of the Business Process Management suites, which we found to be quite differing between the evaluated Business Process Management suites.
- **Overall criterion:** Ease of use of the product. While we focus on functional criteria in our evaluation method, there are a few criteria that take into account the ease of use of some capabilities of the products, such as defining KPIs or business rules, but this is only a small portion of the capabilities of the Business Process Management suites. Since the ease of use differs significantly between the products, more ease of use criteria should be added to our evaluation method.

9.3 General conclusions

The evaluations of the three Business Process Management suites show that the evaluation method is capable distinguishing between different Business Process Management products and is not too general, which would have resulted in approximately the same scores for each of the products. However, comparing multiple products in the way we did, takes up a lot of time. It should be taken into account that we did not have any experience with the products beforehand, which required a lot of time to master them. Someone knowledgeable with the products can evaluate them in a much shorter time.

Therefore we suggest using our evaluation method as a basis for a proof of concept, in which different vendors are asked to show their product. The vendors themselves can then walk through our evaluation method, while independent people observe and rate the products based on the rating method included in our evaluation method. This should result in an objective evaluation in which all relevant points are evaluated.

For the testing of our evaluation method, we have used Business Process Management suites and not single products. This gives a good impression of the Business Process Management vendor's capabilities, but using only these suites limits the evaluation of the vendor, since the contents of these suites varies. Some vendors exclude some Business Process Management capability from their suite while they provide this capability through products that are not included in the suite. For example, Oracle provides technical monitoring, but does so through its service bus, which is not included in the Business Process Management suite. To get a better impression of the Business Process Management capabilities from a certain vendor, we suggest not to limit the scope of the evaluation method to the Business Process Management suite, but also to include other products of the vendor in the evaluation.

9.4 Further Research

The following areas have been identified for further research:

- **Map entire BPM framework to the evaluation method.** For this initial version of the method we have focused on a few of the phases from the framework and have omitted some criteria for various reasons. Our evaluation method should be extended so that it includes all criteria from the BPM framework. This will require the case to be modified and new information to be added, concerning the testing and rating of these added criteria.
- **Evaluate more products.** In our demonstration we have evaluated three Business Process Management suites. Performing more evaluations based on our evaluation method will produce more feedback which can be used to improve the evaluation method.
- **Validate the rating method.** The rating method included in our evaluation method is based on our understanding of the Business Process Management field. In further research this rating method can be validated by Business Process Management experts.

- Extension of the framework. The Business Process Management field is constantly moving and new topics are added to it. These new topics should also be present in the framework and in the evaluation method. Therefore the framework and the evaluation method have to be updated once in a while. Examples of new topics that are currently emerging in the Business Process Management field include:
 - Complex Event Processing (CEP). CEP deals with the processing of (simple) events from multiple systems and tries to identify meaningful complex events that are made up of multiple simple events. The processing of these events can result in an automatic reaction of the Business Process Management product or the involvement of a user.
 - Human Interaction Management (HMI). HMI is a set of management principles, patterns and techniques that provide process-based support for innovative, adaptive, collaborative human work. Currently, Business Process Management focuses on simple, automatic processes or on processes which involve some human participation. HMI, on the contrary, focuses on the complex processes that are mainly executed by humans.
- Comparing the evaluation criteria with those of market research companies. Gartner and Forrester are two well respected market research companies which make their own comparison of Business Process Management suites. These companies are not completely open. However, they do share the criteria that they use for their comparison. In another study, the criteria from our evaluation method can be compared to those of Gartner and Forrester, in order to improve our evaluation method.

List of Definitions

Business process	A business process is a set of linked and related activities (organizational tasks) which are designed to take an input and to transform it into a specific output.
Business process execution environment	The business process execution environment consist of a business process execution engine, which is able to execute the executable business process models, ways of interaction, which users can use to interact with the executable business process models, and management functionality, which is used to manage the environment and the executable business process models.
Business Process Management (BPM)	Business Process Management is a management discipline that takes a systematic, structured approach to support explicit process management using methods, techniques and tools, involving humans, organizations, applications, documents and other sources of information, with the aim of reaching the organization's objectives by aligning the business processes with its objectives.
Business process model	A business process model is an explicit representation of a business process. The business process model is formal if it is modelled in a business process modelling language.
Executable business process model	The executable business process model contains technical data necessary for the execution. It is modelled using a business process execution language.
Process instance	The process instance is an executable business process model which is executed. One executable business process model can relate to multiple process instances.

