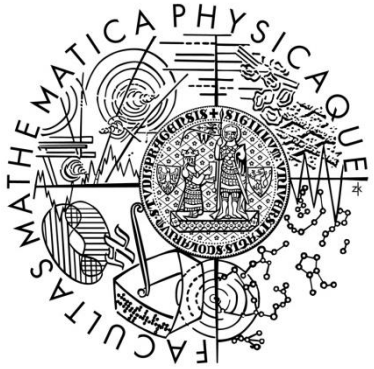
Charles University in Prague

Faculty of Mathematics and Physics

MASTER THESIS



Bc. Jiří Mach

Comparison of BPM Suites and Their Application in Enterprise Architecture

Department of Software Engineering

Supervisor of the Master Thesis: Mgr. Martin Nečaský, Ph.D.

Study Program: Computer Science

Specialization: Software Systems

Prague 2012

I declare that I carried out this Master Thesis independently and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act as amended, in particular the fact that the Charles University in Prague has the right to conclude a licence agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague on 7.12.2012 Jiří Mach

Název práce: Porovnání BPM nástrojů a jejich implementace v podnikové architektuře

Autor: Bc. Jiří Mach

Katedra (ústav): Katedra softwarového inženýrství

Vedoucí práce: Mgr. Martin Nečaský, Ph.D.

Abstrakt: Rozvoj servisně orientované architektury a centralizace služeb umožnil vznik a rozvoj nástrojů pro řízení obchodních procesů (angl. Business proces Management - BPM). Centralizované řešení pro řízení procesů přináší vyšší míru flexibility, ale zároveň pomáhá budovat kvalitní podnikovou architekturu. V současné době jsou na trhu stovky různých řešení podporující BPM a není snadné si vybrat ten správný. Zároveň se jedná o velmi rychle se rozvíjející oblast, která sebou přináší řadu nových postupů a standardů. Hlavním cílem mé práce je pomoci specialistům z oblasti IT ale i businessu při výběru implementaci BPM nástroje.

Klíčová slova: BPM, BPEL, BPMN, SOA, architektura podniku, obchodní procesy

Title: Comparison of BPM Suites and Their Application in Enterprise Architecture

Author: Bc. Jiří Mach

Department / Institute: Department of Software Engineering

Master Thesis Supervisor: Mgr. Martin Nečaský, Ph.D.

Abstract: The evolution of services oriented architecture and the centralization of services enable the expansion of business process management tools. The centralized solution for BPM brings more flexibility, and it also helps to build high-quality architecture. Currently there exist several solutions that support the BPM, and the selection of the proper tool can be difficult. Simultaneously, it is a rapidly growing industry which also brings several standards and principals. The main goal of my thesis is to help IT and business specialist with the selection and application of BPM into enterprise architecture.

Key words: BPM, BPEL, BPMN, SOA, enterprise architecture, business processes

Contents

1. Introduction

In the last few decades the companies tried to automate their business processes using specialized tools. From 1990s tools for automation of concrete domain of their business started to appear (for example: resource planning, customer relationship management, transactions processing, etc.). With the growing number of individual applications the companies needed to integrate them. From basic file and data sharing the evolution of integration technologies goes to providing services based on standards. The SOA and ESB concepts minimised the relations between systems and centralized services provided by the systems. This approach enables the automation of business processes which are going across business domains and company departments.

Within the last few years the number of tools and vendors that support business process management (BPM) rapidly increased. The separate products were successively put together into suites to provide a complex functionality to customers. The first BPM suites were primarily designed for IT and business users only defined the requirement on process. But with the evolution of the BPM technology BPM suites started to be more business oriented and involved business users directly into processes. This also made vendors and standardization organization to change the view on BPM and update the current standards and methodologies.

Nowadays tens of BPM suites and standards exist. And it is not easy for customers to choose the right product or standards, which they will support. Also the implementation and integration of a BPM suite into enterprise architecture has a big impact on a whole company and with regard to a size of enterprise architecture in modern companies it rightfully belongs to one of the most complicated projects.

1.1. Goal of the Thesis

The main goal of the thesis is to help business and IT specialists with implementation of a business process management.

The first goal of the thesis is to define the business process management, its history and describe the most used standards. The second goal is to describe the methodology of implementing BPM suites into enterprise architecture and to closely define the relations between enterprise architecture, SOA and BPM. The third goal is to define the criteria or checklist used for the comparison of BPM

suites. And finally to compare the BPM suites (commercial and open source[1]) based on these criteria including the results from implementation of a business process.

1.2. Structure of the Thesis

The whole work is organised into four main chapters with this first Introduction chapter and the last chapter – Conclusion. At the end of the thesis you can find the list of used references, images and abbreviations.

The second chapter defines the basic terms and it is primarily devoted to business process management, its definition, history and life-cycle. This chapter also describes the standards used in BPM.

The third chapter is intended for the application of BPM into enterprise architecture. The chapter starts with the overview of architecture, integration patterns and service oriented architecture. Then it describes the enterprise architecture, its relation to BPM and finally the application of BPM in modern enterprises.

The fourth chapter contains definition of BPM suite and selection of both commercial and open source BPM suites for comparison. In this chapter the key components of BPM suite are listed and then they are mapped to components of concrete products.

The fifth chapter specifies the comparison criteria or checklist for easier choice between BPM suites. And finally, it compares the commercial and open source BPM suites to these criteria and real implementation of business process.

The Conclusion summarises the achieved outputs and goals of the whole thesis.

1.3. Sources

The whole thesis was made with the use of three information channels.

The first information channel is represented by books, online sources and other external documents such as technical manuals of concrete products or evaluations from technology research and advisory companies. The complete list of used references is mentioned at the end of the thesis.

The second source of information is based on my personal experience from commercial work. During my professional career I have worked on SOA and

[1] The products released with one of the licenses mentioned in the list (36) will be treated as open source for the purpose of this thesis.

BPM projects at various positions: the developer, the analyst, the architect, the project manager and the sales consultant. This experience allows me to see BPM and its application to enterprise architecture in wide perspective. I gained most of my experience within the projects for the following financial, insurance and medical insurance companies:

- "VZP": analysis and implementation of business processes in BPEL
- "Raiffeisenbank": integration analysis and methodology for online and batch integration platform, SOA governance and service catalogue definition
- "ING Pojišťovna": SOA governance, Enterprise architecture analysis

The thesis was finalised also thanks to the knowledge gained from the conferences, trainings and consultations with my supervisor Mgr. Martin Nečaský, Ph.D., and also with both business and IT specialists primarily from following companies: "ING pojišťovna, Oracle, Profinit, Raiffeisenbank and Telefonica".

2. Business Process Management

This is the introductory chapter to BPM. The definitions of business process and the BPM acronym are presented to make the boundaries of this management discipline clear. This chapter also briefly mentions the history of BPM which goes to the beginning of 20th century. Since then many of standards were published and nowadays we have tens of standards. Therefore at the end of the chapter there the key standards used in BPM are chosen and described.

2.1. Definition of BPM

Before I define the business process management, I should make clear what the business process is.

Definition of "**Business Process**": "A business process is an ordered sequence of activities, which proceed in series or parallel, which are carried out by people or applications (practical activities, semi-automatic, automatic) and succeed to an awaited result" (see (1), page 6).

Technically, the process consists of three basic components:

- **Inputs**: They start the process. For example, if you are applying for loan, the inputs are parameters of loan and optionally confirmed level of income.
- **Activities**: They transform inputs into a specific output for a particular customer or market. In loan example, activities would consist of evidence of the loan, verifying the income, approving or disapproving the loan.
- **Outputs:** The outputs are awaited results of activities. In this example, the result can be providing money to client or rejection of application.

A part of this thesis is also implementation of business processes using the BPM suite. In chapter 5.2.1 you can find an example of business process.

Now I can clarify the definition of the acronym BPM. It stands for three different meanings:

- **Business Process Modeling**: "Business process modeling is a set of practices or tasks that companies can perform to visually depict or describe all the aspects of a business process, including its flow, control and decision points, triggers and conditions for activity execution, the context in which an activity runs, and associated resources" (see (1), page 82).

- **Business Performance Management:** "Business performance management focused on various processes that are proven effective in helping organizations optimize business performance" (see (2), page 16). This meaning is often known as Enterprise Performance Management (EPM).

- **Business Process Management**: There are many definition of Business Process Management. I chose the definition from the Association of Business Process Management Professionals: "Business Process Management (BPM) is a disciplined approach to identify, design, execute, document, measure, monitor, and control both automated and non-automated business processes to achieve consistent, targeted results aligned with an organization's strategic goals. BPM involves the deliberate, collaborative and increasingly technology-aided definition, improvement, innovation, and management of end-to-end business processes that drive business results, create value, and enable an organization to meet its business objectives with more agility. BPM enables an enterprise to align its business processes to its business strategy, leading to effective overall company performance through improvements of specific work activities either within a specific department, across the enterprise, or between organizations" (3).

This mentioned definition of BPM may seem to be very long and unintelligible but it includes all aspects of BPM. In short, the BPM should be defined as a set of principles, methods and tools used to identify, design, execute, monitor and control business processes.

For the purpose of this thesis I will always use the acronym BPM for Business Process Management.

2.1. History of Business Process Management

BPM does not belong to three or four letter acronyms which have been defined during previous years (like BPMN[2] or BPEL[3]). Evolution of business process management started at the beginning of 20th century.

The first mention of business process management appeared in Frederick Winslow Taylor's book "Shop Management" in 1903 (see (1), page 26). Taylor and his colleagues developed modern industrial engineering and process

[2] BPMN = Business process model and notation, see chapter 2.3.1 for more details

[3] BPEL = Business process execution language, see chapter 2.3.3 for more details

improvement. This stream is characterized by optimization of the business processes and standardization of the activities. The Taylorist principles were mostly used in the first half of 20th century (for example, one of the first users was Henry Ford).

In 1980s new methodologies with considerable focus on business process improvement and quality appeared. Total Quality Management (TQM) and Six Sigma belong among these methodologies. The first mentions of TQM came into sight in 1951 when Doctor Armand V. Feigenbaum wrote a book: "Quality Control: Principles, Practice and Administration" (see (1), page 27). Several years later the idea of "Six Sigma" was born in Motorola Company. "Six Sigma" had strong emphasis on statistical techniques and process improvement partly derived from preceding methodologies like TQM.

The nineties were strongly influenced by business process change. It can be dated from the Business Process Reengineering (BPR) movement that began in 1990 (see (4), page 9). Thomas H. Davenport and James R. Short can be considered as founders of this management philosophy. They argued that the combined use of IT and business process redesign could transform organizations and improve business processes (see (5), page 30). At the same time Michael Hammer had published his radical concept. The reengineering in its extreme form does not improve the existing process. It supposes that the existing processes are completely ineffective and have to be redesigned prior to the wide usage of IT. But this radical nature of BPR was often associated with failure.

In the mid and late 1990s, the Enterprise Resource Planning (ERP) systems gained organizational focus. ERP systems are information systems that handle the data, information and communication requirements of the whole organization. Proper use of ERP systems is a key to successful Supply Chain Management (SCM). In the end of the 1990s and in the early 2000s systems with strong emphasis on customer were introduced by IT companies. The Customer Relation Management (CRM) came into sight of various companies.

From the beginning of 1990s to the 2000s several approaches, methodologies or systems were introduced but they were focused on certain part of business (for example, CRM – customers, ERP – sources etc). Many people realized that they need to integrate the number of systems and technologies that appeared in previous years. The new wave of business process management was introduced in 2002 in a book: "Business Process Management: The Third Wave" (see (6), page 2). They proposed to combine separate tools inside and outside the enterprises: "Companies could use one formalism to express all processes in the value chain, paving the way for the development of systems that could execute such process directly" (see (6), page 13).

At Figure 1 you can see the history of BPM plotted into hype cycle.

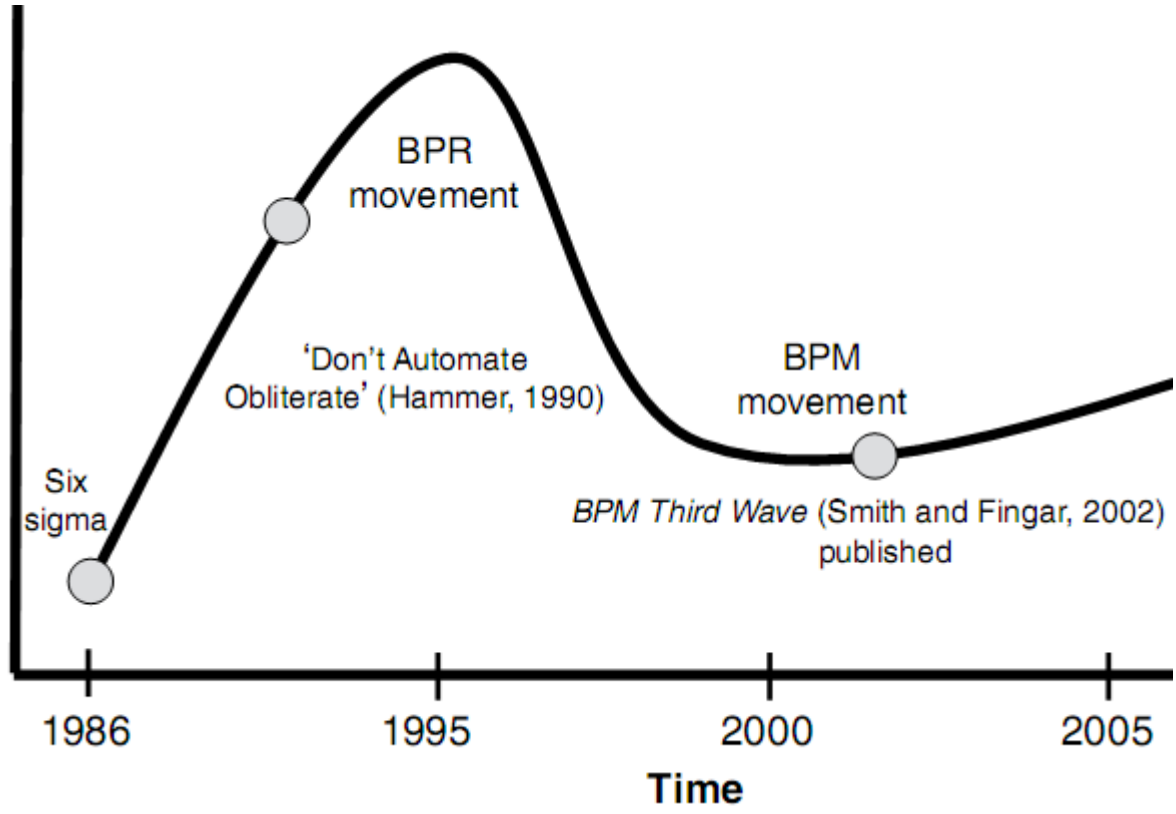


Figure 1: BPM hype cycle (see (7), page 5)

According to a new view to business process management it was necessary to develop several standards and tools (for example: standardized language for expressing complex end-to-end business processes, environment for execution and monitoring of business processes etc.). In 2000s several standards like BPML, BPEL or BPMN appeared as well as business rules engines, business activity monitoring systems, business process engines and many others (these standards and technologies are closely described in chapter 2.3).

By the end of 2005, Business Process Management was defined by Garner[4] as a technology base for business users and IT professionals to create and manage dynamic business processes (see (2)). And the set of tools, systems and technologies were referred to as BPMS – Business Process Management System or Suites.

2.2. Life-cycle of Business Process Management

[4] Garner is the world's leading information technology research and advisory company.

In this chapter I will go through the whole life cycle of business process management. The life cycle consists of several phases which are organised in the cycle. The number of phases is different in different sources but it only relies on granularity. The content of phases is the same. I will describe the life cycle of business process management in four phases[5]. For each phase I will introduce relevant concepts, standards and technologies from BPMS. Complete overview of life cycle phases you can see at Figure 2.

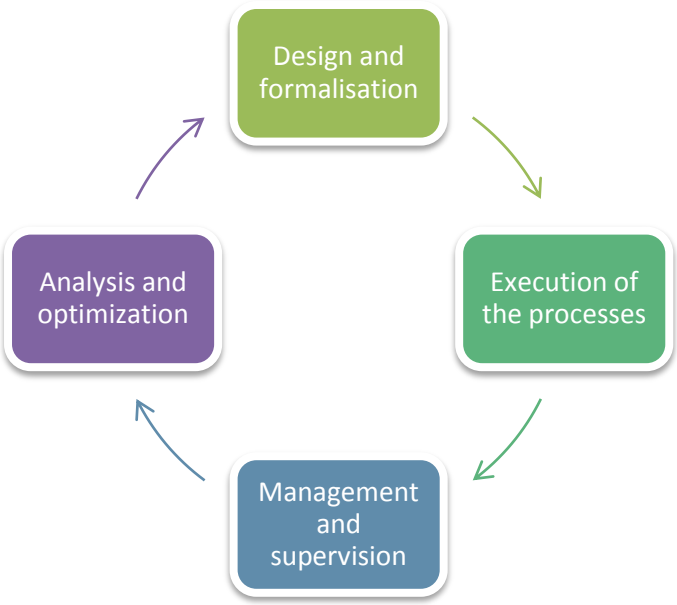


Figure 2: BPM Life-cycle

2.2.1. Design and Formalization

This phase can be divided into two sub-phases – identification of the process and process modeling.

The first sub-phase starts with identification of the business processes. The business processes must be documented, reviewed and validated. In this phase the business owner should be assigned to the processes. Business owner is responsible for the process throughout its life cycle. Business owner also identifies cooperation with other processes, services, users, documents or other resources.

When the business processes are reviewed and validated the second sub-phase can start. The business analysts describe in more detail the business processes and express the processes in a graphical notation. There exist several standard

[5] The same granularity for BPM life cycle uses for example Jean-noel Gillot in (1), page 92.

modeling notations as BPMN or UML[6]. Nowadays BPM suites offer very sophisticated graphical tools enabling modeling of business processes and some of them also support these standards (example of graphical tool is shown in Figure 3).

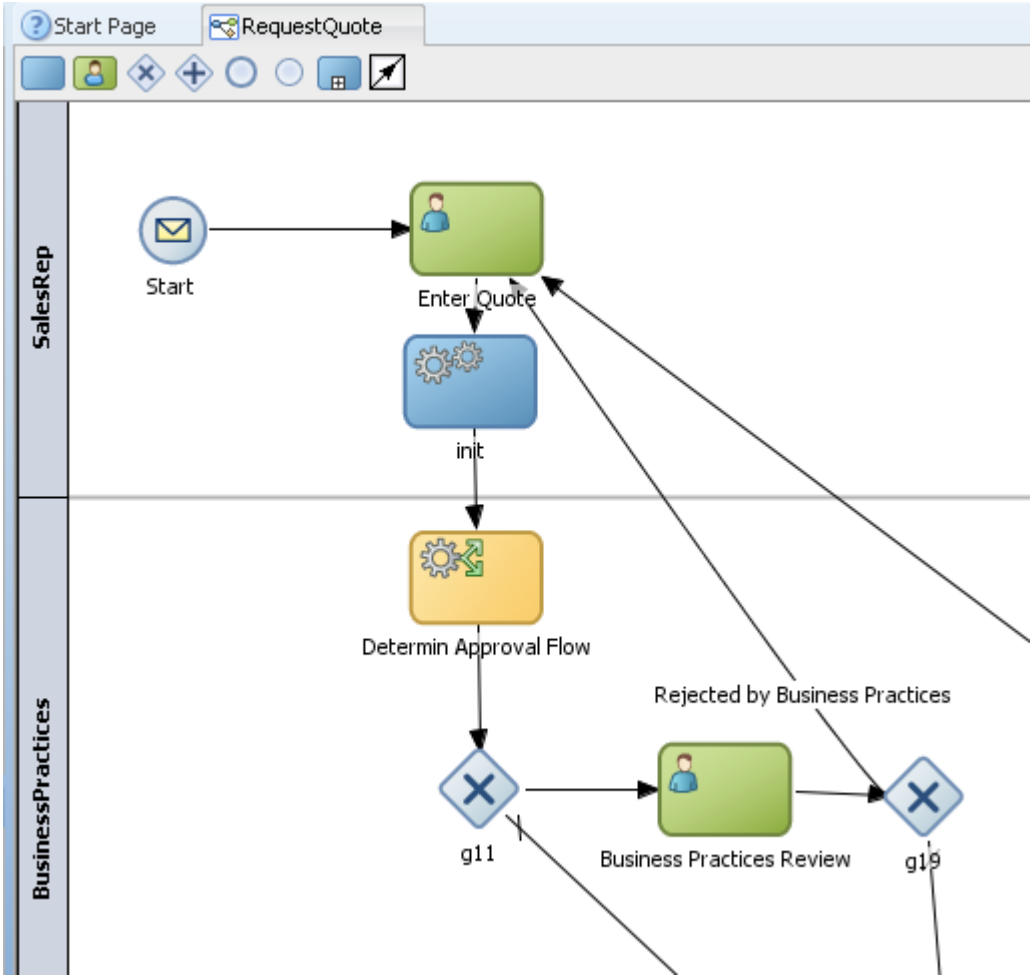


Figure 3: Oracle BPM Studio (9).

Defining the business rules[7] and human tasks[8] can be another part of this phase. But it depends on the chosen BPMS and their technologies.

Once the business process is designed, it needs to be validated. One of the good techniques for validation is workshop with business owner, analyst and other involved people. The participants of the workshop discuss the parts of business process and that is why the standardized model of process can make things

[6] Unified Modelling Language is OMG's most-used specification, and the way the world models not only application structure, behaviour, and architecture, but also business process and data structure (see (38)).

[7] For more information about business rules see chapter 4.1.3

[8] For more information about human tasks see chapter 4.1.4

easier. In order to facilitate the validation part, the BPM systems already offer the simulation environment for business processes at this phase of life cycle.

2.2.2. Execution of the Processes

When the business process is designed and validated, the implementation phase can start. The design of business process has to be enriched with technical information which enables integration into the enterprise architecture and involvement of persons from the organizational structure. Many of these processes are put into practice with the help of existing systems or by employees.

The business processes can be implemented by the set of rules and policy or they can also be developed within one of your programming language. But there exist several standards that are useful for implementation of business processes: BPEL, BPMN 2.0 and XPDL[9]. These standards preserve the structures and workflows of business processes defined in design phase and also enable to add necessary information that is needed for the execution of business processes. The approach of implementation is similar to the creation of enterprise application in programming language like Java or C#. You need an environment where you can easily write your code, a compiler to create executable output and environment or server where you can run this output. The BPM systems contain these tools that support above mentioned standards. Many of them also include adapters to be easily integrated with existing systems, portal solutions, ESBs etc. They very often contain tools for processing human tasks or business rules (Figure 4 shows human task editor).

[9] For more information about standards used in BPM see chapter 2.3

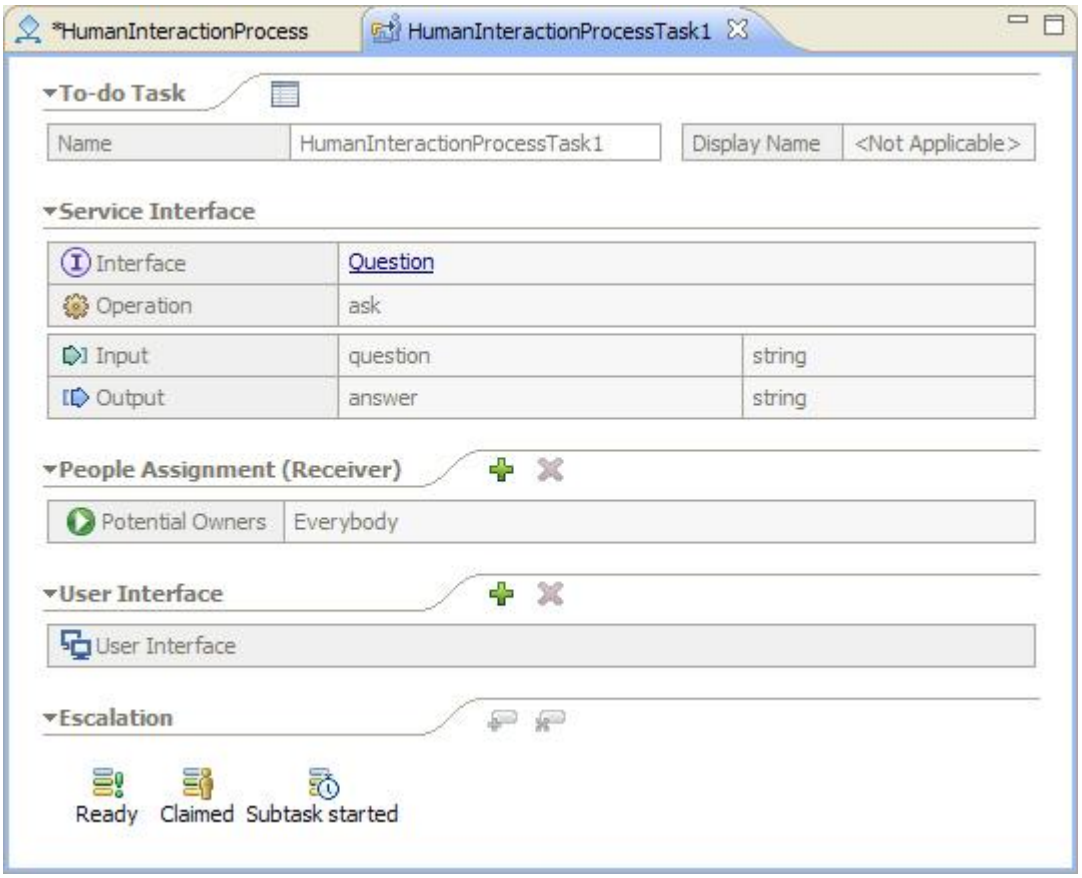


Figure 4: Human Task editor (10)

Once the process is developed, the implementation of process needs to be tested. Traditional testing techniques can be used to check the process. The main emphasis should be put on the integration testing because business processes often integrate many other systems and users. Some of BPM suites have tools and technologies that make testing and debugging easier.

