UNIVERSITÄT KOBLENZ · LANDAU

Fachbereich 4: Institut für Wirtschafts- und Verwaltungsinformatik

R4eGov

Making eGovernment work

Evaluation of Open Source Business Process Management Suites in the Context of R4eGov

Diplomarbeit

zur Erlangung des Grades eines
Diplom-Informatikers
im Studiengang Informatik mit
Anwendungsfach Wirtschaftsinformatik

vorgelegt von

Dimitri Petruschenko

Erstgutachter: Prof. Dr. Maria Wimmer,
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich 4

Zweitgutachter: Dipl. Inf. Daniel M. Schmidt,
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich 4

Koblenz, im Februar 2009

«Für meine Eltern Flora und Wassili Petruschenko»

«Моим родителям Флоре и Василию Петрущенко»

created with LaTeX and KOMA-Script

Acknowledgments

Mit dieser Arbeit endet die Zeit meines Studentendaseins. Es war eine aufregende, abwechslungsreiche und vor allem eine erfahrungsreiche Zeit. Für diese, möchte ich in diesem Abschnitt danken.

Der Ausdruck meiner Verbundenheit gilt in erster Linie allen Mitarbeitern der Forschungsgruppe Wimmer. Ich danke Frau Prof. Maria Wimmer und Daniel M. Schmidt. In dieser Zeit, habe ich viel über das wissenschaftliche Arbeiten, aber auch über mich selbst gelernt. Mein besonderer Dank richtet sich an Ansgar Mondorf für seine freundschaftliche, konstruktive und motivierende Art.

Ferner bedanke ich mich bei meinen Freunden. Bei Olesia Muntaniol, die den größten Teil dazu beigetragen hat, dass die Erinnerungen an mein Studium unvergesslich bleiben. Andreas Harder und seiner reizenden Frau Nejla möchte ich an dieser Stelle für ihre Freundschaft danken. Mein Dank gilt all meinen Freunden, die mich in dieser Zeit unterstützt haben.

Besonders und zutiefst Verbunden bin ich meiner Familie. Meinen lieben Eltern und Großeltern werde ich für all die Liebe, Fürsorge und Geduld ewig dankbar sein. Schließlich bedanke ich mich bei meiner aller liebsten Schwester Elena, ihrem Lebensgefährten Marco und meinem allerbesten Neffen Eugen für Ihre Liebe und Ihre Unterstützung.

Dimitri Petruschenko

Abstract

The thesis at hand evaluates Open Source Business Process Management (BPM) Systems in the context of the R4eGov[1] Project. The provision of concepts and tools to support and enable interoperability in pan-European networks of pubic administrations is one of the major objectives that R4eGov is aiming at. Thereby a strong focus lies on the interoperability of cross-organizational processes from the viewpoint of modeling, execution and monitoring. BPM can increase the effectiveness and efficiency of cross-organizational processes by restructuring them towards the needs of the entities involved. BPM is dependent on BPM systems that combine technologies of process modeling, business process analysis and execution along with their integration into adequate runtime environments and rule engines. The evaluation that is performed within the thesis investigates how far BPM systems can support several requirements of interoperability that have been developed by the R4eGov project. It also targets at analyzing those BPM system according to generic requirements on BPM and software tools. The investigation is build upon common BPM theories and standards for modeling business processes. It describes the origin and interdependencies of BPM and Workflow Management (WfM), highlighting similarities and differences from the technological and historical perspective. Moreover, it introduces web service standards and technologies that are used to build service-oriented architectures allowing greater flexibility in BPM. In addition the thesis introduces methods and best practices to evaluate software tools. It contains an evaluation framework for BPM tools that has been based on the software product evaluation standard ISO/IEC 14598. The evaluation framework comprises the definition of an R4eGov scenario and a catalogue of criteria for evaluating a set of selected Open Source BPM systems. The definition of the catalogue of criteria is build upon generic requirements on BPM systems and those that are specifically to R4eGov. The chosen methods and the core elements of the evaluation framework will be applied to the selected BPM systems Intalio BPMS,

[1] http://www.r4egov.eu/

NetBeans IDE, and JBoss jBPM. Finally the results of the applied R4eGov scenario and of the applied catalogue of criteria are being discussed by highlighting individual strengths and weaknesses of the systems.

Contents

List of Figures

List of Tables

List of Acronyms

BAM	Business Activity Monitoring
BPA	Business Process Analysis
BPD	Business Process Diagram
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPMN	Business Process Management Notation
CBP	Collaborative Business Process
IDE	Integrated Development Environment
EAW	European Arrest Warrant
EPC	Event-Driven Process Chain
EPML	EPC Markup Language
ESB	Enterprise Service Bus
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
JEMS	JBoss Enterprise Middleware System
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
PVM	Process Virtual Machine
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UDDI	Universal Discovery, Description, Integration

UML	Unified Modeling Language
WfM	Workflow Management
WfMC	Workflow Management Coalition
WSDL	Web Services Description Language
XML	Extensible Markup Language
XPDL	XML Process Definition Language

1. Introduction

1.1. Problem Scope and Challenges

This research work deals with the evaluation of open source Business Process Management systems. The evaluation concerns the special requirements for the R4eGov project which main objective is to provide concepts for the overall interoperability in public administrations across systems organizations and countries. The understanding of interoperability in R4eGov is based on the R4eGov Interoperability Framework as depicted in Figure 1.1, which illustrates three different dimensions [Wimmer et al., 2006]. One of these dimensions describes the collaboration among heterogeneous administrations on three levels, adapting eGovernment applications to local, national or EU /International characteristics. The second dimension (Seamless eAdministration) depicts the collaboration of several processes and services that have to work smoothly. The last dimension describes three different levels of interoperability: technical, semantic, and organizational. The technical level describes protocols and components supporting the semantic level with data and content, and the organizational level describes process and organization. Processes within and across organizations contain a huge potential in terms of optimization. Harmut F. Binner discusses in [Binner, 2004] the advantages of the process-oriented approach within organizations and highlights its benefits. So R4eGov deals with cross-organizational processes and their modeling as an important method for handling them. In fact, models can serve as a common language among different organizations. Furthermore they can be used to understand, analyze and optimize cross-organizational business processes [Freiheit et al., 2007]. The discipline to handle business processes is Business Process Management (BPM). Gartner descries BPM as a management discipline that treats business processes as assets to be valued, designed and exploited in their own rights. It describes a structured approach employing methods, policies, metrics, management practices and software tools to manage and continuously optimize an organization's activities and

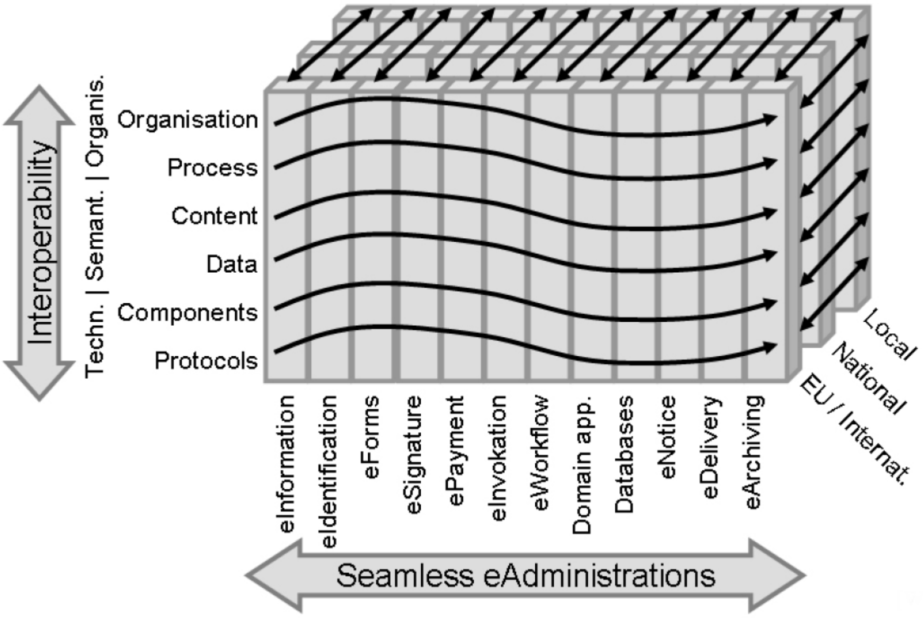


Figure 1.1.: R4eGov Interoperability Framework [Wimmer et al., 2006]

processes [Hill et al., 2006a]. The main instruments of BPM are process modeling and its executing. Thus, BPM leads to the benefit of BPM systems that combine technologies of process modeling including business process analysis tools and executing the process with integration technology, a runtime environment, and rule engine.

Process modeling and executing the process consider nowadays variety of business process modeling standards as well as business process execution standards (compare [Owen and Raj, 2003, Keller and Partner, 1999, Juric et al., 2004, Peterson, 1977]). It means that different organizations may use different standards for the definition of common processes and require the ability to interoperate with each other.

The usage of different standards in process design and execution illustrates the challenge for the definition of cross-organizational processes among different parties and thus for their interoperability. This challenge leads to the particular requirements on BPM systems and there ability to cope with them.

This work deals mainly with the question: *How far BPM systems can support the interoperability in the context of R4eGov?*

There are a lot of different vendors for BPM tools. This work pays attention to the Open

Source software that can provide some advantages to the public sector in general and thus for R4eGov as well (compare [Di Maio, 2007a]). The choice for Open Source can have several drivers [Di Maio, 2007b]. Most of all open source developers use open standards and have open development process. Not less important for the public administrations is being independent of vendors and being flexible in terms of products. The market of Open Source software grows fast and offers nowadays competitive solutions to proprietary software.

The next section will introduce the methodology and approach addressing the issue of the research question defined above.

1.2. Methodology and Approach

Based on the problem scope and challenges this section provides the approach that will be used to answer the research question. The core task of this thesis is to evaluate open source BPM systems in context of R4eGov. Evaluation is part of empirical research and it supplies methods of research concerning the estimation of concepts, plans of investigation etc. [Bortz and Döring, 2002, Rossi et al., 2004]. Evaluation offers methods to investigate different kind of objects. For instance, Wottawa and Thierau list a multitude of evaluation objects. This list contains different items that can be evaluated (e.g. People, Products, Objectives, Systems/Structures) [Wottawa and Thierau, 1990]. Since BPM systems are software products, it is necessary to analyze existing evaluation methods on that area. This analysis provides an overview and supports the decision choosing one of the introduced and most suitable approaches. This decision builds a framework for the evaluation of the BPM systems.

Before the evaluation can be executed, there are three main points that need to be defined (Figure 1.2):

- Scenario description
- Defining the catalogue of criteria
- Identification of Open Source BPM systems

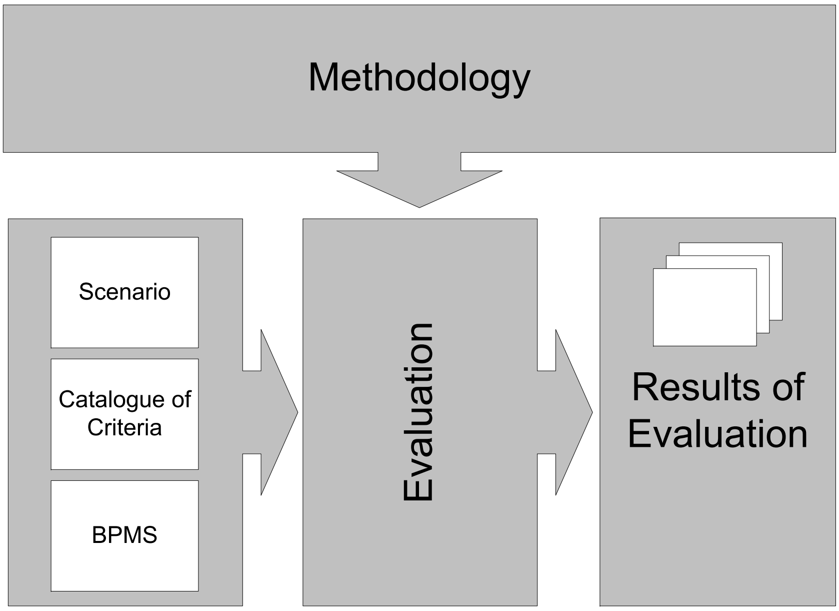


Figure 1.2.: Methodology and Approach of the Thesis

The scenario depicts a fragment process of European Arrest Warrant. This business process will be modeled in every BPM system and it will provide useful information about the tools and their behavior. The most important point of the evaluation is the catalogue of criteria. It is based on the theories and standards of BPM as well as identified requirements on BPM systems. The last point deals with the identification of Open Source BPM systems. Therefore it requires the definition of special criteria for the selection of the tools. After the main components are defined the evaluation can be executed. The scenario and the catalogue of criteria will be applied to the identified set of BPM systems. Each system will be evaluated accordingly to the catalogue of criteria modeling the scenario. The results of the evaluation will be gathered in the overall summary for each tool. These results build the basis for comparison in the last part of the thesis. This part will discuss the strengths and weaknesses of the tools opposing them to each other. The comparison of the BPM systems leads to the answer of the research question that was defined in 1.1.

The application of the methodology and approach of this thesis results in the structure introduced in the next section.

1.3. Structure of the Thesis

This section depicts the structure of the thesis providing an overview of the chapters and describing their contents.

The second chapter describes the theories and standards of business process management. It depicts BPM and Workflow Management (WfM) highlighting their similarities and differences from the technological and historical point of view. Moreover, this chapter includes the introduction to nowadays increasing technology Service Oriented Architecture (SOA) that supports BPM in terms of flexibility of creating and management of business processes. This technology leads in the next section to the use of web services which are a particular part of SOA. The last section of this chapter introduces selected Business Process Modeling standards such as: Petri Nets, Unified Modeling Language (UML) Activity Diagram, Event-Driven Process Chain (EPC), Business Process Management Notation (BPMN), XML Process Definition Language (XPDL), and Business Process Execution Language (BPEL). This chapter builds the theoretical basis of the thesis.

The third chapter deals with the analysis of the evaluating methods. In this part the existing method for evaluation of software tools are identified and discussed. This discussion leads to the selection of the ISO/IEC 14598 and describes the standard applying the provided methodology to the approach of the evaluation. This standard depicts the framework of the evaluation and is shown in detail in the last section of this chapter.

The chapter, Definition of Evaluation Criteria, presents the conceptual part of this. It defines the catalogue of criteria considering and discussing the requirements from R4eGov as well as general requirements on BPM systems. Also the description of the scenario occurs in this part and will be used for the further evaluation. This section introduces the European Arrest Warrant (EAW) and defines the business process that is examined for the modeling within the evaluation. Moreover, this part deals also with the identification of the Open Source BPM systems.

The fifth chapter, Evaluation, is the practical part of the thesis. This chapter contains the application of the catalogue of criteria and of scenario to the identified BPM systems (Intalio BPMS, NetBeans IDE, and JBoss jBPM). The evaluation is applied to every system separately and finishes with overall summary of results.

The last chapter deals with the comparison of the evaluated tools. The first part of this chapter discusses the aims of the evaluation, facing and comparing the achieved results. The section, Conclusion, contains a general review of the whole work accordingly to the research question that is defined out of the problem scope and challenges of this thesis.

2. Theories and Standards of Business Process Management

2.1. Business Process Management & Workflow Management

Business Process Management (BPM) has received a lot of attention in the industrial engineering and management literature. As a matter of fact, the public sector is using BPM in as an information management solution. However, most of the publications about BPM come from the private or academic sectors and little has been published on topic. But actually, BPM can bring benefit to the public sector as well; it can increase effectiveness and efficiency by restructuring the organization along cross-functional and cross-organizational processes [Gulledge Jr and Sommer, 2002]. BPM gives the opportunity for public sector organizations to establish a standard IT infrastructure allowing leaders and their staff to develop and deploy business on ”as needed” basis. Contrary to software packages approach, BPM completely disocciates the process design from the IT infrastructure design [Smith and Fingar, 2002].

The denotation BPM emphasizes business processes. A Business Process is [Coalition, 1999]: ”a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships” . There are several points of view regarding BPM and its content. In the literature, BPM is often described as a lifecycle going through various stages (compare [Allweyer, 2005, Jost and Kruppke, 2004, Miserez, 2006]).

Van der Aalst faces in [van der Aalst, 2004] BPM and its relation to the Workflow Management (WfM). The author describes BPM-lifecycle in four phases (see Figure 2.1). In the design phase, the processes are modeled and designed. Then, a system

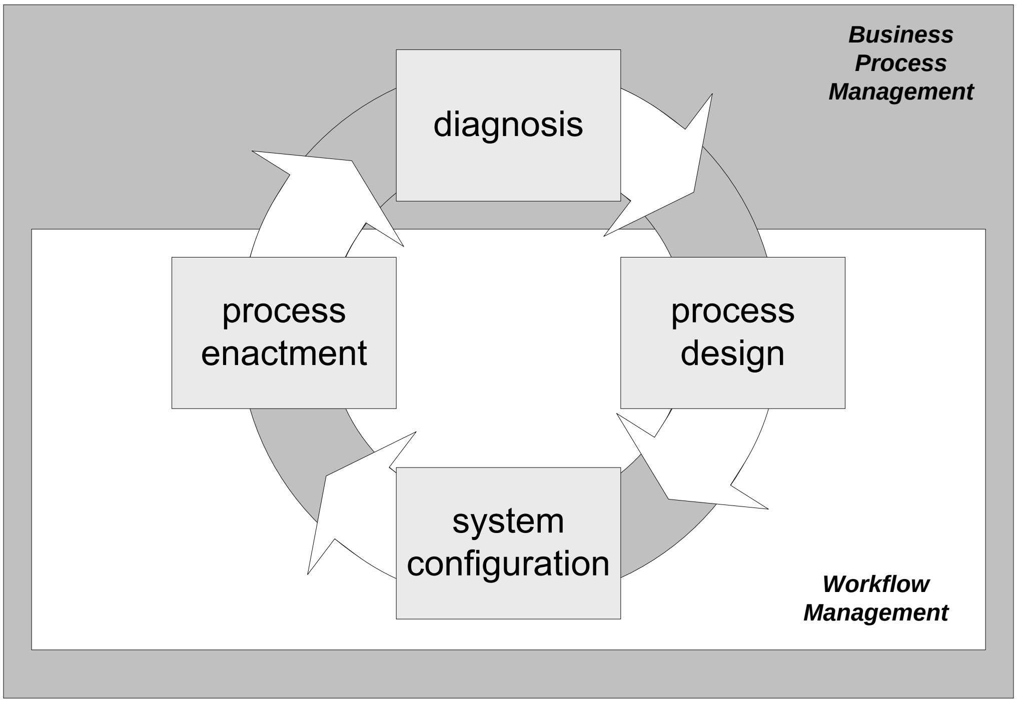


Figure 2.1.: BPM Lifecycle

configuration phase includes the integration of information systems, data source or other technology into business process (e.g. a Service Oriented Architecture composing application frameworks like Web-based applications). The enactment phase starts when the operational business processes are executed using the system configured and turning models into real-world action (e.g. using Business Process Execution Language (BPEL) or other execution languages). In the last phase, diagnosis, the operational processes are analyzed to identify potential problems and find out solutions to improve the overall process.

That view makes a clear distinction between BPM and WfM. The author sees BPM as an extension of the traditional WfM approach that supplies support for the diagnosis phase. According to this observation, he defines BPM as follows [van der Aalst, 2004]: ”Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information.” There are many definitions of BPM, but in most cases, it clearly includes WfM.