List of References

- Alves, A., A. Arkin, et al. (2007). "Web Services Business Process Execution Language Version 2.0 (OASIS Standard). WS-BPEL TC OASIS." Retrieved September 28th, 2008, from <http://docs.oasis-open.org/wsbpel/2.0/wsbpelv2.0.html>.
- Andrews, T., F. Curbera, et al. (2003). "Business Process Execution Language for Web Services, Version 1.1." Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation.
- BEA. (2008). "BEA AquaLogic® BPM." Retrieved June 18th, 2008, from <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/aqualogic/albpm/>.
- Brambilla, M., S. Ceri, et al. (2005). Exception handling in workflow-driven Web applications. Proceedings of the 14th international conference on World Wide Web. Chiba, Japan, ACM.
- Business Rule Group. (2003, November 1st). "Business Rules Manifesto." Version 2.0. Retrieved October 7th, 2008, from <http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf>.
- Castellanos, M., F. Casati, et al. (2005). iBOM: a platform for intelligent business operation management. Proceedings of the Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on.
- Chang, J. F. (2005). Business Process Management Systems: Strategy and Implementation. Boca Raton, FL, USA, Auerbach Publications.
- Chen, M., D. Zhang, et al. (2007). "Empowering collaborative commerce with Web services enabled business process management systems." Decision Support Systems **43**(2): 530-546.
- Chiu, D. K. W., Q. Li, et al. (1999). "A meta modeling approach to workflow management systems supporting exception handling." Information Systems **24**(2): 159-184.
- Cook, J. E. and A. L. Wolf (1995). Automating process discovery through event-data analysis. Proceedings of the 17th international conference on Software engineering. Seattle, Washington, United States, ACM.
- Cordys. (2008). "Business Activity Monitoring (BAM) - Cordys." Retrieved December 20th, 2008, from http://www.cordys.com/cordyscms.com/business_activity_monitoring.php.
- Cordys. (2008). "Business Process Management Suite (BPMS)." Retrieved December 4th, 2008, from http://www.cordys.com/cordyscms.com/cordys_bpms.php.
- Curtis, B., K. Marc, et al. (1992). "Process modeling." Communications of the ACM **35**(9): 75-90.
- Davenport, T. (1993). Process Innovation - Reengineering Work through Information Technology. Boston, MA, USA, Harvard Business School.
- Dayal, U., M. Hsu, et al. (2001). Business Process Coordination: State of the Art, Trends, and Open Issues. Proceedings of the 27th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc.
- Desel, J. and J. Esparza (1995). Free choice petri nets. Cambridge, UK, Cambridge University Press.
- Eder, J. and W. Liebhart (1996). Workflow recovery. Proceedings of the Cooperative Information Systems, 1996. Proceedings., First IFCIS International Conference on.
- Elzinga, D. J., T. Horak, et al. (1995). "Business process management: survey and methodology." Engineering Management, IEEE Transactions on **42**(2): 119-128.
- Franch, X. and J. M. Ribó (2003). A UML-Based Approach to Enhance Reuse within Process Technology. In: Software Process Technology. Berlin, Heidelberg, Springer. **2786/2003**: 74-93.
- Fülscher, J. and S. G. Powell (1999). "Anatomy of a process mapping workshop." Business Process Management Journal **5**(3).
- Gartner Inc. (2008). "Business Process Management." Retrieved September 8th, 2008, from <http://www.gartner.com/it/products/research/topics/topics.jsp#BPM>.
- Gillot, J.-N. (2008). The Complete Guide To Business Process Management, Independently published.
- Golfarelli, M., S. Rizzi, et al. (2004). Beyond Data Warehousing: What's Next in Business Intelligence? DOLAP'04. Washington, DC, USA.
- Greenfield, P., A. Fekete, et al. (2003). Compensation is not enough. Proceedings of the Enterprise Distributed Object Computing Conference, 2003. Proceedings. Seventh IEEE International.
- Grigori, D., F. Casati, et al. (2004). "Business Process Intelligence." Computers in Industry **53**(3): 321-343.
- Gruhn, V. and R. Laue (2006). Complexity metrics for business process models. In: Business Information Systems, 9th International Conference on Business Information Systems. W. Abramowicz and H. C. Mayr. Klagenfurt, Austria, GI: 1-12.
- Hammer, M. and J. Champy (1993). Reengineering the Cooperation: A Manifesto for Business Revolution. New York, NY, USA, Harper Business.
- Hill, J. B., M. Cantara, et al. (2009). "Magic Quadrant for Business Process Management Suites", R3005 02192010, Gartner Inc.