The final part of this phase is deployment into production environment and execution of business processes. The environment for deploying the processes is also a part of BPMS and its execution part is often called engine, for example BPEL engine or BPMN engine[10].

2.2.3. Management and Supervision

When business processes are deployed into production environment it can be initiated and new instance of process is created. The processes are typically initiated by events, for example: an application for a new loan or a creation of a new order. Large companies can have thousands of business processes which are automated and the number of process instances can go up to millions. The management and monitoring of this amount of processes and instances is very

[10] Execution engines are in detail described at chapter 4.1.2

important in life cycle of BPM. That is why BPM Systems provide dashboards and overviews of both running and closed instances of processes (the example of BPM console you can find at Figure 5). They also enable detail view of the instances in graphical form as they were designed so business managers can recognize the current status of process or find out why error occurred.

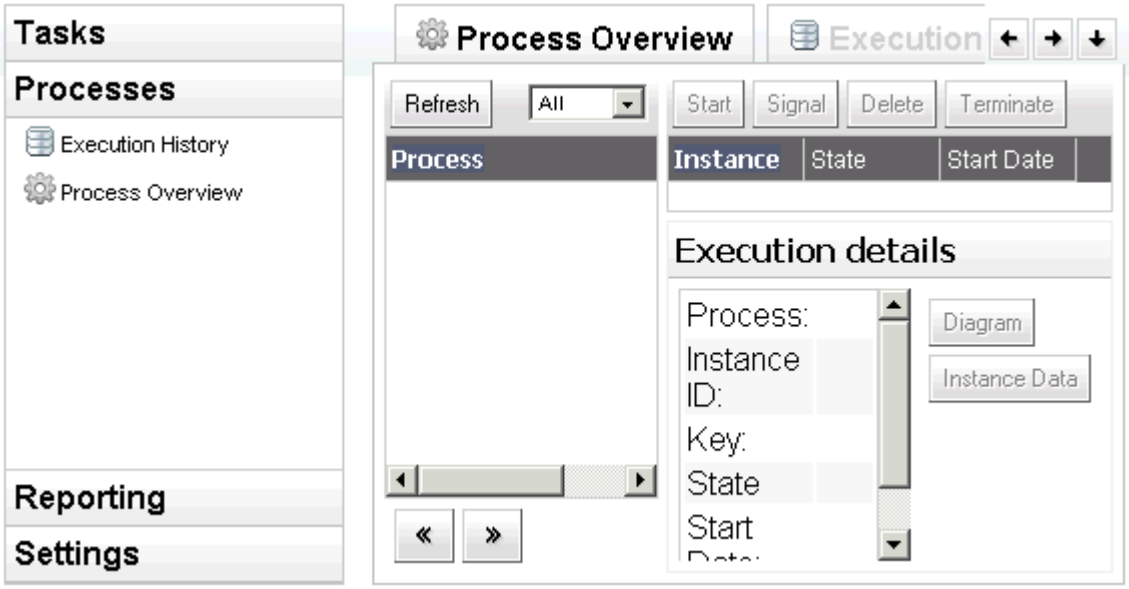


Figure 5: jBPM console

Many of BPM Systems also provide various performance indicators which can be placed into process. During the run of processes BPMS log performance values and make them available for further analysis. BPMS also gather other valuable data and log them (for example event starting/stopping the instance, exceptions, input and output parameters and many others). This information is used especially in last phase of BPM life cycle.

2.2.4 Analysis and Optimization

The collected data from logs are used to analyse the processes. BPM Systems offer several tools (for example Business Activity Monitoring[11]) for analysing traffic, performance indicators or logs. The analyst can easily recognize the source of exceptions, bottlenecks in system load or inefficient business processes. Business analyst can define monitoring criteria for activities and then evaluate then using BAM (Figure 6 shows the monitoring of order processing).

[11] For more information about Business Activity Monitoring (BAM) see chapter 4.1.6

Figure 6: Monitoring of order processing (11).

In some cases the performance of bottlenecks can be improved purchasing the more powerful hardware. But very often the business processes are reengineered and optimized and the life cycle continues with redesign and documentation.

2.3 Standards used in Business Process Management

An important issue for dealing with software and technology is the use of relevant standards. Even though the term business process management is not standardized (see chapter 2.1), the BPM domain has its own standards. The standards are generally very important to the customers. They can easily compare products and switch to another vendor.

But the current situation about BPM standards is still very complex because tens of standards are existing, which are in some way relevant to BPM. There is no general set of standards which can be agreed by vendors or standards consortia.

The key consortia in BPM domain include OMG[12], OASIS[13] and WfMC[14] and among their members you can find all leading BPM vendors.

Based on literature (see (3), page 203; (7), page 209; (4), page 122 and (1), page 91) and my personal experience I chose the following key standards:

- BPMN (OMG)
- BPEL (OASIS)
- XPDL (WfMC)

These standards are described in the following chapters. The last chapter compares the previously mentioned standards with other broadly used standards which are relevant to BPM.

2.3.1 BPMN

"A standard Business Process Modeling Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations. This will ensure that businesses will understand themselves and participants in their business and will enable organizations to adjust to new internal and B2B business circumstances quickly." (14)

---

[12] **OMG** (Object Management Group) is an international, open membership, not-for-profit computer industry consortium since 1989. The most-known OMG's modeling standards are the Unified Modeling Language (UML), Model Driven Architecture (MDA) or Business Process Model and Notation (BPMN). See (26) and (29).

[13] **OASIS** (Organization for the Advancement of Structured Information Standards) is a not-for-profit consortium founded under the name "SGML Open" in 1993. OASIS promotes industry consensus and produces worldwide standards for security, cloud computing, SOA, Web services, the Smart Grid, electronic publishing, emergency management, and other areas. See (27).

[14] **WfMC** (Workflow Management Coalition) is a global organization of adopters, developers, consultants, analysts, as well as university and research groups. Between well-know WfMC's standards belong Wf-XML and XPDL. See (25).

The intent of BPMN is very similar to UML in object-oriented architecture and design. UML activities diagrams can be considered as one of the ancestors of BPMN but these diagrams do not have the necessary expressive power to become a standard for business process modeling and execution.

BPMN was developed by BPMI[15] as a graphical notation for process description expressed in Business Process Modeling Language (see (5), page 10). The first version was published in 2004 by BPMI and in 2008 it was accepted as OMG standard. The current version of BPMN is 2.0 and was published in January 2011. This new version brought several changes among others as well as the change of name to Business Process Model and Notation. Version flow of BPMN is shown at Figure 7[16].

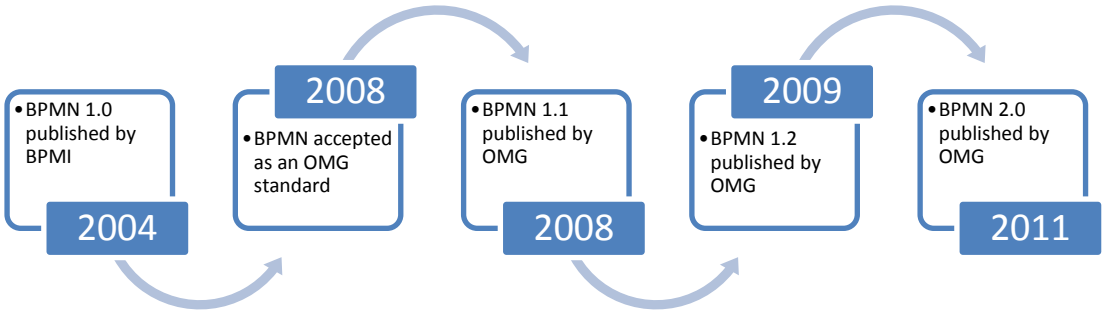


Figure 7 - BPMN history

From the presented versions the most significant changes were introduced in version 2.0 (for more information see the BPMN 1.2 (16) and BPMN 2.0 (17) specifications):

- Change of name to Business Process Model and Notation
- Notation changes
  - Addition of Choreography and Conversation diagram
  - Addition of Non-interrupting Events and Sub-Process Event for Process
- Addition of BPMN Diagram Interchange Meta-model
- Addition of BPMN Execution Semantics

The mentioned changes brought enhanced support for human task and collaboration. Also the added execution semantics enabling simple conversion to execution language BPEL (see chapter 2.3.3). Nevertheless, it is possible to execute directly the detailed BPMN model so it is not necessary to convert the BPMN model to BPEL anymore (for

[15] **BPMI** (Business Process Management Initiative) in 2005 combined its activities with OMG. The combined group has named itself the Business Modeling & Integration (BMI) Domain Task Force (DTF). BPMI's widely used standard for business modeling are Business Process Model and Notation (BPMN), and the Business Motivation Model (BMM). See (29).

[16] Data were primary compiled from (5) page 10, (34) and (29).

example, the Oracle has native BPMN engine). And finally, the BPMN interchange model can be used to transfer models between different BPM tools with a support of this standard.

BPMN consists of four basic types of elements:

- Flow objects
  - Activity – represents work which should be done
  - Gateway – is used for flow control, cycles, parallel flows etc
  - Event – supports asynchronous communication, error handling and controlling logic flow
- Connecting objects
- Swim lanes
- Artifacts

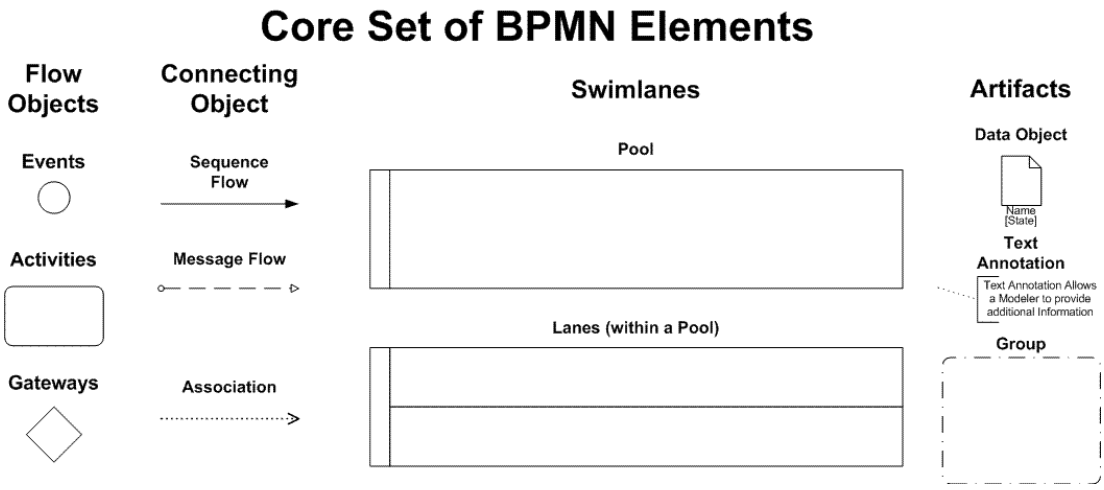


Figure 8 - Graphical representation of basic BPMN objects (18)

The example of BPMN model is shown at Figure 9, the complete definition of BPMN is out of scope of this thesis.

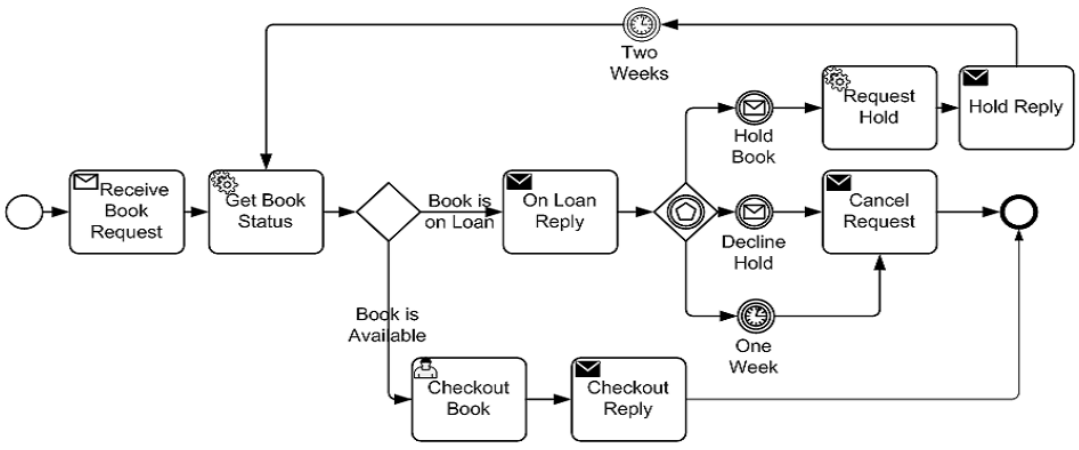


Figure 9 - Example of BPMN model (see (17), page 145)

2.3.2 XPDL

"XML Process Definition Language (XPDL) is the interface between process definition tools and the enactment service. Specifically, processes designed in a tool are exported in XPDL format and loaded into the enactment service for execution, where XML is a suitable external representation. Other purposes of XPDL are to move process definitions between tools. A process definition tool should have both export and import capabilities: export to XPDL, and import from XPDL" (see (6), page 180).

XPDL is a serialisation format for BPMN and it is supported by tens of BPM vendors (many of them use XPDL as an internal representation format). But XPDL unlike BPMN does not contain notation for graphical representation.

In 2002 WfMC published XPDL standard as a replacement of previous standard: Workflow Process Definition Language (WPDL), which was firstly published in 1999. XPDL version 2.1 published in 2008 was intended to be used as a file format for BPMN 1.1. OMG by releasing BPMN 2.0 came with its own serialisation model (for more information see previous chapter). The latest version of XPDL is 2.2 and it is still intended to be used as a representation for BPMN 2.0 but it does not include all parts of BPMN 2.0 (for example: Choreography and Conversations Diagrams). Version flow of XPDL is shown at Figure 10[17].

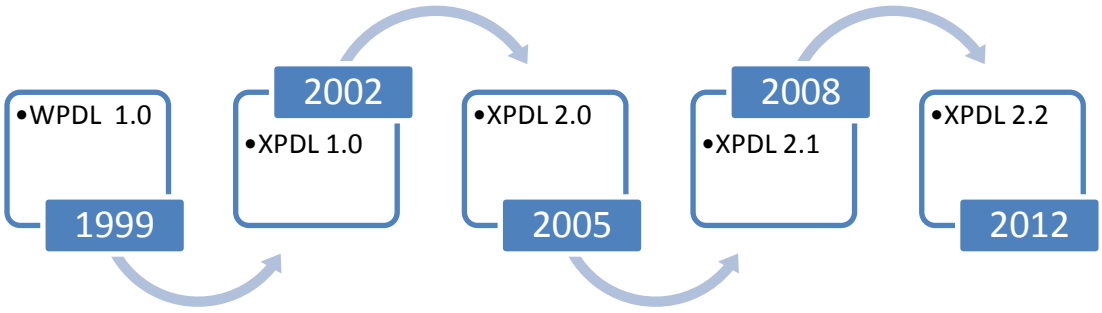


Figure 10 - XPDL history

The XPDL 2.2 specification (see (7)) is divided into two parts:

- Definition of Meta-model: describes the top-level entities, relationships and attributes
- XML process definition language specification: describes the XML structure which represents the definition

[17] Data were primary compiled from (3), page 206 and (24).

The whole specification is supplemented with XML Schema Definition of XPDL representation.

Example of BPMN model (Figure 11) representation in XPDL shows following XML fragments (you can find the whole example at Figure 11).

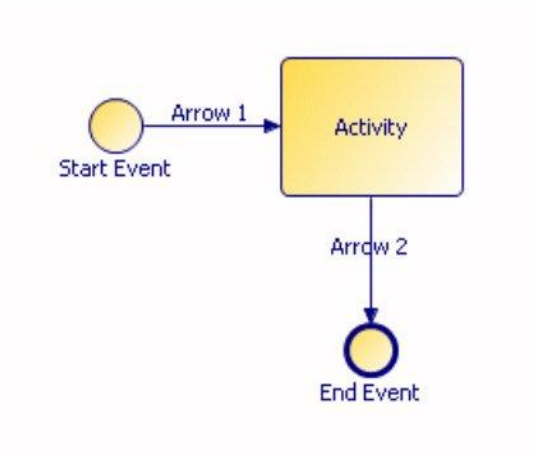


Figure 11 - Example of a process

Representation of activities:

```
<Activities>
  <Activity Id="153189" Name="Start Event">
     ...
  </Activity>
  <Activity Id="153190" Name="Activity">
     ...
  </Activity>
  <Activity Id="153191" Name="End Event">
     ...
  </Activity>
</Activities>
```

Representation of flows:

```
<Transitions>
  <Transition Id="153203" Name="Arrow1?
         From="153189"  To="153190">
    ...
  </Transition>
  <Transition Id="153204" Name="Arrow2?
         From="153190" To="153191">
    ...
  </Transition>
</Transitions>
```

Representation of single activity:

```
<Activity Id="153190" Name="Activity">
  <Implementation><No/></Implementation>
  <Performer>Manager</Performer>
  <Priority/>
  <Documentation/>
```

```
  <TransitionRestrictions>
    <TransitionRestriction>
      <Join Type="XOR"/>
      <Split Type="XOR">
        <TransitionRefs><TransitionRef Id="153204"/>
         </TransitionRefs>
      </Split>
    </TransitionRestriction>
  </TransitionRestrictions>
  <NodeGraphicsInfos>
    <NodeGraphicsInfo LaneId="1" ToolId="Interstage BPM"
        IsVisible="true" Page="1">
      <Coordinates XCoordinate="317.0" YCoordinate="83.0"/>
    </NodeGraphicsInfo>
  </NodeGraphicsInfos>
</Activity>
```

2.3.3 BPEL

"Business Process Execution Language (BPEL) is a language that is utilized to specifically identify the behavioural pattern that results into either an effective business process based on the given web service. WS-BPEL is a platform which allows identification of the two major business processes: The Executable and the Abstract Process" (see (8), page 10).

BPEL represents the business processes using XML format but unlike the BPMN, it does not contain notation for the graphical representation of process. The processes are defined by a set of elements which are closely described later in this chapter. Based on the definition, the execution engine publishes the process on the server where it is accessible through execution engine API (it is not part of the standard) or it is exposed as a standard web service with its own inbound and outbound parameters.

The latest version is called Web Services Business Process Execution Language 2.0 (WSBPEL 2.0) but this standard has more than 10 years of history. BPEL was established as a combination of IBM's Web Services Flow Language (WSFL) and XLANG (notation for the business process automation). The origin name of this standard was Business Process Execution Language for Web Services (BPEL4WS) but then it was renamed by OASIS and issued in 2007 under current name WSBPEL 2.0. In 2007 a set of vendors published WS-BPEL Extension for People (BPEL4People) and WS-Human Task specifications which are designed to involve the users into business process. The version flow of BPEL standards is shown at Figure 12[18].

[18] Data were primary compiled from (6), page 103, (31) and (28).

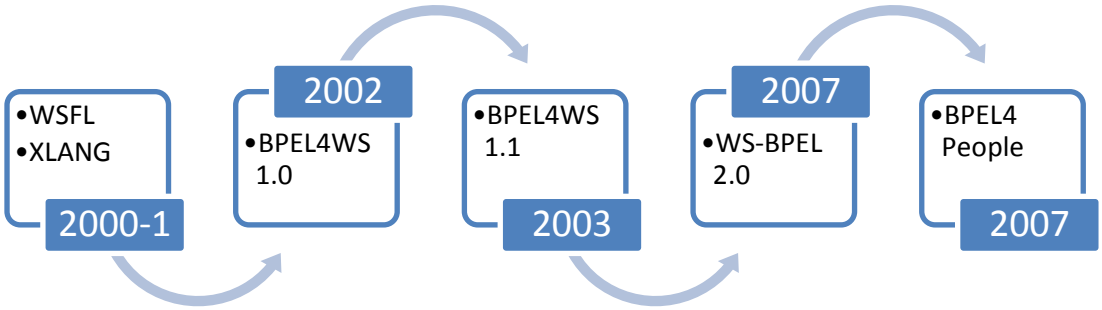


Figure 12 - BPEL History

As it was mentioned above BPEL is represented by XML. In fact the BPEL process is represented by at least two XML documents. The first one is simple XML document and represents the structure of the business process and the second one is WSDL[19] document, which defines the business process interface and interfaces of other called services or processes.

The example of WSDL structure (definition of interfaces is defined in separate WSDL document shippingPT.wsdl):

```
<wsdl:definitions
targetNamespace="http://example.com/shipping/partnerLinkTypes/"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
xmlns:sif="http://example.com/shipping/interfaces/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:import location="shippingPT.wsdl"
      namespace="http://example.com/shipping/interfaces/" />

      <plnk:partnerLinkType name="shippingLT">
            <plnk:role name="shippingService"
            portType="sif:shippingServicePT" />
            <plnk:role name="shippingServiceCustomer"
            portType="sif:shippingServiceCustomerPT" />
      </plnk:partnerLinkType>
</wsdl:definitions>
```

The basic structure of BPEL process (it is encapsulated in *process* element):

- *partnerlinks* – this section defines other processes or services which interact with the process (the interface of these processes or services is defined in WSDL document)

[19] WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information (see (42)).

- *variables* – this section enables to define variables which can be used to maintain the state during process. The type of variables is defined by XML Schema[20] elements.
- *faulthandlers* – this section defines the root fault handlers for processing the exceptions thrown during processing of activities
- *activities* – this part describes the process as a set of activities.

Structure of BPEL process has more sections (see (9)) and also many vendors extend the process definition about their extensions.

The following example shows the basic structure of BPEL process:

```
<process name="shippingService"
targetNamespace="http://example.com/shipping/">
      <partnerLinks>
            <partnerLink name="customer"
            partnerLinkType="plt:shippingLT"
            partnerRole="shippingServiceCustomer"
            myRole="shippingService" />
      </partnerLinks>
      <variables>
            <variable name="shipRequest"
            messageType="sif:shippingRequestMsg" />
            ...
      </variables>
      <sequence>
            list of activities
      </sequence>
</process>
```

WS-BPEL 2.0 specification defines the large number of activities that can be used for process definition. These activities enable the following functionality:

- Synchronous and asynchronous calling of services or processes
- Transformations of data using defined variables, XPath[21] and XSLT[22]
- Flow control, cycles, parallel flows
- Exception handling
- Defining scope inside process with its own variables, exceptions etc

The following example shows the example activities which can be used in sequence element:

```
<sequence>
```

[20] XML Scheme describes the structure of the document.

[21] XPath is a language for addressing parts of an XML document (see (65)).

[22] XSLT is a language for transforming XML documents into XML or other documents (see (43)).

```
      <receive partnerLink="customer" operation="shippingRequest"
      variable="shipRequest">
      </receive>
      <if>
            <condition>bpel:getVariableProperty('shipRequest','props
            :shipComplete')
            </condition>
            <sequence>
                  <assign>
                        <copy>
                              <from variable="shipRequest"
                              property="props:shipOrderID" />
                              <to variable="shipNotice"
                              property="props:shipOrderID" />
                        </copy>
                  </assign>
                  <invoke partnerLink="customer"
                  operation="shippingNotice"
                  inputVariable="shipNotice">
                  </invoke>



            </sequence>
            <else>
                  ...
            </else>
      </if>
</sequence>
```

3. Enterprise Architecture and Business Process Management

This chapter describes how the business process management and enterprise architecture are related. But firstly, it describes architecture itself and its evolution. This evolution of architecture and application enables developing the specialized application focused on the certain part of business. It also brings new requirements on application integration. The following chapters presents the integration patterns, service oriented architecture and the whole concept of enterprise architecture and the relations to BPM. The last chapter specifies the application of BPM in the enterprise architecture.

3.1 Software Architecture

I will cite the definition of software architecture from the book: A Practical Guide to Enterprise Architecture.

“The software architecture of a system or a collection of systems consists of the important design decisions about the software structures and the interactions between those structures that comprise the systems. These design decisions support a desired set of qualities that the system should support to be successful. The design decisions provide a conceptual basis for system development, support, and maintenance” (see (10), page 28).

Based on this general definition we can look at the software architecture from several points of view. The architecture of simple application will be certainly different from architecture of all company’s applications (this view on architecture is often called enterprise architecture and is described in more detail in chapter 3.4).