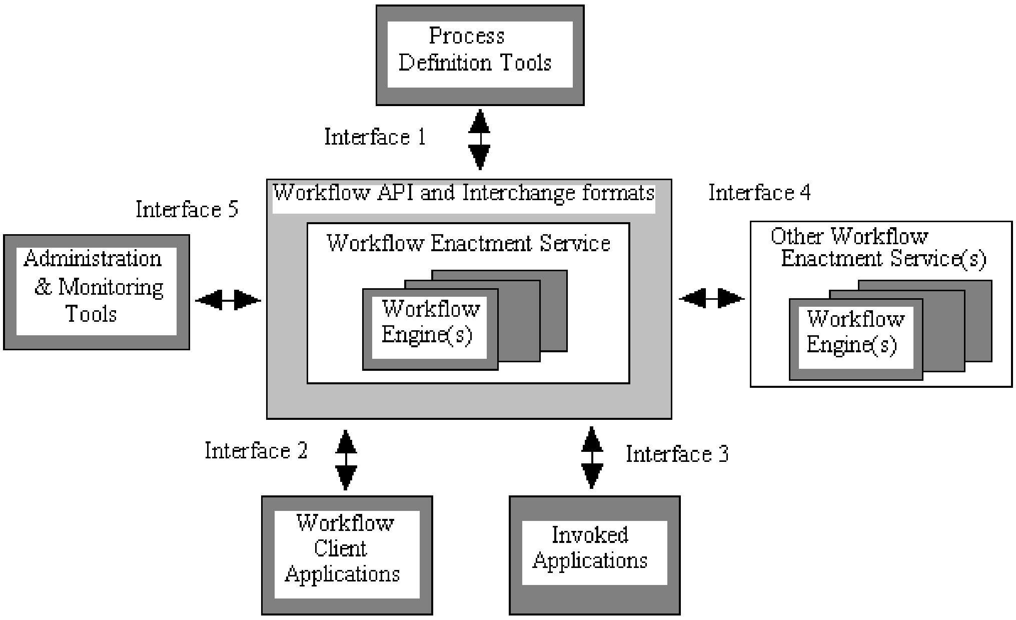

Process Definition Tools
Interface 1
Workflow API and Interchange formats
Workflow Enactment Service
Workflow Engine(s)
Interface 5
Administration & Monitoring Tools
Interface 4
Other Workflow Enactment Service(s)
Workflow Engine(s)
Interface 2
Interface 3
Workflow Client Applications
Invoked Applications

Figure 2.2.: Workflow Reference Model

WfM belongs to the main field of the Workflow Management Coalition (WfMC). The WfMC is a non profit organization that aims at creating new opportunities for the exploitation of workflow technology through the development of common terminology and standards. The core term of this field is a workflow that is defined as [Coalition, 1999]: ”The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” Moreover, WfMC provides a workflow reference model that describes a generic workflow application structure (see Figure 2.2).

The workflow reference model defines five components of workflow, referring to the five interfaces to the workflow enactment service (or invocation engine). The interface 1 is the process definition. Various tools exist on the market and can use this interface for process definition and further execution (e.g. WfMC proposes XML Process Definition Language (XPDL) for process definition). The interface 2 focuses on the client application – the program that invokes the workflow process. Accordingly to the history of WfM, the workflow tools are more concentrated on human interface than BPM, traditionally focused on processes with less human interaction. The third interface is designed for the programs invoked by the business process. Concerned applications are usually web services which can be offered by a third party (it can be a part of a Service Oriented

Architecture). The interface 4 is realized as a communication interface between workflow systems. This interface allows initiating a work on another system and it can be seen as an interoperability interface. Last, but not least, interface 5 is dedicated to administration and monitoring of the system. Even if WfM is seen in [van der Aalst, 2004] as a system with weak diagnosis part, the workflow reference model includes this step as well, like BPM [Fischer, 2005].

It is important to consider that BPM and WfM have different historical backgrounds. In IT business literature, a quite widespread opinion mentions that BPM is more flexible than WfM. It is based on the idea that the computing model of WfM is older than BPM [Smith and Fingar, 2002]. It is true that the initial purposes of these models were different. In fact, BPM was originally focused on computer transaction whereas WfM was focused on content requiring human judgment or processing. But both BPM and WfM allow a process to be designed, tested, and used. WfMC points to fact that they come from different origins, and thus have different strengths. ”The key is to look beyond the product name, and find the function that will best serve the business” [Fischer, 2005].

After the views of BPM and WfM were depicted it can be summarized that both focus on exceptional process flexibility which allows workflows to be determined in real-time by the events or outcomes within the process. The flexibility can be achieved avoiding hard-coded logic of processes using an integration technology that loosely couple the applications and resources that make up the process. One of the technologies that increased attention these days is Service Oriented Architecture (SOA) considered as realization for BPM [Noel, 2005, Decker et al., 2006].

2.2. Service Oriented Architecture

SOA is a business operations strategy for leveraging information to meet the organization’s objectives, such as increasing overall revenue, boosting customer satisfaction, improving product quality, and enhancing operational agility [Durvasula et al., 2006]. SOA can be called as an architectural style in which systems are modular and their components are distributable, have defined interfaces and are loosely coupled and shareable. A SOA can provide a complete view of the independent software system. The services within SOA are connected and can be invoked through Enterprise Service Bus (ESB). An ESB is an integration platform based on standards which combines messaging, web services, data transformation, and intelligent routing of diverse applications [Chappell, 2004].

There are five characteristics which describe a SOA implementation [Schulte, 2008, Richter et al., 2005]:

1. **Modular**: The system has two or more components (usually dozens), including at least one component that acts as a service consumer and another that acts as a service provider (e.g. Web Service).

2. **Distributable**: The components can run on disparate computers and can communicate with each other by sending messages over a network at runtime. SOA relies on program-to-program communication.

3. **Defined interfaces**: Component interfaces are documented using metadata that specifies an explicit contract between consumers and providers. This metadata describes the messages that are exchanged and other characteristics of the agreement among the components (e.g. Web Services Description Language (WSDL)).

4. **Loosely coupled**: A provider component can be swapped out for another component that supplies the same service without changing or recompiling the consumer (or consumers), because the interface is separated from the service provider's implementation (the provider component's internal code and data).

5. **Shareable**: A service provider component can be used successively by disparate consumer components (sometimes called ”reuse”).

The main focus of an SOA is the definition of a business infrastructure. The business infrastructure is based on services, either as service provider or service consumer including the key concepts of service, service repository, and already mentioned service bus. Service repository contains part of the information of service (e.g. Web Service) and operation in form of WSDL, service owner, and access rights on a particular service. In addition, there are application front-end which belong also to a SOA, initiating and controlling all activity of the enterprise systems [Woods and Mattern, 2006].

SOA focuses on business-centric services with business level transaction granularity and not on technology-oriented entities. Furthermore, SOA represents an architectural proposal, which can be realized with different technologies. The purpose is to define cleanly cut service contracts with a clear business orientation [Krafzig et al., 2004].It can reduce the time, effort and cost needed to implement or change distributed application systems compared with other approaches [Schulte and Abrams, 2007].

2.3. Web Services

A Web Service can be included as a part of a service oriented architecture supporting several of its characteristics (e.g. modular, distributable, defined interfaces etc.). In [Schlimmer et al., 2002], a Web Service is defined as a software application, whose interfaces and binding are capable of being defined, described and discovered by Extensible Markup Language (XML) artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols. It means that web service is designed to support interoperable machine-to-machine interaction over a network. The interaction occurs via defined interfaces (e.g. WSDL) using Simple Object Access Protocol (SOAP)-messages, typically conveyed using Hypertext Transfer Protocol (HTTP) with an XML serialization in conjunction with other Web-related standards [Haas and Brown, 2003]. A Web Services can be published and discovered in Universal Discovery, Description, Integration (UDDI) directory.

The functionality and elements of a web service are depicted in Figure 2.3[1]. Thus, web services contain following components [Fensel et al., 2006]:

- A shared and accepted transport protocol (e.g. SMTP, FTP or HTTP).
- A message description format which is an independent platform (e.g. SOAP to exchange XML-coded messages via HTTP).
- A web service interface description that describes which messages and operations a service can offer (e.g. WSDL for XML-based description of interfaces).
- A registry for publication and query of available web services (e.g. UDDI to publish, browse and query existing web services).

SOAP defines the runtime message and is independent of any particular transport and implementation technology. WSDL describes a Web Service and the SOAP Message providing

[1] source: `http://www.w3.org/TR/2002/WD-ws-arch-20021114/`

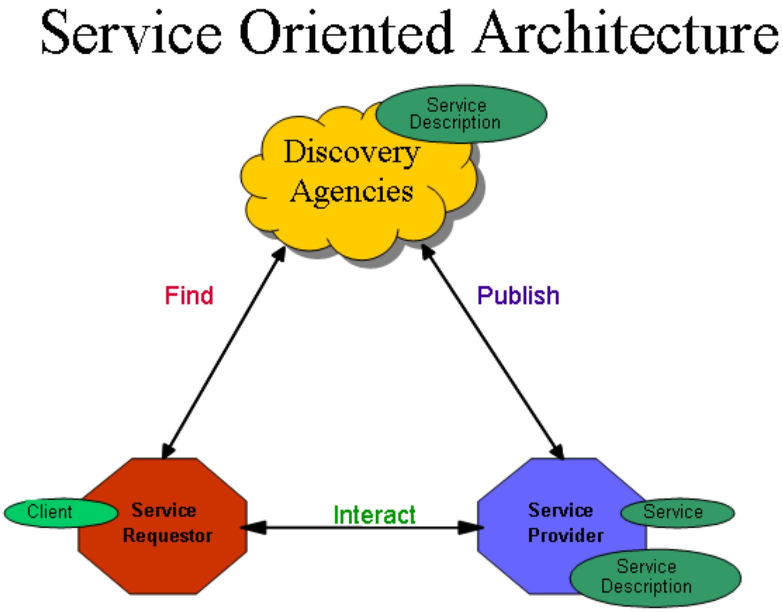


Figure 2.3.: SOA and Web Services

a programmatic way to describe what a service does. UDDI is a cross industry initiative to create a standard for service discovery together with a registry facility that facilitates the publishing and discovery processes [Arroyo et al., 2004, Leymann et al., 2002, Papazoglou and Georgakopoulos, 2003].

After SOA and Web Services were introduced it is also important to mention and explain the web services collaboration. A common way to describe this collaboration refers to the expressions orchestration and choreography. Orchestration concerns executable business process which may depict the interaction between web services, and describes the way the web services can interact including the execution order and business logic of the interaction. Choreography describes collaboration among a set of relevant web services according to the order which is required to achieve a common goal. This covers the interactions from a global point of view of all participating web services [Peltz, 2003, Barros et al., 2005].

2.3.1. Simple Object Access Protocol

In the technology of web services, SOAP plays the role of a standardized packaging protocol for the messages. This protocol is shared by the applications which may contain the orchestration of web services depicted by business processes [Snell et al., 2002]. It was intended for exchanging structured information in a decentralized, distributed environment. SOAP is based on XML technologies providing a message construct that can be exchanged according to the underlying protocols. Thus, SOAP depicts a framework that is independent of any other particular programming model and other implementation specific semantic [Gudgin et al., 2007].

SOAP Version 1.2 is defined through four parts. The first part is SOAP Processing Model that defines the rules for processing a SOAP message. SOAP Extensibility Model defines the concepts of SOAP modules and SOAP features. The third part SOAP Protocol Biding Framework, defines a binding to an underlying protocol for exchanging SOAP messages between nodes. And the last part is SOAP Message Construct, defining the structure of a SOAP message [Gudgin et al., 2007].

These four parts of SOAP were designed to meet two major goals. On one hand, keep the whole framework as simple as possible and on the other hand, to guarantee extensibility supporting the development of new features and modules.

2.3.2. Universal Discovery, Description, Integration

UDDI depicts a platform-independent registry that allows publishing, retrieving, and managing information about web services described therein. Its development is led by the Organization for the Advancement of Structured Information Standards (OASIS) consortium of enterprise software vendors and customers. UDDI supports several established industry standards (e.g. HTTP, XML, SOAP, and WSDL) and is a central component of service oriented architecture. UDDI brings several benefits to both service providers and service consumers managing following tasks [OASIS, 2004]:

- Publishing information about Web services and categorization rules specific to an organization

- Finding Web services (within an organization or across organizational boundaries) that meet given criteria
- Determining the security and transport protocols supported by a given Web service and the parameters necessary to invoke the service
- Providing a mean to insulate applications (and providing fail-over and intelligent routing) from failures or changes in invoked services

The information of a Web service and the parameters, necessary to invoke that service, include also a link to a WSDL file that describes the interfaces of the web service itself.

2.3.3. Web Services Description Language

WSDL is an XML used to describe a web services interface and also specify the location of the service and the operations (or methods) the service exposes. It was created by IBM and Microsoft and is currently developed by W3 Consortium's Web Services Description working Group. WSDL enables to separate the description of the abstract functionality offered by a service from concrete details of a service description such as ”how” and ”where” that functionality is offered [Chinnici et al., 2004].

A WSDL document defines services as collections of network endpoints, or *ports*. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: *messages*, which are abstract descriptions of the data being exchanged, and *port types* which are abstract collections of *operations*. The concrete protocol and data format specifications for a particular port type constitute a reusable *binding*. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service.

Thus, the structure of WSDL shown in Figure 2.4[2] contains following elements [Christensen et al., 2001]:

- **Types**: a container for data type definitions using some type system (such as XSD).
- **Message**: an abstract, typed definition of the data being communicated.

[2] source: http://www.ibm.com/developerworks/web/library/wa-aj-end2end3/

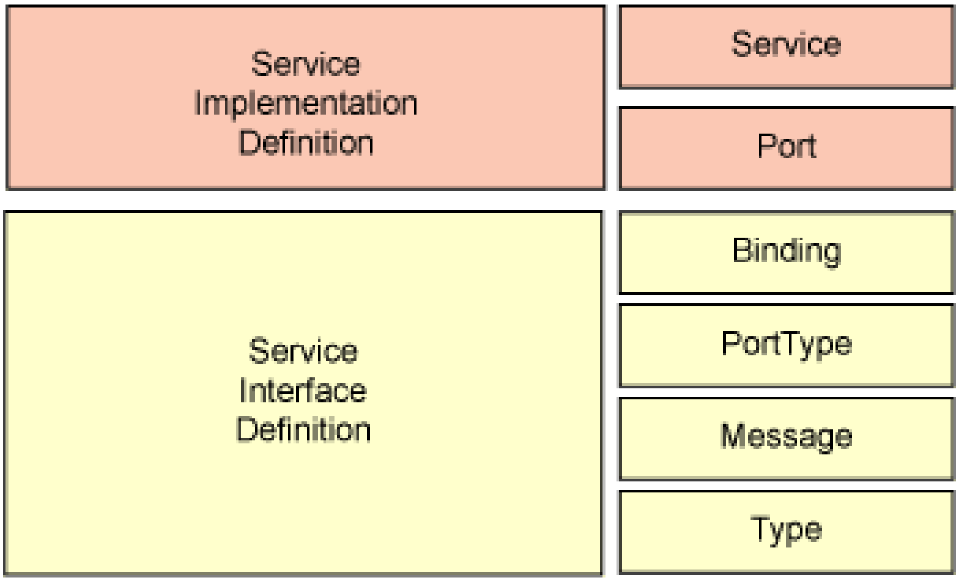

Service Implementation Definition
Service
Port
Service Interface Definition
Binding
PortType
Message
Type

Figure 2.4.: Structure of WSDL

- **Port Type**: an abstract set of operations supported by one or more endpoints.
- **Binding**: a concrete protocol and data format specification for a particular port type.
- **Port**: a single endpoint defined as a combination of a binding and a network address.
- **Service**: a collection of related endpoints.

WSDL is independent of other protocols, languages or implementations used in other standards such as XML and Internet Protocol (IP), which are prerequisites for Web services. This independence allows WSDL to be the single basic Web services standard used across different languages, bindings and platforms [Plummer et al., 2004].

2.4. Selected Business Process Modeling Standards

2.4.1. Petri Nets

The concept of the Petri net was developed by Carl Adam Petri in his dissertation that was submitted to the faculty of Mathematics and Physics at the Technical University of

Darmstadt.

A Petri net is a directed, connected, bipartite graph [Hamadi and Benatallah, 2003]. Petri nets can represent finite-state machines or their state diagrams. Also parallel activities or concurrency can be easily expressed in terms of Petri nets. Furthermore Petri nets can be used to represent flow of control as well as the flow of data. It is also possible to depict communications protocols, synchronization control etc. using the Petri nets (see [Murata, 1989]). A Petri net contains following elements as place nodes (circles), transition nodes (bars), and directed arcs connecting places with transitions.

Petri nets became interesting in different research fields. So in terms of workflows (business processes) were made some considerations. Van der Aalst introduces in his paper [van der Aalst et al., 1998] workflow management as an application domain for Petri nets, presenting state-of-the-art results with respect to the verification of workflows, and highlighting some Petri-net-based workflow tools. Hamandi and Benatallah go further, proposing and discussing in [Hamadi and Benatallah, 2003] the use of a Petri Net-based model for Web Service composition. The Web Service and its functionality are introduced in the section 2.3.

For more elaborate introduction to Petri nets, the reader is referred to [Peterson, 1977, Peterson, 1981, Reisig, 1985].

2.4.2. UML Activity Diagram

Unified Modeling Language (UML) is a family of graphical notations that help in describing and designing software systems. The UML is controlled by the Object Management Group (OMG), an open consortium of companies. The OMG was formed to build standards that supported interoperability, specifically the interoperability of object-oriented systems. [Fowler, 2004] The first version of UML (0.9) was developed by Booch, Rumbaugh, and Jacobson in 1996 [Rumbaugh et al., 1996]. This development has not stopped and the most recent version UML 2.0 includes 13 modeling notations considering behavior diagrams and structured diagrams.

One area of UML that has received particular attention is that of Activity Diagrams (AD), which provide a high-level means of modeling dynamic system behavior. Activity diagram, a graph with directed edges, contains following core elements [Jeckle et al., 2004]:

- One or several activities
- Actions
- Object nodes
- Control elements for the workflow
- Connecting edges

This diagram is interesting in terms of suitability for BPM as well as a workflow specification language [Russell et al., 2006, Dumas and Hofstede, 2001].

2.4.3. Event-Driven Process Chain

Event-Driven Process Chain (EPC) are introduced in [Keller et al., 1992] to represent temporal and logical dependences in business process. EPC were developed at the Institute for Information Systems at the Saarland University in Saarbrücken, in cooperation with SAP.

The development occurred within the framework of ARIS in order to model business processes [Nüttgens et al., 1998]. SAP AG has been using them to express their SAP reference model [Keller and Partner, 1999].

EPC is an intuitive graphical business process description language. That language is quite popular among business people since it is easy to understand and use. With the notation of the EPC, it is possible to describe the business logic within the business process.

The notation includes first of all the core elements (see Figure 2.5). Then, there are functions which form the active elements within the EPC. The function nodes include a task activity, but can also include the whole business process. They can be activated by an event and can create other events. Events are passive elements of EPC. An event describes the state of business process in a model. Events can trigger one or several functions simultaneously. Furthermore there are logical connectors which can split or join a control flow. Each connector node supplies three types of logical operations (AND, OR, XOR). All these core elements are linked via control flows arcs.

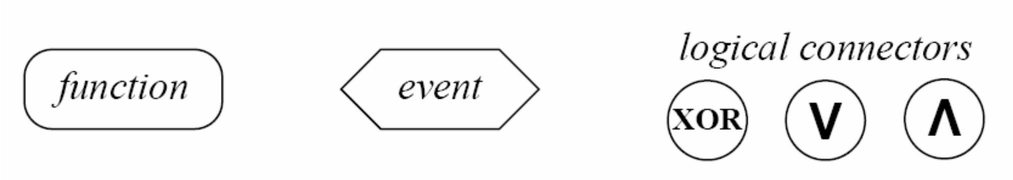


Figure 2.5.: EPC core elements

Moreover it is possible to extend event-driven process chains with entities, business objects, and organizational units. This extension makes the EPC or eEPC more expressive and powerful in terms of modelling [van der Aalst, 1999, Nissen et al., 2008].

EPC have their own interchange format EPC Markup Language (EPML). EPML was developed to serve as a tool-neutral interchange and intermediary format for EPC business process models. For further EPML related papers the reader is referred to [Mendling and Nüttgens, 2006, Mendling and Nüttgens, 2004].

2.4.4. Business Process Management Notation

Business Process Management Notation (BPMN) was developed by the Business Process Management Initiative (BPMI) with the purpose of becoming a standard in business process modeling. Now BPMN is maintained by the OMG since the two organizations merged in 2005. The goal of the BPMN is to provide a set of tools, understandable and useable by all business users. Business analysts use BPMN to create the initial drafts of the processes. BPMN is a graphical notation for drawing business processes in a workflow. This notation is depicted by the Business Process Diagram (BPD) that is based on a flowcharting technique for creating graphical models of business process operations. Also, it provides BPMN with an internal model (specification) that allows partial mapping to BPM execution languages such as BPEL. The language is intended mainly for technical developers who are responsible for implementing the technology using the definition of business processes. Thus, BPMN should build a standardized bridge for the gap between the business process design and process implementation [White, 2004b]. However, there is still a long way to come directly from a business model to an executable code [Wohed et al., 2006].