- Hunt, V. D. (1996). Process Mapping. New York, NY, USA, John Wiley & Sons.
- Hwang, S.-Y. and W.-S. Yang (2002). "On the discovery of process models from their instances." Decision Support Systems **34**(1): 41-57.
- IBM. (2009). "IBM - WebSphere Business Process Management." Retrieved January 8th, 2009, from <http://www-01.ibm.com/software/websphere/products/businessint/>.
- Jin, L. J., F. Casati, et al. (2001). "Business Process Simulation with HP Process Manager." Simulation Series **34**: 291-295.
- Johansson, H. J., P. McHugh, et al. (1993). Business Process Reengineering: BreakPoint Strategies for Market Dominance. Chichester, UK, John Wiley & Sons.
- Johnson, H. T. (1988). "A blueprint for world-class management accounting." Manage. Account. **72**(June): 23-30.
- Kalle, L., M. Lars, et al. (1998). "Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches." Information Systems Research **9**(3): 275-301.
- Leymann, F. (2003). Web Services: Distributed Applications Without Limits -- An Outline. Proc Database Systems for Business, Technology and Web, BTW 2003. Leipzig, Germany, Springer.
- Leymann, F., D. Roller, et al. (2002). "Web services and business process management." IBM Systems Journal **41**(2).
- List, B. and B. Korherr (2006). An evaluation of conceptual business process modelling languages. Proceedings of the 2006 ACM symposium on Applied computing. New York, ACM. **SAC '06**.
- List, B., J. Schiefer, et al. (2002). Multidimensional Business Process Analysis with the Process Warehouse. In: Knowledge Discovery for Business Information Systems: 211-227.
- Mărușter, L., A. J. M. M. Weijters, et al. (2006). "A Rule-Based Approach for Process Discovery: Dealing with Noise and Imbalance in Process Logs." Data Mining and Knowledge Discovery **13**(1): 67-87.
- Metastorm. (2008). "Metastorm website." Retrieved 11th November, 2008, from <http://www.metastorm.com>.
- Nuno Melão, M. P. (2000). "A conceptual framework for understanding business processes and business process modelling." Information Systems Journal **10**(2): 105-129.
- Oracle. (2008). "Oracle Business Activity Monitoring." Retrieved November 16th, 2008, from <http://www.oracle.com/technology/products/integration/bam/index.html>.
- Oracle. (2008). "Oracle Business Process Management Suite." Retrieved August 19th, 2008, from <http://www.oracle.com/technologies/bpm/bpm-suite.html>.
- Ouyang, C., M. Dumas, et al. (2007). "Pattern-based translation of BPMN process models to BPEL web services." International Journal of Web Services Research (JWSR), **5**(1): 42-62.
- Papazoglou, M. P. and P. M. A. Ribbers (2006). e-Business. Chichester, UK, John Wiley & Sons Ltd.
- Pasley, J. (2005). "How BPEL and SOA are changing Web services development." Internet Computing, IEEE **9**(3): 60-67.
- Petzmann, A., M. Puncoschar, et al. (2007). Applying MDA Concepts to Business Process Management. In: 2007 BPM and Workflow Handbook: Methods, Concepts, Case Studies and Standards. L. Fischer. Lighthouse Point, FL, USA, Future Strategies Inc.
- Pritchard, J. P. and C. Armistead (1999). "Business process management - lessons from European business." Business Process Management Journal **5**: 10-35.
- Recker, J. C. and J. Mendling (2006). On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In: Proceedings 18th International Conference on Advanced Information Systems Engineering: 521-532.
- Ricken, J. (2007). Top-Down Modeling Methodology for Model-Driven SOA Construction. In: On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. Berlin, Heidelberg, Springer. **4805/2007**: 323-332.
- Ross, R. G. (2003). Principles of the Business Rule Approach. Boston, MA, USA, Addison-Wesley Professional.
- Russell, N., A. H. M. ter Hofstede, et al. (2004). "Workflow Data Patterns", QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane.
- Russell, N., A. H. M. ter Hofstede, et al. (2004). "Workflow Resource Patterns", BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven.
- Russell, N., A. H. M. ter Hofstede, et al. (2003). "Workflow Control-Flow Patterns: A Revised View", BPM Center Report BPM-06-22, BPMcenter.org.
- Russell, N., W. M. P. van der Aalst, et al. (2006). "Exception Handling Patterns in Process-Aware Information Systems", BPM Center Report BPM-06-04, BPMcenter.org.
- Savvion. (2008). "Savvion." Retrieved November 11th, 2008, from <http://www.savvion.com/>.
- Sayal, M., F. Casati, et al. (2002). Business Process Cockpit. Proceedings of the 28th international conference on Very Large Data Bases. Hong Kong, China, VLDB Endowment.
- Smith, H. and P. Fingar (2003). Business Process Management (BPM): The Third Wave. Tampa, FL, USA, Meghan-Kiffer Press.