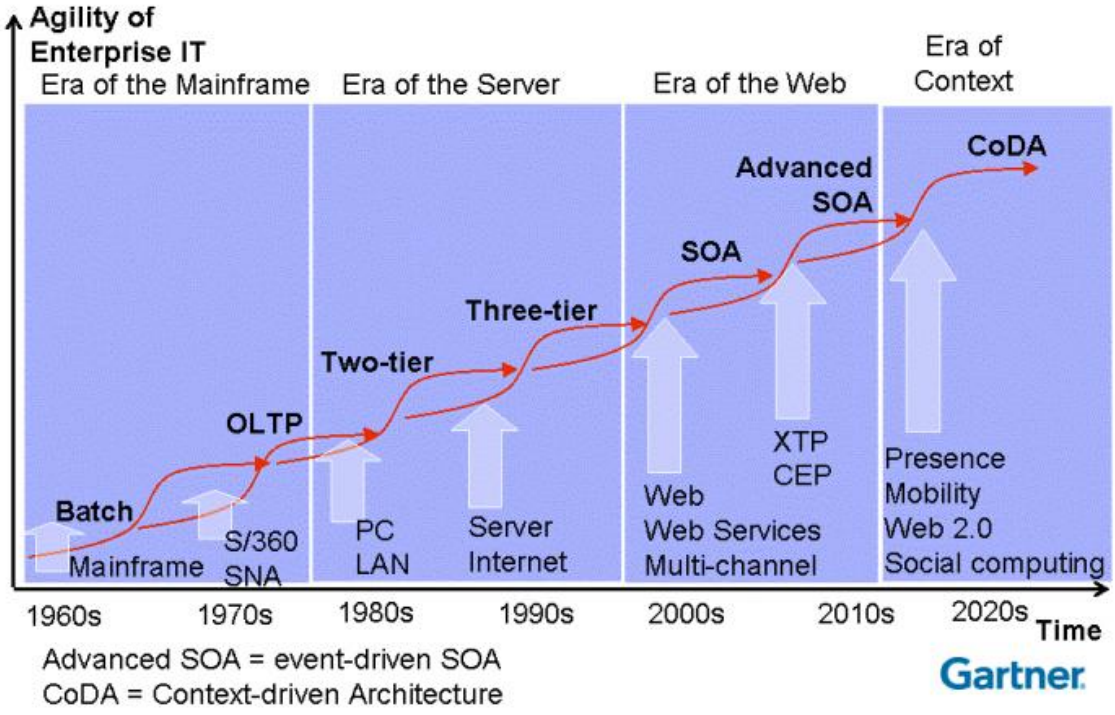


Figure 13 - Software Architecture in the Context of History (see (11))

The definition of software architecture also implies that software architecture is evolving together with evolution of software application generally (the evolution of architecture shows Figure 13). First applications were developed from scratch and developers had to code basic functionality for each application. Architecture of these applications contained decisions and interactions about both processing of applications and memory management, persistent storage access etc. The common logic was in the course of time subsequently devoted to stand alone applications communicating through interfaces. The first earmarked applications were operating systems which encapsulated the access to hardware. Then there appeared particular applications for data processing – database systems followed by many other specific applications. This division enabled the creation of very complex company solutions containing tens or hundreds applications but also brought new requirements on their integration. The first integration patterns were based on sharing files. Then there appeared more sophisticated methods for data sharing like database systems and also new integration patterns for real-time sharing of functions or services (for example RPC, messaging or SOA). These patterns are described in next chapter.

3.2 Integration Patterns

As was mentioned in previous chapter, with the growing numbers of individual applications in companies new requirements on integration of these applications appeared. The technologies that are used for application integration are very often called middleware.

Middleware is system services software that executes between the operating system layer and the application layer and provides services. It connects two or more applications, thus providing connectivity and interoperability to the applications (see (4), page 36).

During the evolution of software many integration patterns or middleware appeared and large number of them is still used nowadays. Every type of bellow mentioned patterns has its pros and cons and it is suitable for different types of situation.

From books SOA Approach to Integration (See (4), page 37) and Enterprise Integration Patterns (See (12), page 29) I mention the most common forms of middleware:

- **File Transfer** – it is one of the simplest ways of integration because files are universal store mechanism and no other integration tool is needed. One application stores a file to a shared storage and other applications can retrieve and process the stored file. But there are also several issues you have to solve: standard format of transferred data, synchronization, concurrent access and you cannot achieve the real-time integration.

- **Database Access Technologies** – due to several libraries for various technologies the common database can be used as an integration pattern. The main advantages are transactional system and wide-spread use of SQL, reliability and multi-user support. On the other hand, one of its biggest problems is the definition of common data model and its maintenance. Also there can be problems with performance (in case of frequent reads and updates of the same data) and data are still not shared in real-time mode.

- **Remote Procedure Calls (RPC) and Object Request Brokers (ORB)** – these technologies allow the client to call procedure or function which is located on remote server as if it was located on local server. But still, there are several differences in invoking remote methods: for example performance and reliability which is caused by network communication.

  In addition, ORBs also supports communication between distributed objects or components with location, protocol and operating system transparency. The disadvantages are very similar to RPC. The communication is typically synchronous, but some implementations also provide asynchronous mode. The ORB architecture is shown at Figure 14.

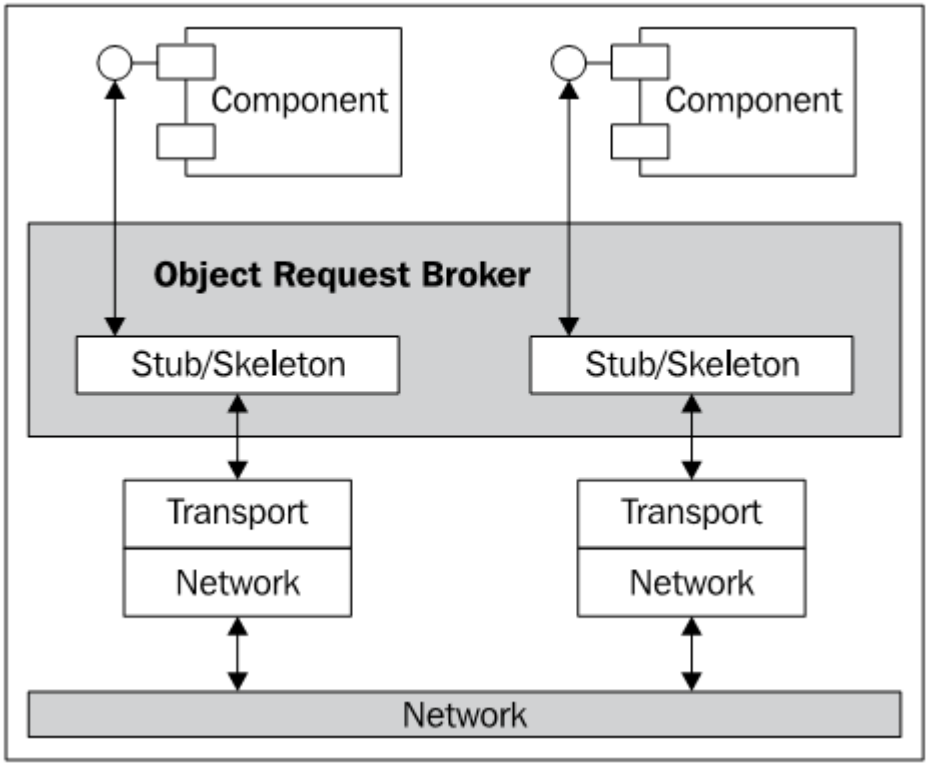


Figure 14 - Object Request Broker (see (4), page 41)

There exist several implementations of RPC and ORB (.NET Remoting, Java Remote Method Invocation, Corba).

- **Message-oriented Middleware (MOM)** - is pattern based on asynchronous messaging using queues or channels. Source application creates a message and sends it to MOM. Messaging system transfers the message to destination application which can read and process it. The whole system is very robust because messages are queued and persisted on the side of MOM. This pattern also enables both point-to-point and publish-subscribe integration. The disadvantage of this solution is limitation to asynchronous communication and also the platform dependency.

- **Transaction Processing Monitors and Application Servers** - transaction processing monitors represent the first version of application servers. They coordinate transactions among different resources and provide performance management and security services. They are based on remote produce calls but the provided API is very complex.

  Application servers refer to the principles of transaction processing. They provide all of already mentioned functions in a management environment called container. The container provides applications support for transactions, security and resource management and also

contains solutions for MOM, web services etc. Application servers can be used as a platform for both integration and application development.

- **Web Services** - they are in some way very similar to RPC and ORB, interoperability between application regardless of operating system, platform or programming language. But they are supported with wide spectrum of software platforms and use standardized format for data representation – XML. They enable both synchronous and asynchronous communication based on concept of request/response. Another advantage over the RPC and ORB is using standard Internet protocols as a transport layer (HTTP, SMTP, FTP...) and therefore they can go through company firewall more easily. Amongst disadvantages of web services belong the performance and less support for security, transactions, correlations or routing.

- **Enterprise Service Bus** – it is a very robust service oriented solution for enterprise integration. This brings us to the concept of service oriented architecture (SOA) where ESB is its backbone. The next chapter presents this concept and closely describes an ESB.

- **Service Component Architecture (SCA)** - is a set of specifications which describes a model for building applications and systems using a Service-Oriented Architecture (SOA). SCA extends and complements prior approaches to implementing services. SCA builds on the open standards such as Web services (26).

3.3 Service Oriented Architecture

As Nicolai M. Josuttis stated in his book SOA in Practice (see (1), page 11): "It is very hard to define the Service oriented architecture because there are many definitions from different point of views. But the common part of most definitions is that SOA is a paradigm that leads to concrete architecture".

Because this thesis is focused on BPM Suites and their application in enterprise architecture, I will mention the definition which is more related to enterprise architecture.

"A Service-Oriented Architecture (SOA) is a software architecture that is based on the key concepts of an application frontend, service, service repository, and service bus (see Figure 15). The service consists of a contract, one or more interfaces and an implementation" (see (13), page 42).

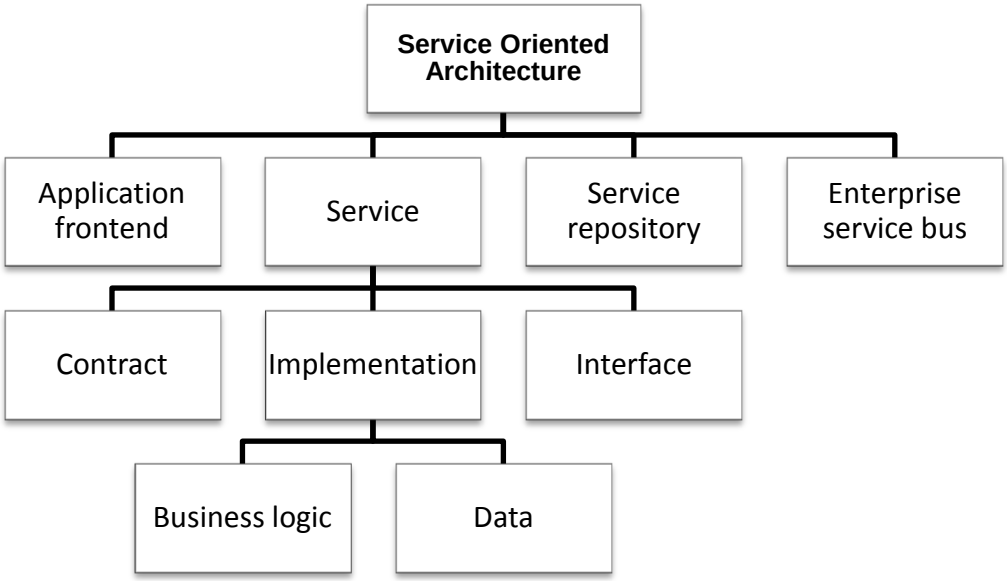


Figure 15 - Key concepts of SOA

The SOA is very often defined using an analogy with a real world and a set of principles which must be met. For example Thomas Erl[23] in his book Service-Oriented Architecture: Concepts, Technology, and Design (see (28), page 32) uses a cosmopolitan city and individual companies which provide distinct services that can be used by multiple consumers. But to make this work we have to define several conventions or standards which companies and consumers will adhere (common currency, same language, etc.). In this analogy individual companies which offers their services are the first part of application frontend = providers. The residents and other companies which use these service are the second part of application frontend = consumers. Service catalogue is in this analogy a city portal, maps or a directory and the central part is ESB which contains infrastructure and city governance (rules, policies, standards...).

The key concepts of SOA are closely described in next chapters.

3.3.1 Application Frontend

Nowadays enterprises have tens or hundreds of applications which cover different business or technical domains (for example CRM, ERP, transaction and financial systems, portals, business-to-business tools, etc.). These applications provide several services that can be consumed and on the other hand they act as consumers of services from other applications.

[23] Thomas Erl is a best-selling IT author and founder of SOASchool.com® and CloudSchool.com™. He is also the CEO of Arcitura Education Inc. and SOA Systems Inc. and the founding member of the SOA Manifesto Working Group and author of the Annotated SOA Manifesto.

3.3.2 Services

The formal definition of service is: “A service is a software component of distinctive functional meaning that typically encapsulates a high-level business concept. It consists of three parts: contract, interface and implementation” (see (13), page 44).

- **Contract** - the service contract provides an informal specification of the purpose, functionality, constraints and usage of the service.
- **Interface** - the business logic of the service is exposed by the interface to consumers that can use the service.
- **Implementation** - the service implementation provides the business logic and data.

To complete the definition of service we have to mention the standards that services have to fulfil (similarly as in the cosmopolitan city the companies and consumers use the same language or currency). The complete set of these standards or design principles is stated in another book from Thomas Erl SOA: Principles of Service Design (see (29), page 64):

- **Standardized Service Contract** - the services have to share one standard which describes the services themselves and the conditions for information exchange
- **Service Loose Coupling** - the services must be designed to interact with other services without having close links between them
- **Service Abstraction** - the only visible part of service abstraction is a service contract. The business logic and particular implementation is encapsulated and invisible for service consumers
- **Service Reusability** - regardless of whether you can see the immediate occasions to reuse the service, services must be designed to be reusable
- **Service Autonomy** - the encapsulated business logic must be implemented independently on other services
- **Service Statelessness** - the services should not persist any information about their state because this can decrease the loose coupling of services
- **Service Discoverability** - the service contracts should be accessible to consumers in an easy way to enable the reusability of services

- **Service Composability** - the services should be designed with regard to the possibility of combining services which enables the creation of different granularity of services

3.3.3 Service Repository

To fulfil principles of discoverability and reusability described in previous chapter you need a catalogue or a repository of services. The particular implementation of repository depends on enterprise requirements and amount of services. You can use simple document, wiki or special software for registering services.

The service repository should provide following functionality:

- **Cataloging and Discovery** - service management is the key function of repository. The service repository should contain both business and technical information. There should be stored relations between services and also other parts of enterprise systems such as applications, processes, etc. The particular scope of stored entities relies on enterprise's requirements. The catalogue should also include search engine to enable reusability and discoverability of service.

- **Operational Support** - for operational department there should be stored information about SLA[24], business and technical owners, escalation matrix which is used for passing information about failures.

- **Impact Analysis Support** - repository should easily provide information about dependencies and relation between services and other stored entities.

- **Life Cycle Management** - the whole life cycle of a service should be covered in repository. Several different roles should use the catalogue during service life cycle:

    - Integration Architects – before implementing a new system, replacing the existing system or developing new service integration, architects can use impact analysis or try to find similar service and reuse it.

    - Analysts – after analysis of new project, solution or requested change, analysts should update service repository and add new services.

[24] SLA is a formally defined contract which guarantees specified parameters of service to consumers (for example: quality, availability, throughput, etc.)

- Developers and Testers – newly added or updated services are implemented by developers. The status of services has to be changed and hand over to testers.
- Operational Employees – tested services are deployed to production environment and operation updates the status of service. The operational employees also use service catalogue during failures of system to inform people defined in escalation matrix

- **Versioning** - sometimes it is necessary that two or more versions of service have to coexist at the same time (for example, not all clients update interface in order to consume new interface of service). Therefore service catalogue should contain logic for maintaining several versions of services.

Besides technical requirement on service catalogue there is one more process requirement. The service repository can be used only if it contains current information about services. Therefore the updating of repository has to be defined in release and management process of enterprises has to be changed.

3.3.4 Enterprise Service Bus

Thomas Erl defines an enterprise service bus as an environment designed to foster sophisticated inter-connectivity between services. It establishes an intermediate layer of processing that can help to overcome common problems associated with reliability, scalability and communication disparity (see (30), page 704).

In modern computing environment there exist several applications providing services through different interfaces (for example web services, remote method invocation, messaging, etc). The ESB enables interoperability across these communication protocols and provides functionality to manage this integration layer easily.

I will mention the key characteristics of ESB. They are compiled on the basis of characteristics published in SOA Based Enterprise Integration (see (14), page 134), Service-Oriented Architecture Compass (see (15), page 36) and Enterprise Service Bus (see (16), page 7):

- **Standards-Based Integration** - an ESB is a backbone for service oriented architecture and enables communication between service

providers and consumers through different interfaces. This is the key reason why an ESB should support large amount of standard communication protocols.

- **Data Validation, Transformation and Enhancement** - supporting several communication protocols requires mapping between different data types of messages. An ESB is very often implemented using XML as a foundation for representing data. This implementation enables standardized validation of input and output data and transformation using XSLT and other standards from XML family. Optionally ESB provides functionality to enhance transferred data (for example security context or policies).

- **Routing and Process Flow** - based on the content of the message and routing rules an ESB can choose the correct provider of a service or distribute service response to several consumers. The possibilities which provide particular ESBs vary from simple content based routing to designing sophisticated process orchestration with parallel calls of individual services and their processing. Sometimes the functionalities of an ESB and a BPMS overlap. But the ESB cannot fully replace the BPMS therefore it provides capabilities for integration with BPMS.

- **Synchronous and Asynchronous Mode** - an ESB should support both synchronous or real-time and asynchronous communication and allow switching between them with regards to underlying protocol.

- **Reliability and High Availability** - as it was already mentioned, an ESB is backbone of SOA which implies it is also a backbone of the entire company. As a mission critical system it must enable high availability and must be reliable. Service requests and responses have to be delivered in required quality and it must assure that they are always delivered even in case of ESB failure.

- **Security** - security is very important issue in SOA. An ESB can contain services providing sensitive information (card numbers, account balances, personal information etc) that must be secured. Among security mechanisms are encryption, authorization, authentication of consumers and data integrity mechanism etc. The advanced implementations of ESB offer the possibility of managing user context and pass them between different applications (for example credentials passed through user certificate transfer to WS-Security header and invoke a service with this user context).

- **Management and Monitoring** – administration of large amount of services published on an ESB is very complicated process. An ESB should

contain the set of features which simplifies the administration. The main feature is monitoring of ESB. Operations should observe messages, loading of individual services or tracking of potential problems. ESBs can also store operational data and provide them to 3rd party system for further analysis. An ESB should also provide other functions like categorization of services and integration of service catalogue or versioning and simple deployment features.

In order to implement functionalities described in previous section, number of components is required in an ESB. The logical view of these components is shown at Figure 16.

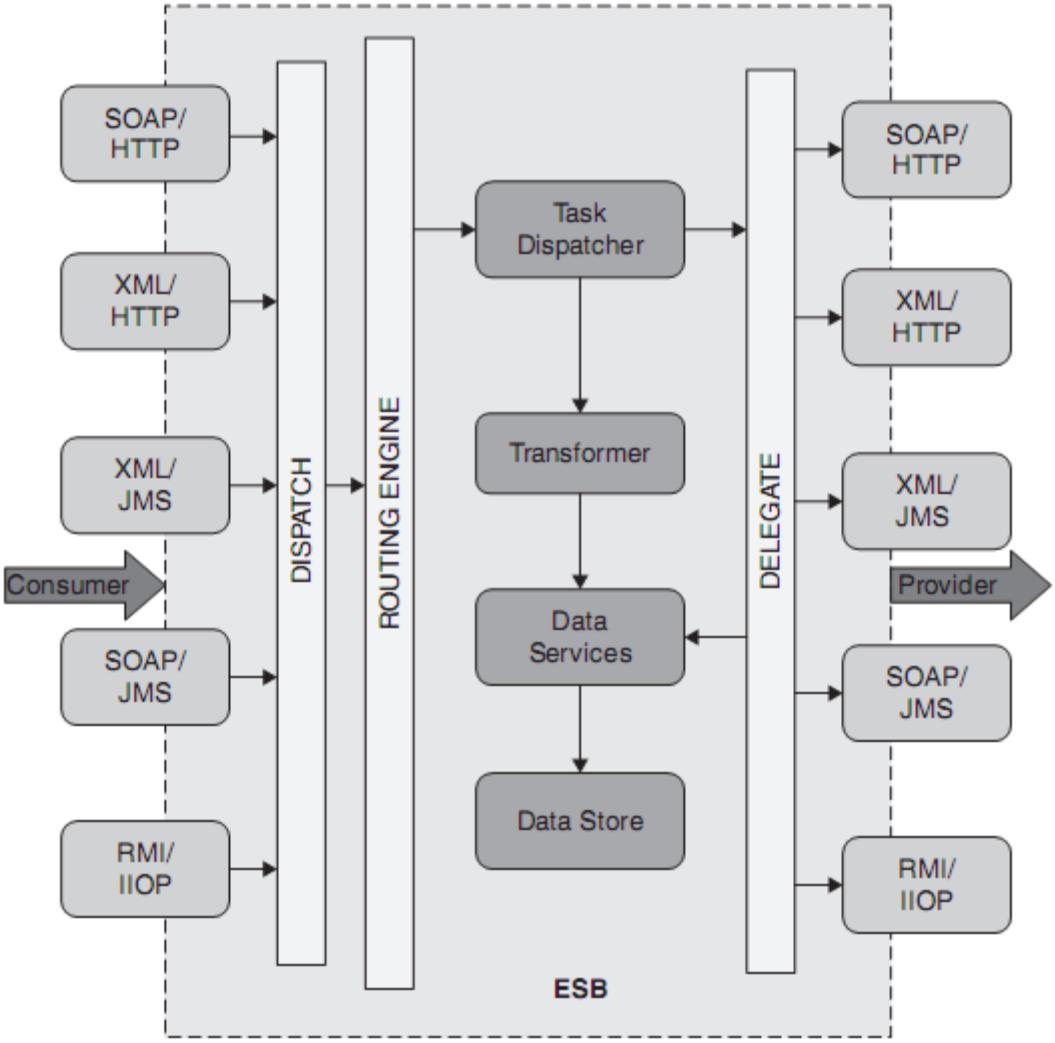


Figure 16 - Some of the logical components of an ESB (see (14), page 145)

The scope of implementation of an ESB and its usage in SOA is determined by enterprise's environment. The book: SOA Getting It Right presents three simple rules when to use an ESB (see (17), page 44):

- When the number of interdependencies between Services, Processes and Schemes becomes more than twice the aggregate number of those elements

- When the process objectives of the SOA begin to span multiple geographically distributed locations and/or federated organizational boundaries
- When you need to integrate services using disparate interaction models

3.4 Enterprise Architecture

The evolution of application architecture enables the creation of very complex business specific applications. Nowadays companies have tens and hundreds of applications and IT becomes the one of the key pillars of their business. The architecture of entire company is called Enterprise Architecture (EA).

The book Enterprise Architecture at Work Modeling, Communication and Analysis describes the EA as the essentials of the business, IT and its evolution and presents following definition:

> "Enterprise architecture: a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure" (see (35), page 3).

The Figure 17 shows that EA consists of four main disciplines: business, application, information and technology architecture. And in a relation to SOA the EA is a superset of SOA.

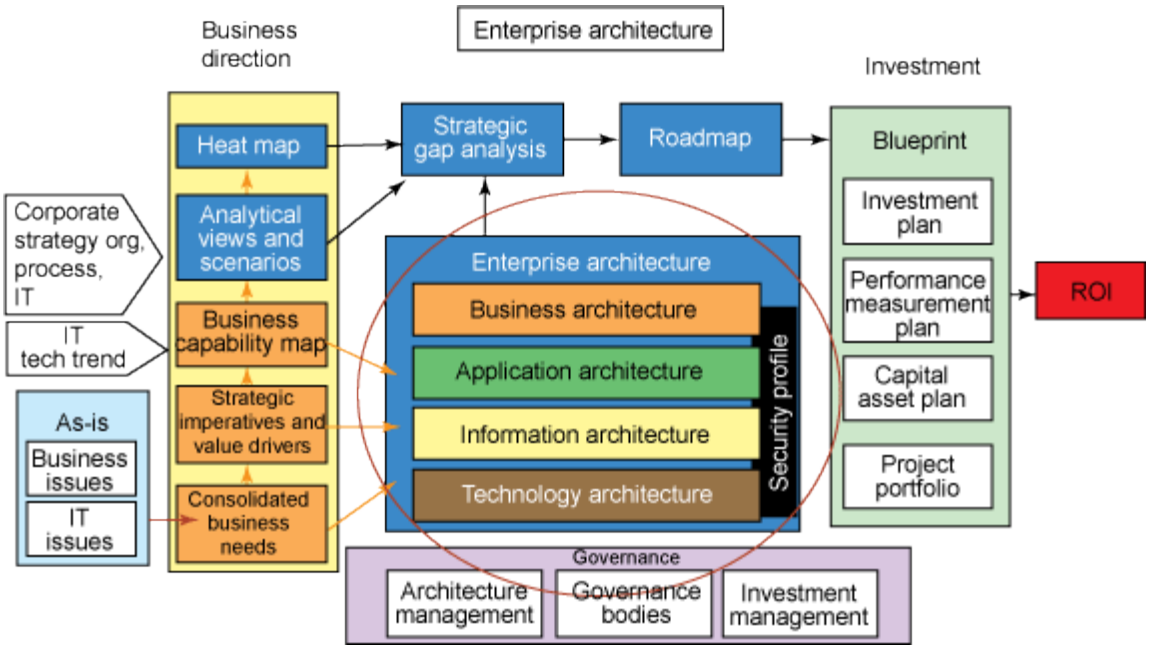


Figure 17- Enterprise architecture (36)

3.4.1 Enterprise Architecture Frameworks

Based on previous definition the scope of EA is very complex and affects the whole company and its business success. During the evolution of EA vendors and IT consortiums tried to define methods and principles (generally marked as EA frameworks) which can help companies with EA. The first mention about EA framework is dated to an article published in the IBM Systems Journal in 1987, titled "A framework for information systems architecture" by J. A. Zachman (see (37)). From that time it was updated several times and today this it is known as Zachman Framework. Since then several frameworks appeared. I will choose and closely describe two of the most used frameworks to complement the overall picture of EA (sources: (35), page 20 and (38)).