BPMN is able and allows depicting two basic types of models within the Business Process Management. On one hand, it is possible to model Collaborative (Public) B2B Processes which concerns the interaction between two or more business entities. This type of diagrams generally use a global point of view without paying attention to any particular participant but showing the interaction between two or more of them. The interactions are shown as a sequence of activities. On the other hand there is an Internal (Private) Business Process which generally focuses on the perspective of a single business organization. In fact, although internal processes often show interactions with external participants, they define the activities that are not normally visible to the public and are, therefore, private activities [White, 2004b].

BPMN provides four basic categories of elements which include *Flow Objects*, *Connecting Objects*, *Swimlanes*, and *Artifacts*. Flow objects build the main graphical elements which define the behavior of a business process. The core elements of flow objects are Events, Activities, and Gateways. The relation between flow objects is defined by connecting objects which provides three ways of connection such as Sequence Flow, Message Flow, and Association. The grouping of modeling elements can be done through the use of swim lanes (e.g. pools and lanes). The BPMN model can be enhanced through an artifact that is an additional process information like data objects, annotations and groups [White, 2004a].

Modeling with BPMN is essential to understand the business processes within an enterprise. This modeling technique enables a firm to understand and design its enterprise architecture, and therefore to react to change quicker, and in a safer manner [Owen and Raj, 2003]. However, discussing the advantages of BPMN, we should mention that the translation of BPMN model to BPEL code is not working smoothly. One of the reasons is the absence of congruent ability among BPMN and BPEL which makes the mapping not as simple as it is often claimed to be [Wohed et al., 2006].

2.4.5. XML Process Definition Language

XML Process Definition Language was published by WfMC in 2002 and is based on the Workflow Definition Language (WPDL). WPDL was designed as an interchange format to allow two WPDL-compliant BPMS products to exchange process definitions. The idea was for the process definition to be executable by another BPMS that can import the WPDL process definition, once a WPDL definition has been exported [Chang, 2005]. Then, WPDL was replaced and extended by XPDL. However, XPDL is not only a conversion of WPDL. XML Process Definition Language uses an XML-based syntax, specified by an XML schema and contains the specification for Web service support that is not included in WPDL. Furthermore XPDL has been enhanced to additionally serve as an interchange format for BPMN (see 2.4.4). Since October of 2005 the version 2.0 of XPDL includes also new concepts adapted from BPMN including e.g. Pools, Gateways or Events.

The main elements of the language are [van der Aalst, 2003]: Package, Application, Process, Activity, Transition, and Participant.

The *Package* serves as a container for all information associated with a process definition including Pools, Processes, Participants, Applications, Data Fields and Data Types. The definition of Application provides descriptions of the IT applications or interfaces. It could be a generic industry tool, a specific departmental or enterprise services or a localized procedures implemented within the framework of process/workflow management system.

The *Process* Definition provides contextual information. That information relates to other entities within the process. The Process Definition includes the process itself and provides information in association with administration contents (creation date, author etc.) or to be used during process execution (execution priority, time limits to be checked, person to be notified, etc.). A process consists of one or more activities.

An *Activity* represents work, which will be performed by a combination of resources or any IT applications. Furthermore, other optional information may be associated with activities such as information, whether it is to be started/finished automatically by the process or workflow management system or its priority relative to other activities where contention for resource or system services occurs. Activities are related one to another via flow control conditions. There are different types of Activities. The Task/Tool activity describes an activity that is executed automatically, without involving human beings. The Route activity specifies join and split conditions. Similar to a Gateway, its sole purpose is to represent complex control flow conditions. It can specify both data and event-based branching conditions. The Block Activity defines an embedded sub-process that executes an Activity Set. Different types of Events are adapted from BPMN as special types of Activities.

The flow control conditions depict *Transition* information. Each individual transition has three elementary properties, the from-activity, the to-activity and the condition under which the transition is made. The transition from one activity to another may be conditional or unconditional, while the transitions within a process may result in the sequential or parallel operation of individual activities within the process. The information related to associated split or join conditions is defined within the appropriate activity, either split as a form of ”post activity” processing in the from-activity, or joined as a form of ”pre-activity” processing in the to-activity.

Participant declaration provides descriptions of resources that can act as the performer of the various activities in the process definition. The particular resources that can be assigned to perform a specific activity, are specified as an attribute of the activity: the participant assignment, which links the activity to the set of resources (within the participant declaration) that may be allocated to it. The participant declaration does not necessarily refer to a human or a single person, but may also identify a set of people with appropriate skills or responsibilities, or machine/automates resources rather than humans. The meta-model includes some simple types of resources that may be defined within the participant declaration [Shapiro, 2005].

XPDL aims at storing and exchanging process diagrams. It does not, however, guarantee the accurate execution semantics. In contrast to XPDL, web service composition languages, like WS-BPEL, that are designed from the perspective of developers support the interaction between web services. In WS-BPEL the human only serves as an operator of the user interface. Thus it lacks to directly support manual activities because the attribution of resources, roles and organizational units is missing. In contrast XPDL focuses on the definition of manual activities that are performed by participants such as humans, organizational units and roles. Since the WfMC released version 2.0 of XPDL, it also covers graphical elements like pools, swim lanes, gateways and events and thus allows a bidirectional interchange with BPMN while BPEL only offers unidirectional interchange (from BPMN to BPEL) [Bartonitz, 2005].

2.4.6. Business Process Execution Language

BPEL for Web Services provides a means to formally specify business processes and interaction protocols. BPEL4WS results from the merging between WSFL (from IBM) and XLANG (from Microsoft), and thus combines the best of both specifications. The initial specification of BPEL4WS was released in 2002 [Andrews et al., 2003]. In 2003 the BPEL4WS was submitted to an OASIS technical committee so that the specification could be developed into an official, open standard. The current specification is called Web Services Business Process Execution Language Version 2.0 (WS-BPEL 2.0) and it was completed in May 2007 [Alves et al., 2007].

BPEL is a language which is used for composition, orchestration and coordination of web services. It has rich vocabulary which allows expressing the behavior of business processes [Juric et al., 2004]. There are different language components but the BPEL processes always start with process element and the condition of that element is to include at least one activity. There are two kinds of Activities: *Basic Activities* and *Structured Activities*. Every basic activity has several standard attributes and elements that can be used to specify certain properties (e.g. Invoke, Receive, Reply and Waiting). Structured activities on the other hand offer a way to structure a BPEL process using *sequence*, *switch* and *while* activities.

BPEL also defines executable and abstract processes which depict different views. The abstract process is not executable and it specifies the internal message exchange behavior from the perspective of a single organization or composite service. On the other hand an executable process defines the execution order of a set of activities including either involved partners and exchanged messages between them or exception and fault handling [Ouyang et al., 2007]. The execution itself occurs on a BPEL server which provide run-time environment for that [Juric et al., 2004]. The main benefits of BPEL are summarized in [Alves et al., 2007] highlighting the importance and convenience provided by this standard according to the SOA.

3. Methods for Evaluating Software Tools

3.1. Analysis of Existing Evaluation Methods

There are several methods to evaluate software. Choosing the right one is very important and can be the crucial factor for the evaluation results.

In [Brown and Wallnau, 1996], Brown and Wallnau distinguish identify two classes of evaluation. On the one hand there is product-oriented evaluation, which deals with selecting among a set of products that provide similar functionality. On the other hand there is process-oriented evaluation, which deals with assessing the impact of a new technology on existing practices to understand how it will improve performance or increase quality. The class of product-oriented methods suits to the aims of this work. According to this classification the authors introduce a framework for systematic evaluation of software technologies. This framework, called Technology Delta Evaluation Framework, consists of three phases (Descriptive Modeling Phase, Experiment Design Phase, and Experimental Evaluation Phase), which divides the process of evaluation in different activities.

Also Hegner examines in [Hegner, 2003] the methods for software evaluation, mainly distinguishing subjective and objective methods. The author provides criteria for the right selection of a particular evaluation method. These criteria are related to the different purposes of the evaluation (e.g. functionality, usability, suitability for the task, self-descriptiveness etc.).

Another approach is the knowledge based evaluation of software systems [Stamelos et al., 2000]. Here, the authors discuss software evaluation problems and they propose a solution in

Multiple-Criteria Decision Aid. This technique provides a model with elements (Multiple-Criteria) depicted as a 7-ple $\{A, T, D, M, E, G, R\}$:

- A is the set of alternatives under evaluation in the model
- T is the type of the evaluation
- D is the tree of the evaluation attributes
- M is the set of associated measures
- E is the set of scales associated to the attributes
- G is the set of criteria constructed in order to represent the user's preferences
- R is the preference aggregation procedure

The software evaluation methods introduced here are very useful and they could also provide a framework for this work. However, the best solution was identified ISO/IEC 14598 Software Product Evaluation standard. It provides a strict and clear structure for the evaluation process, considering different existing roles within an evaluation (e.g. developers, acquires etc.)

The following section introduces Software Product Evaluation standard ISO/IEC 14598 in details.

3.2. Methodolgy of Evaluation: ISO/IEC 14598

For the evaluation of the Business Process Management (BPM)-Systems the Information Technology - Software product evaluation standard (ISO/IEC 14598) was chosen. The standard consists of six parts:

1. General overview
2. Planning and Management
3. Process for Developers
4. Process for Acquirers
5. Process for Evaluation

6. Evaluation modules

The first part of ISO/IEC 14598 introduces the other parts. It contains general requirements for specification and evaluation of software quality, defines the technical terms used in the other parts and clarifies the general concepts [ISO, 1999a]. The part of Planning and Management provides a guide to people involved in managing a quantitative software product or project evaluation within an existing quality management system [ISO, 1999b]. The third standard provides guidelines to clarify quality requirements and implement and analyze software quality metrics. This part applies to all software at all phases of the life cycle development [ISO, 1999c]. The part of process for acquirers should be used by organizations that are planning to acquire or buy a software product that will be developed or has been developed. The fifth part provides requirements and recommendations for the practical implementation of software product evaluation. This process defines the activities needed to analyze evaluation requirements, to specify, design and perform evaluation actions and to conclude the evaluation of any kind software product [ISO, 1999d]. The last part defines the structure and content of the documentation to be used to describe an Evaluation Module that is intended to be used in evaluation technology such as research institutes, testing laboratories and others when producing new evaluation modules [ISO, 1999e]. Thus, the ISO/IEC 14598 standard provides guidance and requirements for the evaluation process in three different situations (Development, Acquisition, and Evaluation) [ISO, 1999a]. This work focuses on the process for evaluators (ISO/IEC 14598-5) and is intended for people who perform independent evaluation. The evaluation process contains the five activities listed below [ISO, 1999d]:

- **analysis** of evaluation requirements;
- **specification** of the evaluation based on the evaluation requirements and on the description of the product provided by the requester;
- **design** of the evaluation which produces an evaluation plan on the basis of the evaluation specification; this activity takes into account the components of the software product to be evaluated and the evaluation methods proposed by the evaluator;
- **execution** of the evaluation plan which consists of inspecting, modeling, measuring and testing the products and its components according to the evaluation plan; the

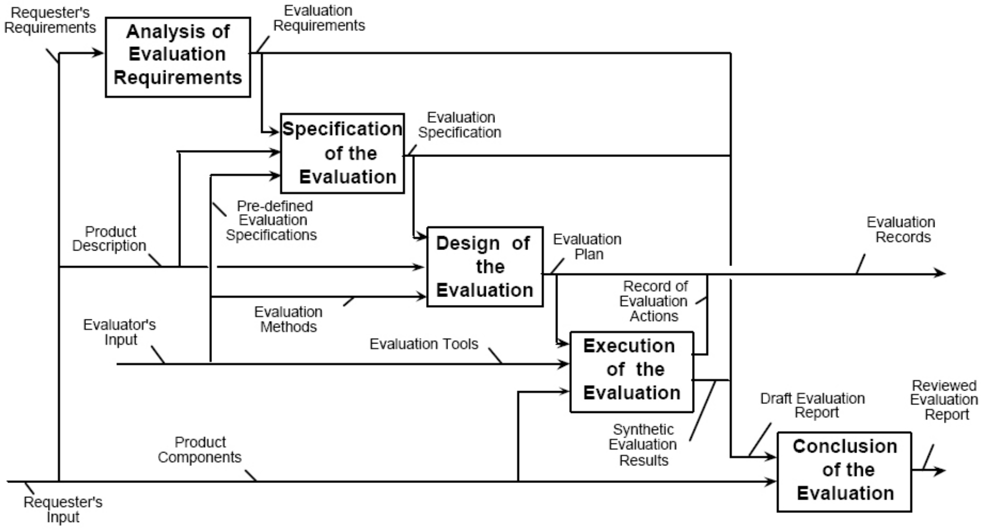


Figure 3.1.: The evaluation process [ISO, 1999d]

actions performed by the evaluator are recorded and the results obtained are put in a draft evaluation report;

- **conclusion** of the evaluation, which consists of the delivery of the evaluation report and the disposal by the evaluator of the product evaluated as well as its components when they have been transmitted independently.

The following Figure 3.1 shows the evaluation process and the dependences between several activities and resulting documents (e.g. evaluation specification or evaluation plan).

The approach for the further work is leaded from this standard. The chapters 2 and 4 build the basis for the analysis of requirements, specification of the evaluation, and the evaluation design. The core elements required for the evaluation are the catalogue of criteria (see 4.3), the set of identified BPM-Systems (see 4.4), and the scenario introduced in 4.2.2.

The creation of catalogue of criteria considers the identification of requirements that are specified in 4.1. This identification contains the R4eGov requirements on Collaborative Business Process (CBP) as well as requirements on Business Process Management Systems (BPM-Systems). The identification of BPM-System introduced in 4.4 is based on particular criteria that are leaded form a list of attributes for mature Open Source software. The scenario describes an example process that will be modeled in the process of evaluation. This process is supposed to provide a knowledge about the used tools for the process design, and thus for the used modeling language.

As proposed in the evaluation process, the results are discussed after the execution of the evaluation for each tool. Afterwards, the last chapter contains the comparison of the tools and discusses their strengths and weaknesses.

4. Definition of Evaluation Criteria

4.1. Idetification of Requirements on Business Process Management Systems

4.1.1. R4eGov Requirements on Collaborative Business Processes

The definition of Collaborative Business Process (CBP) (also called cross-organizational business process) is built on the definition of business process (see 2.1): A business process is a goal-driven sequence of activities. Using this definition R4eGov defines in [Freiheit et al., 2007] a cross-organizational business process as follows: *A cross-organizational business process is a sequence of activities executed by two or more organizations aiming to realize a shared goal.*

Accordingly to the definition above R4eGov identifies different requirements on CBP. These requirements are divided into general requirements for CBP concepts and requirements for CBP modeling language. Requirements for CBP modeling language contain general requirements as well as specific requirements for modeling language.

R4eGov concluded four requirements of inter-organizational workflows for CBP concepts. The first requirement *Preserve legacy systems* describes the importance of keeping own entire IT infrastructure without expecting of change. The next requirement *Support of the service oriented architecture* leaded from the requirement above depicts the need of the concept called Service Oriented Architecture (SOA). SOA provides a powerful opportunity of making business processes collaborative without replacing an exiting IT infrastructure. The third requirement *Privacy respect principle* deals with preservation of the know-how of each collaborative partner although allowing cooperation and improving productivity. For that purpose cooperation require a certain degree of workflow inter-visibility. The last requirement of this part is *Flexible support*. It describes the need of a dynamic character of cooperation. Organizations can be geographically distributed and there must be a possibility to join and leave the virtual organization as its state changes over time.

Next point of requirement identification depicts general requirements for CBP modeling language (Table 4.1).

As already mentioned above beside general requirements R4eGove defined also specific requirements for CBP modeling. These requirements contain seven points that are specific to the workflows that are supposed to collaborate with each other.

The first requirement *Keep private information private* describes the need of publishing the relevant information only. It can not be expected that each partner publishes its entire workflow and all contained information. There are two approaches presented. On the one hand it is possible to distinct between public and private processes (public and private views). On the other hand there is an option to generate an ”interface specification” describing the required input, interaction pattern etc. The next requirement is *Specify the interfaces of the partners formally.* It depicts the importance of the comprehensive information accordingly to the interface specification. *Mapping the CBP to executable processes* displays the requirement for modeling language that should be able to transfer the CBP from business level (e.g. business process model) into IT-oriented workflow model on technical level (e.g. execution language). The fourth requirement *Support of the information flow* depicts the importance of information flow particular its ability being represented by the modeling language. Especially a description of the needed input of the partner in order to execute their process parts is necessary. The requirement *Support of organizational units and roles* explains the need of organizational units and roles. Organizational units contain the communication and reporting relationships of different partners involved in CBP. The term ”role” describes a certain type of organizational unit with clearly defined qualifications and skills. *Support of the analysis of the CBP* expects the modeling language to supply the ability for measurement and examination of running and finished collaborative processes. This point allows changing the process model in the modeling phase. Last requirement *Support of semantic annotation* discusses the concept of ontologies. Accordingly to R4eGov, ontology can supply a common set (dictionary) of terminologies, a set of relations between terms and their transformations to private processes/terminologies.

Learnability	In order to be used by the potential users the modelling language has to be easy to learn and to understand.
Visualization	The modelling language provides a graphical notation that helps to understand the model and provides an appropriate visualization of the process.
Extensibility	Extensibility is important in two dimensions: models have to be easy to extend and new features have to be easy to integrate into the language and thus into the models.
Hierarchy	Hierarchy describes a concept of choosing to model and visualize different levels of granularity. There is a need of support to model and visualize the process in different abstraction levels in order to get an overall view of the process or the get very detailed views of parts of the process.
Expressability	Seveeral important featues, properties and specifications of process have to be includet into the model, such as: • Local events and local states and their causal relations (transitions) • Roles and organizational structures • Pre- and post-conditions of events • Objects within the process, e g. documents, e-mails (material and information flow) Besides these features other features might be important, such as access control or temporal aspects. However, if the language is extensible as required above, such other features should be easy to integrate.
Executability	Either the models have to be able to be executed or there has to be existed a transformation component that transforms a model into an executable format and is also able to send back information about transitions to the model such that the dynamic process and its changes can be visualized.
Analyzability	Analyzability includes the verification of certain properties of the process, such as the possibility of correct termination and the absence of dead-locks and undesired states as well as the possibility of performance measuring and optimization.

Table 4.1.: General requirements for CBP modeling

4.1.2. Requirements on Business Process Management Systems

The chapter 2 provided an overview of the theoretical background that will be used for the identification of relevant requirements here. The relevance of these requirements will be investigated according to the deployment to Business Process Management (BPM)-Systems. The requirements listed in 4.1.1 depict the view of CBP and consider BPM-Systems only as a part within the whole concept. It means that BPM-Systems can not be investigated using all those defined requirements. This section identifies corresponding criteria for the further evaluation of BPM-Systems.

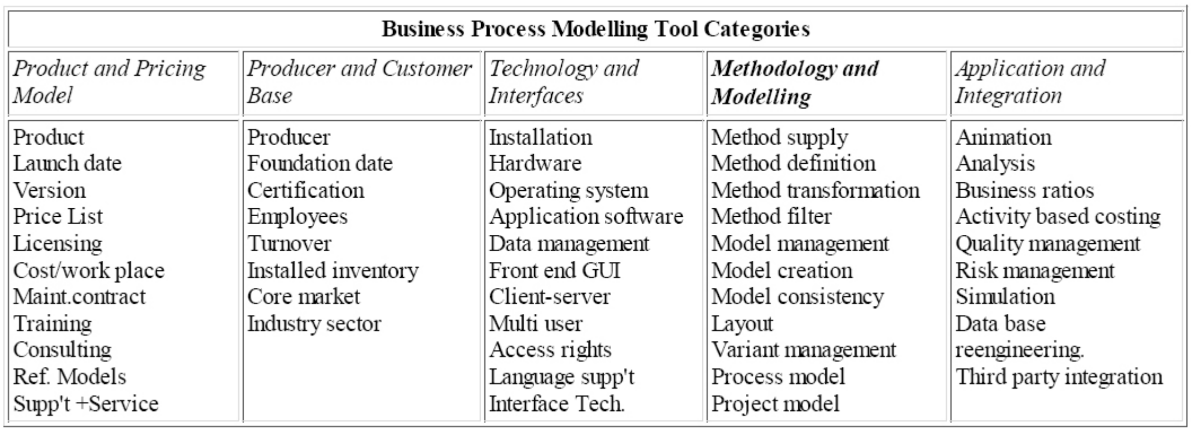
Gartner Research provides a research paper which includes selection criteria details for BPM suites. In this research were identified and presented 10 major areas of functionality as selection criteria [Hill et al., 2006b]:

- Human task support: Executing human-focused process steps
- Business process/policy modeling and simulation environment
- Pre-built frameworks, models, flows, rules and services
- Human interface support and content management
- Collaboration anywhere support
- System task and integration support
- Business Activity Monitoring (BAM)
- Runtime simulation, optimization and predictive modeling
- Business policy/rule management support
- Real-time agility infrastructure support

These criteria partly correspond with R4eGov requirements on CBP. One of the described requirements is the support of collaboration anywhere. SOA is an architecture in which systems are modular and their components are distributable and is able to provide this characteristic. This attributes can be realized using the technology of Web Services. The usage of Web Services within the process design and their creation will be investigated during the evaluation. Both SOA and Web Services were introduced in detail in 2.2 and 2.3.