- Smith, H. and P. Fingar. (2004). "Workflow is just a Pi process." Retrieved 14th November, 2008, from <http://www.bptrends.com/publicationfiles/01-04%20Workflow%20is%20just%20a%20Pi%20Process%20Smith-Fingar.pdf>.
- The Eclipse Foundation. (2008). "Eclipse.org." Retrieved December 4th, 2008, from <http://www.eclipse.org>.
- Tibco. (2008). "iProcess Suite." Retrieved June 18th, 2008, from http://www.tibco.com/solutions/bpm/iprocess_suite.jsp.
- TIOBE Software BV. (2009). "TIOBE Software: Tiobe Index." Retrieved January 7th, 2009, from <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- van der Aalst, W. M. P., F. Leymann, et al. (2007). "The role of business processes in service oriented architectures (Editorial)." International Journal of Business Process Integration and Management 2: 75-80.
- van der Aalst, W. M. P., A. ter Hofstede, et al. (2003). Business Process Management: A Survey. In: Business Process Management: 1019-1019.
- van der Aalst, W. M. P., B. F. van Dongen, et al. (2003). "Workflow mining: A survey of issues and approaches." Data & Knowledge Engineering 47(2): 237-267.
- Wang, M. and H. Wang (2006). "From process logic to business logic--A cognitive approach to business process management." Information & Management 43(2): 179-193.
- Warner, B. W. and W. Trahan (2000). Business Climate Shifts: Profiles of Change Makers. Boston, MA, USA, Butterworth Heinemann.
- Weske, M. (2007). Business Process Management: Concepts, Languages, Architectures. Berlin, Heidelberg, Springer.
- Weske, M., W. M. P. van der Aalst, et al. (2004). "Advances in business process management." Data & Knowledge Engineering 50(1): 1-8.
- White, S. A. (2004). "Introduction to BPMN." Retrieved November 11th, 2008, from <http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>.
- Zhuge, H. (2002). "A process matching approach for flexible workflow process reuse." Information and Software Technology 44(8): 445-450.
- zur Muehlen, M. (2001). Process-driven Management Information Systems - Combining Data Warehouses and Workflow Technology. Proceedings of the Fourth International Conference on Electronic Commerce Research (ICECR-4), Dallas, TX.
- zur Muehlen, M. and M. Rosemann (2000). Workflow-based Process Monitoring and Controlling - Technical and Organizational Issues. Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii.