The Zachman Framework for Enterprise Architectures

The description of the Zachman Framework is based on John A. Zachman's own web site zachman.com and complemented by information from (35), page 24; (39), page 69 and (40), page 99.

The Zachman framework is typically depicted as a bounded 6 x 6 "matrix". The columns are composed of six interrogatives which cover following aspects: data (what), function (how), network (where), people (who), time (when) and motivation (why). These columns of the matrix are divided into six layers or views on entire enterprise: scope, enterprise/business model, system model, technology model, detailed representation and functioning enterprise. The individual cells represent the intersection of these views and aspects. For example, the function aspect and logical view can be represented by process diagram. The whole concept of Zachman Framework is illustrated the Figure 18.

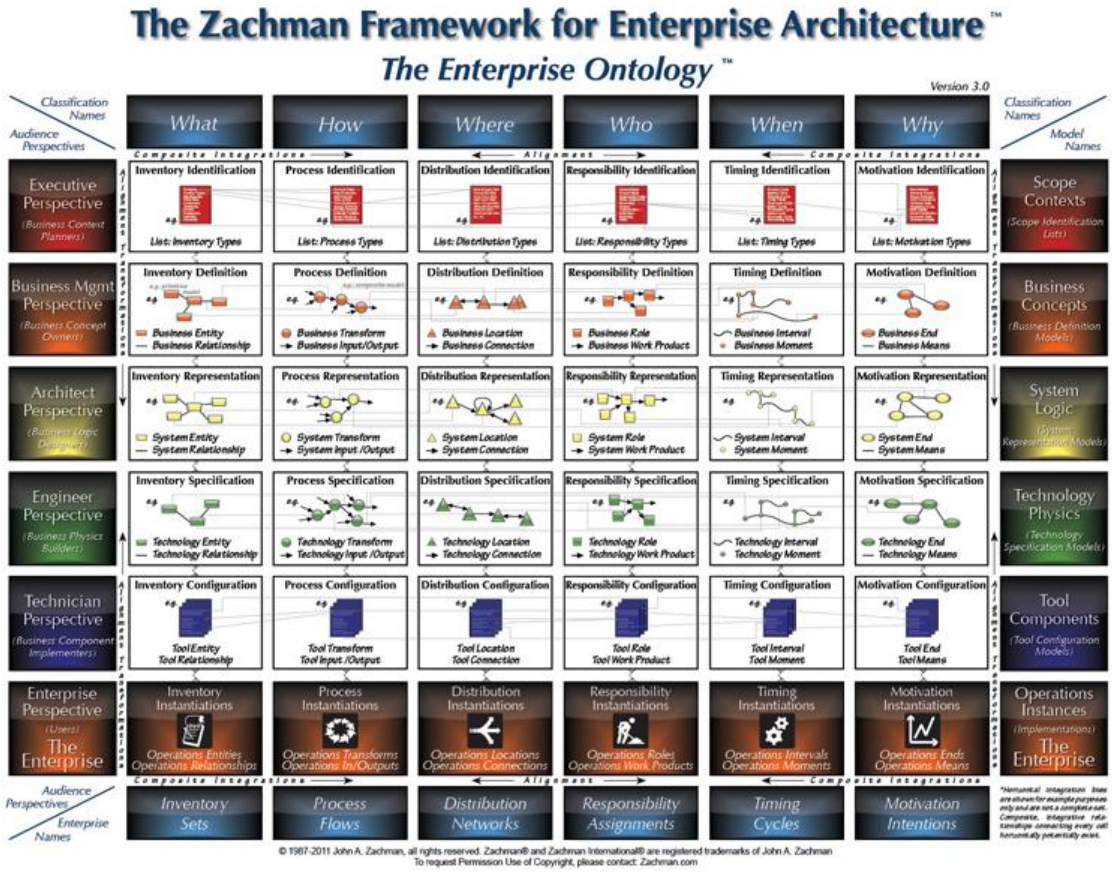


Figure 18 - The Zachman Framework for Enterprise Architecture (41)

Advantages of the Zachman framework are that it covers the entire enterprise ant is easy to understand. It is also defined independently of tools, methods, frameworks or methodologies which can become obsolete during the time. But it also has several problems. Zachman framework is not a methodology but it is ontology for describing the enterprise. Therefore it does not tell how EA should be done and in some cases it can lead to a documentation heavy-approach.

The Open Group Architectural Framework (TOGAF)

The main source for the description of TOGAF is TOGAF documentation issued by The Open Group (see (42)).

TOGAF is a framework for developing an enterprise architecture which is developed and maintained by members of The Open Group[25]. The first version of TOGAF was released in 1995, and since then several versions appeared. The latest version, which is nowadays available, is TOGAF 9.1.

[25] The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, we have a diverse membership that spans all sectors of the IT community (see (46)).

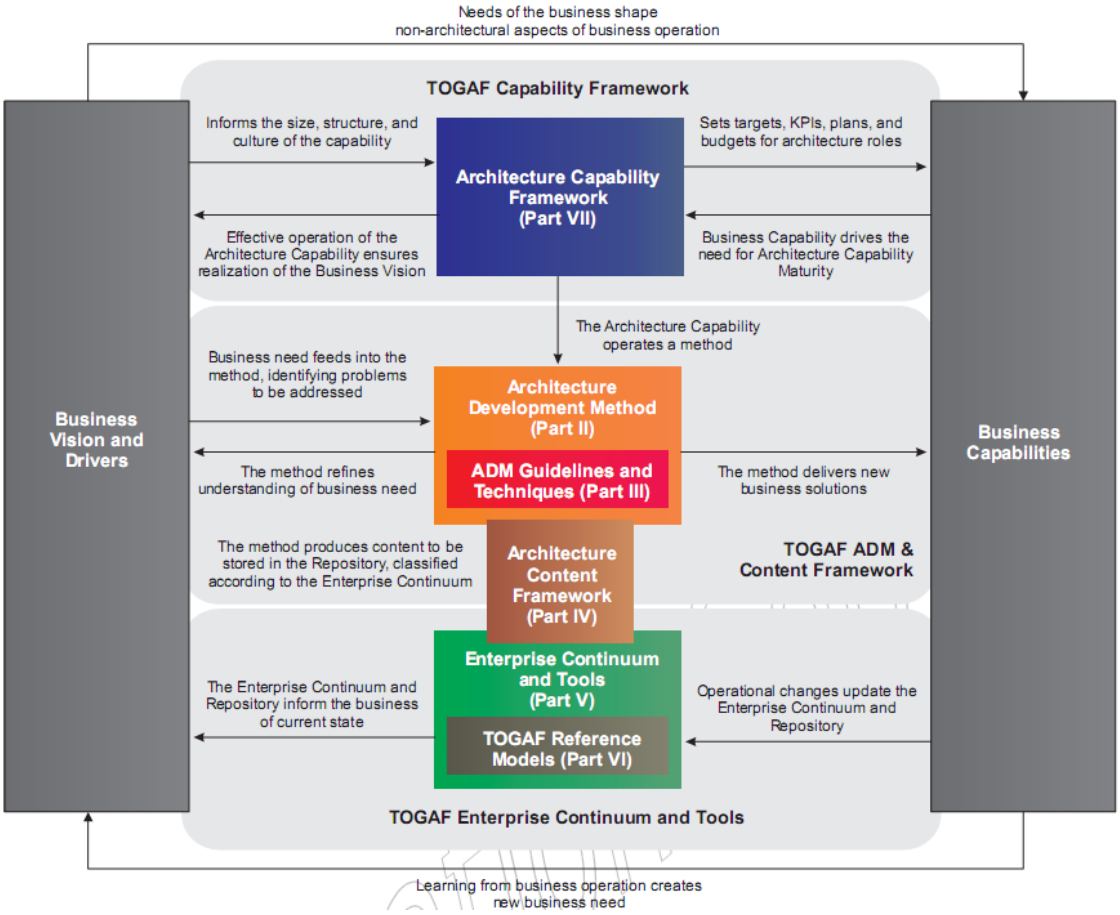


Figure 19 - Structure of TOGAF Document (see (43), page 4)

The exact definition of TOGAF is: "TOGAF provides the methods and tools for assisting in the acceptance, production, use and maintenance of enterprise architecture. It is based on an iterative process model supported by best practices and a re-usable set of existing architecture assets" (see (43), page 10).

The overall framework is divided into several components which are shown at Figure 19.

- **Architecture Development Method (ADM)** – The ADM is primarily process framework for developing architectures. It includes following phases:
    - Preliminary Phase - preparation and initiation activities
    - Architecture Vision – initial phase of architecture development cycle
    - Business, Information Systems and Technology Architecture – describes the development of architecture at these levels.

- Opportunities and Solutions - initial implementation planning and the identification of delivery vehicles for the previously described architecture levels.
- Migration Planning – plan for moving from baseline to the target architecture
- Implementation Governance – architectural oversight of the implementation
- Architecture Change and Requirements Management – process for managing the new change requests and requirements to the architecture

- **Architecture Content Framework –** During execution of the ADM a large number of outputs is created (for example process flows, architectural requirements, project plans, etc.). The Architecture Content Framework divides the outputs into three categories:
  - Deliverable – it is typically documentation that is formally reviewed, agreed and approved by stakeholders[26].
  - Artifact – it is product that describes an aspect of the architecture (for example requirements catalogues, interaction matrix, use-case diagram)
  - Building Block – it is component that delivers architecture or solution (for example processes, data, application software or network)

  The Architecture Content Framework provides a structural model that allows these three categories to be consistently defined, structured, and presented.

- **Enterprise Continuum –** The Enterprise Continuum provides an architecture repository of all architecture assets and methods for classifying architecture and solution artifacts. The Enterprise Continuum enables the organization of re-usable architecture artifacts and solution assets to maximise the enterprise architecture investment opportunities.

  The architecture repository is used to store architectural output at different levels of abstraction, created by the ADM. The repository operates at multiple levels within the organization and facilitates understanding between stakeholders and practitioners.

[26] Stakeholder is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system (See (35), page 2).

- **Architecture Capability Framework –** The Architecture Capability Framework consists of two core concepts:
    - Establishing and Maintaining an Enterprise Architecture Capability **-** In order to achieve effective enterprise architecture an appropriate business capability for architecture, through organization structures, roles, responsibilities, skills, and processes must be put in place.
    - Establishing the Architecture Capability as an Operational Entity **–** The TOGAF states that running the enterprise architecture like any other operational entity leads to a successful enterprise architecture practice. This comprehends the establishment of following areas: financial, performance, service, risk, resource, communication and stakeholder, quality, supplier, configuration and environment management.

The TOGAF provides a standard way, using best practices, to enable the link between a business and its technology infrastructure. TOGAF is flexible about the actual generated architecture and merely describes how to generate an enterprise architecture, not necessarily how to generate a good enterprise architecture.

3.4.2 Enterprise Architecture, SOA and BPM

As it was mentioned in previous chapter the EA is a set of principles, methods and models that are used in the design and realization of an organizational structure, business processes, information systems and infrastructure. From this definition and from the definition of SOA (see 3.3) and BPM (see 2.1) is clear that both SOA and BPM are parts of enterprise architecture.

The Table 1 shows the disciplines of EA (see 3.4) where the SOA and BPM are included.

Enterprise architecture		
Business architecture	BPM	
Application architecture		SOA
Information architecture	BPM	SOA
Technology architecture	BPM	SOA

Table 1 - Classification of SOA and BPM

The SOA and BPM are also very tightly connected as it is shown at Figure 20. The ESB which is a part of the SOA provides services which can be consumed by BPM solutions and vice versa the implementation of a business process can be exposed as service. The BPM and SOA also share the policies, security concepts

and enterprise repository. It follows that SOA and BPM are central and key components of enterprise architecture.

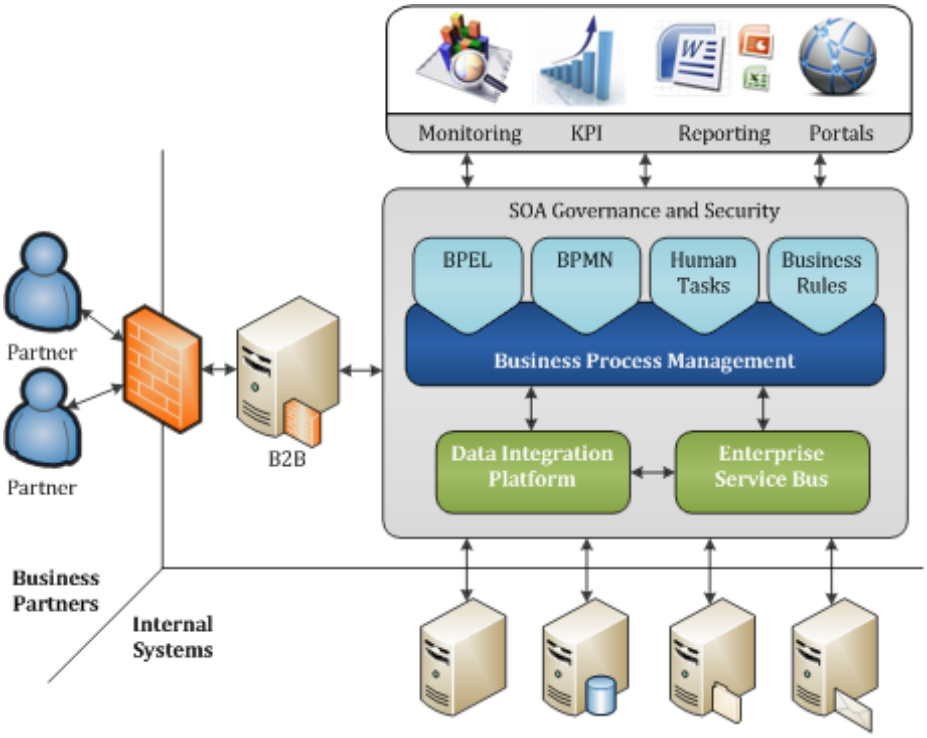


Figure 20 - SOA and BPM

3.5 Application of Business Process Management

For the successful implementation of the BPM projects is required both robust BPM suite and an effective BPM methodology.

The BPM methodology is based on my experience and complemented with OMG Best Practices (see (44)). The methodology provides full-lifecycle and pragmatic approach to the application of BPM in modern enterprises. It is divided into five phases which are closely described below the methodology overview (see Figure 21).

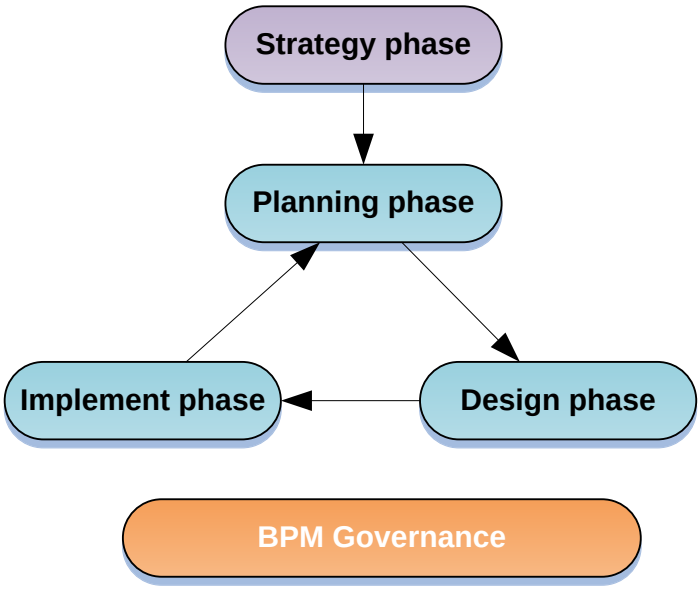


Figure 21 - BPM methodology phases

Strategy phase

This is the initial phase of establishing the BPM in a company. The main goal of this phase is to gather the high level requirements, goals, strategy and define the vision of BPM. To successfully start the BPM project the support from executive sponsors is essential. During this phase is needed to set up the BPM steering committee. The steering committee should be composed of CIO, executive business sponsors and BPM program leader.

Planning phase

After the basic establishment of the BPM program the initial BPM project should be chosen. The application of BPM is incremental process – the vision should be maximalist but it is better to start with small BPM projects. There are several reasons: minimization of risks, better and easier managing of scope and financial budget, faster evaluation of the program, etc.

The processes for the project should be selected with regard to:

- maximization of the processes automation using BPM tool
- low complexity of the processes
- high importance of the processes

The main goal is to select the set of processes where their automation brinks the great benefits. This has positive impact on application of BPM and acceptance by company employees (sometime this is called a quick-win strategy).

Together with the definition of project scope there should be formed the project team containing: BPM project manager (this is not the same role as program manager), business owners of selected processes, business analyst and technical specialist.

The last part of this phase is a selection of a BPM suite which should support the needs and requirements of project and be compliant with the vision of BPM. The literature very often presents the selection of a tool in later phases. But from my experience holds true that the tool must be selected before the BPM project starts. This selection influences the total costs of ownership and the business case of the project.

The key outputs of this phase are: scope of the new BPM project with business case, definition of project team and selection of BPM suite.

Design phase

The previous phase starts the new BPM project. The following phases will correspond with a life-cycle of business process management (see chapter 2.2).

This phase is primary about analysis of processes. It can optionally start with the analysis of current state (as-is analysis). The next steps are requirement analysis, modeling and optimization of processes. The optimized models go as a one of the inputs for the next phase.

In this phase the process repository should be set to enable sharing and collaboration of project members.

Implement phase

The implement phase in this point view represents the adding of technical details, business rules and task implementation, testing, deploying, running the processes and reports creation.

This phase ends with successful acceptance of the project and deployment of all processes to production environment. After the closure of the project the BPM steering committee should analyze the impacts of the project and edit the vision and goals of BPM program. After successful adoption of the BPM project the others should be iteratively started.

BPM governance

The BPM is constantly evolving platform which should be governed by BPM program manager. One of the outputs of the first BPM projects should be methodologies, minimum standards, best practices, templates, etc. These artifacts are the basement for the BPM governance. Besides the documents, the processes for approving the newly developed processes, meeting the defined

standards and best practices, etc. should be set. And to each process the responsible roles should be assigned.

4. Business Process Management Suites

BPMS is a new class of software that allows organizations to devise process-centric information technology solutions. Process-centric means that BPMS solutions are able to integrate people, systems, and data (see (3), page 50).

Gartner defines top four BPMS usage scenarios (see (18)):

- Implementation of an Industry-Specific or Company-Specific Process Solution – the organization wants to improve business performance through automation and better coordination of business processes
- Support for Continuous Process Improvement Program – a BPMS is selected to provide a stable platform to deliver continuous process improvements
- Support for Business Transformation Initiatives – a BPMS should support quick business process re-engineering to fulfil the needs of rapidly changing market.
- Support for Process-Based SOA Redesign – upgrading of a SOA platform leads IT departments into enriching the platform with BPM benefits.

4.1. Key Components of a BPM Suite

A BPMS integrates the following BPM-enabling technologies, many of which are also available as independent products, into a suite (see (2)): Graphical Business Process Modeling, Orchestration Engine, Process Analysis Tool, Rule Engine, Process Registry/Repository, Simulation and Optimization, Integration, Document/Content management technology.

The following chapters describe the individual parts of BPMS. The exact list of products which should be included in BPMS doesn't exist. Therefore the list of BPM-enabling products reflects current usage of BPM suites defined by Gartner (see (18)), BPM & Workflow Handbook (see (46), page 75) and my personal knowledge. The individual parts of BPMS are named based on its main purpose – the concrete names of products in BPM Suites might be different.

4.1.1. Integrated Development Environment

The main goal of this part of BPMS is to create visual representation of the business process. The processes are designed by business analyst (see chapter

2.2) and therefore the modeling environment should be understandable for non-technical users. The environment should also support IT professionals' perspective for adding greater technical details to business processes. The integrated development environment (IDE) should also support other parts of business process lifecycle such as simulation, testing, debugging and deploy.

The graphical environment contains a design palette with a set of components, services or other processes from the enterprise repository that can be used in the process. Nodes in a business process model represent activities or actions performed (service calls, transformations, flow controls, etc.). Current graphical environment also should support easy integration with human tasks, business rules or a middleware platform. The IDE should support current standards for business process modeling – primarily the BPMN (for more information see 2.3.1).

The IDE should besides the designing the processes also enable them to be tested and simulated. The simulation and its outputs are directly influenced by usage of data and processes. Therefore the IDE should provide function which easily creates the environment for testing and simulation. It should also offer tool for analyzing the results of simulation and testing. The advanced debugging and optimization possibilities simplify the development of business process and save time.

Finally, the IDE should provide function for packaging the business process and deployment to the execution engine.

4.1.2. Execution Engine

Once the processes are designed and simulated they need to be passed to production environment. The execution of business processes is provided by an execution engine. The main difference between process execution engines and traditional EAI products (enterprise service bus, message broker, messaging platforms, etc.) is the state management as well as human tasks and rules integration.

Before the designed processes can be executed, they have to be enriched by technical details and transformed to the execution language. One of the most known standards is BPEL which was published in 2002 (for more information see chapter 2.3.3). When business processes were modeled with the use of UML or Microsoft Visio, the reuse of business models was minimal. After publishing the BPEL standard vendors defined their own graphical representation for BPEL and tried to maximize the reuse of models. But the BPEL models were not understandable to business users. Current BPMS very often use BPMN 2.0 as a modeling standard. This version also contains the standard transformation to BPEL and therefore it is executable on BPEL engine. BPMN 2.0 has its own

execution semantics and the vendors started to implement BPMN engines directly without any need of transformation to BPEL.

Apart from the process execution, the execution engines provide functions for version management. Versioning of business processes is very important when you want to update your process definition. After deployment of a new version the running instances are not affected, but the new process instances are started according to updated definition. Some of the engines also provide the option to update running process instances to the new definition in case it is compatible with the old one.

Another group of functions are instance control and monitoring. These functions are used by management console or other business process monitoring tools (more information about monitoring you can find in chapter 4.1.5).

4.1.3. Business Rules

Some vendors extend capabilities of BPMS by providing support for business rules engines or complex event processing.

Business Rules Group (see (19)) defines business rules from two perspectives:

- Business Perspective - a business rule is guidance if there is an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere.
- Information System Perspective - a business rule is a statement that defines or constrains some aspects of the business. It is intended to assert business structure, or to control or influence the behaviour of the business.

BPMS can directly contain business rules engines (BRE) or they can provide interface to one of the existing BREs. BRE extension of BPM suite consists of design environment for business rules, repository for storing rules and tools for execution. Current BPM suites typically provide direct BRE extension. Users can in one design studio model both business processes and rules, and also execution engine or repository can work with both entities.

The big advantage of business rules support is that when you want to change the constraints of business process you can do it simply by editing the rule without any need for changing or redeploying the process.

4.1.4. Human Tasks

The large part of business processes is not supported by systems or applications. There are many human interactions and manual activities. It is necessary to integrate people into business processes. The first specification of human task integration was presented as a BPEL4People. Today the specification is under the control of OASIS and it has two parts:

- BPEL4People – is the extension for BPEL which enables to address human interactions from BPEL process
- Web Services Human Task (WS-Human Task) - introduces the definition of standalone human tasks, including their properties, behaviour and the operations used to manipulate them. Capabilities provided by WS-Human Task may be utilised by web service based applications under WS-BPEL process (see (20), page 165).

The current BPM suites provide more features than the BPEL4People specification covers. Besides human interactions the BPMS provides deadlines, escalations and notifications for human tasks. Also one of the parts of BPMS is a portal for users where they can manage their tasks. The whole solutions can load the current organization structure and within the business process you can assign tasks not only to single person, but also to department or to a group of your employees.

4.1.5. Enterprise Repository

The process governance is one of the crucial thinks for successful implementation of BPM in enterprise architecture. The enterprise or component repository is a key tool for process governance (similarly as service repository for SOA – see 3.3.3). The enterprise repository can be upgraded to service repository and it can store both SOA (web services, endpoints, etc.) and process components (business rules, processes, workflows, etc.) or it can be a separate tool only for business process management.

The Enterprise repository is the central repository of individual components. It contains artifacts of components as well as metadata about them. Similarly as the service repository the enterprise repository provides following functions:

- Catalouging and Discovery
- Operational Support
- Impact Analysis Support
- Life Cycle Management
- Versioning

4.1.6. Monitoring and Process Management

The previously mentioned components are primarily used in process design, implementation or execution phase. The management and supervision phase (see 2.2.3) requires tools for process management, monitoring and reporting. The BPMS should provide these capabilities to both business and IT users.

The enterprises have hundreds, thousands or even more processes based on company size. Each process can have several instances, but for example the mortgage application can also have hundreds or even thousands of parallel instances. The BPMS has to offer functions to manage and monitor the large number of parallel executing instances of different processes.

These functions are typically provided by a management console. The console offers following features:

- Managing of processes: grouping to logical partitions, deployment, versioning,…
- Managing of instances: reports over instances (by status, duration, bottlenecks,…), process instance life cycle detail,…
- Human tasks console or portal can be an individual component of BPM suite or it can be one part of the management console

The management console also contains features for business analytics or reporting. Business Activity Monitoring (BAM) which determines abnormal events in the process and creates alerts when events occur. The BPM suites also provide very important set of information about your company. These data can be used with business intelligence or analytics tools and the business users then can easily react to abnormal events or use advanced reporting to monitor department, clients and segments based on data retrieved from a BPMS.

4.2. BPMS Products

In previous chapter the key components of BPM suite were described. Now I would like to choose the concrete BPMS products and map their parts to defined key components. The selected products will also be compared based on criteria defined in chapter 5.