Furthermore, R4eGov identified languages according to the requirements for CBP modeling. This identification considers the following languages for process design: Petri Nets, Unified Modeling Language (UML), Event-Driven Process Chain (EPC), and Business Process Management Notation (BPMN). Moreover, XML Process Definition Language (XPDL) and Business Process Execution Language (BPEL) were added to the selection as a possible executable formats that could be generated using a modeling language. These BPM standards are introduced and discussed in detail in 2.4. Although, BPM standards were identified and cover the requirements for CBP modeling it is important to investigate the import and export ability of the BPM-systems. This feature should allow import/export of a given or required format.

Moreover, there is a categorized set of requirements that was identified by Nüttgens in [Nüttgens, 2002]. This work is called: *"Evaluation Framework for Business Process Modeling Tools"* and it consists of five main categories (see Table 4.2). These main categories are further operationalized through multi-level sub-categories and include about 350 attributes at the lowest level. The framework was developed to be used for evaluation of modeling tools for BPM. It is important to mention that the criteria cover only modeling tools for Business Process Management and not the BPM-systems.



Business Process Modelling Tool Categories				
Product and Pricing Model	*Producer and Customer Base*	*Technology and Interfaces*	***Methodology and Modelling***	*Application and Integration*
Product	Producer	Installation	Method supply	Animation
Launch date	Foundation date	Hardware	Method definition	Analysis
Version	Certification	Operating system	Method transformation	Business ratios
Price List	Employees	Application software	Method filter	Activity based costing
Licensing	Turnover	Data management	Model management	Quality management
Cost/work place	Installed inventory	Front end GUI	Model creation	Risk management
Maint.contract	Core market	Client-server	Model consistency	Simulation
Training	Industry sector	Multi user	Layout	Data base reengineering.
Consulting		Access rights	Variant management	Third party integration
Ref. Models		Language supp't	Process model	
Supp't +Service		Interface Tech.	Project model	

Table 4.2.: Evaluation Framework for Business Process Modelling Tools

One of the requirements was not mentioned in Chapter 2, but is included in the topic of this thesis. It deals with licensing and in particular the usage of open source software. This attribute will be considered for identification of open source BPM-Systems in 4.4.

Other requirements are not listed in this section because they are not relevant for the evaluation or beyond the scope of this thesis. The set of identified requirements will be gathered and categorized in a catalogue of criteria (see 4.3) that builds a basis for the evaluation.

4.2. Scenario Description

4.2.1. European Arrest Warrant

Two agencies - Europol and Eurojust - have been set up to help the EU member states co-operate in the fight against cross-border organized crime. Co-operation in criminal matters is a subject dealt with in the "third pillar" of the EU (Title VI of the Treaty of the European Union). Eurojust stands for the European Judicial Cooperation Unit whereas Europol refers to the European Police Office.

Europol and Eurojust carry out very specific tasks in the context of dialogue, mutual assistance, joint efforts and co-operation between the police, customs, immigration services and justice departments of the EU member states.

Europol became fully operational in 1999 whereas Eurojust was set up by a Council Decision in 2002. Both agencies are based in The Hague, The Netherlands. Only recently (June 2007), a secure IT connection between Eurojust and Europol has been established to allow efficient collaboration between their respective IT systems. The detailed technical requirements of these systems have been investigated and are presented in this document [Huys, 2007].

One of the objectives of the European Union is to move towards an area of freedom, security and justice using European Arrest Warrant (EAW). After the attacks on New York and Washington, the enactment of the European Arrest Warrant and the surrender procedures between the Member States became a top priority for the EU's political leaders [Blekxtoon and van Ballegooij, 2005].

The European Union is replacing lengthy extradition procedures with a new efficient way of bringing back suspected criminals who have absconded abroad and for people convicted of a serious crime who have fled the country. EAW will enable such people to be returned within a reasonable time to attend their trial or to be put in prison according to their sentence.

The Europa Glossary[1] defines the EAW as: ”The European arrest warrant is a judicial decision issued by a Member State with a view to the arrest and surrender by another Member State of a person being sought for a criminal prosecution or a custodial sentence.”

4.2.2. Scenario

The EPC-model bellow (see Figure 4.1) describes the process used for the further evaluation. The main reason for choosing EPC as default modeling language is the widespread acceptance among business analysts for this standard. This process will provide an overview in terms of process design and supported standards within the evaluation. It means that the illustrated process will be modeled in the selected BPM-Systems using their supplied environment and modeling languages.

The process describes a request for a suspected person. The model contains three parties depicted by the organizational units. The first unit represents an employee of an administration who requires the information about a particular suspected person (e.g. personal details, warrants etc.). The employee starts the request (e.g. using a web form). This request goes to the Organization A that hosts the processes and it can represent e.g. a BPM-System. The Organization A invokes a service (e.g. a web service) from Organization B that delivers the requested information about the suspected person. The response goes back to the employee who started the request.

4.3. Creation of the Catalogue of Criteria

4.3.1. Categorization of Requirements

This section deals with a meaningful categorization and structuring of criteria that have been leaded from the identified requirements (see 4.1). In 4.1.2 existing approaches were introduced for the BPM-Systems evaluation. Gartner Research provides 10 major areas of functionality as selection criteria [Hill et al., 2006b]. Furthermore, Nüttgens identified

[1] http://europa.eu/scadplus/glossary/arrest_warrant_en.htm

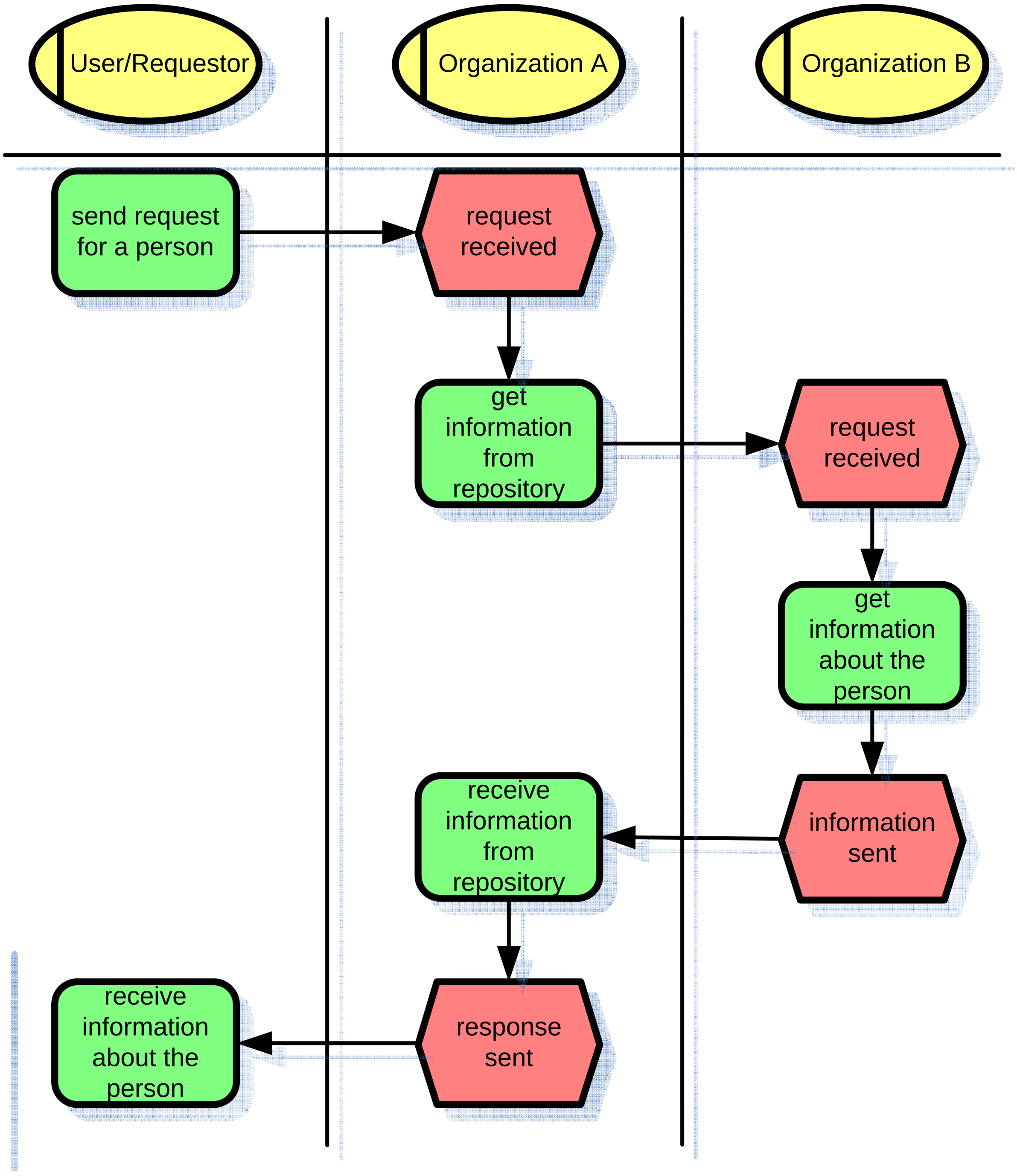

User/Requestor
Organization A
Organization B
send request for a person
request received
get information from repository
request received
get information about the person
information sent
receive information from repository
response sent
receive information about the person

Figure 4.1.: The Scenario (modeled with EPC)

an *Evaluation Framework for Business Process Modeling Tools* [Nüttgens, 2002]. Both approaches are useful for the creation of a catalogue of criteria but categorization does not cover the need of this thesis in the required way.

The catalogue of criteria will be built on the BPM-lifecycle. This categorization appears as a suitable way to structure identified requirements and thus to investigate the described research question (see 1.1).

Thus, the catalogue of criteria distinguishes four categories:

- Process design
- System configuration
- Process enactment
- Diagnosis

The first category, *process design*, contains the criterion "modeling languages". This part deals with the investigation of the language support which is based on identified and selected BPM standards in 2.4. It means that the BPM-systems will be checked for Petri Nets, UML, EPC, BPMN, XPDL, and BPEL as a process design language.

The category *system configuration* contains the criteria "SOA support" and "import- and exportability". The first of these two criteria includes to check if web services can be used in terms of orchestration with the own engine. This part considers the usage of web services description language. The second criterion investigates on import and export abilities of the selected BPM standards. The more BPM standards can be imported or exported, the more flexible and interoperable the BPM-system might be.

The third category, *process enactment*, investigates the execution language of the BPM-system. There are two languages considered and checked: BPEL and XPDL. Also it is important to mention that out of these two standards, only BPEL was developed for execution (compare 2.4.6 and 2.4.5) and is seen as an execution standard.

The last category of the BPM lifecycle, *diagnosis* will not be considered in this thesis. This phase is beyond the scope of the evaluation that mainly focuses on the process design and its execution. The main points of this part would be Business Process Analysis (BPA) and is one of the emerging areas: BAM.

Additionally, there are three other criteria that will be depicted: *support*, *system requirements*, and *licensing*. These criteria are not mainly related to the research question, but they will provide a better overview of the investigated products. The first criterion "support" contains the overview of the provided hotline, existing email, FAQs, and download-area. The next criterion describes the system requirements; it contains the information about operating system and hardware requirements. And the last criterion describes the license information of the investigated product.

4.3.2. Specification of Comparison

The section 3.2 introduced the approach and methodology of the evaluation. The categorization of requirements occurred in 4.3.1. It is important to provide a basis that makes the systems comparable and thus allows to depict all investigated attributes. ISO/IEC 14598 standard provides the guidance for the evaluation and it emphasizes on the need for software quality characteristics and metrics that are defined in ISO/IEC 9126 [Jung et al., 2004]. This standard offers a quality model and evaluation methodology. ISO 9126 contains six distinct quality characteristics between external and internal quality [Hörbst et al., 2005]: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. These characteristics cover the attributes for software quality but they do not reflect the objective for evaluation of BPM-systems in context of R4eGov.

The categories used in section 4.3.1 will be applied for the comparison. These categories and their including criteria represent the requirements according to BPM-systems in the context of R4eGov. The criteria will not be measured as proposed in ISO/IEC 9126; however, they will be investigated upon to check if they exist and then commented and evaluated. The investigation in is case does not require the usage of measurement and metrics. The focus of the evaluation is to discover which of the selected standards are supported.

Thus, the following tables depict the categories within the criteria the way they will be applied to each of the BPM-system:

- Process Design (see Table A.1)
- System Configuration (see Table A.2)

- Process Enactment (see Table A.3)
- Additional Criteria (see Table A.4)

4.4. Identification of Open Source Business Process Management Systems

This section deals with the identification of open source BPM-systems. There are various tendencies for the usage of “free” software in the public sector. In [Georgiev et al., 2004] the authors discuss important points in terms of general understanding for open source software including software structures and license models. Moreover they depict strengths and weaknesses as well as chances and risks related to the adoption of this software. Also the European Commission is working on encouraging good practices in the use of open source software in public administrations providing e.g. ”Guideline for public administrations on partnering with free software developers” [Ghosh et al., 2004].

One of the main characteristics of open source software is licensing. To be considered as ”Free / Open Source”, a license must comply to a series of conditions that will basically grant four freedoms [Dusollier et al., 2004]:

1. Run the program, for any users or purpose (e.g. commercial or not);
2. Access to source code to study how it works, and adapt it according to any need;
3. Freedom to redistribute copies;
4. Freedom to improve the program, and release improvements if whished.

The identification of open source BPM-systems is partly based on the characteristics of mature open source software defined in [OpenBRR.org, 2005]. This list contains 25 characteristics that should describe mature open source software. The following points have been considered while selection of BPM-systems:

1. The software is backed by a foundation, a corporation, or a strong community.
2. The project has existed at least 1 year.

3. The project's license is acknowledged by the Open Source Initiative[2].
4. Well-known large-scale deployments (e.g. Wikipedia for mediawiki).
5. Ported across multiple platforms (Linux, Windows, Solaris, and Mac).

Considering the above depicted characteristics the selection accordingly to the BPM-systems was made on: Italio | BPMS, NetBeans IDE, and jBPM from JBoss. The following sections introduce the several products in context of the considered characteristics adopted from [OpenBRR.org, 2005].

4.4.1. Intalio BPMS

Intalio[3] was started in 1999 as a pure Open Source venture. The company profile is described as follows:

> ”*Intalio is the leading vendor of Open Source BPM & SOA software. The Intalio Business Process Platform empowers organizations of all sizes to develop process driven applications faster, better, and cheaper.*”

Intalio is Open Source and has already more than seven years of research and development experience. The products are built on Eclipse, which provides an Integrated Development Environment (IDE) allowing the usage of existing components that have been developed by third-party vendors. Furthermore Intalio provides a Designer that offers the environment to create a process model and also a Server that handles the execution of processes. Moreover, Intalio promises the portability across multiple platforms.

The investigation focuses on Intalio | BPMS that consists of Inalio | Designer 5.2 and Intalio | Server 5.2. Further information, forums, product downloads, and other services can be found on the official BPMS Intalio website[4].

[2] http://www.opensource.org

[3] http://www.intalio.com

[4] http://bpms.intalio.com

4.4.2. NetBeans IDE

NetBeans[5] was started as a student project in the Czech Republic, in 1996. Originally NetBeans was called Xelfi, because the students wanted to write a Delphi-like Java IDE in Java. The original plan for the business was to develop network-enabled JavaBeans components, which lead to the name *NetBeans*. In October 1999, Sun did officially acquire NetBeans. Less than six months later, the decision was made that NetBeans would be open sourced, and NetBeans thus became the first Open Source project sponsored by Sun. Also NetBeans supplies portability across multiple platforms.

The product that will be observed is the NetBeans IDE 6.1 that contains different features. The feature that this work will pay attention to is SOA including several tools (Web Services BPEL Designer etc.) These tools should allow to create and manage web services, to create and modify a process, and to execute and test it.

Further information of NetBeans IDE and the containing features can be found on the official website or other resources (like e.g. the book of Adam Myatt [Myatt, 2007]).

4.4.3. JBoss jBPM

JBoss[6] was started as a project in 1999. JBoss Group, LLC was incorporated in 2001 in Atlanta, Georgia. JBoss became a corporation under the name JBoss, Inc. in 2004 and counted as the global leader in open source middleware. In 2006 JBoss was bought by Red Hat.

The company describes itself as a not any ordinary open source project. Jboss is seen as one of the few open source projects that turned into commercial success stories without betraying their open source roots. Although JBoss is freely available for any purpose, it is backed by a real company that provides support and training for those who need the reassurance of having strong vendor backing [Richards and Griffith, 2005].

JBoss provides many solutions in the area of application server. However, there are also projects in the area of integration, such as jBPM project, that are particularly interesting for this work. In fact, jBPM is a platform for executable process languages ranging from

[5] http://www.netbeans.org

[6] http://www.jboss.org

BPM over workflow to service orchestration. This platform promises not only to focus on one particular language but also to support multiple languages processes. All of the jBPM sub projects are supposed to work in the supplied environment.

The information of JBoss and jBPM can be found on the official website[7].

[7] http://www.jboss.org/jbossjbpm/

5. Evaluation of Open Source Business Process Management Systems

5.1. Evaluation of Inalio | BPMS

5.1.1. Process Design

Intalio | Designer is built on Eclipse and it deals with BPMN Design, Web Service integration, and data mapping. As mentioned earlier Intalio | Designer 5.2 uses BPMN for the process design.

The first step to create a business process is to create a *Business Process Project*. This *project* contains the whole information (diagrams, web service description, documentation etc.) which is related to the business process. Intalio | Designer provides a process explorer view where all the business process projects are listed.

Once the project has been created, the new diagram can be added. The designer provides a palette with all the BPMN shapes divided in basic BPMN shapes, start events, intermediary events, gateway shapes, and artifacts (see Figure 5.1). Intalio supports the latest version of BPMN – BPMN 2.0. No other modeling language from the catalogue of criteria is supported, which probably considers BPMN as a standard notation for business analysts. The following Figure 5.2 illustrates the scenario process. The process is created with Intalio | Designer using the supplied BPMN syntax. The pools named User/Requestor, Organization A, and Organizatio B represent interacting instances. The main process takes place in the *Organization A*, which is depicted by a start event, an activity, and an end event. The task *get information from repository* invokes the web service from Organization B. A web services can be seamless integrated into the business process (see

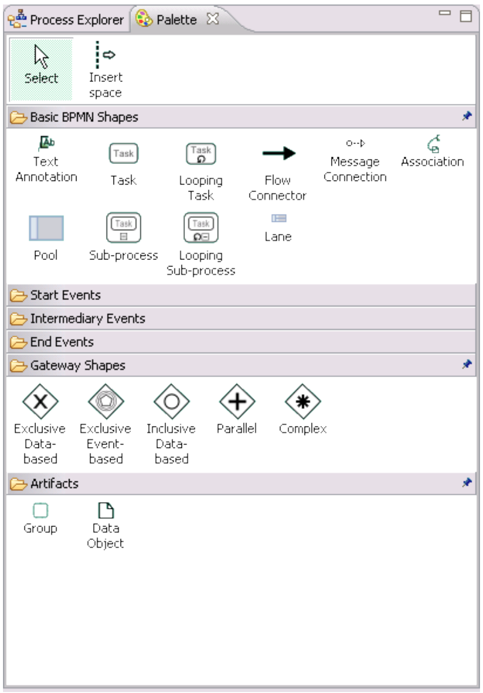


Figure 5.1.: Intalio BPMN Palette

5.1.2) ant thus it allows to create real processes choosing from the set of existing services the required ones.

The files of the created model are saved in a folder with an extension ”bpm”, as for instance the files of this diagram are located in the folder ”EAW-Diagram.bpm”. The sructural information of the model is placed in the file ”modeler.bpmn”, which describes the process using the XML syntax.

The use of BPMN within Intalio is intuitive and it allows getting very fast familiar with the tool. However, it support only one of the selected BPM standards.