List of Figures

Figure 1 – Problem bundle	1
Figure 2 – Research approach.....	2
Figure 3 – BPM lifecycle	5
Figure 4 – BPM lifecycle in other notation	6
Figure 5 – Process management lifecycle by (Petzmann, Puncochar et al. 2007).....	7
Figure 6 – Process lifecycle by (Smith and Fingar 2003).....	7
Figure 7 – BPM lifecycle with highlighted phases	9
Figure 8 – Strategy development phase.....	10
Figure 9 – Linking objectives to high-level business processes, picture adapted from (Ricken 2007)	10
Figure 10 – Discovery phase	11
Figure 11 – ARIS business architect	11
Figure 12 – Process mining.....	12
Figure 13 – Modelling phase.....	14
Figure 14 – Example of a simple business process modelled in BPMN (White 2004)	15
Figure 15 – Implicit use of business rules	16
Figure 16 – Explicit use of business rules	16
Figure 17 – Design phase	18
Figure 18 – Example of a part of a business process modelled in BPEL (graphical representation and the underlying BPEL code).....	19
Figure 19 – Deployment phase.....	22
Figure 20 – Operation phase.....	23
Figure 21 – Execution sub-phase.....	23
Figure 22 – Interaction phase	26
Figure 23 – Example of a BPM portal (Metastorm 2008).....	26
Figure 24 – Monitoring and Control phase.....	28
Figure 25 – Example of a monitoring dashboard (Savvion 2008)	29
Figure 26 – EPC diagram of the mobile subscription application business process.....	33
Figure 27 – Gartner’s Magic Quadrant for BPM suites (Hill, Cantara et al. 2009).....	39
Figure 28 – Tools from the Oracle BPM Suite mapped to our BPM lifecycle	40
Figure 29 – Business process modelled in Oracle BPM Studio.....	41
Figure 30 – Imported business process	41
Figure 31 – Behavioural view of the business process	42
Figure 32 – Informational view of the business process	42
Figure 33 – Business process in UML notation.....	43
Figure 34 – Implicit business rules in Oracle BPM Studio.....	43
Figure 35 – Explicit business rules in Oracle BPM Studio.....	44
Figure 36 – Explicit business rules in Oracle BPM Studio.....	44
Figure 37 – Risk KPI in Oracle BPM Workspace	45
Figure 38 – Using a Java class in Oracle BPM Studio	46
Figure 39 – Using the Oracle BPM Workspace.....	46
Figure 40 – Defining roles in Oracle BPM.....	47
Figure 41 – Searching the status of an application.....	48
Figure 42 – Changing business rules during runtime in the BPM Workspace.....	48
Figure 43 – Radar chart with rating of the Oracle BPM Suite.....	51
Figure 44 – Cordys BPMS components (Cordys 2008)	52
Figure 45 – Components from Cordys BPMS mapped to our BPM lifecycle	53
Figure 46 – Business process modelled in Cordys Business Process Designer	53
Figure 47 – Imported business process from ARIS into Cordys	54
Figure 48 – Behavioural view of the business process in Cordys	54
Figure 49 – Organizational view of the business process in Cordys	54
Figure 50 – Defining explicit business rules in Cordys.....	55
Figure 51 – Cordys KPIComposer.....	56
Figure 52 – Cordys XForm editor with preview	57
Figure 53 – Role management in Cordys	57
Figure 54 – My Inbox of the Cordys portal.....	58
Figure 55 – Monitoring SOAP processors.....	58
Figure 56 – Monitoring the web gateway.....	58

Figure 57 – Passive monitoring of business processes	59
Figure 58 – Radar chart with rating of Cordys BPMS	62
Figure 59 – IBM WebSphere BPM products mapped to our BPM lifecycle.....	63
Figure 60 – Business process modelled in IBM WebSphere Business Modeler.....	64
Figure 61 – ARIS information imported into IBM WebSphere Business Modeler	64
Figure 62 – IBM WebSphere Business Modeler alternative notation	65
Figure 63 – Defining business rules in IBM WebSphere Integration Developer	66
Figure 64 – Adding a business measure in IBM WebSphere Business Modeler	66
Figure 65 – Defining a trigger in IBM WebSphere Integration Developer	67
Figure 66 – Defining a metric in IBM WebSphere Integration Developer.....	67
Figure 67 – Defining a KPI in IBM WebSphere Integration Developer	68
Figure 68 – Accepted Applications Percentage KPI in the IBM WebSphere Business Space.....	68
Figure 69 – Executable business process model made in IBM WebSphere Integration Developer	69
Figure 70 – Synchronization function of the IBM WebSphere Integration Developer	69
Figure 71 – DetermineApplicationRisk interface	70
Figure 72 – Generating human task user interfaces using IBM WebSphere Integration Developer	70
Figure 73 – Generated JSF form.....	71
Figure 74 – Activity list as part of the generated portal	71
Figure 75 – Defining a user in the administrative console of the IBM WebSphere Process Server.....	72
Figure 76 – Assigning a task to a user using IBM WebSphere Integration Developer	72
Figure 77 – Informing the supervisor through email.....	73
Figure 78 – The administrative console providing advice on reducing the CPU load	74
Figure 79 – The administrative console showing servlets load.....	74
Figure 80 – Alert manager from IBM WebSphere Business Space	75
Figure 81 – Passive monitoring in the IBM WebSphere Business Space.....	75
Figure 82 – Changing business rules during runtime using the Business Rules Manager.....	76
Figure 83 – Radar chart with rating of IBM WebSphere BPM.....	79
Figure 84 – Radar chart comparing the phases of the evaluated BPM suites	80
Figure 85 – Radar chart comparing the ten criteria of the evaluated BPM suites	80