The market offers tens of BPM suites from different vendors and also there are many open source BPM suites. For commercial products the research and advisory companies (ex. Gartner, Forrester) regularly prepare their comparison (ex. Gartner Magic Quadrant, Forrester Wave). On the other hand the open source products typically cover some parts of BPM suites (process engine,

business rules, etc.) but very few of them contain all components. My goal is to choose one commercial product and one open source product which cover all components of BPM suite. The comparison of products from different domains will be very interesting and may help business and IT specialists to decide which platform they will use for implementation of BPM. Also the comparison will be done with regard to the Czech market unlike the comparison from Gartner of Forrester (some vendors even have not got company representation or partners at the Czech market).

4.2.1. BPM Suite Selection

Based on my experience and the comparison documents from research and advisory companies I chose the following criteria for product selection:

- **Product or portfolio completeness** - the vendor or open source community should provide all key components mentioned in chapter 4.1. The components should be packed into the BPM suite.
- **Personal experience –** the BPM suites are very complex solutions and therefore one of the criteria is personal experience with the suite. By virtue of personal experience I will compare and describe the suites also from the practical and usable point of view.
- **Availability of sources (commercial products)/Availability of support (open source products)**
    - One of my goals of the thesis is to compare products in practice and implementation the defined processes. But the plenty of vendors and also open source communities don't enable to download their product. The possibility to download and try the product is typically restricted to their partners and customers. These products are not suitable for the comparison because it is very hard to get access to their developer or trial edition.
    - The open source products provide their sources with downloading but not all of them provide support for their products. For application at the enterprise architecture in companies this is one of the key decision factors because the BPM suite becomes one of the core systems in the enterprise architecture. The companies need to have support for its products, install patches of bugs or upgrade the BPM suite.
- **Representation on the Czech market –** the project for the application of BPM is quite complex and requires support from the product's vendor.

The vendor should provide technical and sales consultants, courses, certifications, training materials, etc.

Each of the mentioned criteria will be evaluated by three levels (low, medium and high) and percentage representation of criteria importance. The meaning of each level and the concrete percentage is shown at the following table. The importance of each criterion was set based on my experience and consultations with SOA and BPM experts.

Selection criteria	Importance	Evaluation level		
		High (5)	Medium (3)	Low (1)
Product or portfolio completeness	50%	Vendor provides all components	The product covers at least 6 components	Other
Personal experience	10%	Commercial experience	School or non-commercial experience	Other
Availability of sources (commercial products)	30%	Developer edition of product is available to download and it is for free	Not applicable	Developer edition is not available
Availability of support (open source products)	30%	Support is available.	Not applicable	Companies cannot buy product support
Representation on the Czech market	10%	Vendor or open source community has representation on the Czech market	Partners have representation on the Czech market	Other

Table 2 - Selection criteria weighting and importance

The result is calculated as a sum of evaluation levels for each criterion multiplied by importance.

The comparison result gives information about suitability of product to the comparison but does not compare the products themselves.

Commercial products

The list of commercial products was compiled on the basis of Gartner's Magic Quadrant for Business Process Management Suites from 2010 (see (18)) and my personal experience.

The result of product selection shows Table 3. For each vendor the latest version of product available in October 2012 was selected.

Vendor	Product or portfolio completeness	Personal experience	Avail-ability of sources	Representa-tion on the Czech market	Result
	50%	**10%**	**30%**	**10%**	
Pegasystems	High	Low	Low	Medium	3,2
IBM	High	Medium	Low	High	3,6
Software AG	High	Medium	Low	High	3,6
Progress	High	Low	Low	Medium	3,2
Appian	High	Low	Low	Low	3
Oracle	**High**	**High**	**High**	**High**	**5**
OpenText	High	Low	Low	Medium	3,2
Adobe	Medium	Low	High	High	3,6
EMC	Medium	Low	Low	Medium	2,2
Fujitsu	High	Low	Low	High	3,4
Microsoft	Low	Medium	High	High	2,8
Symantec	Low	Low	Low	High	1,4
HP	Low	Low	Low	High	1,4
Kofax	High	Low	Low	Medium	3,2
TIBCO	High	Medium	Low	Medium	3,4
K2	High	Low	Low	Low	3
SAP	Medium	Low	High	High	3,6
Intalio	High	Medium	High	Medium	4,6
BizAgi	High	Low	High	Medium	4,4
AgilePoint	Medium	Low	Low	Medium	2,2
Handy Soft	Medium	Low	Low	Low	2
Perceptive Software	Medium	Low	Low	Medium	2,2
PNMSoft	High	Low	Low	Medium	3,2
Active Endpoints	High	Medium	Low	Medium	3,4
Newgen Software Technologies	High	Low	Low	Medium	3,2
Cordys	High	Low	Low	Medium	3,2

Table 3 - The selection of commercial product

Based on the selection criteria Oracle BPM suite was selected as a commercial product for comparison. The detailed information about Oracle BPM suite is mentioned in chapter 4.2.2.

Open source products

The list of open source products was compiled on the basis of Patterns-based Evaluation of Open Source BPM Systems: List of top open source BPM / workflow solution (see (49)) and my personal experience.

The result of product selection shows Table 4. For each open source community the latest version of product available in October 2012.

Vendor	Product or	Personal	Avail-	Representa-	Result

	portfolio completeness	experience	ability of support	tion on the Czech market	
	50%	**10%**	**30%**	**10%**	
Intalio[27]	Low	Medium	High	Medium	2,6
ProcessMaker	Low	Low	High	Low	2,2
BonitaSoft	High	Low	High	Low	4,2
Activiti	Medium	Low	Low	Medium	2,2
uEngine	Low	Low	Low	Low	1
Cuteflow	Low	Low	High	Low	2,2
jBPM	**High**	**High**	**High**	**High**	**5**
Enhydra Shark	Low	Low	High	Low	2,2
Apache	Low	Medium	Low	High	1,6
Joget	Low	Low	High	Low	2,2
jSonic	High	Low	Low	Low	3
NexusBPM	Low	Low	Low	Low	1
Orchestra	Medium	Low	High	Low	3,2
WSO2	Medium	High	High	Medium	3,8

Table 4 – The selection of open source product

Based on the selection criteria JBoss jBPM suite was selected as an open source product for comparison. The detailed information about JBoss jBPM suite is mentioned in chapter 4.2.3.

4.2.2. Oracle Business Process Management Suite 11g

Oracle Business Process Management Suite 11g (the last version was released in February 2012 under version: 11.1.1.6.) is developed by Oracle Corporation.

The BPM suite contains complete set of tools for creating, executing, and optimizing business processes. These tools cover the whole life cycle of business process management and enable all roles to engage in a business process.

Oracle BPM suite is based on Oracle's SOA Suite[28]. This enables full integration of these products and reuse of Oracle SOA Suite components in business processes developed in Oracle BPM. The architecture of Oracle SOA and BPM suite is described at Figure 22.

[27] Intallio Community Edition is considered for comparison of open source products.

[28] Oracle SOA Suite provides a comprehensive suite of products for developing, securing, and monitoring service-oriented architecture (SOA). Oracle SOA Suite 11g provides a unified runtime engine based on the SCA standard. See (58).

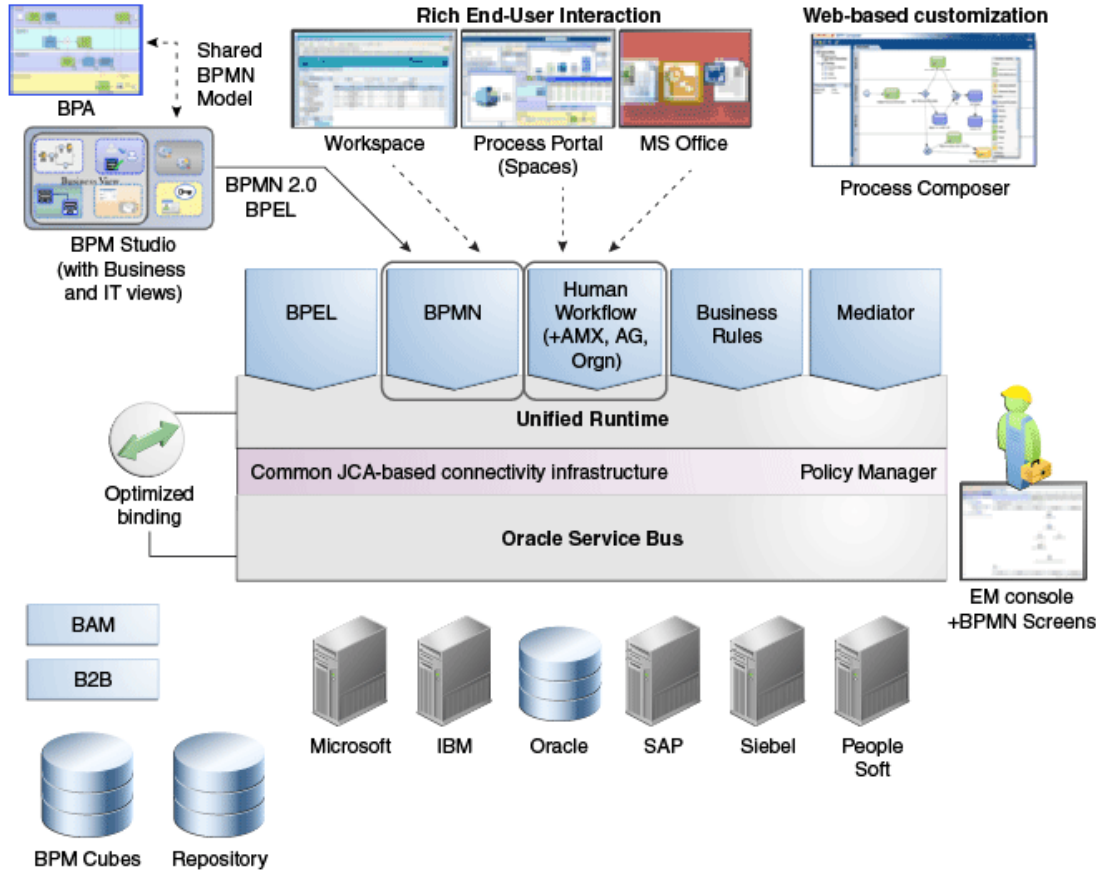


Figure 22 - Oracle SOA and BPM Suite architecture overview (50)

Oracle BPM Suite 11g tools

The Figure 22 shows architecture of SOA and BPM Suite together. But in this thesis I will be focused only to BPM Suite. In despite of Figure 22 the separate Oracle BPM Suite doesn't contain for example BPEL Engine, Enterprise Service Bus, BAM, B2B etc. Oracle BPM Suite 11g consists of six tools which are closely described below:

- **BPM Studio**
  - Is a part of the Oracle JDeveloper IDE
  - Environment where process analysts can design business process models with BPMN 2.0 and run process simulations.
  - Enables implementation of business processes with other components such as adapters, human workflow and business rules.
  - Enables deployment of business processes to BPM server
- **Business Process Composer**

- Process Composer is the new web-based BPMN tooling that lets business users customize BPMN processes created in BPM Studio
- Business user changes can be deployed directly from Composer without engaging IT

- **Run-time Engine**
  - The Oracle BPM Suite 11g has a native BPMN 2.0 run-time engine
  - Business processes modeled and implemented in BPM Studio can be run natively on the BPMN run-time engine without any need of translation to BPEL. The native BPMN engine was added just at the 11g version. The previous version has only BPEL engine which is now part of Oracle SOA Suite 11g
  - The engine also execute business rules and human workflows
- **Oracle Metadata Service (MDS) Repository**
  - Provides a repository that is used to store data about processes and their components
  - Repository is integrated from both BPM studio and Business process composer
  - Enables to share projects and project templates between process analysts and process developers
- **Enterprise Manager for BPM**
  - Enterprise Manager allows rich end-to-end tracking and monitoring of BPMN processes, including exception management
- **Process Modeling Spaces**
  - Process Spaces is a collaborative workspace built on top of Web Center Spaces
  - Company employees manage their tasks and business processes in which they are engaged using this portal
  - Process modeling spaces also enables to define process and activity dashboards.

The Table 5 maps Oracle BPM tools to key components of BPM suite defined in chapter 4.1.

Key Components of BPM Suite	Oracle BPM Suite tool
Graphical process modeling	BPM Studio Business Process Composer
Execution engine	BPMN Run-time Engine
Business rules	BPM Studio Business Process Composer Run-time engine Process Modeling Spaces
Human task support	BPM Studio Business Process Composer Run-time engine Process Modeling Spaces
Enterprise repository	Oracle Metadata Service Repository
Monitoring and process management	Process Modeling Spaces Enterprise Manager for BPM

Table 5 - Mapping of Oracle BPM to the key components of BPM

4.2.3. JBoss jBPM 5.4

jBPM is open-source BPM suite distributed under Apache license. It is written in Java and it belongs to Red Hat company. jBPM is also easily integrated with a variety of frameworks – for example: Seam, Spring, OSGi.

It allows you to design, execute and monitor business processes, throughout their life cycle. And similarly as Oracle BPM suite it contains all key components of BPM suite.

The architecture of jBPM is shown at Figure 23.

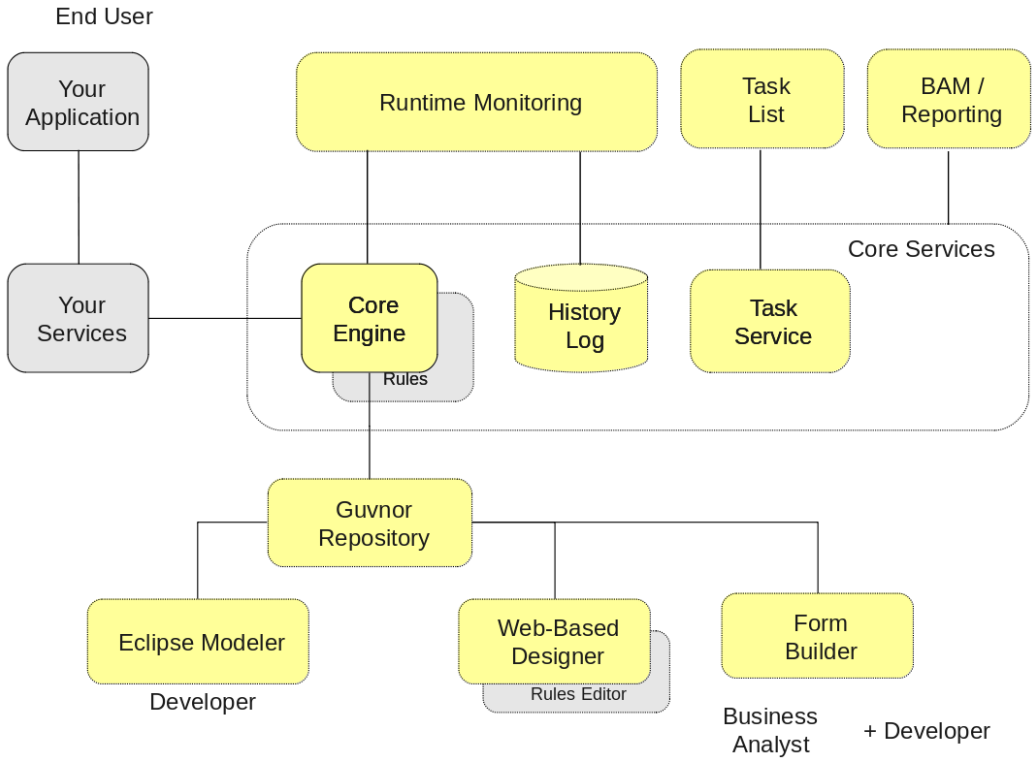


Figure 23 - jBPM architecture overview (51)

jBPM 5.4 consists of six tools which are closely described below:

- Core engine
    - jBPM from its version 5 support similarly as Oracle BPM 11g BPMN 2.0 and it contains native BPMN engine without no need of translation to BPEL
    - Part of the core engine is also human task service which manages the life cycle of human tasks and enables the company employees to participate on business process
    - Unlike the Oracle BPM the business rules engine isn't directly part of core engine but jBPM can be easily combined with the Drools[29] project to support processing of rules and events.
- Eclipse editor
    - The desktop development environment is based on Eclipse IDE and jBPM plugin. This allows creating business processes, business rules, human tasks etc. The Eclipse editor is more technical tool and it is not convenient for business users.

[29] Drools 5 introduces the Business Logic integration Platform which provides a unified and integrated platform for Rules, Workflow and Event Processing. See (60).

    - The editor also enables to validate, test and debug the implemented processes using graphical debugger.

- Web-based designer
    - The web designer is more targeted to business users.
    - It offers the web editor for viewing and editing the business processes similarly as Eclipse editor.

- Form builder
    - This web-based tool is intended to managing forms for human task.
    - It enables to create and edit form for both starting a process and completing human task using WYSIWYG editor.

- Guvnor Repository
    - The repository supports to store business processes and other artifacts. It provides functions for collaboration and versioning across all designers: Eclipse editor, Web-based designer and Form builder
    - Currently the Web-based designer and Form builder are parts of Guvnor Repository

- Web-based management console
    - It allows getting list of processes, running instances, their states, etc. You can visually display the business processes and inspect the business activity or current state.
    - The company employees can start business processes or complete their tasks through the management console.
    - The management console also offers business activity monitoring and creation of customizable reports.

The Table 6 maps jBPM tools to key components of BPM suite defined in chapter 4.1.

Key Components of BPM Suite	**jBPM suite tool**
Graphical process modeling	Eclipse editor Web-based designer
Execution engine	Core engine
Business rules	Eclipse editor Core engine Web-based management console

Human task support	Eclipse editor Form builder Core engine Web-based management console
Enterprise repository	Guvnor Repository
Monitoring and process management	Web-based management console

Table 6 - Mapping of jBPM to the key components of BPM

5. Comparison of Business Process Management Suites

As it was mentioned in previous chapter currently there exist tens of BPM suites from both commercial vendors and open source communities. The selection of proper solution is very difficult because the requirements of individual companies are very different.

This chapter summarizes the criteria which should be considered for BPM suite comparison. Before the final comparison the criteria are completed with practical experience from business process implementation. This part focuses on features of both products. Based on the defined criteria and practical experience the products are compared.

5.1. Comparison Criteria

In this chapter I would like to summarize the criteria which should help business and IT specialists with finding the convenient BPM suite.

The complete list of criteria was compiled on the basis of Software Evaluation: Tutorial-based Assessment (see (52)), Criteria for measuring and comparing information systems (see (53)). To fully meet the BPM suite specialties the criteria were completed with my personal experience and consultations.

The criteria are divided into four groups. The order of the criteria is random and doesn't correspond with their importance.

5.1.1. Organization Criteria

The organization criteria are primarily interesting from the point of view of top and middle management.

Company standards - international companies can prescribe the company standards for a particular software tools or they can even bought the licenses and resell to local branches of company. The local branches of company have to consider this criterion before selecting the BPM suite.

Total cost of ownership (TCO) – this criterion definitely belongs among the most important ones. Every large company calculates the cost of the solution before it starts the new project – in this case to apply BPM into the enterprise architecture. The TCO is as standard calculated to five years and it is divided into two main parts OPEX – operational costs and CAPEX – capital costs.

Representatives on the Czech market – the application of BPM suite is very complex project which involves the whole company. Due to its complexity it is typically implemented together with IT company. Therefore the company needs to have sales and technical consultants who can help with the setup of the project and implementation of the BPM suite. This doesn't necessary mean the number of partners which can implement the BPM suite but also the availability of consultants on the labor market.

Support – the BPM suite definitely belongs among the core systems of a company. Therefore the requirement of service level agreement (SLA) or paid support very often belongs to key criteria. This limits many open source projects but from the company perspective it is very logical decision criterion.

References – as I mentioned previously the companies typically demanded the IT companies to implement the BPM suite. From my personal experience the references were one of the key criteria in the selection of the appropriate software. The companies need to be assured that the IT company has a real reference of implementation and application in the enterprise company.

Vendor Lock-in – the criterion represents how easy or difficult it should be to change the BPM suite. The level of vendor lock-in reduces the number of the supported standards and the complete documentation. Some vendors also provide tools for automation migration from different vendors.

5.1.2. Individual Criteria

These criteria were defined from the user's point of view. For BPM suites these are the company employees (mostly business users) who work with tasks, update business rules, create reports, etc. Some of these criteria are quite subjective and the easiest way to express them is the comparison with other BPM suites.

Localization – the majority of BPM suites is developed by foreign vendors or communities and therefore their native language is typically English. But for company employees this should not be convenient and the localization requirement can be very important.

Learnability – how easy is it for users to get to know the working environment

Efficiency – represents the amount of time to complete the task.

User friendliness – partly supports the learnability and efficiency but it also defines how pleasant is the usage of BPM components (mainly processes and tasks management)

Device support – the users can access to a BPM suite using different devices. Those can be web browsers, desktop applications, but also the mobile or smart phones. For example for task management this is a very important requirement.

5.1.3. Information Criteria

This set of criteria relates to quality and access of information about a BPM suite.

Documentation - documentation should cover all aspects and features of the product. The documentation should contain installation and administration manual, documentation for technical, business and target users. Besides the classical documentation, the tutorials, case studies, best practices, methodologies and examples are very useful too.

Courses and certifications - the courses can prepare your employees for rapid development of business processes using the concrete BPM suite. The courses should be tailored to the course attendants. The vendor or community should also issue certificates to simplify the recruitment of new consultants.

Quality and completeness – this criterion is very important for all users. Required information should be easy to find and also clear and comprehensive.

Communities – both the open source and commercial products typically have several communities where you can discuss your issues. This is very helpful when you need to solve any problems or bug. And it can be another source of information besides the paid support.

Accessibility – the access to the information is very different. Typically the commercial vendors provide for free only the basic set of information. After buying their product they add other documents and finally after buying a support you have the full access to documentation, known bugs and issues.

5.1.4. Technology Criteria

The technology criteria include primarily technical requirements on BPM suites which support the whole life cycle from design to operation. On the other hand, this group of criteria doesn't contain requirements on BPM components and their completeness (these criteria are mentioned in following chapter).

Product stability – attributes of BPM suite such as product age, backward compatibility, regular releases and others were placed under this criterion.

Extensibility – is it possible to easily extend the possibilities of BPM suite. For example: easy integration with both vendor/community and third party

software (ESB, BAM, IDM…), possibility to create custom plugins or connectors, etc.

Interoperability – BPM suite should support a wide range of hardware platforms, application servers, database and operation systems

Security Support – initiation of new process or changing its status should be restricted to concrete roles. The BPM suite should support definition of the security roles and rules into business processes.

Identity Management Integration – this criterion relates to previous point. The security roles should be loaded from central identity management tool (IDM). The task can be assigned to whole departments or escalated after expiration of time period. And this requires access to the organization structure which is typically also part of an IDM.

Performance – the business processes are typically long-running and the bottle neck is typically tasks management and not the performance of a suite. But very important parameters of BPM suites are the number of running instances, response time of web-based designers or management consoles, number of parallel working people, etc.

Clustering and High Availability – the BPM suite belongs among core systems of each company. Therefore this solution should be deployed in high availability mode which means that two or more independent servers are connected into cluster. When one of the servers shuts down the other should be able to run its processes without any failure.

Standards support – the BPM suite should support standards at least on the level of chapter 2.3.

Connectors/Adapters – as the BPM suite is highly integrated software it should support many standards for integration. The basic ones are: web services and other middleware technologies, databases and file systems. For easier integration there also exist connectors for widespread applications (for example: CRM, ERP, DMS or ECM tools).

5.2. Business Process Implementation

In this chapter I would like to summarize my personal experience from the implementation of the business process using Oracle BPM 11g and jBPM 5.4.

The defined process was selected to contain both basic and advanced activities, human tasks, business rules and external service calls. The complete definition of the process is presented in the next chapter 5.2.1

The implementation will be described in the following life-cycle phases:

- Installation and configuration – the basic installation of the product
- Design – design of the process (business point of view)
- Implementation – technical implementation (technical point of view)
- Testing – unit test of business process
- Deploy – packaging and deployment on execution engine
- Execution and monitoring – processes and tasks management
- Reporting – creation of custom reports

The system configuration for the business process implementation has following parameters:

- Operation system: Windows 7
- RAM: 4 GB
- Processor: Intel i5 - Dual Core (2,5 GHz)
- HDD: 250 GB

5.2.1. Definition of Business Process

The process used for implementation contains all elements of business process: business activities, external services, human tasks and business rules.

The business process implements solution for submitting sales quotes across several departments. The process was defined based on Sales Quote Processing presented in Getting Started with Oracle BPM Suite 11gR1 (see (22), chapter 5) and complemented to enable the complete comparison.

The process definition:

- The business process is initiated by Sales Representative who enters the details about the quote.
- The business rule is applied to decide whether the quote should be reviewed by Business Practices.
- After review from Business Practices the quote has to be approved for the deal structure and terms by Approvers.

- Final part of the process is the finalization of a contract by Contractors and calling an external service to save the quote.
- When anyone of the roles doesn't approve of the quote the process goes to the activity for entering the quote details and process continues from the beginning.