5.1.2. System Configuration

SOA Support

Intalio | Designer supports the web service involvement into the process design. It integrates a full Web Services Description Language (WSDL) Visual Browser that allows the introspection of WSDL documents. Moreover there is an option to import the web

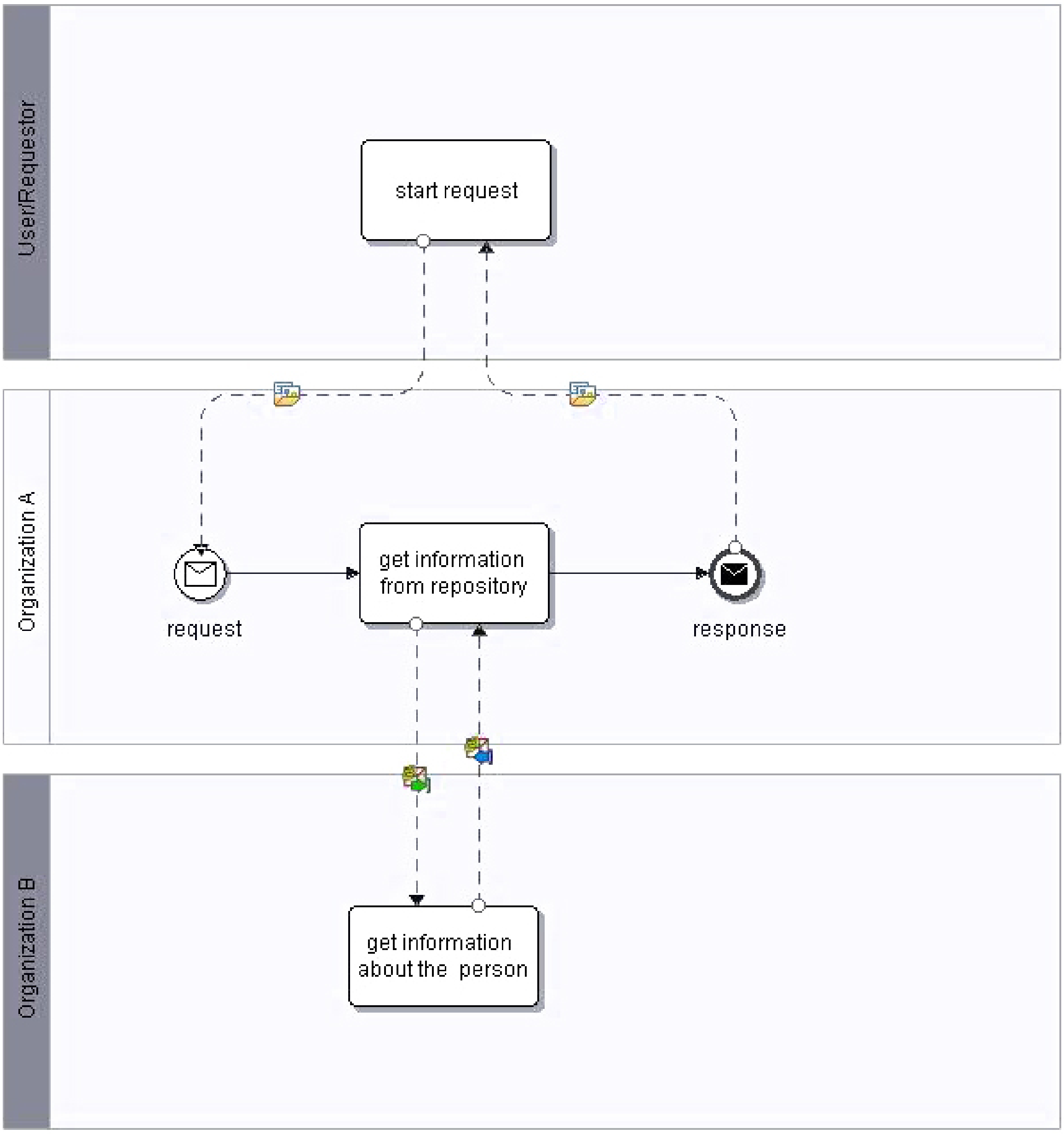
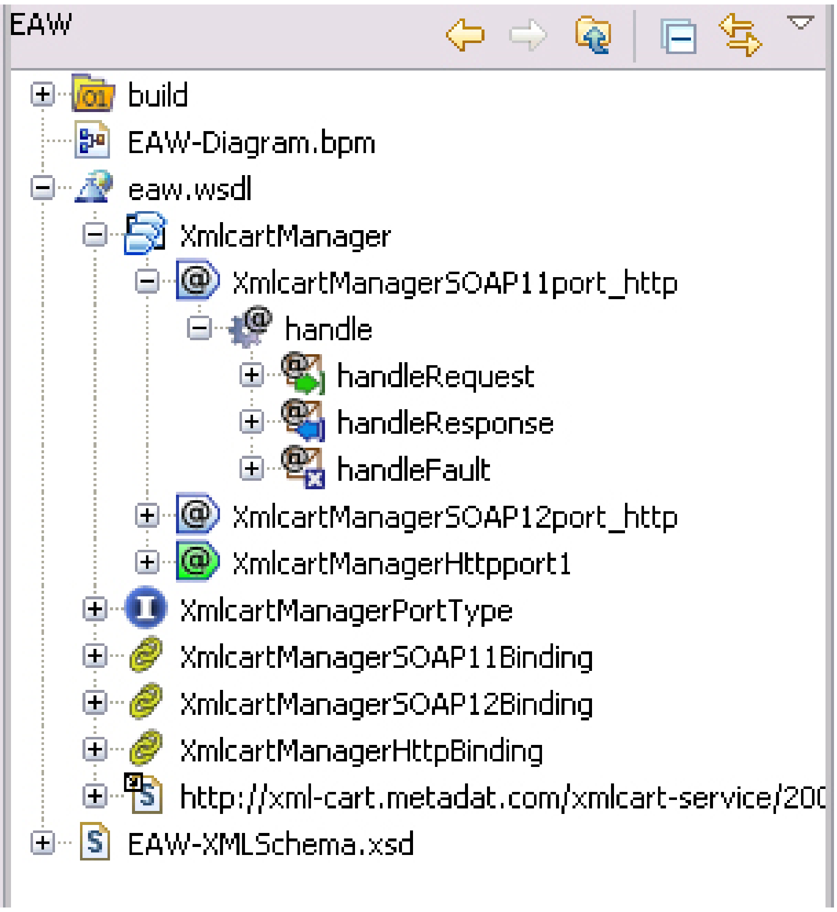


Figure 5.2.: Intalio - Modeling the Scenario Process



```
EAW
build
EAW-Diagram.bpm
eaw.wsdl
  XmlcartManager
    XmlcartManagerSOAP11port_http
      handle
        handleRequest
        handleResponse
        handleFault
    XmlcartManagerSOAP12port_http
    XmlcartManagerHttpport1
  XmlcartManagerPortType
  XmlcartManagerSOAP11Binding
  XmlcartManagerSOAP12Binding
  XmlcartManagerHttpBinding
  http://xml-cart.metadat.com/xmlcart-service/20(
EAW-XMLSchema.xsd
```

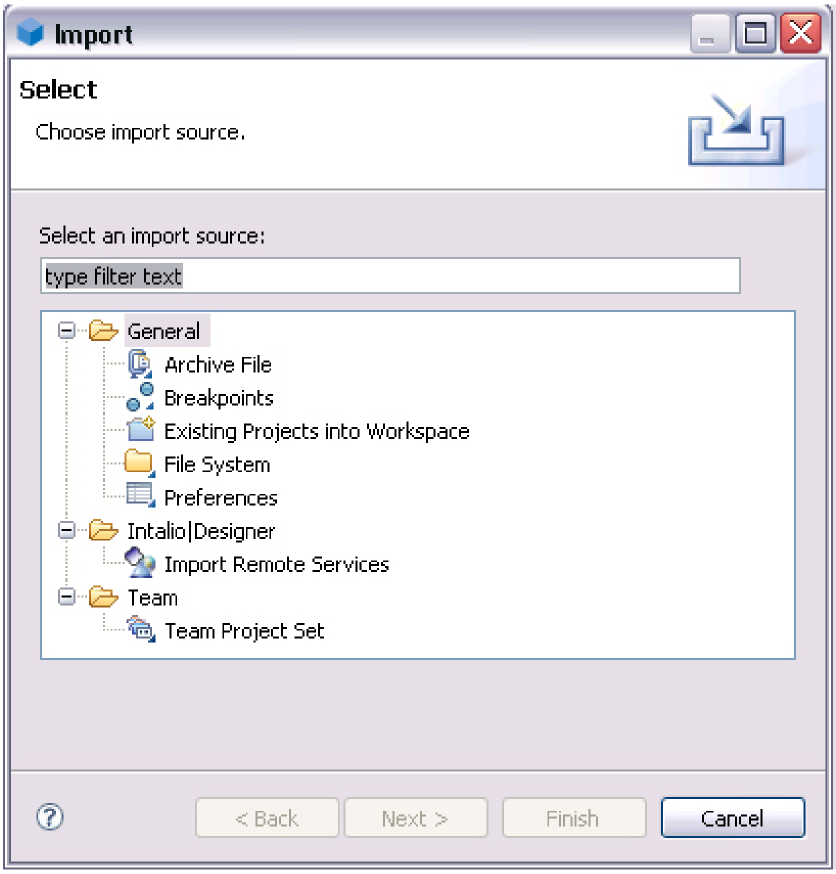
Figure 5.3.: Intalio EAW-WSDL in Process Explorer

service using remote resources. For this the URL of the web service is required only and it allows to synchronize the web service description every time the service changes.

The tool does not provide any option to publish a web service, and therefore other tools are required (e.g. Axis2[1] can be used to create a web service). But it is possible to create own WSDL-file and thus to specify a web service itself and its corresponding operations.

Intalio | Designer uses WSDL for the web service description. It allows expanding each WSDL document from the process explorer (see Figure 5.3) - to access the different services defined in the WSDL document - to the operations exposed for each service.

[1] Apache Axis2 is the core engine for Web services.



Import

Select

Choose import source.

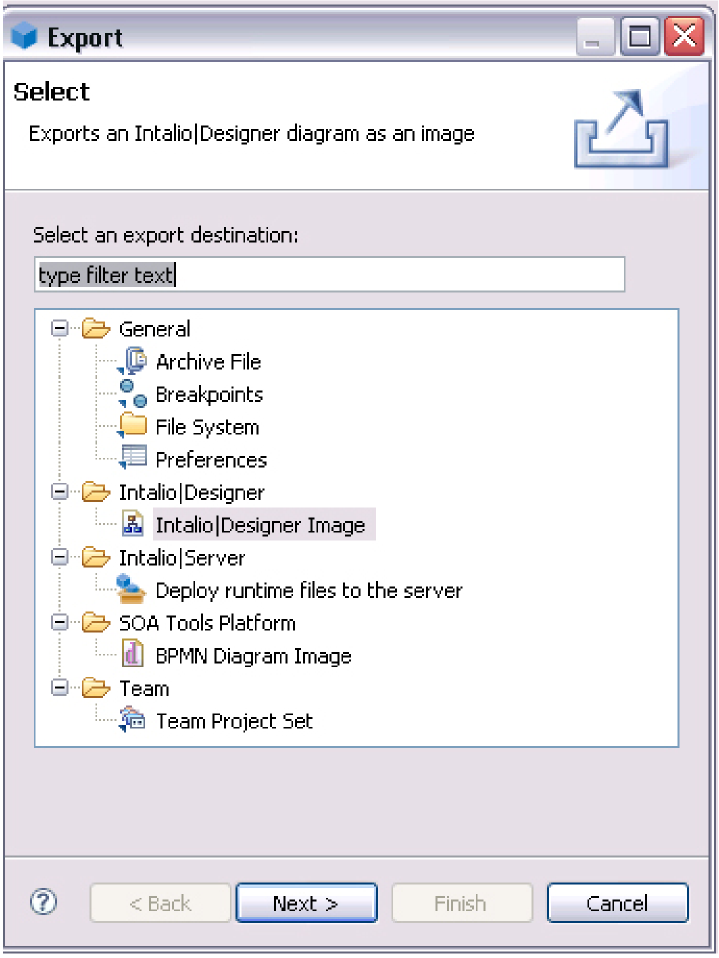
Select an import source:

type filter text

Figure 5.4.: Intalio Import Dialog

Import

Intalio | Designer provides the option to import three types of resources (see Figure 5.4): General, Intalio | Designer, and Team. The section *General* contains *Archive File*, *Breakpoints*, *Existing Projects into Workspace*, *File System*, and *Preferences*. The second section contains the option *Import Remote Service* that allows using a URL to integrate a web service into the process design. The last section contains an option to import a *Team Project Set*. Intalio does not provide the possibility to import any of the defined criteria as e.g. EPC, BPMN, and BPEL etc. The import feature focuses exclusively on own existing projects.



Export

Select

Exports an Intalio|Designer diagram as an image

Select an export destination:

type filter text

- General
  - Archive File
  - Breakpoints
  - File System
  - Preferences
- Intalio|Designer
  - Intalio|Designer Image
- Intalio|Server
  - Deploy runtime files to the server
- SOA Tools Platform
  - BPMN Diagram Image
- Team
  - Team Project Set

Figure 5.5.: Intalio Export Dialog

Export

Intalio | Designer provides the option to export five kinds of resources (see Figure 5.5): General, Intalio | Designer, Intalio | Server, SOA Tools Platform, and Team. The sections *General* and *Team* are consistent with corresponding sections in the Import part.

The tool provides within the Intalio | Designer section the feature to export the process diagram as an image. It supports several image formats as GIF, BMP, JPG, SVG, and PNG. The next section, Intalio | Server, contains the feature to deploy runtime files to the server. That feature deals with process enactment, but it means also that the executive files can be extract from the workspace. The generated files within the configuration phase include also a BPEL file that can be deployed to the Intaio | Server. Intalio supports, since the Intalio | BPMS 5.0, the BPEL 2.0 version. The last section, SOA Tools Plattform, allows the user to export the business process diagram as an image using the same image formats as in Intalio | Designer section.

5.1.3. Process Enactment

After using the described feature Deploy runtime files to the server, the designed process can be executed. The execution occurs mainly deploying the BPEL file and other required documents as e.g. WSDL. The standard that is used for the execution is BPEL 2.0. The Intalio | Server can be accessed via BPMS-Console[2] where all executable and uploaded processes are listed. From that list a particular process can be chosen and executed. XPDL and other execution languages are not supported.

5.1.4. Additional Criteria

Support

Hotline

Intalio website provides contact information such as phone and fax numbers. However, these numbers are not part of the expected hotline. The international hotline or the hotline itself can not be found on the official website. The companies that integrate Intalio products might provide this kind of service, but the information is not mentioned by Intalio.

Email

There is an option to send an email to Intalio using an email form in the contact section. Furthermore there are several addresses that are displayed for different purposes (e.g. sales or training)

FAQs

Intalio provides a solid Forum environment including different sections e.g. FAQ, forums in several languages, documentation etc. Intalio registered area contains 31.825 (accessed

[2] E.g. http://localhost:8080/bpms-console/

on 30.09.2008) users. Such a large amount of users can be very helpful in solving any kind of problems.

Download-Area

The download area provided by Intalio is well structured and offers both Intalio | Server and Intalio | Designer including different platform distributions for download. Some of the items are available only for registered users. The download area displays the current releases, but the past releases can not be found in these pages.

System Requirements

Operating System

The Intalio BPM-suit runs on following operating systems: Microsoft Windows XP or Microsoft Windows 2000 or Microsoft Windows 2003 Server or Linux or Mac OS X. The test environment for the investigated system has been Microsoft Windows XP Pro SP2.

Hardware

The website provides information for the required memory which is, as a minimum, 512 MB. There are no notes for the recommended capacity. It is also the case for the required disk space which is, as a minimum, 200 MB (but there are no further recommendations). No information about the CPU is available at all.

Licensing

Intalio's Open Source business model is based on layering and dual licensing. Intalio | BPMS is distributed in three different editions:

- Open Source Edition under Mozilla Public License (MPL)
- Community Edition
- Enterprise Edition

Open Source Edition which is published under MPL is supposed to cover approximately 95% of the code used for the Community Edition and the Enterprise Edition. This Open Source Edition is designed to be deployed on top of the Apache Geronimo J2EE application server, and deployed alongside the MySQL database. The Community Edition is prepackaged with IBM WebSphere Application Server Community Edition, and supports deployments alongside the MySQL database. IBM WebSphere Application Server is used for building and managing Java applications. The Enterprise Edition[3] can be deployed on other application servers and databases, and provides advanced features for clustering and transaction processing.

The description of the licenses provided by Intalio shows that Open Source Edition does not include the complete code. Furthermore the deployment of the Community Edition includes the usage of:

- IBM WebSphere Application Server
- MySQL

It means that there is no option to use another database or application server if needed. The usage of other software with Intalio | BPMS is possible only with the Enterprise Edition that is already not free.

5.1.5. Overall Summary

For process design, Intalio supports BPMN as a modeling standard. It allows to easily create processes in the graphical way. The BPMN palette provides a good overview of the elements corresponding to the current BPMN 2.0 specification. However, when using Intalio | Designer, there is only one modeling language available for process design. Except BPMN, no other standards identified in 2.4 are supported. Intalio focuses on BPMN and places it as *the* business process management language.

The scenario process can be designed easily without any particular technical knowledge. The clear advantage of the Intalio | Designer is the ability to model with real services.

It is possible to integrate various web services into the business process. The visual browser allows the introspection of WSDL documents, visualizing the messages, protocols,

[3] http://www.intalio.com/company/open-source/

bindings, and types of required variables. This view allows using web services for the process design as the BPMN elements (e.g. Tasks) from the palette. WSDL depicts a standard for the description of web services and Intalio | Designer makes it convenient to use. According to the publishing of web services, there is no direct support.

Intalio allows importing a remote service. It means that the allocation of the WSDL file does not require to be local. The web service description will be accessed remotely, using a URL. But according to the investigated criteria, it means that there is no congruence. In fact, none of the expected standards can be directly imported into the system. Thus, the export ability allows to export a business process diagram in various graphic formats (e.g. GIF, JPG etc.), but does not provide any option for direct export of BPM standards. It is possible to extract from BPMN generated BPEL code that is in the project workspace. Thereby it is important to consider that the BPEL code is generated after the business process in BPMN is consistent and ready to be deployed.

Intalio provides a runtime for the execution of the business processes based on the generated BPEL code. This code is produced from the before created BPMN model and after a consistency check it can be deployed to the server.

Intalio | BPMS is a powerful tool that allows easily getting familiar and creating first business processes. But the use of the community edition (”free” available) implies the use of IBM WebSphere application server and MySql database, because they are the components of this edition, since the use of other software is not possible in the community edition.

5.2. Evaluation of NetBeans IDE

5.2.1. Process Design

NetBeans IDE provides a BPEL Designer for the creation of business processes. First of all it is necessary to create a new project choosing the SOA category and to select the BPEL Module. After the corresponding project is built the new BPEL document can be created. This document is illustrated with BPEL-Designer that provides two views: Source and Design view. The BPEL diagram (BPEL Design View) is the visual representation of the

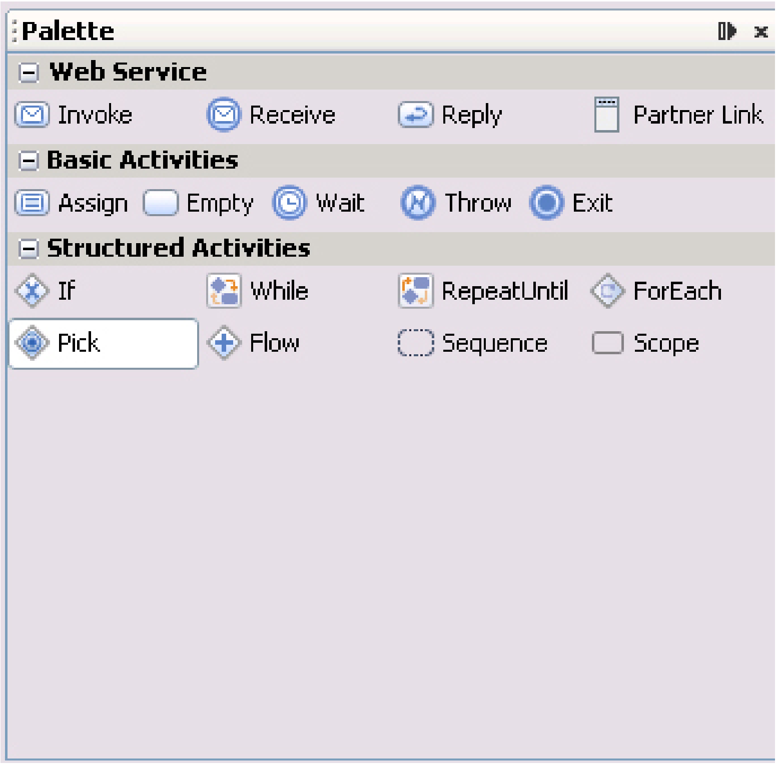


Figure 5.6.: Netbeans BPEL Palette

BPEL Process. On the diagram, various activities can be added and configured. NetBeans IDE supports BPEL 2.0. The BPEL elements are depicted in a palette (see Figure 5.6).