List of Tables

Table 1 – Strategy development criteria.....	10
Table 2 – Discovery criteria	13
Table 3 - Harvesting and reuse mechanisms (Franch and Ribó 2003)	14
Table 4 – Business process modelling using only business rules.....	17
Table 5 – Modelling criteria.....	17
Table 6 – Design criteria	21
Table 7 – Deployment criteria	22
Table 8 – Execution criteria	25
Table 9 – Interaction criteria	27
Table 10 – Monitoring and Control criteria.....	30
Table 11 – Analysis criteria.....	31
Table 12 – Selected criteria and the method of rating.....	36
Table 13 – Oracle BPM Suite evaluation summary	49
Table 14 – Cordys BPMS evaluation summary.....	60
Table 15 – IBM WebSphere BPM evaluation summary	76
Table 16 – Strengths and weaknesses of the evaluated BPM suites.....	81

Appendix A: Criteria used by our method

	Strategy development criteria
1.	Support for capturing the organization's objectives (for example, using a Balanced scorecard).
2.	Support for creating a high-level overview of the business processes (for example, using a value-added chain diagram).
3.	Support for linking the organization's objectives with the high-level business processes.
	Discovery criteria
1.	Support for process mapping (manual discovery of business processes) <ul style="list-style-type: none"> I. Support for popular techniques and notations (for example, flowcharting and cause-and-effect diagrams). II. Support for popular methods like IDS Sheer's the ARIS methodology and CSC's Catalyst.
2.	Support for process mining <ul style="list-style-type: none"> I. Ability to handle incomplete logs. II. Ability to handle noise. III. Ability to handle duplicate activity names. IV. Types of structures supported (graph or block). V. Ability to handle timestamps. VI. Ability to handle basic parallelism. VII. Ability to handle non-free choice constructs. VIII. Ability to handle basic loops. IX. Ability to handle arbitrary loops.
3.	Support for defining a process architecture.
	Modelling criteria
1.	Support for harvesting/reuse. <ul style="list-style-type: none"> I. Support for retrieval methods (exact/inexact). II. Support for harvesting/reuse mechanisms. III. Ability to handle consistency problems. IV. Ability to handle redundancy problems. V. Ability to handle residue problems.
2.	Support for Business Process Modelling. <ul style="list-style-type: none"> I. Support for different business process modelling languages (for example, AD, BPDM, BPMN, EDPC, IDEF3, Petri Nets, RAD and BPEL4WS). II. Support for interoperability between different business process modelling languages. III. Support for interoperability between simple process modelling methods and business process modelling languages. IV. Support for different views when modelling a business process. V. Support for different perspectives of a business process. VI. Support for modelling with implicit business rules. VII. Support for modelling with explicit business rules. VIII. Support for modelling business processes using explicit business rules. IX. Ease of use of defining explicit business rules.

3.	Support for KPIs. I. Flexibility of KPI model. II. Ease of use of defining KPIs.
	Design criteria
1.	Support for creating executable business process models. I. Supported business process execution languages. II. Support for manual translation between business process modelling languages and business process execution languages. III. Support for automatic translation between business process modelling languages and business process execution languages. IV. Support for linking the business process execution model with the business process model.
2.	Support for adding SLA and linking SLA KPIs to business processes.
3.	Supported exception handling methods (for example, ACID, Saga or compensation handlers).
4.	Support for integration. I. Underlying middleware technology. II. Support for orchestration. III. Support for choreography.
5.	Supported programming languages for implementing services/applications.
6.	Support for designing user interfaces. I. Support for separate applications for data entry. II. Support for portal technology. III. Support for an activity list.
7.	Support for user management. I. Way of describing roles (by names/description or by capabilities). II. Support for determining user capabilities based on organizational role. III. Support for importing organizational structure from other systems. IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities). V. Support for manually assigning a user to a role. VI. Support for automatically assigning a user to a role, based on his capabilities. VII. Support for activity permissions based on roles.
	Deployment criteria
1.	Support for distributed execution I. Support for multiple business process execution engines. II. Support for distributed business process execution engines.
2.	Support for workflow resource patterns.
3.	Support for informing the user of a new executable business process model.
	Execution criteria
1.	Support for optimized execution according to some measurable criteria.
2.	Efficiency of KPI computation.