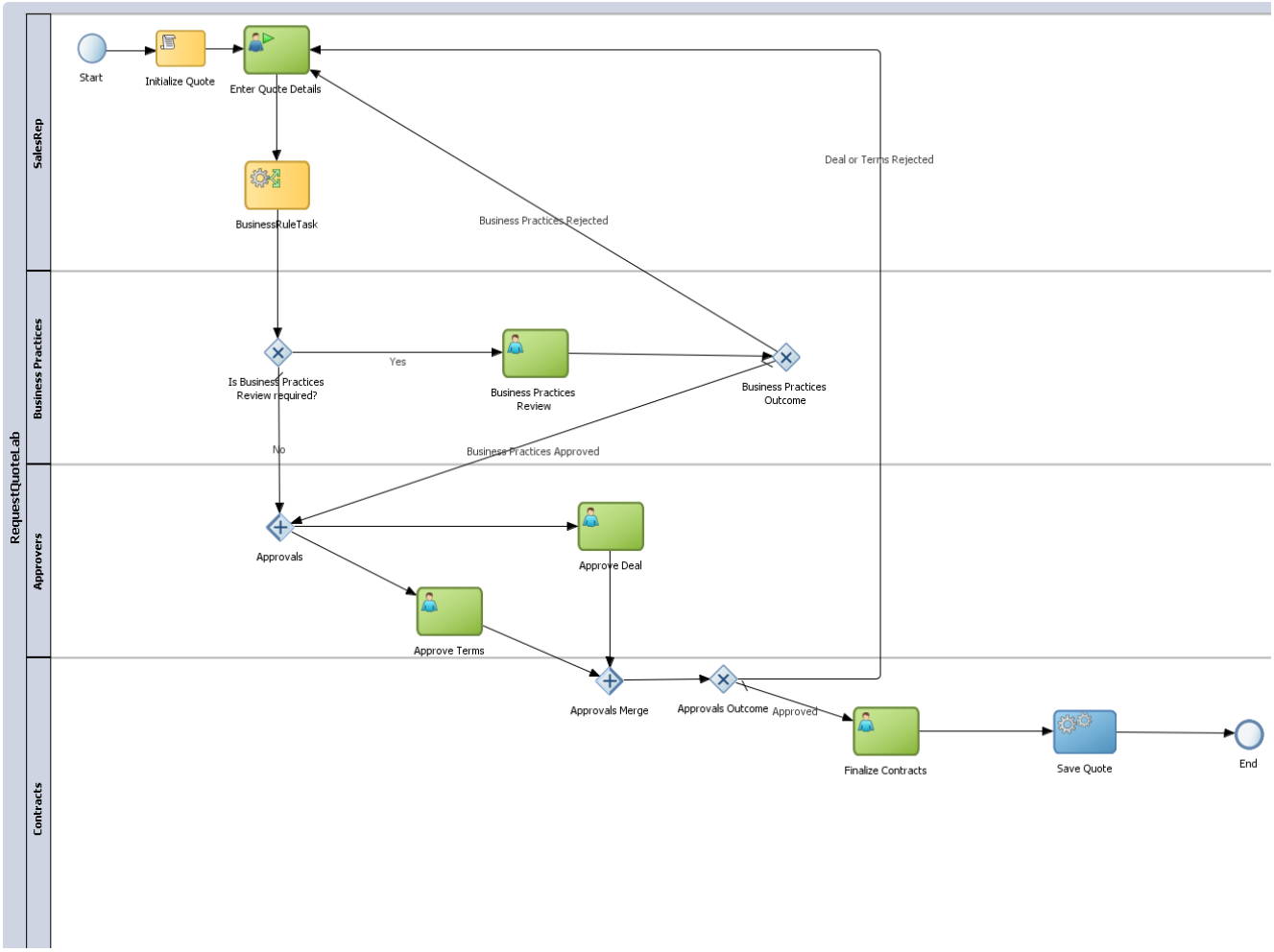


Figure 24 - Sales Quote Process

Diagram showing the process is shown at the Figure 24.

The following chapters describe the experience from the implementation of this business processes with the use of two different platforms.

5.2.2. Implementation in Oracle BPM Suite 11g

Installation

The installation process is divided into several activities. The complete overview of tasks is shown at Figure 25. The main parts are:

- Installation of database and schema configuration
- Installation of Weblogic application server
- Installation of Oracle BPM Suite
- Installation of JDeveloper with SOA and BPM options

The complete installation package has about 7 GB. The installation process of all parts is mostly automated. The Oracle BPM can be installed using scripts but it also has graphical installer. Despite of the graphical installer and high-quality installation manual the whole installation is very complicated and took several hours based on the hardware performance (my experience was 12 hours to complete installation and verification).

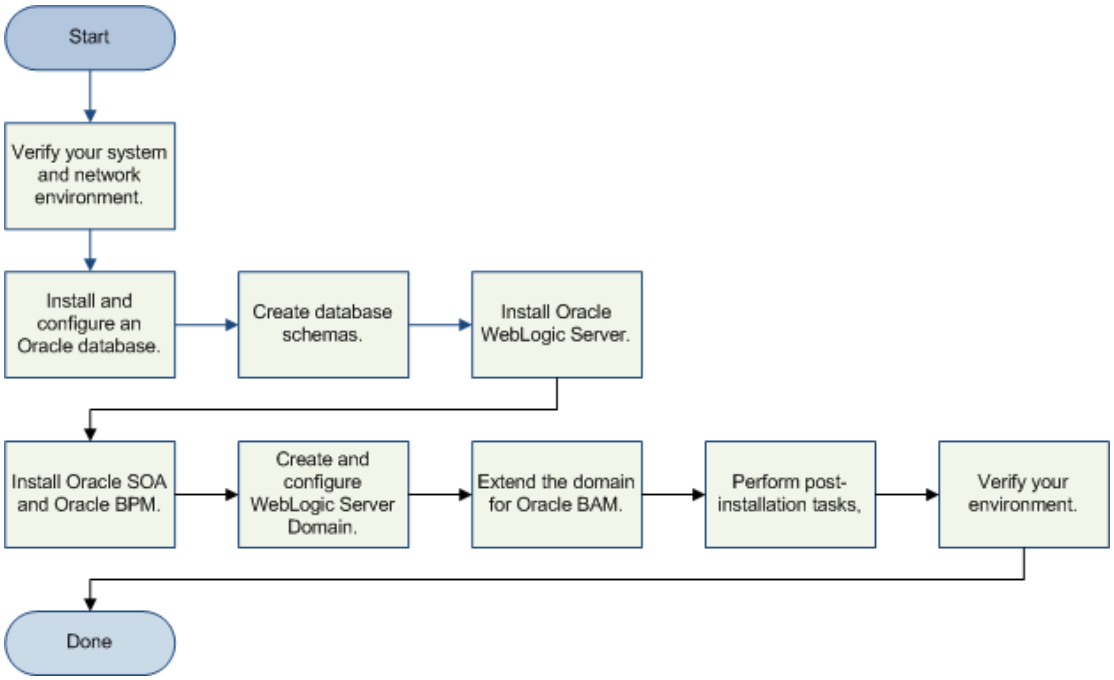


Figure 25 - Schema for installing Oracle BPM suite (see (23))

Besides its own products Oracle supports also other vendors for both application and database servers (for example IBM, Microsoft, etc.), plenty of them are also certified by Oracle. During the installation you can configure several parameters and set many options but you can also postpone the configuration of the BPM suite.

The installation manual is available in several versions: the simple version for standard installation usage Oracle products and extended versions for running Oracle BPM Suite on different vendors' platforms. The complete documentation is available at (23), the installation manual at (56).

To run the BPM suite you have to start the database, WebLogic Administration server and WebLogic BPM Server. You can start them using a script or the installer also creates the shortcuts for running the servers.

Design

Oracle BPM Suite has two tools to design the business processes by business owners. I will describe them separately:

- **Design studio**

  Designing a business process using JDeveloper is very easy and intuitive. The design studio consists of several panels: on the left side there is a project explorer and a structure view, in the middle there is main canvas for editing the processes and on the right side is formed by component pallet and resource panel.

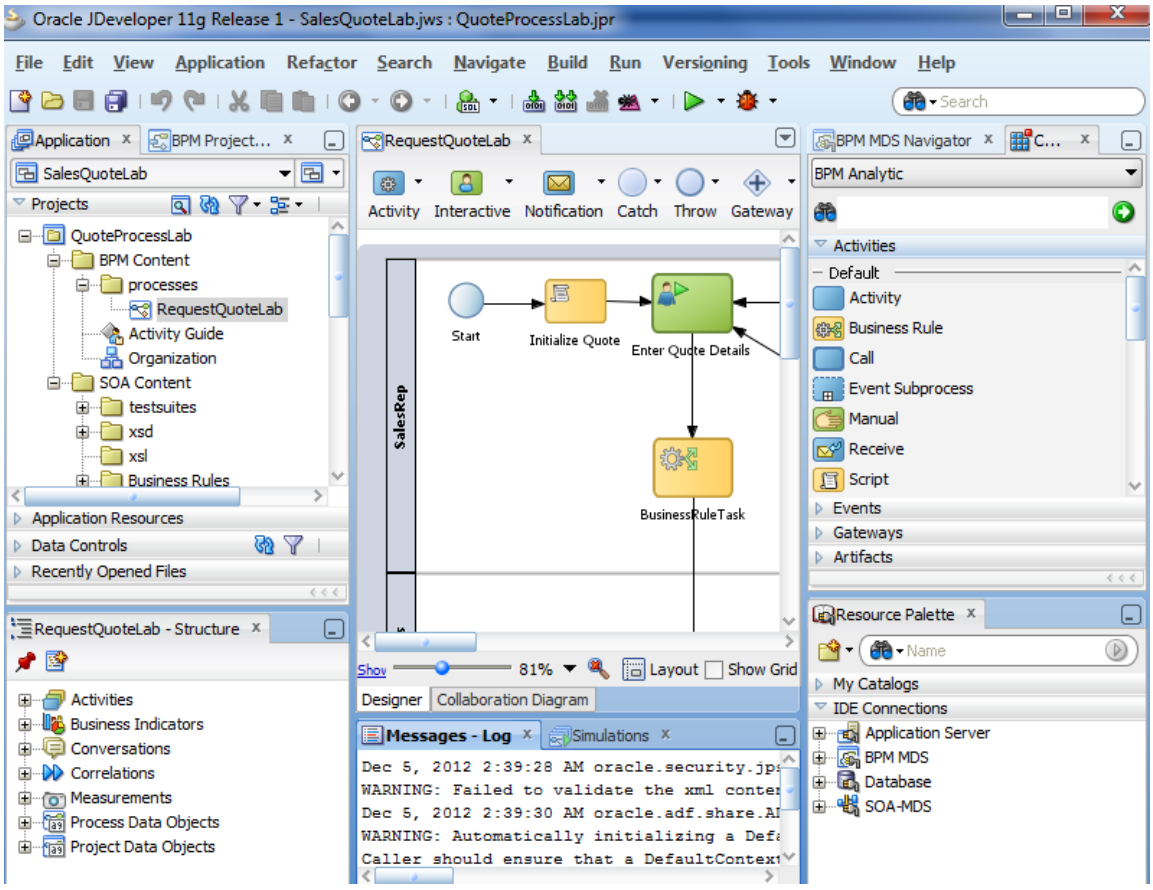


Figure 26 - BPM Studio

The JDeveloper IDE is completely described in documentation (see (23)).

The first of all is creation of the swim lanes and user roles. The BPM Suite also supports the integration with LDAP and other IDM systems to automatically load the organization structure, roles and users. Then we put activities and gateways from component explorer to model a

business processes according to requirements defined in previous chapter.

The whole process (including the swim lanes, activities, human tasks, business rules, decision points, etc.) was designed using drag-and-drop in a one hour. The model contained only BPMN elements without any technical information, data mapping, forms for human tasks, etc.

Design studio also enables real simulation of a business process. You can define the simulation model. For each element of BPMN process it is possible to set duration, resources, costs, queue limit and probability for outgoing flows (it makes sense for gateways). The definition of simulation model is shown at Figure 27. After the model definition the duration of simulation must be set (optionally you should also set the capacity of resources) and then using simulations panel you should run the simulation. The great advantage is the possibility to connect directly to BPM server and use the runtime data with makes the configuration of simulation much easier. Design studio provides several charts and also table outputs of simulations but it should be more comprehensible. It is not as easy to find for example the bottlenecks of business process.

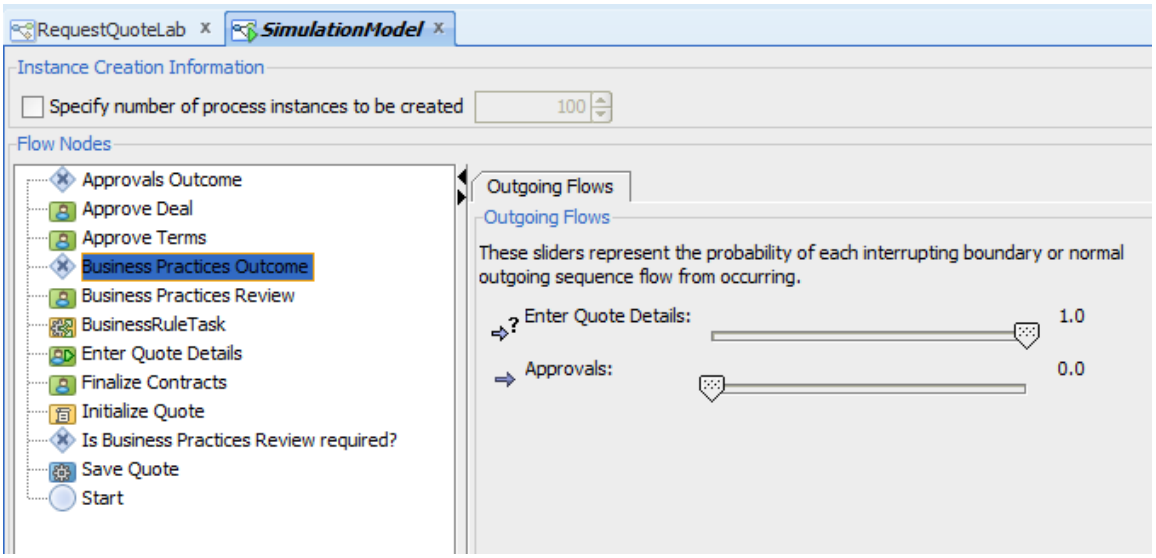


Figure 27 - Simulation model

In BPM Studio you can set whether you want to show compilation errors and warnings. For business users this should be switched off. Even if working with JDeveloper is intuitive and well-documented this is still designed as a tool for developers and for business users it might be very strange to use it.

- **Business Process Composer**

This web-based tool is devoted for business users to view, edit and create business processes. It enables to browse and manage the artifacts stored in MDS repository. The exchange of information between Business Process Composer and Design Studio goes always through the MDS repository.

The viewing of business processes is quite simple. You only find the business process in a repository and the Business process composer shows the exactly same BPMN diagram as JDeveloper (see Figure 28).

The business users can also use this tool as modeling software for business processes. The usability of this web-based tool is comparable with JDeveloper and the process design took approximately the same amount of time. You can also use the drag and drop feature, component pallet, business catalogue and so on. The model or design of business process than has to be stored to MDS repository and complemented about technical details in JDeveloper.

But the configuration of MDS repository for synchronizing the Design studio and Business Process Composer is not so well-documented and it took me about two hours.

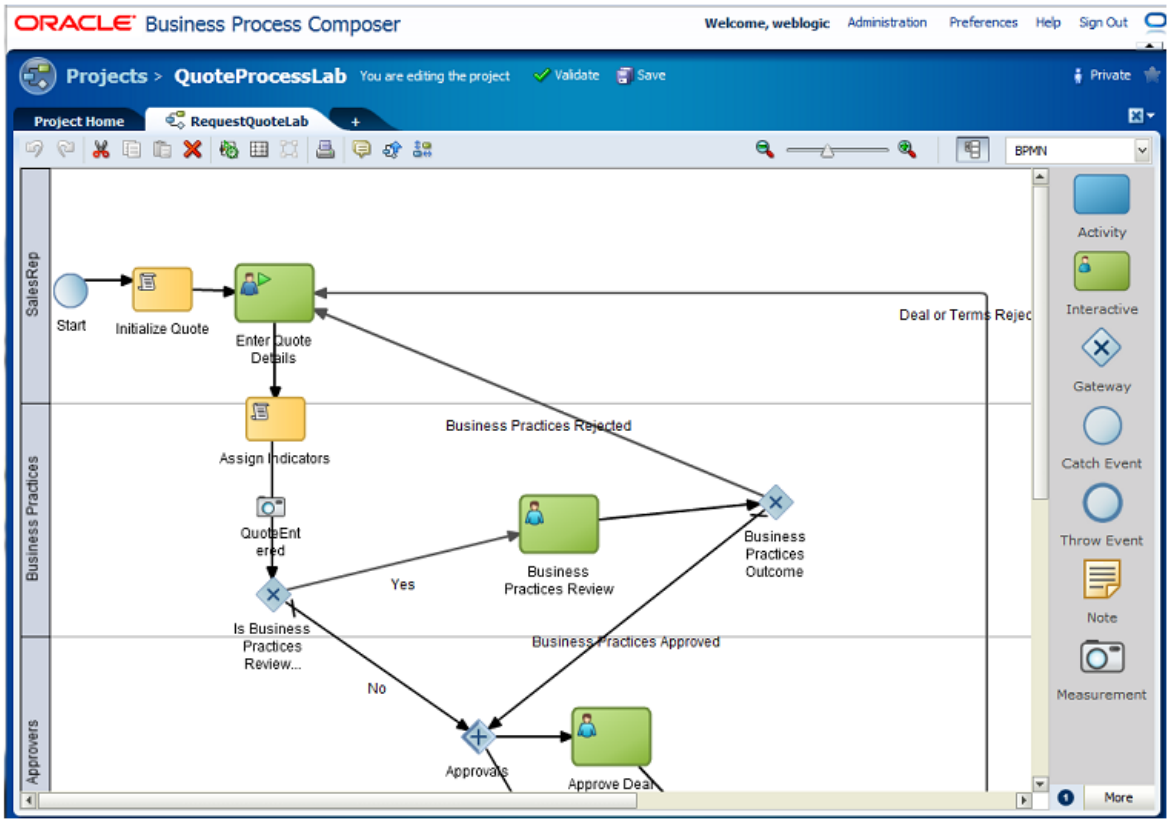


Figure 28 - Business Process Composer

Implementation

Similarly as for design I will describe the development tools separately:

- **BPM Studio**

  When we have high level design/model of business process we can now add the technical information in order to execute it. The process must be enriched with technical information which is described below.

  Business and data objects – the Oracle defines two types of. The business objects represented by the complex data types. The structure of this object is represented by XSD. You can create the scheme directly in JDeveloper or you can use the existing one.

  The data objects represent variables that you can use in the solutions. The type of the object can be simple type from XSD or it can be complex type which is defined using a business object. The data object has three scopes where they can be defined: project, process and activity scope.

  The definition of both business and data objects is very easy (see Figure 29) and it took me several minutes (but I have prepared the XSD).

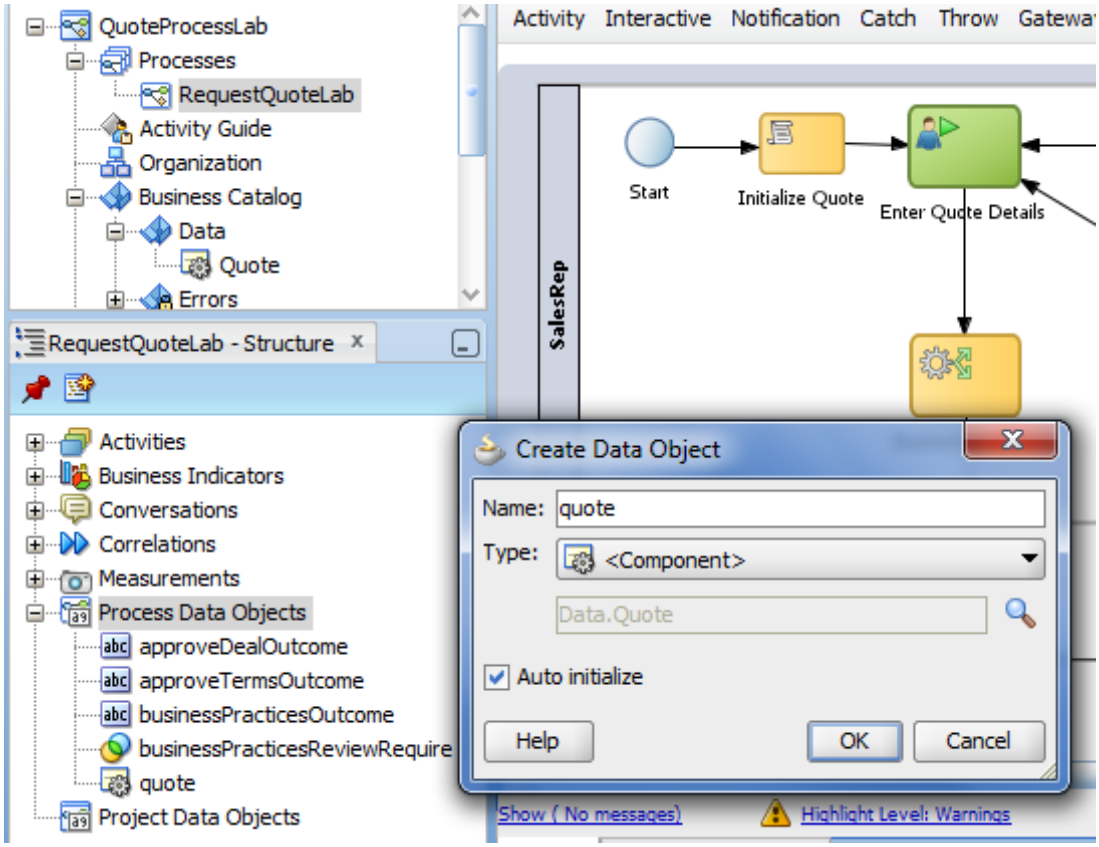


Figure 29 - Business and data objects

Data mappings – the definition of data mapping has change from previous version. The BPEL has the activity "assign" which must be placed before and after the each activity. And the final model of BPEL

doesn't correspond with activity diagram (input from business users) and therefore the BPEL representation was unusable for them.

The BPMN defines the inputs and outputs sets as a part of an activity. The mapping is accessible through properties of each activity and the link Data Associations which open new window for mapping both input and output variables. The mapping is very powerful and enables to drag and drop, XSL transformation. You can also use several functions, add or remove elements in list variables and many others. Example of assigning the data is presented at the Figure 30.

The mapping of variables is really very easy and intuitive and for this process took me about one hour.

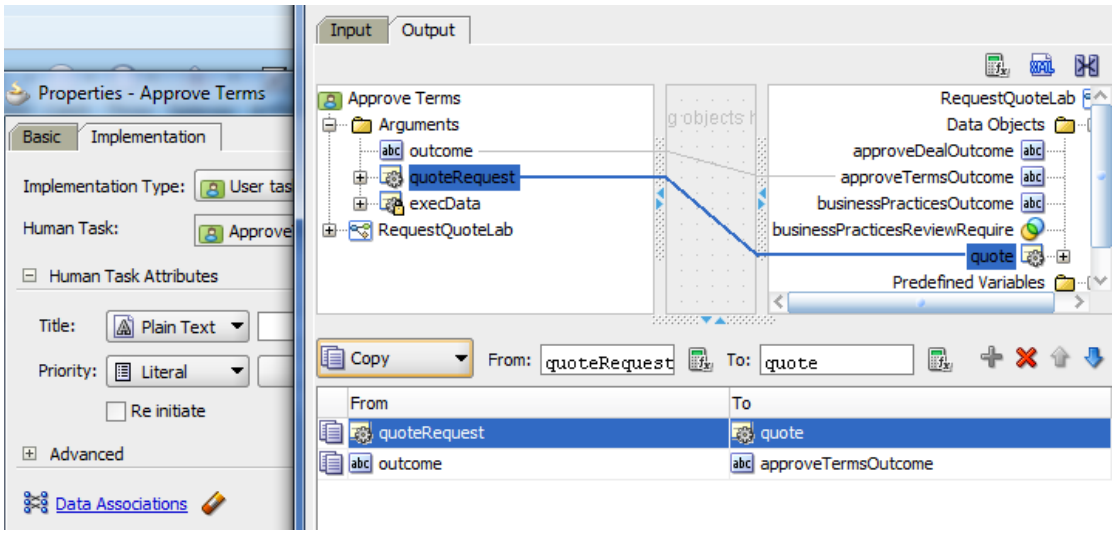


Figure 30 - Data mapping

Conditions – the next step was to add conditions for the gateways. The studio enables you to create very sophisticated conditions using graphical expression builder. In my opinion it is not so logical to add conditions to the flows I expect that this will be part of gateway properties. As I mentioned the express builder is very powerful but for beginners it should be more documented and contain a function for an easy test.

The adding of conditions took me about one hour may be less.

Human tasks – the support for human tasks was also in previous version of Oracle BPM but at the version 11g it was completely reworked and not the suite provides comprehensive solution for tasks management.

In the defined process there are five human tasks (initiation task, three decision tasks and task for finalization). The human tasks for these

activities are created using the property window of activity. On the implementation tab you have to create a new human task. In newly open window you have fill inputs and outputs of parameters and define which one is editable (the graphical environment is shown at Figure 31). After confirmation the human tasks is created. By default the participants correspond do swim lane and the human task owner which should reassign the task is by default the owner of the process.

When the human task is created then you need to create form for editing the task. In business catalogue you open the created human task (for better usability it should be opened directly after human task creation) and create the form. The studio can generate the form automatically based on its type (initiation, decision, etc.) or you can create it manually.