The following Figure 5.7 illustrates the scenario process modeled in BPEL. The diagram depicts three participating instances from the perspective of Organization A. This view can be referred to the fact that BPEL mainly deals with the execution of the processes. It means also that neither the User/Requester nor the Organization B do not play the main role in the process of Organization A. Both of them are inserted as PartnerLinks into the whole process. PartnerLinks depict a connection to the remote service.

The modeling with the NetBeans BPEL-Designer requires a solid technical background and it depicts a challenge for business analysts. The process design with BPEL considers a block structured view. In contrast, such standard as EPC, Petri nets or UML AD are graph oriented and they are easier to use by the people with less technical understanding of BPEL.

NetBeans supports modeling of various UML diagrams, but only for the Java development.

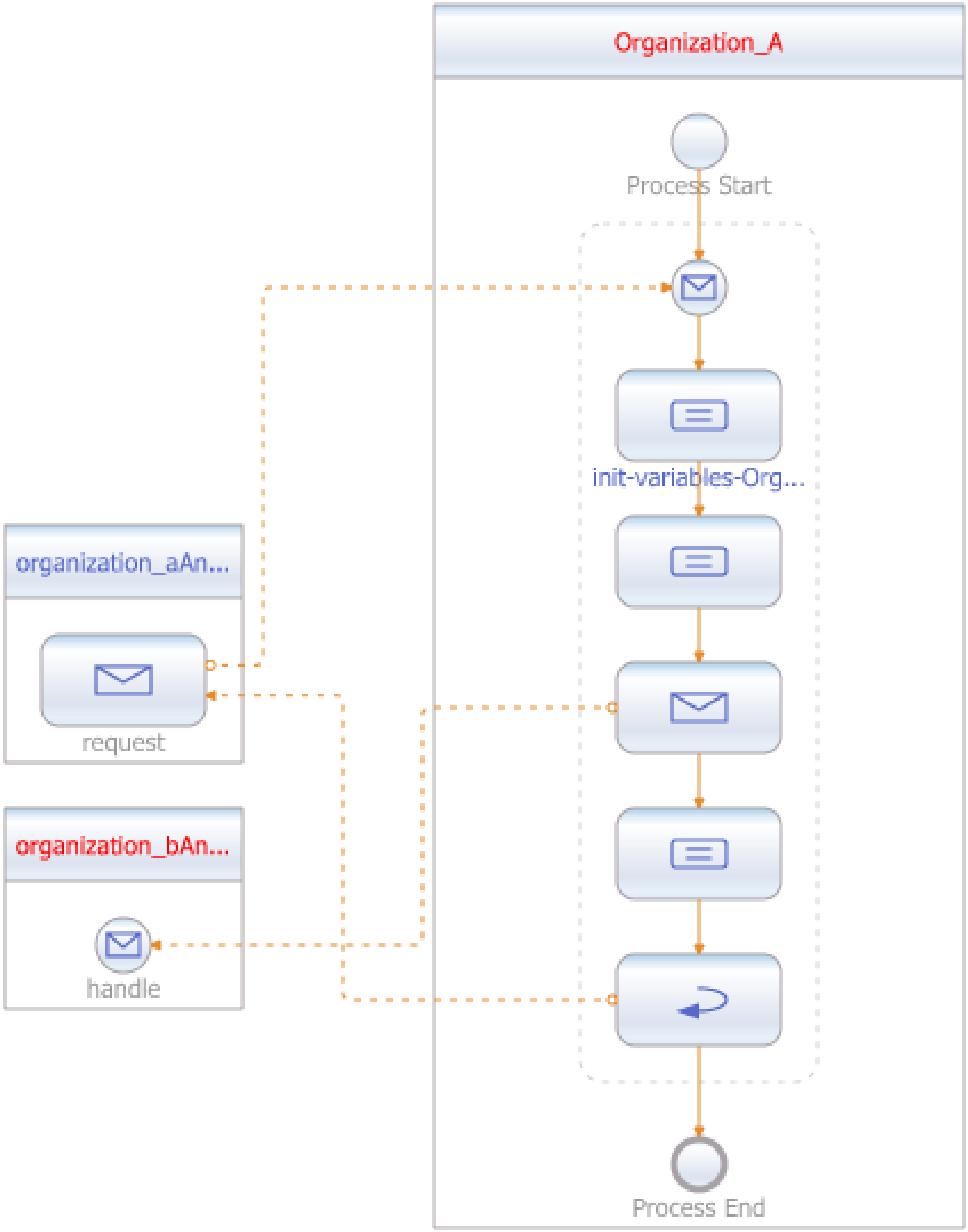


Figure 5.7.: NetBeans - Modeling the Scenario Process

The UML models can not be used within NetBeans for the process design with consideration to be executed. Thus, NetBeans supports only BPEL 2.0 and no other identified BPM standards for the process design.

5.2.2. System Configuration

NetBeans IDE BPEL Designer supports the web service involvement into the process design. It allows inserting the web service directly into the diagram and using it as a part of a BPEL process. The web service is illustrated as a block and can be integrated via messages into the modelled process.

The NetBeans tool provides also the environment for the web service development. And it is possible to build own Web Application in JAVA. The web service description will be generated automatically after the deployment of the application on the server.

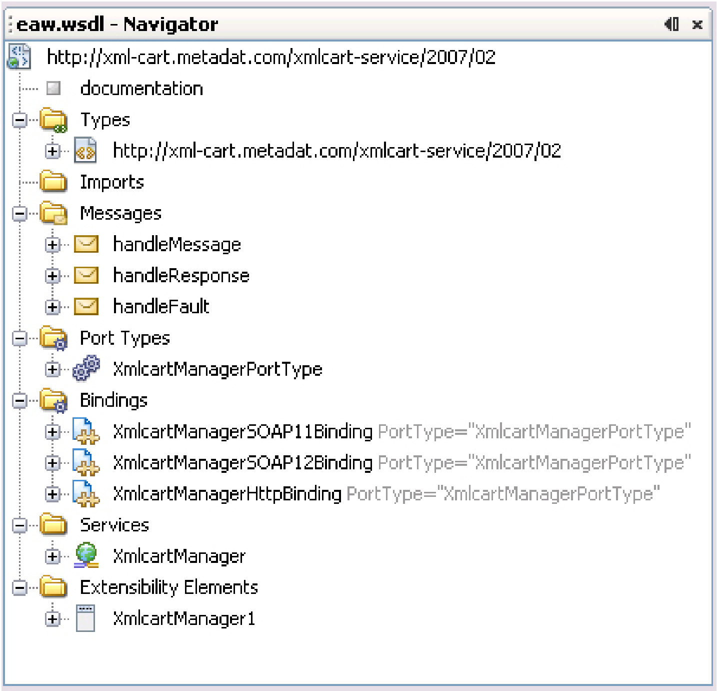
The web service description uses WSDL. On one hand, a WSDL file can be inserted into the project folder and thus can be used for the integration of the web service into the process. On the other hand there is an option to create a new WSDL document and to specify the required fields (e.g. messages and protocols) as needed. The Figure 5.8 depicts the Navigator view of the web service description of the EAW Scenario.

Import

There is no feature provided to support any of the defined criteria. Only one standard that can be included in the project is a BPEL file. This file and the corresponding WSDL can usually be adopted.

Export

Also, there is no option provided by NetBeans IDE to export any of the defined. The way to share a defined business process is manual extracting of the existing BPEL file from the project. There is no other option supplied.



```
eaw.wsdl - Navigator
http://xml-cart.metadat.com/xmlcart-service/2007/02
  documentation
  Types
    http://xml-cart.metadat.com/xmlcart-service/2007/02
  Imports
  Messages
    handleMessage
    handleResponse
    handleFault
  Port Types
    XmlcartManagerPortType
  Bindings
    XmlcartManagerSOAP11Binding PortType="XmlcartManagerPortType"
    XmlcartManagerSOAP12Binding PortType="XmlcartManagerPortType"
    XmlcartManagerHttpBinding PortType="XmlcartManagerPortType"
  Services
    XmlcartManager
  Extensibility Elements
    XmlcartManager1
```

Figure 5.8.: NetBeans Navigator EAW-WSDL

5.2.3. Process Enactment

In the design phase, NetBeans focuses already on further executions. This consideration means that the process design uses BPEL standard for modeling. However, BPEL was developed mainly for execution. The BPEL runtime engine is integrated with the GlassFish[4] application server and supports BPEL 2.0 specification.

5.2.4. Additional Criteria

Support

Hotline

NetBeans does not provide any phone numbers. The contact section focuses on providing different links of community forums.

[4] GlassFish is an open source application server project led by Sun Microsystems for the Java EE platform.

Email

The email contact is possible only using the contact form. But it can not be used for any Java questions or use of NetBeans. Besides that, there is no further option to get in touch with NetBeans.

FAQs

NetBeans supplies a solid support in form of FAQs, forums, learning trails etc. Moreover the company subdivides the FAQ into User FAQs and Developer FAQs. The information is well structured and does not require any registration to access. The documentation of the product is provided in a very well organized way.

Download-Area

The download area of NetBeans is well structured and provides a download for different platforms. It is possible to download the required bundles choosing suitable packs. Moreover the past releases of the IDE are all accessible via the archive and can be downloaded as well.

System Requirements

Operating System

The NetBeans IDE runs on the following operating systems[5]: Microsoft Windows (2000 Pro SP4, XP Pro SP2, Vista), Linux (Ubuntu 7.x, Red Hat EL 4) or Solaris OS (10) or Macintosh (10.4.9 Intel & PPC). The test environment for the investigated system has been Microsoft Windows XP Pro SP2.

Hardware

The minimum requirement for the CPU is 800 MHz and recommended frequency is 2.6 GHz. The minimum memory required is 512 MB and the recommended capacity is 1 GB. Also the there is some information about the disk space (minimum 650 MB, with 1 GB of recommended disk space).

[5] http://www.netbeans.org/community/releases/index.html

Licensing

The NetBeans IDE 5.5 code is available under a dual license consisting of the Common Development and Distribution License (CDDL) and the GNU General Public License (GPL). The GPL license provides an additional option to vendors that are unable to work with NetBeans under the CDDL license and makes it even more Linux friendly. The prior versions of NetBeans were made available under the terms of the Sun Public License (SPL). Both licenses CDDL and SPL are based on the Mozilla Public License (MPL). NetBeans describes the CDDL as more reusable license that was written to be more readable. Furthermore CDDL and MPL contain the same attributes:

- Sources licensed under them may be reused in commercial products
- Changes made directly in the sources - bug fixes or enhancements - must be contributed back to the netbeans.org, but new source files written with links to NetBeans code do not need to be.

5.2.5. Overall Summary

NetBeans IDE supports only one of the defined criteria. BPEL is here considered as process design language. This fact makes the transformation from a process modeling language to an executable language dispensable, because BPEL is already an execution language. BPEL was developed as an execution standard and is bock based. In contrast, all the business process standards that are considered for modeling are graph based (e.g. Petri nets, EPC, BPMN etc.). It can be a problem for business analysts, because they are used to apply graph based modeling languages for business processes. The modeling of the scenario process depicted exactly these issues. NetBeans BPEL-Designer does not provide an easy high-level view of the process and it requires a solid technical knowledge designing the processes or integrating web services.

The web services can be integrated into the process using the BPEL Designer mentioned earlier. It allows to simultaneously design business processes and choose the required web services. Moreover, the web services must be described in WSDL. Navigator provides well structured overview of the WSDL file illustrating all required information for the web service integration into the business process.

The tool does not provide particular features to import any kind of defined standards. A BPEL process could be adopted only including it with corresponding WSDL into the project directory. In the same way the BPEL process can be exported only after the process design is finished.

NetBeans pass on different levels of design and execution. BPEL is considered as the language for both phases, on one hand, as a design, and on the other hand as an execution language. In fact this should provide an advantage, because it does not require any transformation from a modeling language to an execution language using only BPEL. But it can also constitute a big challenge for business analyst or people with less technical background.

5.3. Evaluation of JBoss jBPM

5.3.1. Process Design

JBoss jBPM supplies the Graphical Process Designer (GPD) for the process design. GPD is a set of plug-ins for Eclipse that makes it possible to edit graph based languages. The process designer can represent and edit graph based executable languages such as jPDL and Seam[6] pageflow. The elements of jPDL are illustrated in Figure 5.9.

The diagram illustrated in Figure 5.10 shows the scenario process. The process is built using the supplied jPDL process language. The language provides various simple elements tat can be used quickly to create business process. The node "Node" can carry different semantic. It can represent an event or a task. There is no visual option to define separate organizations or participants within the graph. This specification can be done specifying the nodes separately. The purpose of jPDL is to provide a language for business analysts, but with this grade of abstraction it can also bring the challenges for developers, who have to implement the created by jPDL processes.

Although, jBPM provides support for design supplying two languages, but they do not exactly correspond with the defined criteria of BPM standards. Accordingly design there

[6] Seam is a powerful open source development platform for building rich Internet applications in Java. Seam integrates technologies such as BPM into a unified full-stack solution, complete with sophisticated tooling. (`http://www.seamframework.org/`)

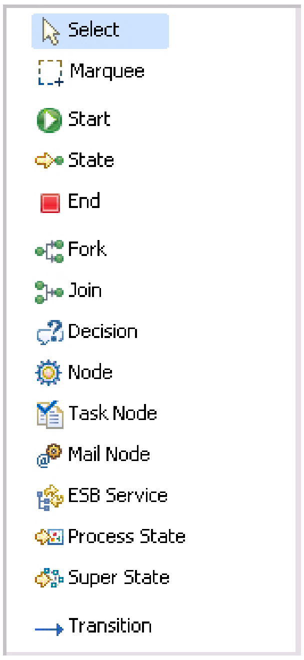


Figure 5.9.: JBoss jBPM jPDL Palette

is no support for the identified and required standards. However, jBPM recommends the use of the Eclipse BPEL Designer that can be integrated into BPM-lifecycle within jBPM environment. But Eclipse BPEL Designer is a project of Eclipse and not of JBoss. Moreover, the consideration of use graph based languages could include the adoption of Petri nets, EPC, UML AD, and BPMN. All these mentioned standards are graph based. Thus, this adoption has to be investigated and is beyond the scope of this work.

5.3.2. System Configuration

jBPM GPD does not consider graphical integration of web services into the process design. The user focuses on the designing the process using the supplied elements by the jPDL. The system provides an option to integrate web services into a process, but it does not occur graphically.

A web service can be implemented and published in Java using the Eclipse JBoss IDE. That requires the installation of additional modules from JBoss application server[7].

[7] http://www.jboss.org/jbossas

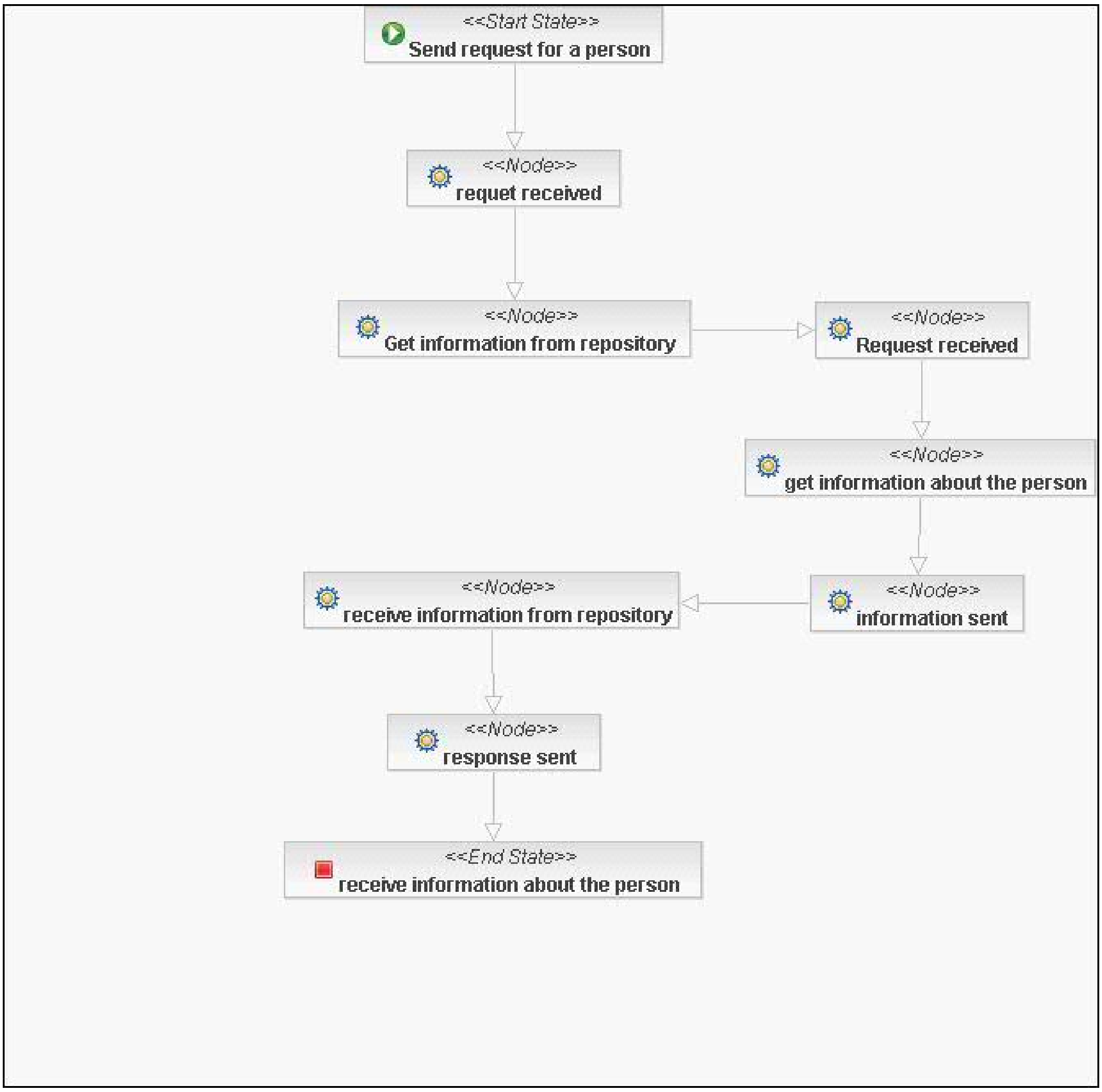

<<Start State>>
Send request for a person
<<Node>>
requet received
<<Node>>
Get information from repository
<<Node>>
Request received
<<Node>>
get information about the person
<<Node>>
receive information from repository
<<Node>>
information sent
<<Node>>
response sent
<<End State>>
receive information about the person

Figure 5.10.: JBoss jBPM - Modeling the Scenario Process

jBPM supports WSDL combined with jBPM BPEL for service orchestration. Although JBoss does not supply own environment to produce BPEL documents, but the company refers to Eclipse BPEL designer[8] or any other editor which is able to do that.

In fact jBPM can deal (import) with one of the defined criteria. BPEL can be deployed to the runtime engine even if jBPM do not provide an own tool for BPEL design. None of the defined standards can be export.

5.3.3. Process Enactment

The jBoss Process Virtual Machine (PVM) is a simple Java library for building and executing process graphs. This serves as a basis for all kinds of workflow, BPM and orchestration process languages. The PVM is the basis for multiple process languages. Native support for any process language can be build on top of the PVM.

JBoss uses own standard for the process design and execution jPDL. However, there is jBPM BPEL runtime engine that compiles with the BPEL 2.0 standard. Moreover, native support for any process language can be build on top of the Process Virtual Machine.

5.3.4. Additional Criteria

Support

Hotline

The official website does not provide any kind of information about a hotline. There are phone numbers of headquarters in different countries but no hotline or phone support in general.