3.	Support for process state persistency.
4.	Supported modes of routing.
5.	Support for calendar management.
6.	<p>Ability to cope with failures.</p> <ul style="list-style-type: none"> I. Ability to cope with business process engine failures. II. Ability to cope with activity failures. III. Ability to cope with system failures. IV. Ability to cope with semantic failures. V. Ability to cope with behavioural failures. VI. Ability to cope with communication failures.
7.	Compliance with workflow exception patterns.
8.	Compliance with workflow control-flow patterns.
9.	Compliance with workflow data patterns.
10.	Support for version management.
Interaction criteria	
1.	<p>Support for informing the user.</p> <ul style="list-style-type: none"> I. Support for active informing. II. Support for passive informing.
2.	<p>Support for exception handling.</p> <ul style="list-style-type: none"> I. Support for synchronous notification of exceptions. II. Support for asynchronous notification of exceptions. III. Support for providing suggestions when handling exceptions. IV. Support for saving exception handling solution. V. Support for exception handling while maintaining execution behaviour. VI. Support for exception handling while modifying execution behaviour.
Monitoring and Control criteria	
1.	Support for technical monitoring and control.
2.	<p>Support for business monitoring.</p> <ul style="list-style-type: none"> I. Support for active monitoring. II. Support for passive monitoring. III. Support for various level of detail. IV. Support for different views of monitoring information.
3.	<p>Support for business control.</p> <ul style="list-style-type: none"> I. Support for prediction of unwanted situations. II. Support for process instance evolution. III. Support for explicit procedural exception handlers. IV. Support for changing business rules. V. Support for changing activities. VI. Support for changing the workload balance between users.

	Analysis criteria
1.	Support for modelling-time testing. <ul style="list-style-type: none"> I. Support for verification. II. Support for verifying compliance with rules. III. Support for traditional simulation. IV. Support for innovative simulation.
2.	Supported types of historical data available for analysis (execution logs, computed KPIs, application data, status of the IT infrastructure).
3.	Support for simple visual data representation tools (cost-time profiles, pie charts, histograms, Pareto charts, and bar graphs).
4.	Support for complex methods of quantification (traditional accounting or activity-based costing).
5.	Support for defining processes in taxonomies.
6.	Support for suggestions on improvement.

Appendix B: Oracle BPM Suite rating

	Modelling phase: 6.37	60.5 out of 95 points
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	8
	II. Support for interoperability between different business process modelling languages.	
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	8
	IV. Support for different views when modelling a business process.	7.5
	V. Support for different perspectives of a business process.	1
	VI. Support for modelling with implicit business rules.	10
	VII. Support for modelling with explicit business rules.	10
	VIII. Support for modelling business processes using explicit business rules.	0
	IX. Ease of use of defining explicit business rules.	6
2.	Support for KPIs.	6 for our own KPIs 6 for the ease of use
	Design phase: 7.79	109 out of 140 points
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	6
	II. Support for manual translation between business process modelling languages and business process execution languages.	10
	III. Support for automatic translation between business process modelling languages and business process execution languages.	
	IV. Support for linking the business process execution model with the business process model.	10
4.	Supported programming languages for implementing services/applications.	4
5.	Support for designing user interfaces.	10
	I. Support for separate applications for data entry.	10
	II. Support for portal technology.	10
	III. Support for an activity list.	7
6.	Support for user management.	
	I. Way of describing roles (by names/description or by capabilities).	5
	II. Support for determining user capabilities based on organizational role.	10
	III. Support for importing organizational structure from other systems.	10

	IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	0
	V. Support for manually assigning a user to a role.	7
	VI. Support for automatically assigning a user to a role, based on his capabilities.	
	VII. Support for activity permissions based on roles.	10
Interaction phase: 9.00		18 out of 20 points
7.	Support for informing the user.	
	I. Support for active informing.	8
	II. Support for passive informing.	10
Monitoring and Control phase: 6.83		61.5 out of 90 points
8.	Support for technical monitoring and control.	5
9.	Support for business monitoring.	
	I. Support for active monitoring.	7
	II. Support for passive monitoring.	10
	III. Support for various level of detail.	10
	IV. Support for different views of monitoring information.	2.5
10.	Support for business control.	
	I. Support for process instance evolution.	8
	II. Support for changing business rules.	7
	III. Support for changing activities.	8
	IV. Support for changing the workload balance between users.	4