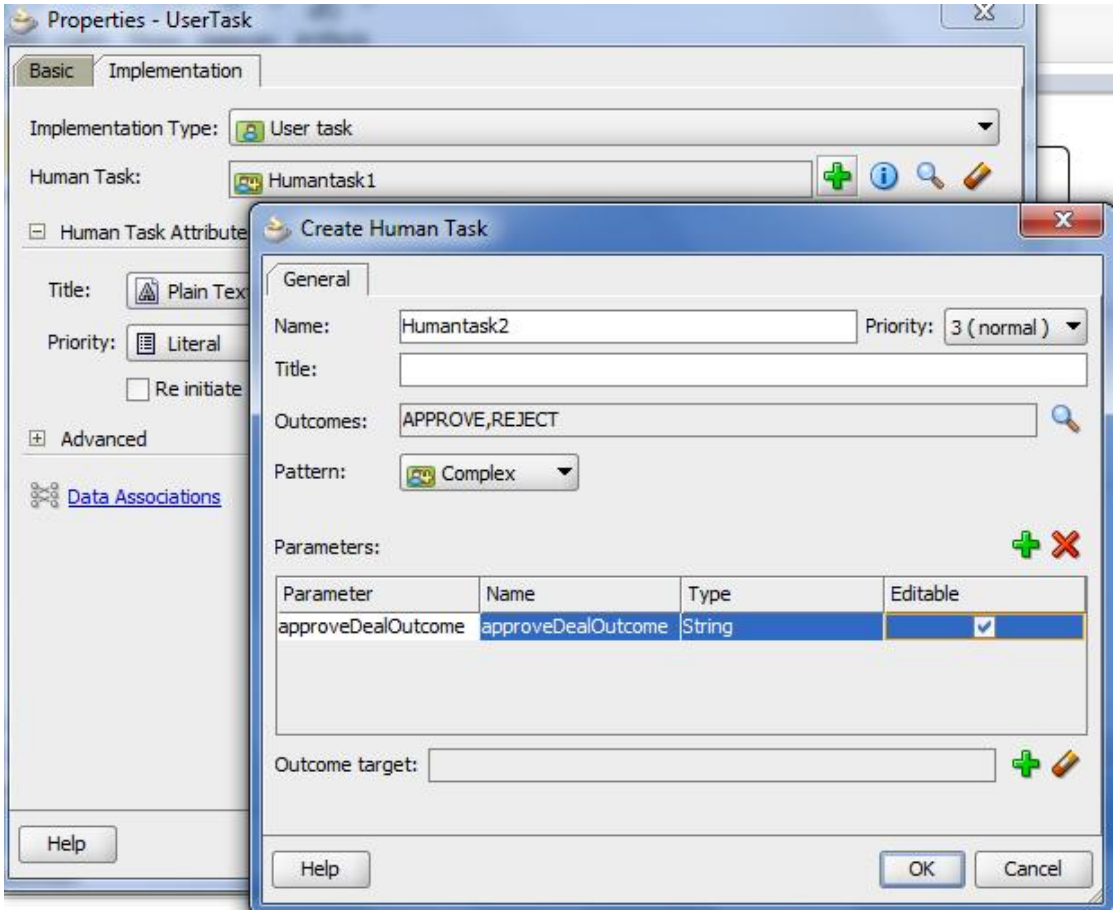


Figure 31 - Human task definition

The studio places all editable elements that you chose in creation of human tasks. The auto-generated form then can be updated to correspond with your ideas. The environment for editing the forms is very rich and it is based on Oracle ADF (see (57)). The illustration of the environment shows the Figure 32.

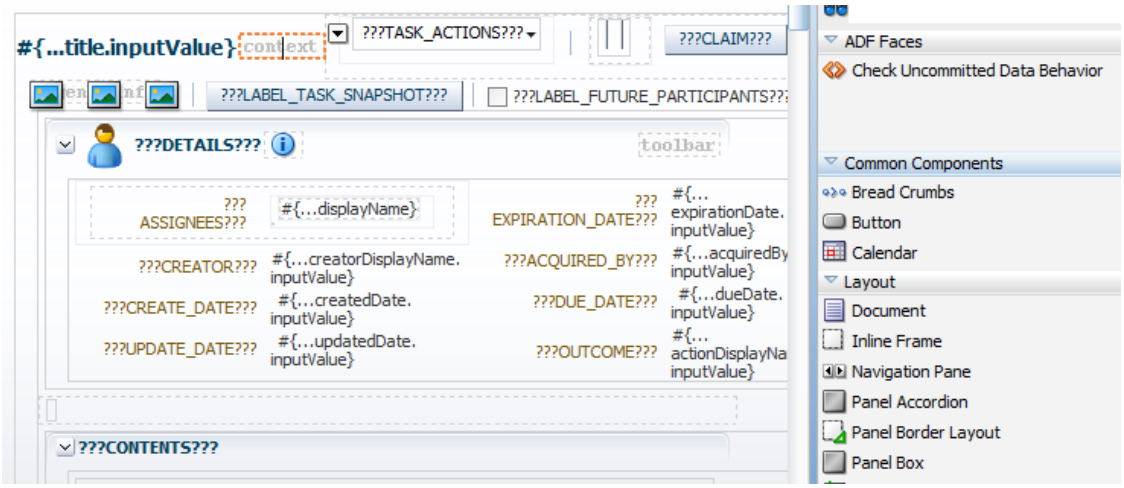


Figure 32 - Environment for editing the forms

Human task editor also enables to define deadline, escalations, access restrictions and reactions on events and many other options.

This part of BPM suites went through extensive change and now covers all necessary requirements on human tasks.

The implementation of the first human tasks took 3 hours, but the others took about 20 minutes.

Business rules - the defined process contains one business rule: if the totalNetRevenue of quote is greater than 100000 the business practices approval is needed otherwise not. The process for creation a business rule is quite similar to creation of a human task. Firstly at business rules activity the new business rules have to be added (you also defined the used variables). Then in business catalogue open the created business rule and define the condition. You can choose between decision table and simple rule (if/then rule). Then the rule designer is opened and you can edit the business rules. Oracle presents that this tool is also suitable for business users but I don't share this opinion because it took me enough time to create a valid business rule. The business rules editor is shown at Figure 33).

The implementation of new business rules took me about two hours.

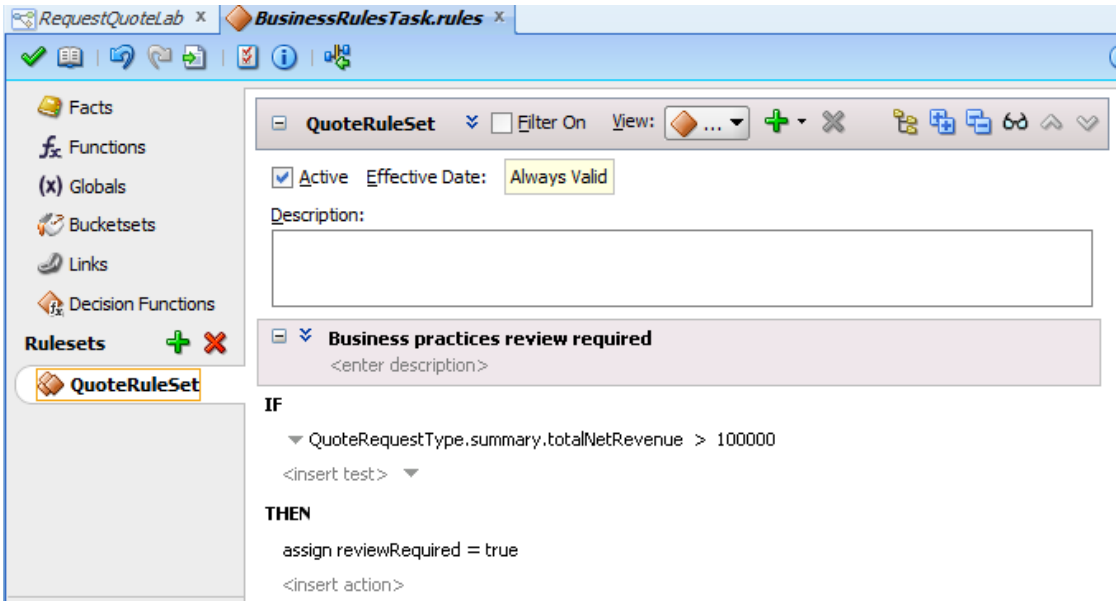


Figure 33 - Business rules editor

Services – the last thing to complete the implementation of business process is to call an external service which stores the approved quote. The process of calling external services is same for technology or application adapters. Therefore the simple file adapter was chosen to store the quote to the file.

The first of all the external service has to be created in the SCA editor (more information about SCA you can find at 3.2). Using drag and drop you can add a new external service. The possible list of external service you can see at the Figure 34**Chyba! Nenalezen zdroj odkazů.**. In several steps you can configure the service, in this case the file name, path primary and the structure of content (the XSD schema is used). Then in process designer you only added the new service task and choose the created external service for storing a file. To complete the calling of an external service you need to map the inputs and outputs.

Implementation of external services is clearly documented and the forms take you through the setting of parameters. It took me less then hour.

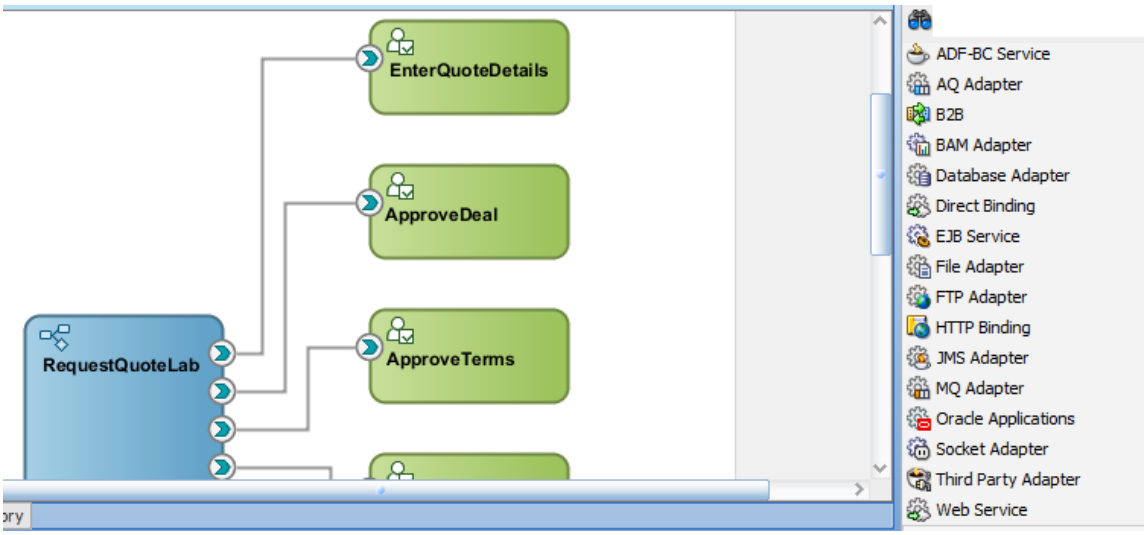


Figure 34 - SCA editor

- **Business Process Composer**

  In my opinion this tool is primary devoted to business users and it is not appropriate for technical implementation of a process because it has worse response time and provides less functions than BPM Studio.

  The editing of an existing process is little bit complicated then designing from a scratch. Before you can start to edit the business processes, it is necessary to create a process template. The template is derived from a business processes in Design studio and it is enriched about permissions. The permissions set the possibilities of changes in Business process composer (you can allow to change only some activities or flows). The template now must be stored to repository to be available for business users to edit. The stored template can now be open in Business Process Composer and the business users can change the permitted parts of a process.

  The business users can also change the processes using Business Process Composer directly on the production environment without technical consultants (of course they can change only the permitted parts). But from my point of view I highly don't recommend this approach because every change should be documented and the business processes are still complex enough to be directly change without IT.

Testing

The support for automated testing and creation of mock object is very poor in comparison to other part of BPM suite. The Oracle BPM provides a functionality that enables to create several configurations for different environments and easily switch the services and endpoints you are calling. For testing

environment you can create manually mocks and updates the process to use this mocks. For this example process the initiation activity has to be change from initiation task to initiation message which enables to start process for example by calling the web service. But this requires many steps and modifications of the original process.

Partly you can test your business processes using simulation but this cannot substitute the automated tests. The Oracle BPM suite should provide a platform for test automation that enables to mock external services and human tasks and run automated tests directly from BPM studio. In my opinion this is one of the most disadvantages of this product.

Deploy

The Design studio already validates the business process during the design and development. The level of reporting about errors and warnings you can set in graphical designer.

The complete processes can be built using the integrated compiler and also deployed to the server. This process is very well documented. The only think what is needed when you first deploy your process is the configuration of server where you wanted to place it. Then this process took together with compilation several minutes.

The studio also contains several views which can help you to find the eventual mistake. But sometimes the error message is too general and the finding of a mistake can take more time.

Execution and monitoring

The Business Process Workspace (is devoted to business users. It enables the monitoring of running instances, task management and also creating custom dashboards (this is closely described in last section – Reporting).

After logging to the Business Process Workspace you should see the screen similar to the Figure 35. The Business Process Workspace contains several tabs for management and monitoring, the last tab: Standard Dashboards contains the predefined reports. The rest of the window is divided into two parts. The panel on the left side contains the deployed process. Running of the process is very easy – the clicking on the name of the process causes the opening of new window where you can fill attributes of the quote and after submitting you will start a new instance of a process (the generated form is shown at the Figure 36).

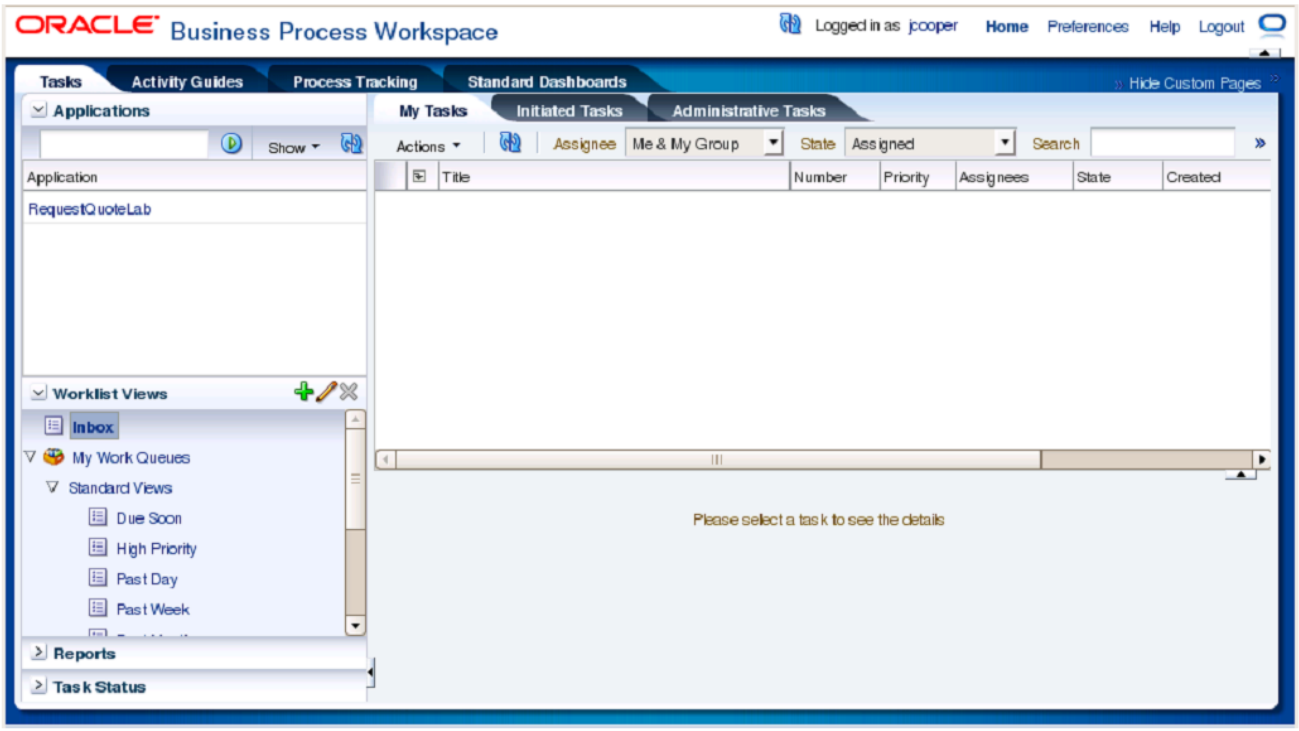


Figure 35 - Business Process Workspace

The right side of the window contains the list of assigned tasks. When the instance of a process is comes back to state for entering the quote details it will be displayed in the list. The same list of action as you define in the development phase is then used in Business Process Workspace to manage the assigned task. The detail of the task also shows the model of the whole process and marks the current path and the last activity for better orientation and tracking.

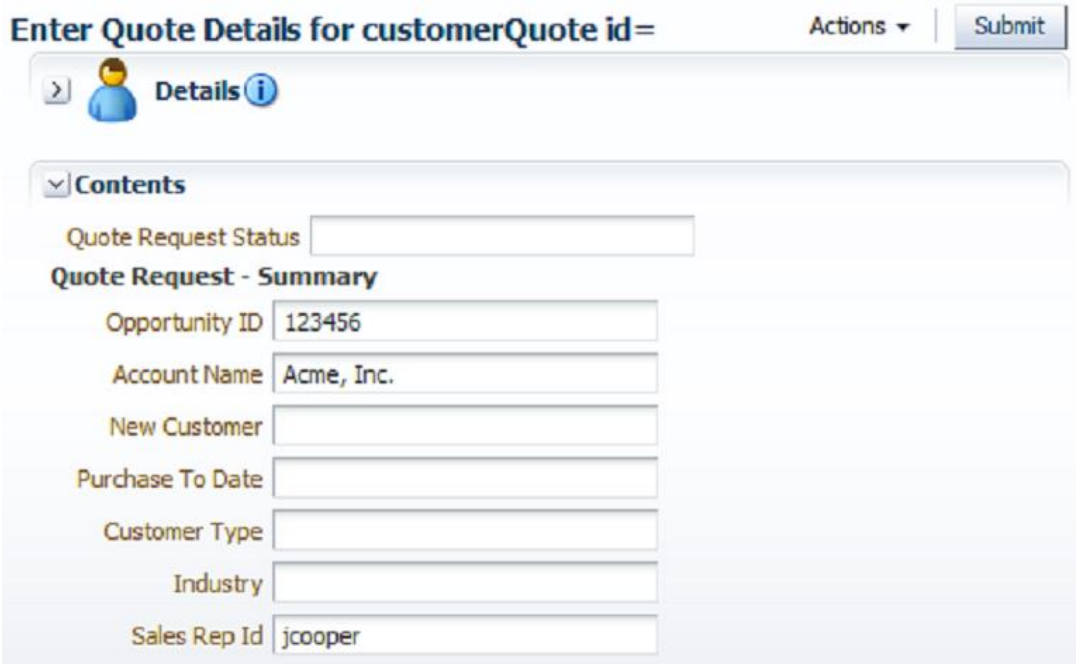


Figure 36 - Generated form for entering the quote details

The orientation in this workspace is very intuitive and for task management I don't need any documentation. The standard localization is English but the workspace also supports the other languages. The configuration is a little bit difficult but it is possible.

The second tool for management of BPM Suite is the Enterprise Manager. This tool is dedicated to technical administrators and it is a standard web-based tool form monitoring and administration of many Oracle products. This tool enables the setting parameters of BPMN engine, integration with other Oracle products. In this console you can also see the list of processes and their instances, but the detail of an instance show much more information than Business process workspace. For example: for every activity you can display the input and output payload, instance failures and technical error messages. The Enterprise Manager also provides several statistical data but this tool doesn't contain features for their processing.

Reporting

The Business Process Workspace contains a set of predefined reports which are available on Standard Dashboards tab.

For creating the custom reports you have to set the business indicators to the process. The dimensions are of the three types: measures (the numerical data that are interesting for analysis), dimensions (represents type or range for numerical data) and counters (for example the number of iterations). The adding of these data is very easy. Using the Design studio you can add all of the business indicators. After adding of business indicators you have to assign data using to them. In BPMN process you choose an element where you want to collect the data and using the properties tab you will set the source for an indicator. Then you have to redeploy the process to start collection of data. The data are stored in process cubes in Oracle database and their accessible through the Business Process Workspace but also using the Oracle Business Activity Monitoring (see (58)) which is not a part of BPM Suite.

The creation of custom report is very simple. In the Business Process Composer you click to: Add Custom Page. The wizard will take you through the whole creation of a new report. Firstly the data source (process) is selected and then you only choose witch business indicators you want to use in your report. The report can be display as a table or graph. The Figure 37 shows the custom generated report which uses the business indicators defined in a process.

The creation of custom report took 3 hours. The documentation very often leads to Oracle BAM which is very comprehensive solution for business analysis but it is not a part of Oracle BPM Suite.

Figure 37 - Custom report

5.2.3. Implementation in jBPM 5.4

Installation

The installation package has about 500 MB (during the installation some other components are downloaded and finally the size is about 1,3 GB) and requires installed Java SDK and Ant. The documentation is available at (59) and (59) but it is much poorer than Oracle documentation. I had to find a workaround because the installation failed (problem with downloading the Eclipse IDE). The installation took about three hours and it is automated with Ant, graphical installer is not available. During the installation process you don't have too many possibilities to configure the BPM solution. jBPM depends on JBoss application server (JBoss AS) and the only think you can configure is database. By default the jBPM uses the H2 database (for more information see (61)). Besides the H2 database, the jBPM supports the same databases as JBoss AS (for example Oracle, DB2, MSSQL,…).

The jBPM is started with ant script which starts H2 database, JBoss AS, Eclipse and the Human Task Service.

Design

The jBPM has also two tools for a business process development. They will be described separately.

- **The Eclipse editor**

The first problem appears with installation of Eclipse. The jBPM 5.4 contains the Eclipse Helios version which is older than Eclipse Indigo. But the Eclipse Indigo version is also mentioned as a supported IDE in installation manual (see (59)). Another problem was that the installation of Eclipse Indigo and the plugin failed. Finally I used the Eclipse Helios version for process modeling and implementation. The environment is faster than JDeveloper but on the first sight it provides fewer functions than JDeveloper. The Eclipse plugin supports only the basic set of BPMN elements (for example it doesn't support swim lanes).

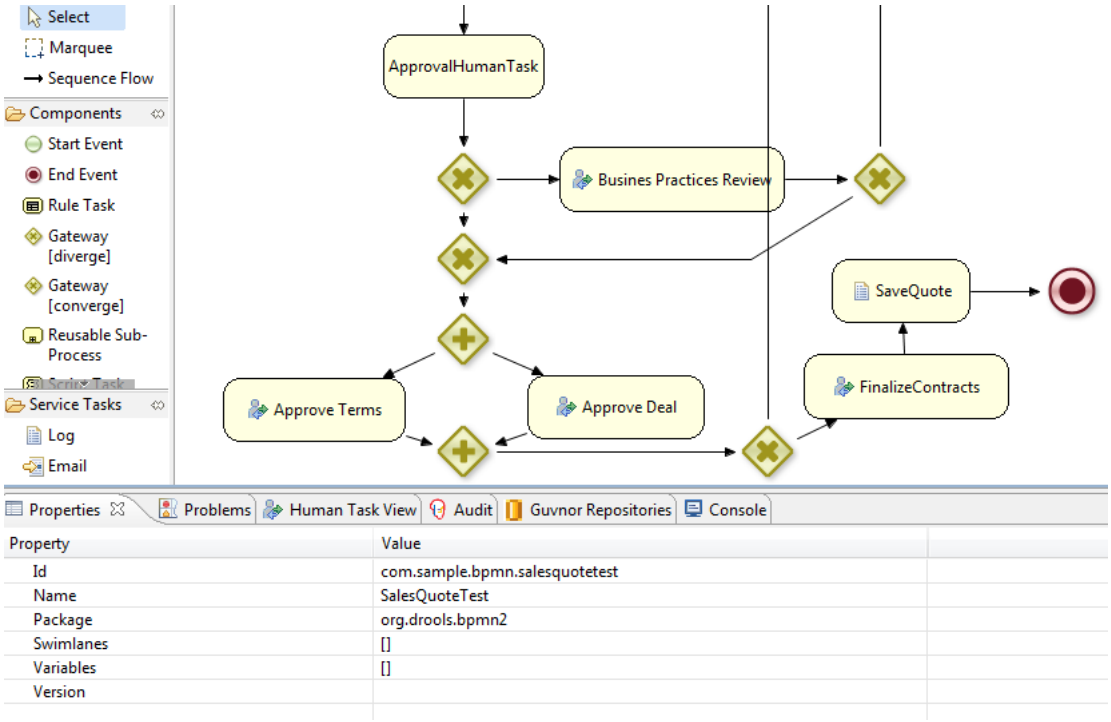


Figure 38 - Eclipse designer

The graphical environment is layered similarly as JDeveloper (see Figure 38). The most important panels are component palette, the modeling space and properties which enable to set the parameters of elements. And there is one more similarity to JDeveloper and it is that this tool is primary design for technical consultants.

Unlike the JDeveloper the Eclipse plugin doesn't provide any possibilities to simulate the designed business process.

The designing of business process model took about one hour.

- **Web-based designer (a part of Guvnor Repository)**

  At the documentation there is at many places mentioned web-based designer as a separate tool, but it is a part of Guvnor Repository. During the installation Eclipse IDE is integrated to the Guvnor Repository and

thus it needn't to be configured. The synchronization between repository and the both web and desktop designer works perfectly. The problem is that the models are mutually importable (the missing swim lanes in Eclipse designer are one of the problems why you cannot load the model from Eclipse to web-based designer and vice versa).

But the creation of business process model in web-based designer from a scratch is very intuitive. The layout is similar to Eclipse IDE but it additionally includes the action panel for easier work with business model (see Figure 39 - Web based designer). Unlike the desktop version, the web-based designer supports more BPMN 2.0 elements and their configuration is easier because it is performed directly in modeling panel.

The disadvantage of jBPM web-based editor is the number of bugs. The most bugs have appeared during the saving and validating the model. When you opened the model it happened that all activities were gone. The only think that you could see was the title of the process.