Email

The email service is supplied for sales purposes only, even though email addresses are available for different division around the world.

FAQs

[8] http://eclipse.org/bpel

JBoss provides a solid support in form of documentation, forums & lists, and wiki containing FAQs to different sections and products of JBoss. The information is well structured and does not require any registration to access. The documentation of the product is provided in very well organized way.

Download-Area

The download area provides a very good overview of JBoss tools. But it is not always obvious which pack is required. The project download area displays the current release, but access to previous releases is also available. However, JBoss does not supply the download from own servers[9].

System Requirements

Operating System

jBPM runs on the following operating systems: Windows, Linux, Unix, and others. The website does not provide any further information about the various distributions.

Hardware

The information about the CPU frequency is missing. The minimum memory required is 512 MB and the minimum disk space required is 300 MB. The recommended capacities are not mentioned either for the memory nor for the disc space.

Licensing

JBoss software is licensed to everyone under GNU Lesser General Public License (LGPL). The company causes the license depicting the advantages[10]:

- Freedom of Use
- No Royalties or license fees
- Ability to distribute
- Modify JBoss Enterprise Middleware System (JEMS) products for internal use

[9] http://sourceforge.net/
[10] http://www.jboss.org/company/licensing/

- Distribute modifications externally
- Inclusion of JEMS products and commercially licensed offerings

5.3.5. Overall Summary

The process design phase and its supported standards in jBPM do not meet the defined criteria. jBPM uses own process definition language (jPDL) and supports the process design in Seam pageflow. jBPM provides a graphical process designer that is supposed to cope with graph based executable languages. jPDL was also used for the design of the scenario process. Some visual elements were missing within the design procedure. There is no visual representation of the pools that are supposed to represent the borders cross-organizational processes. Also the difference between an event and task could not be visually highlighted.

Although jBPM do not directly support the defined criteria for process design, it is important to mention that jBPM is extensible. The fact that jBPM deals with graph based languages allows adoption or transformation of standards selected here, such as Petri net, EPC, UML AD, ad BPMN. Hypothetically, any other design tool can be integrated into jBPM environment such as e.g. Eclipse BPEL Designer. This feature can be useful in term of extensibility as well as interoperability in case the designed model is executable.

According to the integration of web services into process design, it is not clear whether there is a visual support or not. For sure, jPDL uses interfaces and variables seamless from Java. This fact distinguishes strictly the process design and its implementation. It might be challenging though, for a business analyst to make a process run without any action from the side of a developer. Also the publishing of web services requires developer's knowledge. JBoss is built on Java, so web services can be implemented in Java as well and JBossWS is providing them. An interoperable WSDL file can be generated using `java2wsdl`, an Axis tool which comes with the JBossWS. The support of WSDL within the jBPM is conditional, because it is related to the use of BPEL.

jBPM supplies a very flexible module based framework. A very attractive point is that, on top of the PVM, various supports for different executable languages can also be built. BPEL depicts one of these execution languages. Some jBPM forums mention also the support of XPDL which is not confirmed by the official website. But this extensible ability can build a good basis for the interoperability.

6. Lessons learned and Conclusions

6.1. Comparison of Evaluation Results

All three evaluated tools supply a process design environment supporting different BPM languages. In fact, Intalio uses BPMN, NetBeans BPEL, and jBPM focuses on jPDL. None of the tools supports more than one BPM language defined in the catalogue of criteria. Thus, in terms of the expected support for several BPM languages, the tools do not meet the requirements. According to the supported languages, the investigated systems depict differences. The scenario helped to face the challenges and the convenience in terms of business process modeling.

JBoss jBPM offers an option to create a business process, integrating process design environments from other vendors, but default supplied language (jPDL) does not meet the requirements. First of all jPDL is not a common standard for process modeling. On the other hand it was not possible to create graphical borders of the different process participants while modeling the scenario. Moreover, the usage of a node for different purposes (e.g. events and tasks) tends to make the process design more abstract. This fact can lead to miss the target of closing the gap between business analysts and developers because too abstract models include more possibilities for interpretations.

In contrast, NetBeans IDE supports for process design a pure technical BPM language (BPEL). This language, usually used for web services orchestration, is the modeling standard of NetBeans BPEL Designer. The scenario could be modeled from the point of view of one business process participant only. The other parties had to be illustrated as Partner Links within the created process. The reason for missing of the global view is that the process design occurs directly on the technical level and requires subjective observation on own processes. This fact might create a challenge for the business analysts designing a business process. To model with BPEL, it requires some more technical knowledge being able to create an executable business process. Also NetBeans do not satisfy the defined requirements on BPM languages supplying only one of the identified standards.

According to the process design, Intalio supplies a good compromise among systems mentioned above. The process designer of Intalio supports BPMN that depicts a standard for business process modeling, balancing the gap between process design and its execution. The scenario could be modeled from the global point of view integrating all participants within the process. It means that the interacting parties could be illustrated as pools. Within the pools the corresponding sequences of the process could be depicted linked to each other. The sequence of the process that has to be executed locally can be highlighted changing the status of the pool to *executable*. However, also Intalio does not meet the requirement of the catalogue of criteria supporting only BPMN and no other standards.

Although such features as importability and exportability of various standards could increase the interoperability among different systems, none of the investigated tools consider this. There is no considered option to import/export any defined standards the supplied way. On that point the generated files by the tools can be extracted directly from the working space. This option can be interesting if the generated format will be common used standard (e.g. BPEL). BPEL is the only one standard that is used in all three BPM systems for the business process execution. This is also the only one execution standard, of two defined, that is supported. However, JBoss jBPM contains the option to integrate support for other standards on the top of its Process Virtual Machine in the system architecture, which could hypothetical serve as a basis for all kinds of workflow, BPM and orchestration process languages.

As a conlusion, all three evaluated BPM systems have particular strengths and weaknesses. Intalio supplies a strong environment for the process design, which is built on the BPMN standard. This standard represents a suitable solution for business analysts as well as for developers. It is important to highlight that the license of free available community edition of Intalio BPMS limits the use of not designated system components (e.g. Data Base Servers, application servers etc.). NetBeans IDE provides the possibility to design process model using BPEL that is simultaneously the execution standard for this BPM system. On the one hand it creates an advantage, which allows avoiding the transformation of a modeling language to an execution language. But on the other hand it depicts a challenge in terms of business process modeling, which is the task of business analysts. The most interesting candidate of evaluation is JBoss jBPM. Although jBPM meets the defined requirements less than the other evaluated products, it provides very high potential in terms of flexibility. The framework of jBPM allows integration of process design tools from other vendors as well as support for different execution languages.

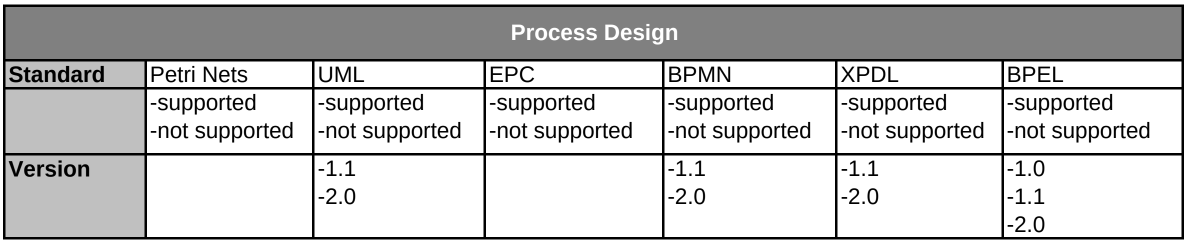
6.2. Conclusion

The main aim of this thesis was to evaluate Open Source BPM systems for the support of interoperability in the context of R4eGov. First of all it is considered to be an identification of a set of Open Source BPM tools. The selection was made and Intalio BPMS, NetBeans IDE, JBoss jBPM were chosen accordingly to the criteria for the mature Open Source Software. Furthermore, it was necessary to make a decision in terms of the methodology for evaluation. There were several methods discussed but the appropriate one depicted software product evaluation standard (ISO/IEC 14598). So, the evaluation of these tools, using ISO/IEC 14598, required a draft of a catalogue of criteria. This catalogue resulted from the analysis of requirements accordingly to R4eGov as well as general requirements on BPM systems. The categorization of identified criteria was based on the BPM lifecycle introduced by van der Aalst in [van der Aalst, 2004]. To support the practical part of the evaluation, a scenario was provided. The scenario, within European Arrest Warrant, describes a process which was applied for modeling using the BPM tools.

Although, the investigated tools have particular strengths, the evaluation could show that the BPM systems are far away from the requirements gathered in the catalogue of criteria. The expectations that one system is able to support several modeling standards were not really met. The vendors do not pay attention to this ability focusing in contrast on the main standard of execution (BPEL) and not on the ”multilingual” feature on the modeling level. This fact can be related to the challenges of the standard transformation from the high-level of abstraction (graphical model) to the technical-level of execution. Therefore vendors try to focus on one particular modeling standard making efforts on the smooth mapping of modeling and execution standards.

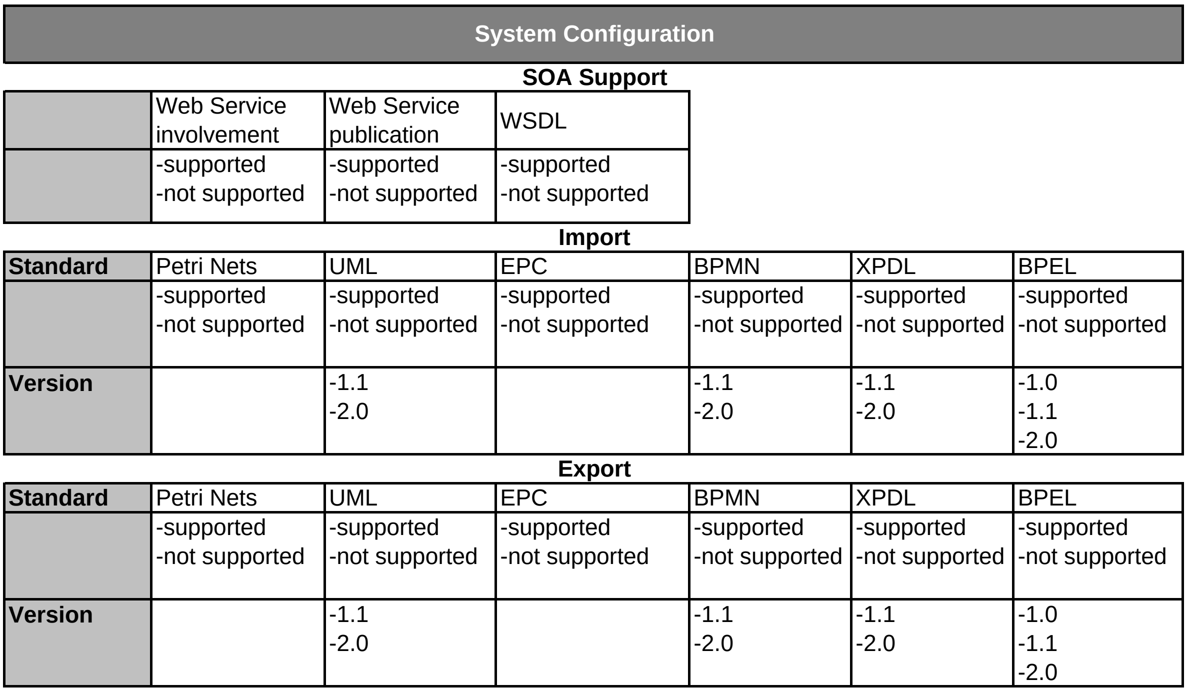
In overall, the investigated systems support the interoperability on the technical level of the execution. It results not from the support of different execution standards, but from the support of the common one, namely BPEL. However, it requires further investigations whether the generated BPEL files are easily adoptable among the different systems. Furthermore, the BPM tools provide low support for interoperability accordingly to the modeling standards. It is not possible to exchange or to share the process models except for BPEL using it simultaneously for process design and execution. However, using BPEL for process design entails technical knowledge and skills.

A. Catalogue of Criteria



Process Design						
Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported
Version		-1.1 -2.0		-1.1 -2.0	-1.1 -2.0	-1.0 -1.1 -2.0

Table A.1.: Process Design



System Configuration

SOA Support

	Web Service involvement	Web Service publication	WSDL
	-supported -not supported	-supported -not supported	-supported -not supported

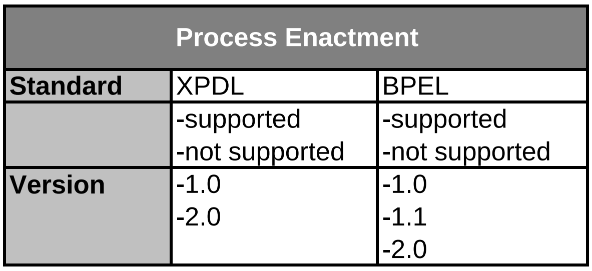
Import

Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported
Version		-1.1 -2.0		-1.1 -2.0	-1.1 -2.0	-1.0 -1.1 -2.0

Export

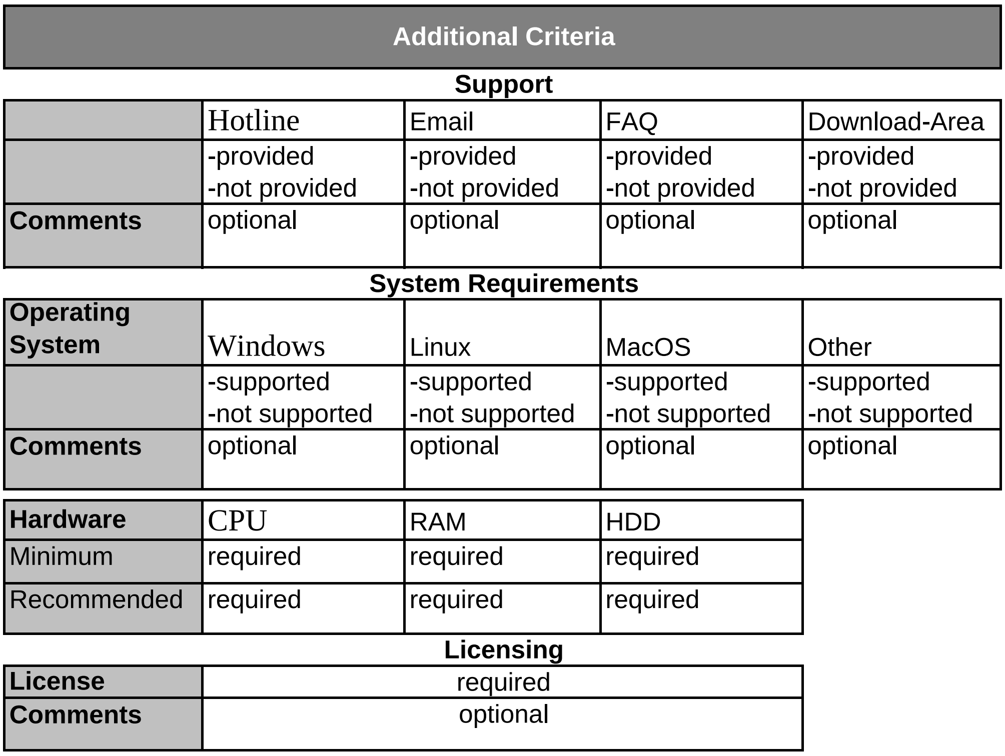
Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported
Version		-1.1 -2.0		-1.1 -2.0	-1.1 -2.0	-1.0 -1.1 -2.0

Table A.2.: System Configuration



Process Enactment		
Standard	XPDL	BPEL
	-supported -not supported	-supported -not supported
Version	-1.0 -2.0	-1.0 -1.1 -2.0

Table A.3.: Process Enactment



Additional Criteria

Support

	Hotline	Email	FAQ	Download-Area
	-provided -not provided	-provided -not provided	-provided -not provided	-provided -not provided
Comments	optional	optional	optional	optional

System Requirements

Operating System	Windows	Linux	MacOS	Other
	-supported -not supported	-supported -not supported	-supported -not supported	-supported -not supported
Comments	optional	optional	optional	optional

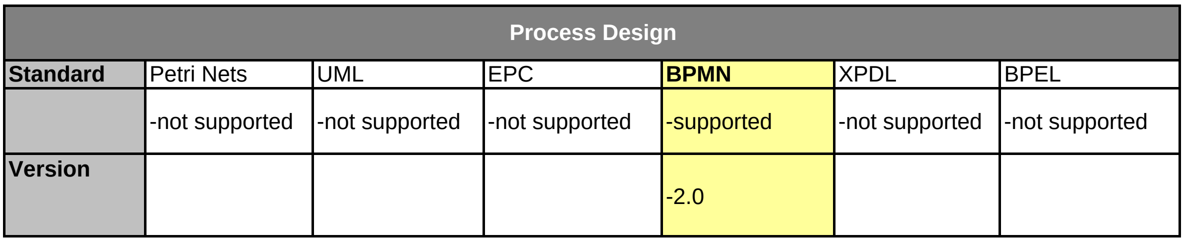
Hardware	CPU	RAM	HDD
Minimum	required	required	required
Recommended	required	required	required

Licensing

License	required
Comments	optional

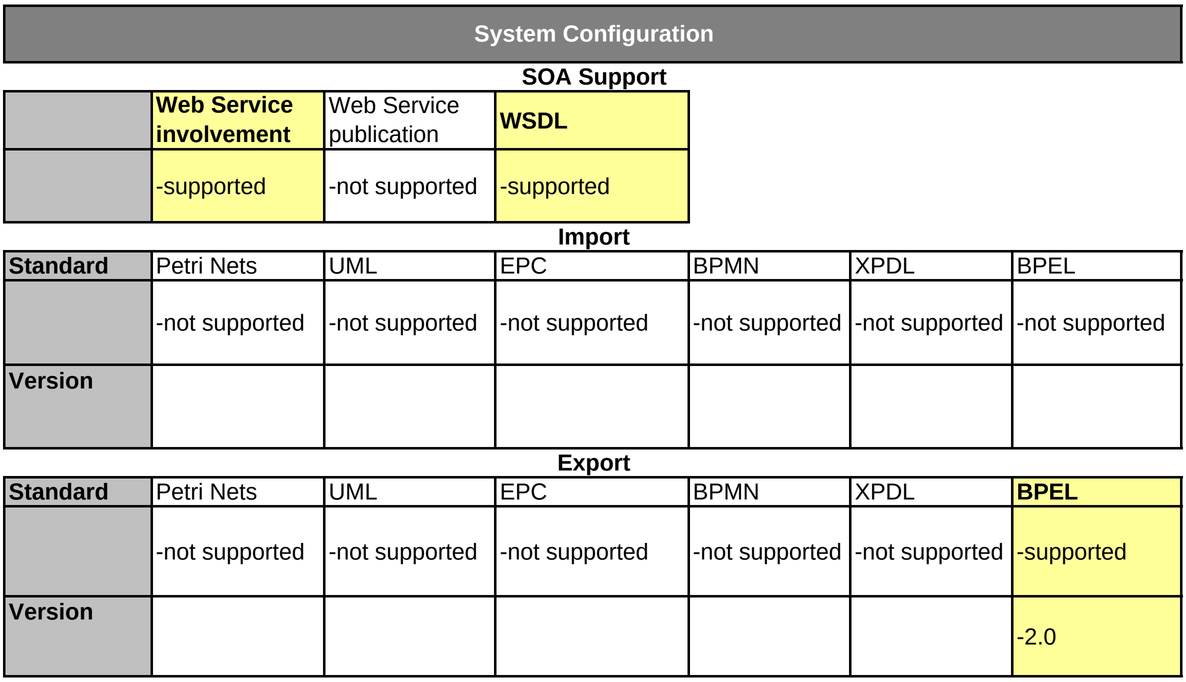
Table A.4.: Additional Criteria

B. Evaluation of Intalio | BPMS



Process Design						
Standard	Petri Nets	UML	EPC	**BPMN**	XPDL	BPEL
	-not supported	-not supported	-not supported	-supported	-not supported	-not supported
Version				-2.0		

Table B.1.: Intalio - Process Design



System Configuration

SOA Support

	Web Service involvement	Web Service publication	**WSDL**
	-supported	-not supported	-supported