Appendix C: Cordys BPMS rating

	Modelling phase: 6.16	58.5 out of 95 points
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	8
	II. Support for interoperability between different business process modelling languages.	
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	6
	IV. Support for different views when modelling a business process.	7.5
	V. Support for different perspectives of a business process.	1
	VI. Support for modelling with implicit business rules.	10
	VII. Support for modelling with explicit business rules.	10
	VIII. Support for modelling business processes using explicit business rules.	0
	IX. Ease of use of defining explicit business rules.	10
2.	Support for KPIs.	0 for our own KPIs 6 for the ease of use
	Design phase: 7.43	104 out of 140 points
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	6
	II. Support for manual translation between business process modelling languages and business process execution languages.	10
	III. Support for automatic translation between business process modelling languages and business process execution languages.	
	IV. Support for linking the business process execution model with the business process model.	10
4.	Supported programming languages for implementing services/applications.	9
5.	Support for designing user interfaces.	7
	I. Support for separate applications for data entry.	10
	II. Support for portal technology.	10
	III. Support for an activity list.	10
6.	Support for user management.	
	I. Way of describing roles (by names/description or by capabilities).	5
	II. Support for determining user capabilities based on organizational role.	10
	III. Support for importing organizational structure from other systems.	0

	IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	0
	V. Support for manually assigning a user to a role.	7
	VI. Support for automatically assigning a user to a role, based on his capabilities.	
	VII. Support for activity permissions based on roles.	10
	Interaction phase: 9	18 out of 20 points
7.	Support for informing the user.	
	I. Support for active informing.	8
	II. Support for passive informing.	10
	Monitoring and Control phase: 7.83	70.5 out of 90 points
8.	Support for technical monitoring and control.	10
9.	Support for business monitoring.	
	I. Support for active monitoring.	8
	II. Support for passive monitoring.	10
	III. Support for various level of detail.	10
	IV. Support for different views of monitoring information.	7.5
10.	Support for business control.	
	I. Support for process instance evolution.	4
	II. Support for changing business rules.	7
	III. Support for changing activities.	4
	IV. Support for changing the workload balance between users.	10

Appendix D: IBM WebSphere BPM rating

	Modelling phase: 8.32	79 out of 95 points
1.	Support for Business Process Modelling	
	I. Support for different business process modelling languages.	8
	II. Support for interoperability between different business process modelling languages.	
	III. Support for interoperability between simple process modelling methods and business process modelling languages.	10
	IV. Support for different views when modelling a business process.	9
	V. Support for different perspectives of a business process.	6
	VI. Support for modelling with implicit business rules.	10
	VII. Support for modelling with explicit business rules.	10
	VIII. Support for modelling business processes using explicit business rules.	0
	IX. Ease of use of defining explicit business rules.	10
2.	Support for KPIs.	10 for our own KPIs 6 for the ease of use
	Design phase: 6.64	93 out of 140 points
3.	Support for creating executable business process models.	
	I. Supported business process execution languages.	8
	II. Support for manual translation between business process modelling languages and business process execution languages.	7
	III. Support for automatic translation between business process modelling languages and business process execution languages.	
	IV. Support for linking the business process execution model with the business process model.	5
4.	Supported programming languages for implementing services/applications.	1
5.	Support for designing user interfaces.	10
	I. Support for separate applications for data entry.	10
	II. Support for portal technology.	10
	III. Support for an activity list.	7
6.	Support for user management.	
	I. Way of describing roles (by names/description or by capabilities).	5
	II. Support for determining user capabilities based on organizational role.	10
	III. Support for importing organizational structure from other systems.	10

	IV. Support for determining user capabilities based on tokens (certain capabilities of a user to execute certain functions, procedures or activities).	0
	V. Support for manually assigning a user to a role.	0
	VI. Support for automatically assigning a user to a role, based on his capabilities.	
	VII. Support for activity permissions based on roles.	10
Interaction phase: 9.00		18 out of 20 points
7.	Support for informing the user.	
	I. Support for active informing.	8
	II. Support for passive informing.	10
Monitoring and Control phase: 8.11		73 out of 90 points
8.	Support for technical monitoring and control.	10
9.	Support for business monitoring.	
	I. Support for active monitoring.	10
	II. Support for passive monitoring.	10
	III. Support for various level of detail.	10
	IV. Support for different views of monitoring information.	5
10.	Support for business control.	
	I. Support for process instance evolution.	8
	II. Support for changing business rules.	7
	III. Support for changing activities.	8
	IV. Support for changing the workload balance between users.	5

About Capgemini



Capgemini is a global player in the field of consulting, technology, outsourcing and local professional services. Capgemini works together with its customers to achieve faster, better and longer lasting results.

At the moment, Capgemini serves a small market in the BPM field, but it estimates that the market is growing. Therefore it wants more knowledge in this field, so it can serve a bigger market once it grows.

Capgemini is headquartered in Paris, France and operates in more than 36 countries. It employs over 86,000 people in North America, Europe, and the Asia Pacific region.

www.nl.capgemini.com