Unlike the Eclipse editor the Guvnor Repository offers the simulation of the processes. You can set up parameters at business activities and then run the simulation from the editor. The report shows also graphical representation of the results.

The creation of BPMN model in Guvnor Repository took me about one hour although I had to start twice from the beginning (because of error in saving the process).

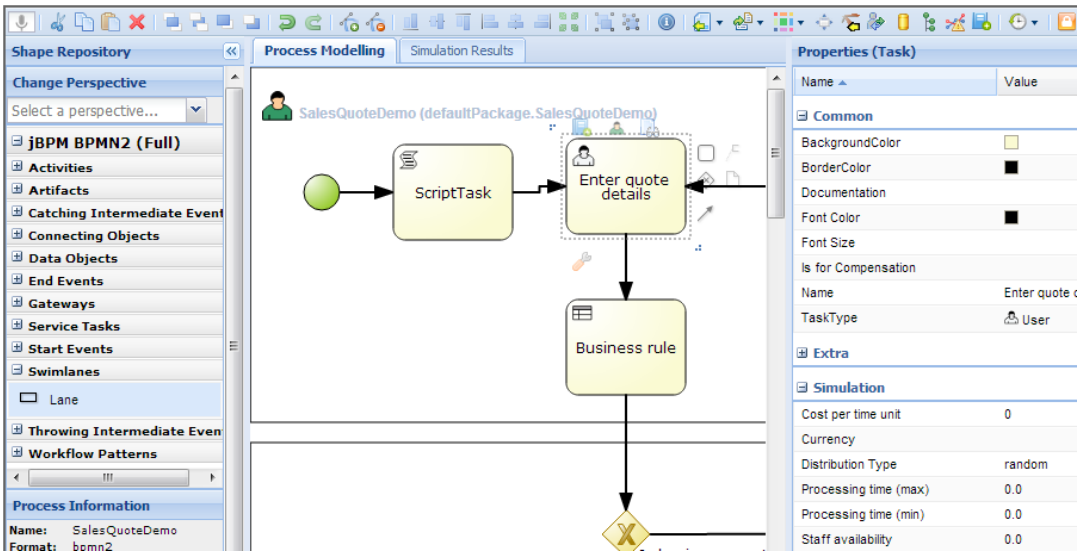


Figure 39 - Web based designer

Implementation

There is big difference between the Oracle and Red Hat approach to implementation of business process. The Oracle's JDeveloper and Business Process Composer are designed to enable the implementation with zero code (of course this is not true, but the number of scripts or manual expressions is minimal). On the other hand the Red Hat's jBPM is closely associated with Java and related frameworks. The jBPM provides very comprehensive set of Java API. The API covers all aspects of BPM – you can manage processes, rules, human tasks, run instances, modify variables, etc. The API is documented using Javadoc and you can find it at (62).

Example of calling business process from Java class:

```
public class ProcessMain {

	public static final void main(String[] args) throws Exception
	{
		// load up the knowledge base
		KnowledgeBase kbase = readKnowledgeBase();
		StatefulKnowledgeSession ksession =
		kbase.newStatefulKnowledgeSession();
		// start a new process instance
		ksession.startProcess("com.sample.bpmn.salesquotetest");
	}

	private static KnowledgeBase readKnowledgeBase() throws
	Exception {
		KnowledgeBuilder kbuilder =
		KnowledgeBuilderFactory.newKnowledgeBuilder();

		kbuilder.add(ResourceFactory.
		newClassPathResource("SalesQuoteTest.bpmn"),
		ResourceType.BPMN2);
		return kbuilder.newKnowledgeBase();
	}

}
```

- **Eclipse designer**

  In a similar way as for JDeveloper I will describe the implementation part of defined process but now for Eclipse designer. The JDeveloper is based on zero-code conception and that's why it has windows and wizards for definition of data objects, mapping, business rules, human tasks, calling services etc. The Eclipse plugin doesn't have these possibilities and it supplements them using the Script task and Java code. The advantage is that Java is very powerful language but the script tasks make the model less clear.

  Variables and data mapping

  The process variables are defined as a project property. As data types of variables are used Java data types (Oracle uses XSD). The mapping of

variables and other data object can be done using the expression language mvel (see (63)) or Java. Some BPMN elements also have properties for data mapping (for example human tasks). For these you can map the variables and the parameters of an element using a simple wizard.

Definitions of variables and data types take about one hour.

Conditions

The conditions are mostly used for gateways which has three nodes: AND (parallel flow), XOR (exactly one outgoing connection will be chosen – therefore the connections have priorities), OR (every outgoing connection which pass the condition will be chosen). The definition of condition for gateway is shown at Figure 40.

The orientation in constraints and gateways should be simplified. The setting of constrains took me about 1 hour.

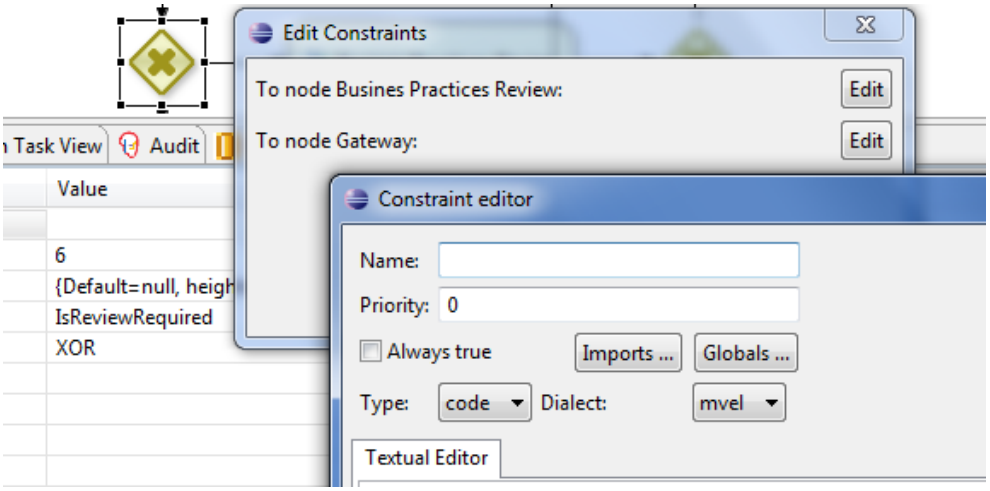


Figure 40- Conditions

Human tasks

The creation of human task is a little bit complicated. The creation of a human task consists of several steps. The first is the definition of task node and setting its attributes. You can assign users and groups, input and output parameters, escalation, notifications etc. The very important attribute is Content, which stores the form data and the TaskName. The second step is creation of forms. jBPM uses ftl forms which are basically HMTL pages. The forms can be generated automatically. The third step is deployment of forms to the Guvnor Repository. The tfl files have to be packed together with screenshot of BPMN diagram into jar file.

The example of generated form definition:

```
<html>
<body>
<h2>Enter Quote Details</h2>
<hr>
<#if task.descriptions[0]??>
Description: ${task.descriptions[0].text}<BR/>
</#if>
Content: ${Content}<BR/>
<form action="complete" method="POST" enctype="multipart/form-
data">
accountName: <input type="text" name="accountName" /><BR/>
effectiveDiscount: <input type="text" name="effectiveDiscount"
/><BR/>
opportunityID: <input type="text" name="opportunityID" /><BR/>
totalNetRevenue: <input type="text" name="totalNetRevenue"
/><BR/>

<BR/>
<input type="submit" name="outcome" value="Complete"/>
</form>
</body>
</html>
```

The process of task definition is very complicated and the original documentation is incomplete. The possibilities of jBPM are far beyond the Oracle BPM.

The implementation took about five hours to run the task. The others take about 30 minutes.

Business rules

The integration of business rules and business processes is very easy. The process consists of two steps. The rule task has to be inserted into business process model. The business rule description is stored in rule resource (the .drl file).

The example of business rule:

```
//created on: 7.12.2012
package rulepackage
import org.drools.runtime.process.WorkflowProcessInstance;
//list any import classes here.


rule "Rule for Business Practices Review"
    ruleflow-group "bpReview"

    when

            $process: WorkflowProcessInstance()

    then
      boolean required =
(Integer)($process.getVariable("totalNetRevenue")) > 100000;
```

```
$process.setVariable("businessPracticesReviewRequired",
required);
end
```

Besides the rules the decision tables and other features of Drools project can be used in business project. But the rule editor from JDeveloper is more user-friendly than Eclipse based editor for Drools.

The creation of the business rule took two hours.

Services

The jBPM doesn't offer technology adapters that can enable easier integration with other application. When you want to call a web service or safe a file you can use BPMN service task and implement its business logic in Java.

- **Guvnor Repository**

  This tool is very intuitive and except for bugs it provides quick environment for modeling business processes using BPMN 2.0. It is business oriented tool and therefore I will not describe the implementation of a process using this tool.

Testing

In this point absolutely exceeds the jBPM over Oracle BPM. jBPM supports both unit testing and creating the mockups. The jBPM API contains the special jBPM JUnit test classes. Using the helper method you can create knowledge base to interact with jBPM engine.

The jBPM API also provides method for mocking the external services and human tasks. You can set their outputs and run automated tests.

Another part of jBPM API is a set of asserts which enriches the standard set and allows making tests on: triggered node and their groups, running or aborting instances, etc.

The testing platform of jBPM is very complex and enables the full-valued testing.

Deploy

The Eclipse validates the business process after saving of any resource. The errors and warnings are shown at standard view: Problem and Console. A part of Eclipse IDE is also a debugger which you can use to find the problem. Beyond, jBPM plugin enhances the debugger about graphical interface to the BPMN project and you can see in debugger mode at which activity the process stopped.

The deployment process is very easy. You only need to store the BPMN processes, eventually also the other resources, to Guvnor Repository. Using Eclipse plugin you can do it directly from context menu or you can use export and import.

Execution and monitoring and reporting

I combined the last section together because jBPM Web-based management console provides only the basic set of functions.

You can use it to show the deployed processes and their running instances. The instance detail contains the graphical preview of BPMN model and shows the last active activity.

The management console also contains the simple task management. The logged users can see assigned task and managed them. They also can start a new instance of a process.

The reporting part contains only few predefined reports. The full integration with jBPM BAM is still not fully supported and this product is not a part of jBPM.

The GUI is not user-friendly and from time to time it fails down or prints a stack trace on the screen (the Figure 41 shows the management console).

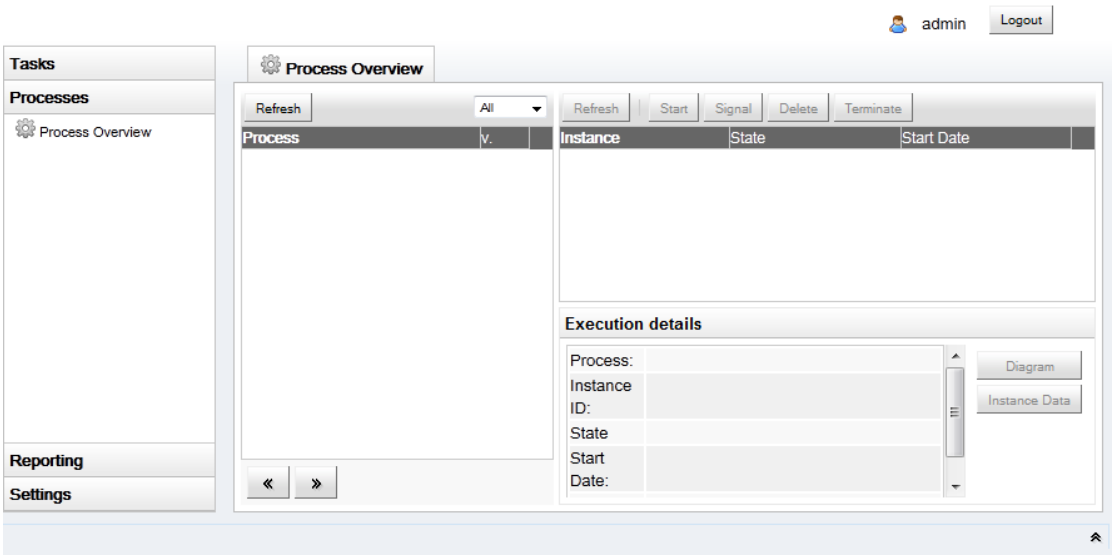


Figure 41 - Management and monitoring console

5.3. Comparison Results

This chapter contains the summarized results from the comparison and the implementation of business process.

Each category of criteria is placed in an own table. The last table contains the evaluation from the practical implementation and it is more oriented on the used components.

The count of asterisks (*****) represents the quality of the BPM suite compared to the other one (the more asterisks the better quality, five is maximum).

The number of asterisks for each criterion is mentioned on the basis of executed comparison and my personal experience.

Organization criteria

Criterion	Oracle BPM Suite 11g	jBPM 5.4
Company standard	Not applicable	
Total cost of ownership	Not applicable	
Representatives on the Czech market	*****	*****
Support	*****	****
References	***	***
Vendor lock-in	***	****

Table 7 - Comparison results - organization criteria

Individual criteria

Criterion	Oracle BPM Suite 11g	jBPM 5.4
Localization	***	**
Learnability	****	**
Efficiency	****	****
User friendliness	****	***
Device support	*	*

Table 8 - Comparison results - individual criteria

Information criteria

Criterion	Oracle BPM Suite 11g	jBPM 5.4
Documentation	****	**
Courses and certification	*****	***
Quality and completeness	****	**
Communities	***	***
Accessibility	****	*****

Table 9 - Comparison results - information criteria

Technology criteria

Criterion	Oracle BPM Suite 11g	jBPM 5.4
Product stability	***	***
Extensibility	****	**
Interoperability	*****	***
Security support	****	**
Identity Management Integration	****	*
Performance	Performance of products wasn't measured.	

Clustering and High Availability	****	**
Standards support	***	****
Connectors/Adapters	****	*

Table 10 - Comparison results - technology criteria

Business process implementation

Criterion	Oracle BPM Suite 11g	jBPM 5.4
Installation	****	**
Design	*****	**
Implementation	****	***
Testing	**	****
Deploy	****	****
Execution and monitoring	****	**
Reporting	****	**

Table 11 - Comparison results – business process implementation

Conclusion

The business process management is a set of principles, methods and tools used to identify, design, execute, monitor and control both automated and non-automated business processes. In this work I tried to explain the complexity of this domain including its history. During the BPM history, which started at the beginning of 20th century, tens of standards had appeared. The first part of this thesis also describes the most important ones used in BPM.

In the following chapter I described the integration patterns, service oriented architecture and the whole concept of the enterprise architecture. Then I clarified the relations between these patterns, architectures, methodologies and BPM. Finally, I describe the methodology for the application of BPM into enterprise architecture. It covers all aspects of the BPM implementation from the planning phase to the implementation of BPM governance.

The first part of the thesis is devoted to the BPM. The second part presents the tools which support the BPM. These tools are typically packed by vendors and designated as BPM suites. The standard list of components or tools, which the BPM suite should consist of, doesn't exist. Based on my personal experience and Gartner's definition of BPM suite, I defined and described the key components. From the commercial vendors and open source communities the representatives of BPM suites were chosen (Oracle BPM Suite 11g and jBPM 5.4). They were closely described and their features were mapped to the defined key components.

The final part of the thesis is the comparison of the selected BPM suites. To be able to compare these tools, I defined the list of criteria which can also help business and IT specialist with finding the convenient BPM suite. These criteria were divided into four groups: organization, individual, information and technology. To provide the complete comparison of BPM suites, the exemplary business process was implemented using both of them. The comparison results then contain evaluation from both the mentioned criteria and the practical use.

The result of a comparison shows that there is a big difference in quality between the commercial product Oracle BPM Suite 11g and an open source project jBPM 5.4. The difference is in the detail and completeness of individual BPM components, also a big contrast was in the documentation, integration with other parts of enterprise architecture and usability. For now the open source solution jBPM is not ready for implementation at the enterprise architecture.

The Oracle solution I can recommend even with the regard to its disadvantages. But before the final resolution you should remember the methodology and also

take into account the price for licenses and maintenance which standardly makes about 20% of product license per year.

There is also one more interesting output from the comparison. The Oracle is built over SOA and XSD to be platform-independent, while the jBpm uses the Java and its frameworks. From this point of view the jBpm is much more convenient as a workflow of rule engine for the individual Java application.

Contribution

The main goal of the thesis was to help business and IT specialist with implementation of business process management. The business and IT specialist can now use the following outputs of the thesis to decide easily which way they will implement a BPM suite:

- Overview of useful standards
- Road map for application of the BPM suite in enterprise architecture
- Key components of BPM suite mapped on both commercial and open source solution
- Comparison criteria or checklist for the right selection of BPM suite
- Detailed comparison results of commercial and open source products

These outputs should help them with better orientation in BPM standards, technologies and products.

References

1. **Gillot, Jean-noel.** *The Complete Guide to Business Process Management.* 2008 : Lulu.com.

2. **Josuttis, Nicolai M.** *SOA in practise.* s.l. : O'Reilly Medi, 2007.

3. Becoming a BPM Professional. [Online] http://www.abpmp.org/displaycommon.cfm?an=1&subarticlenbr=216.

4. **Harmon, Paul.** *Business Process Change.* s.l. : Morgan Kaufmann, 2007.

5. **Verma, Naresh.** *Business Process Management: Profiting from Process.* s.l. : Global India Publications, 2009.

6. **Howard Smith, Peter Fingar.** *Business Process Management: The Third Wave.* s.l. : Meghan-Kiffer Press, 2006.

7. **John Jeston, Johan Nelis.** *Business process management: practical guidelines to successful implementations.* s.l. : Routledge, 2012.

8. **Janelle B. Hill, Jim Sinur.** *Magic Quadrant for Business Process Management Suites, 2006.* s.l. : Gartner, 2006.

9. Introduction to Oracle BPM Studio. [Online] http://docs.oracle.com/cd/E14571_01/doc.1111/e15176/intro_bpm_studio.htm.

10. Human Interaction Sample. [Online] http://publib.boulder.ibm.com/bpcsamp/gettingStarted/humanInteraction/buildItYourself.html.

11. What's new in IBM Business Monitor V8. [Online] http://www.ibm.com/developerworks/bpm/bpmjournal/1209_alcorn/1209_alcorn.html.

12. **Chang, James F.** *Business Process Management Systems – Strategy and Implementation.* s.l. : CRC PressINC, 2006 .

13. **Matjaž B. Jurič, Ramesh Loganathan, Dr P G Sarang.** *SOA Approach to Integration.* s.l. : Packt Publishing, 2007.

14. BPMN Information Home. *BPMN.* [Online] http://www.bpmn.org/.

15. **Allweyer, Thomas.** *Bpmn 2.0.* s.l. : Books on Demand, 2010 .

16. Business Process Model and Notation (BPMN). [Online] 2009. http://www.omg.org/spec/BPMN/1.2/PDF.

17. Business Process Model and Notation (BPMN) 2.0. [Online] 2011. http://www.omg.org/spec/BPMN/2.0/PDF.

18. BPMN Graphical Elements. [Online] http://www.omg.org/bpmn/Samples/Elements/Core_BPMN_Elements.htm.

19. **Havey, Michael.** *Essential Business Process Modeling.* s.l. : O'Reilly Media, 2005.

20. XPDL. [Online] http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-02-24).pdf.

21. **Willis, Tony.** *Bpel 100 Success Secrets - Business Process Execution Language for Web Services.* s.l. : Lulu.com, 2008 .

22. WS BPEL 2.0. [Online] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

23. **McGovern, James.** *A Practical Guide to Enterprise Architecture.* s.l. : Prentice Hall Professional, 2004.

24. **Yefim V. Natis, William Clark, Ray Valdes.** *Context Delivery Architecture, Putting SOA in Context.* s.l. : Gartner, 2007.

25. **Gregor Hohpe, Bobby Woolf.** *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* s.l. : Addison-Wesley, 2012 .

26. SCA. [Online] http://www.oasis-opencsa.org/sca.

27. **Dirk Krafzig, Karl Banke, Dirk Slama.** *Enterprise SOA: Service-Oriented Architecture Best Practices.* s.l. : Prentice Hall Professional, 2005 .

28. **Erl, Thomas.** *Service-Oriented Architecture: Concepts, Technology, and Design.* s.l. : Prentice Hall, 2005.

29. —. *SOA Principles of Service Design.* s.l. : Prentice Hall, 2008.

30. —. *SOA Design Patterns.* s.l. : Prentice Hall, 2009.

31. **Roshen, Waseem.** *SOA-Based Enterprise Integration: A Step-by-Step Guide to Services-Based Application Integration.* s.l. : McGraw Hill Professional, 2009.

32. **Bieberstein, Norbert.** *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap.* s.l. : FT Press, 2006.

33. **Chappell, David.** *Enterprise Service Bus.* s.l. : O'Reilly Media, 2004 .

34. **David Besemer, Paul Butterworth, Luc Clément, Jim Green, Hemant Ramachandra, Jeff Schneider,Hub Vandervoort.** *SOA Getting It Right.* s.l. : Westminster Promotions, 2008.

35. **Lankhorst, Marc.** *Enterprise Architecture at Work: Modelling, Communication and Analysis.* s.l. : Springer, 2005.

36. Service-Oriented Architecture and Enterprise Architecture. [Online] http://www.ibm.com/developerworks/webservices/library/ws-soa-enterprise2/.

37. A Better Path to Enterprise Architectures. [Online] http://msdn.microsoft.com/en-us/library/aa479371.aspx#sessionsfinal100_topic3.

38. A Comparison of the Top Four Enterprise-Architecture Methodologies. [Online] http://msdn.microsoft.com/en-us/library/bb466232.aspx.

39. *Enterprise Architecture: Creating Value by Informed Governance.*

40. **McGovern, James, et al., et al.** *Practical Guide to Enterprise Architecture, A.*

41. Zachman International. [Online] http://zachman.com/about-the-zachman-framework.

42. The Open GROUP. [Online] http://www3.opengroup.org/.

43. **Standard, Open Group.** TOGAF Version 9.1. [Online] http://www.opengroup.org/togaf/.

44. **Miers, Derek.** BPM Best Practices. [Online] http://www.omg.org/news/meetings/ThinkTank/past-events/2006/presentations/05-WS2-2_Miers.pdf.

45. **Jim Sinunr, Janelle B. Hill.** *Gartner Magic Quadrant for Business Process Management Suites, 2010.* s.l. : Gartner, 2010.

46. **Fischer, Layna.** *Workflow Handbook 2007: Methods, Concepts, Case Studies and Standards.* s.l. : Future Strategies Inc., 2007 .

47. Business Rules Group. [Online] 2012. http://www.businessrulesgroup.org/defnbrg.shtml.

48. **Elisa Bertino, Lorenzo Martino, Federica Paci, Anna Squicciarini.** *Security for Web Services and Service-Oriented Architectures.* s.l. : Springer, 2010.

49. List of top open source BPM / workflow solution. [Online] http://www.softwareforenterprise.us/2009/03/13/list-of-top-open-source-bpm-workflow-solution/.

50. Configuring High Availability for Oracle SOA Suite. [Online] http://docs.oracle.com/cd/E27559_01/doc.1112/e28391/ha_soa.htm.

51. jBPM Overview. [Online] http://docs.jboss.org/jbpm/v5.4/userguide/ch.overview.html#d0e66.

52. Software Evaluation: Tutorial-based Assessment. [Online] http://software.ac.uk/sites/default/files/SSI-SoftwareEvaluationTutorial.pdf.

53. **Palmius, Joel.** Criteria for measuring and comparing information systems. [Online] http://www.palmius.com/joel/text/IRIS-30-final.pdf.

54. **Heidi Buelow, Manoj Das, Manas Deb.** *Getting Started With Oracle Bpm Suite 11gr1: A Hands-on Tutorial.* s.l. : Packt Publishing, 2010.

55. Oracle Fusion Middleware documentaion. [Online] http://docs.oracle.com/cd/E23943_01/soa.htm.

56. Quick Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite. [Online] http://docs.oracle.com/cd/E23943_01/install.1111/e14318/qisoa.htm.

57. Oracle Application Development Framework. [Online] http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html.

58. Business Activity Monitoring. [Online] http://www.oracle.com/technetwork/middleware/bam/overview/index.html.

59. jBPM User Guide. [Online] http://docs.jboss.org/jbpm/v5.4/userguide/.

60. jBPM community. [Online] https://community.jboss.org/wiki/jBPM.

61. H2 database. [Online] http://www.h2database.com/html/main.html.

62. jBPM API Documentation. [Online] http://docs.jboss.org/jbpm/v5.4/javadocs/.

63. MVEL. [Online] http://mvel.codehaus.org/.

64. *XPDL.* [Online] http://www.wfmc.org/xpdl.html.

65. *WfMC.* [Online] http://www.wfmc.org/about-us.html.

66. *OMG.* [Online] http://www.omg.org/gettingstarted/gettingstartedindex.htm.

67. *OASIS.* [Online] http://www.oasis-open.org/org.

68. WS-BPEL Extension for People. *BPEL4People.* [Online] 2005 . http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/.

69. Business Process Management Initiative. *BPMI.* [Online] 2008. http://www.bpmi.org/.

70. History of BPEL. *BPEL History.* [Online] 2007. http://bpel.xml.org/history.

71. BPMN. *BPMN history.* [Online] http://www.omg.org/spec/BPMN/.

72. Open Source Initiative. [Online] http://www.opensource.org/licenses/alphabetical.

73. UML. [Online] http://www.uml.org/.

74. WSDL. [Online] http://www.w3.org/TR/wsdl.

75. XSLD. [Online] http://www.w3.org/TR/xslt.

76. DRools. [Online] http://www.jboss.org/drools.

77. XPath. [Online] http://www.w3.org/TR/xpath/.

List of Images

List of Tables