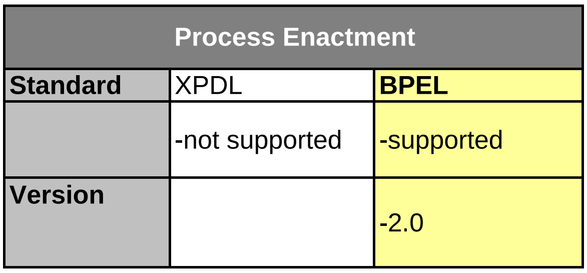
Import

Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-not supported	-not supported	-not supported	-not supported	-not supported	-not supported
Version						

Export

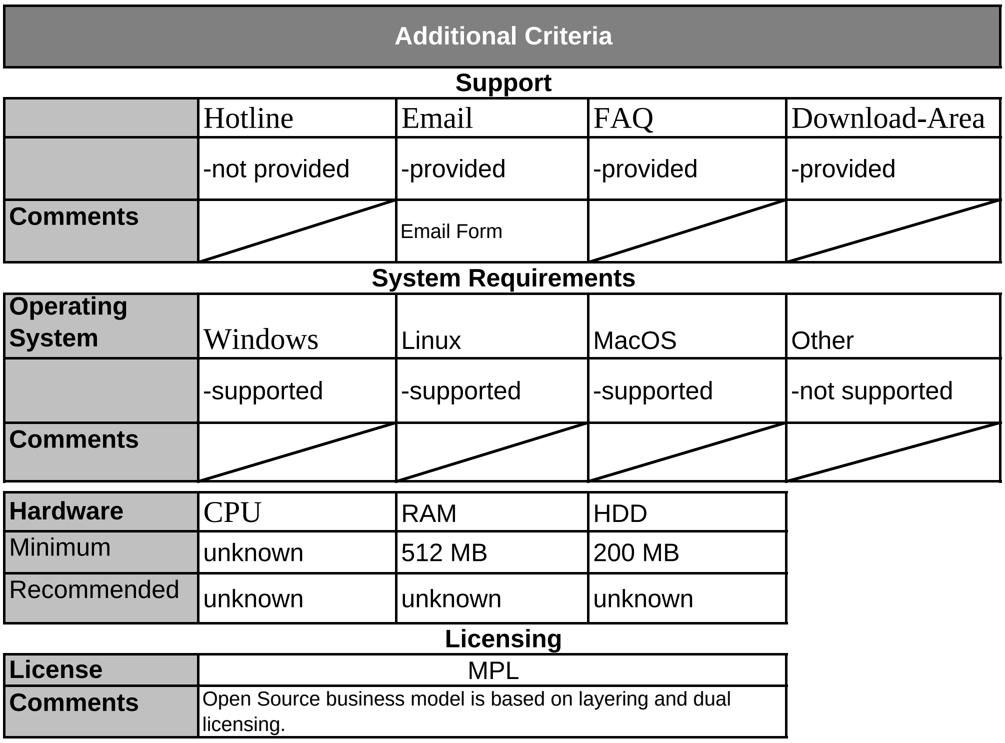
Standard	Petri Nets	UML	EPC	BPMN	XPDL	**BPEL**
	-not supported	-not supported	-not supported	-not supported	-not supported	-supported
Version						-2.0

Table B.2.: Intalio - System Configuration



Process Enactment		
Standard	XPDL	**BPEL**
	-not supported	-supported
Version		-2.0

Table B.3.: Intalio - Process Enactment



Additional Criteria

Support

	Hotline	Email	FAQ	Download-Area
	-not provided	-provided	-provided	-provided
Comments		Email Form		

System Requirements

Operating System	Windows	Linux	MacOS	Other
	-supported	-supported	-supported	-not supported
Comments				

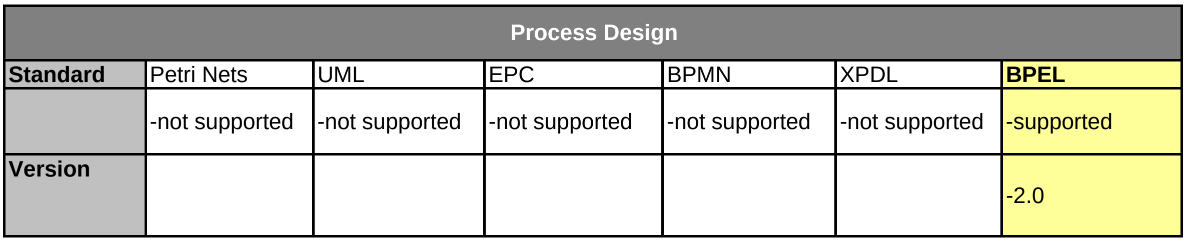
Hardware	CPU	RAM	HDD
Minimum	unknown	512 MB	200 MB
Recommended	unknown	unknown	unknown

Licensing

License	MPL
Comments	Open Source business model is based on layering and dual licensing.

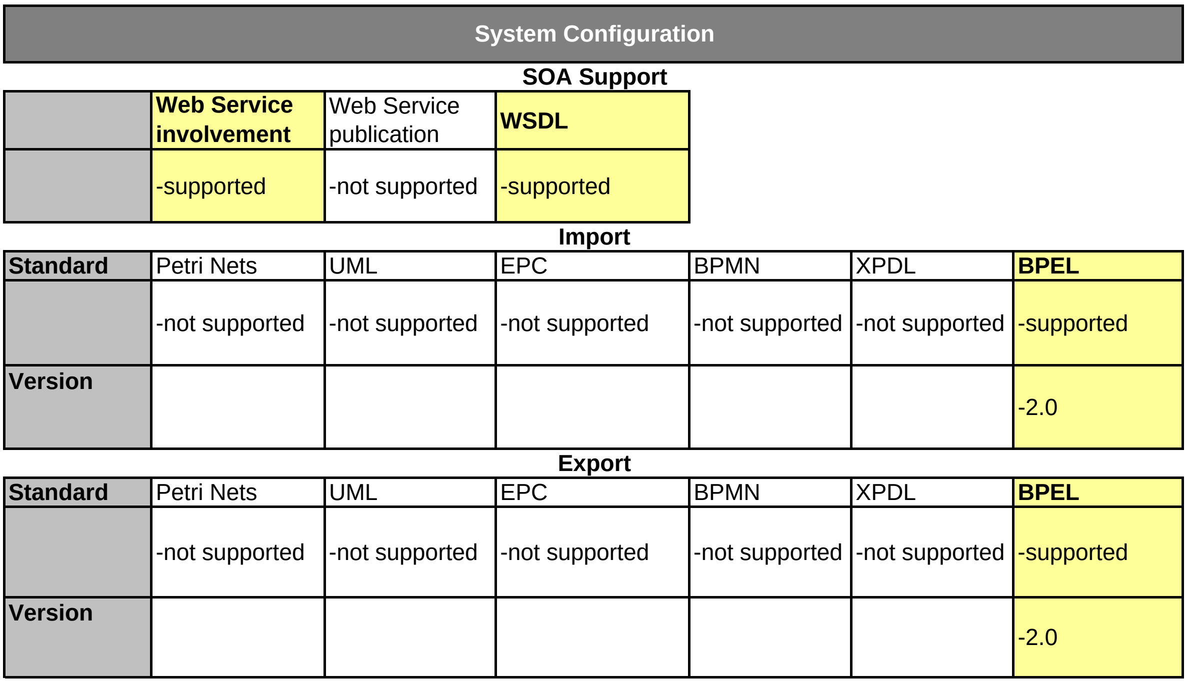
Table B.4.: Intalio - Additional Criteria

C. Evaluation of NetBeans IDE



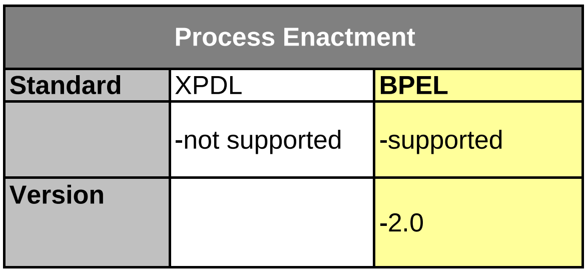
Process Design						
Standard	Petri Nets	UML	EPC	BPMN	XPDL	**BPEL**
	-not supported	-not supported	-not supported	-not supported	-not supported	-supported
Version						-2.0

Table C.1.: NetBeans - Process Design



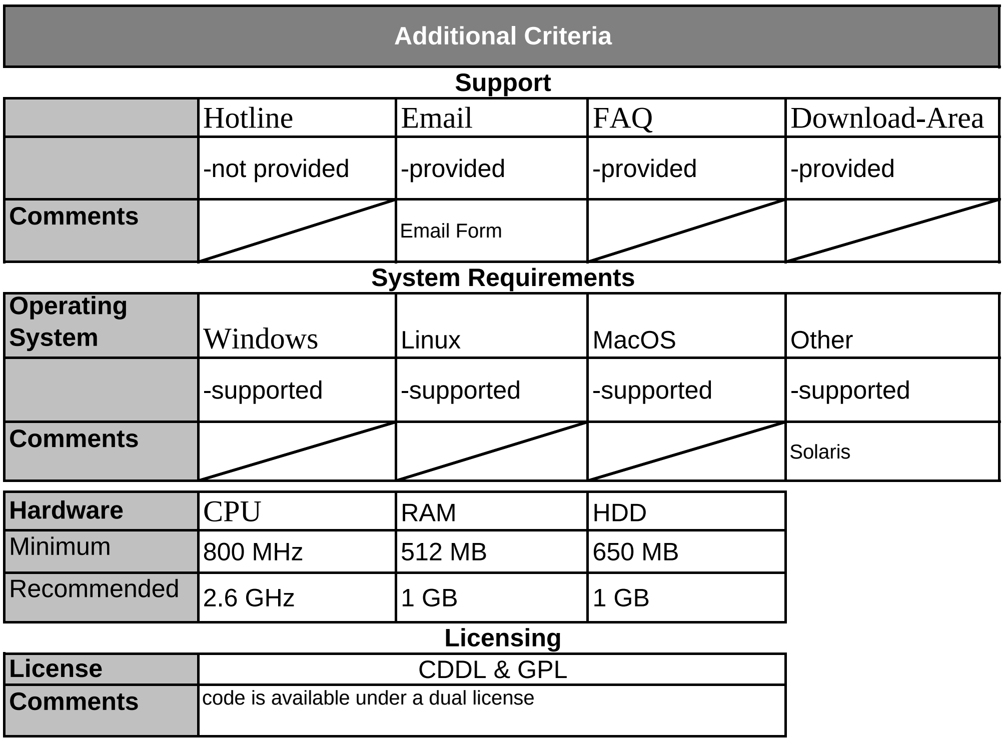
System Configuration						
			SOA Support			
	Web Service involvement	Web Service publication	**WSDL**			
	-supported	-not supported	-supported			
			Import			
Standard	Petri Nets	UML	EPC	BPMN	XPDL	**BPEL**
	-not supported	-not supported	-not supported	-not supported	-not supported	-supported
Version						-2.0
			Export			
Standard	Petri Nets	UML	EPC	BPMN	XPDL	**BPEL**
	-not supported	-not supported	-not supported	-not supported	-not supported	-supported
Version						-2.0

Table C.2.: NetBeans - System Configuration



Process Enactment		
Standard	XPDL	**BPEL**
	-not supported	-supported
Version		-2.0

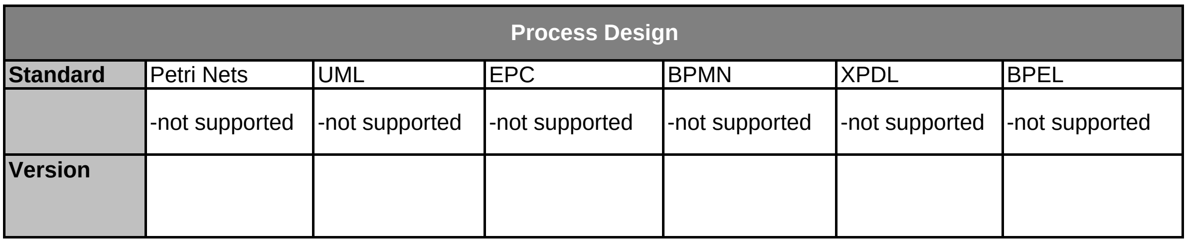
Table C.3.: NetBeans - Process Enactment



Additional Criteria				
		Support		
	Hotline	Email	FAQ	Download-Area
	-not provided	-provided	-provided	-provided
Comments		Email Form		
		System Requirements		
Operating System	Windows	Linux	MacOS	Other
	-supported	-supported	-supported	-supported
Comments				Solaris
Hardware	CPU	RAM	HDD	
Minimum	800 MHz	512 MB	650 MB	
Recommended	2.6 GHz	1 GB	1 GB	
		Licensing		
License	CDDL & GPL			
Comments	code is available under a dual license			

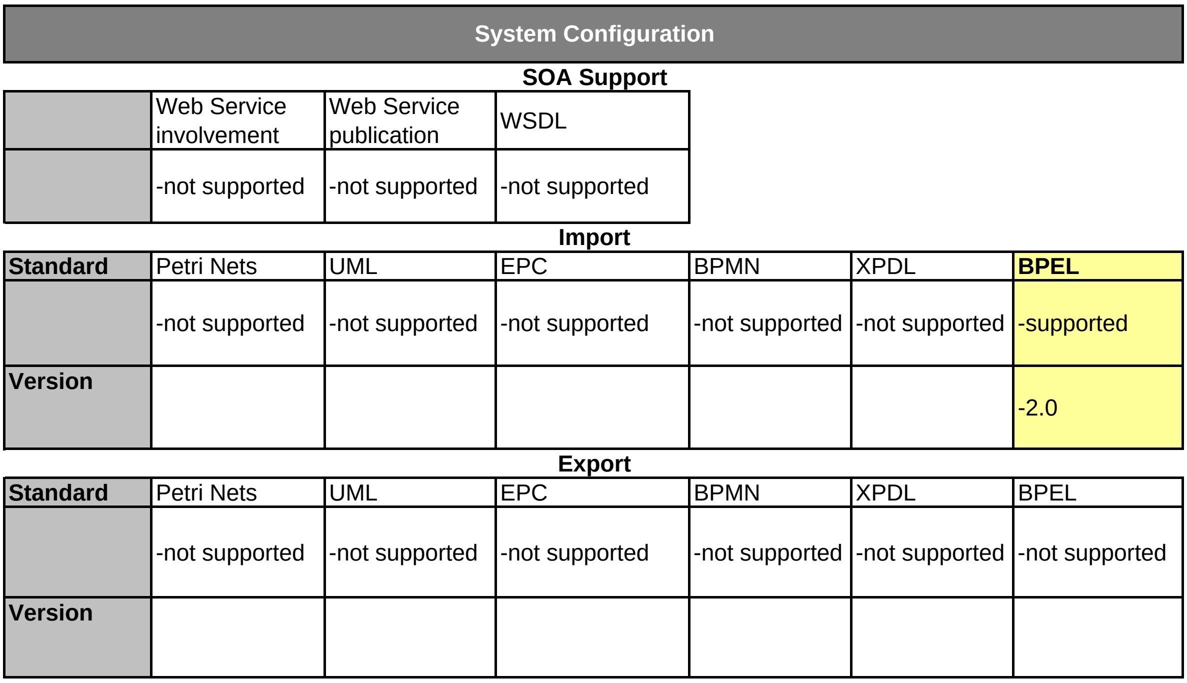
Table C.4.: NetBeans - Additional Criteria

D. Evaluation of JBoss jBPM



Process Design						
Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-not supported	-not supported	-not supported	-not supported	-not supported	-not supported
Version						

Table D.1.: JBoss jBPM - Process Design



System Configuration

SOA Support

	Web Service involvement	Web Service publication	WSDL
	-not supported	-not supported	-not supported

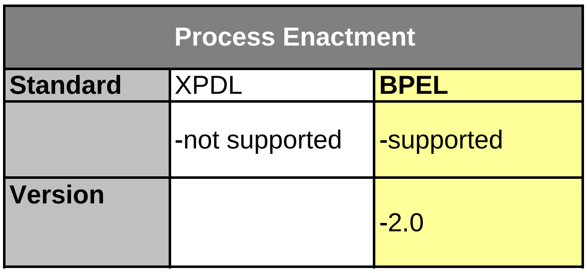
Import

Standard	Petri Nets	UML	EPC	BPMN	XPDL	**BPEL**
	-not supported	-not supported	-not supported	-not supported	-not supported	-supported
Version						-2.0

Export

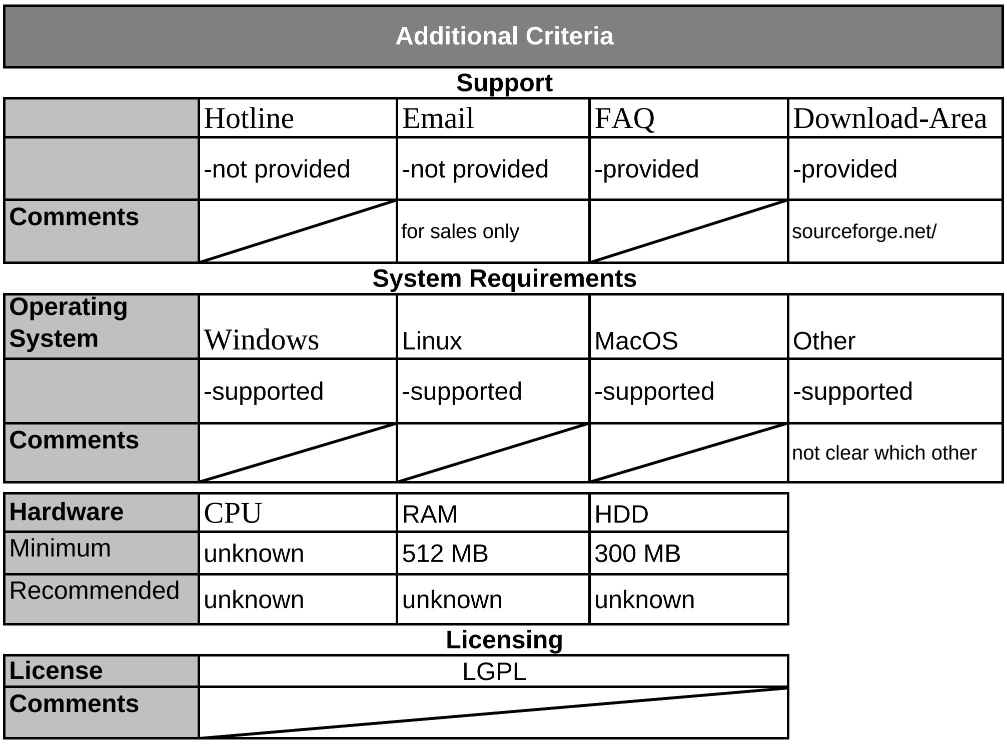
Standard	Petri Nets	UML	EPC	BPMN	XPDL	BPEL
	-not supported	-not supported	-not supported	-not supported	-not supported	-not supported
Version						

Table D.2.: JBoss jBPM - System Configuration



Process Enactment		
Standard	XPDL	**BPEL**
	-not supported	-supported
Version		-2.0

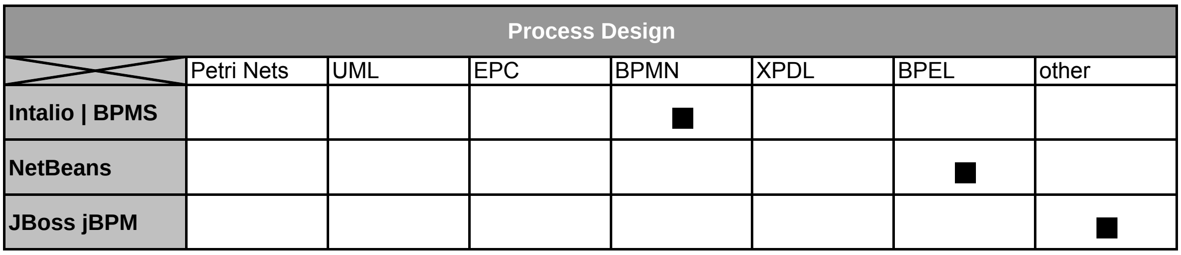
Table D.3.: JBoss jBPM - Process Enactment



Additional Criteria				
		Support		
	Hotline	Email	FAQ	Download-Area
	-not provided	-not provided	-provided	-provided
Comments		for sales only		sourceforge.net/
		System Requirements		
Operating System	Windows	Linux	MacOS	Other
	-supported	-supported	-supported	-supported
Comments				not clear which other
Hardware	CPU	RAM	HDD	
Minimum	unknown	512 MB	300 MB	
Recommended	unknown	unknown	unknown	
		Licensing		
License	LGPL			
Comments				

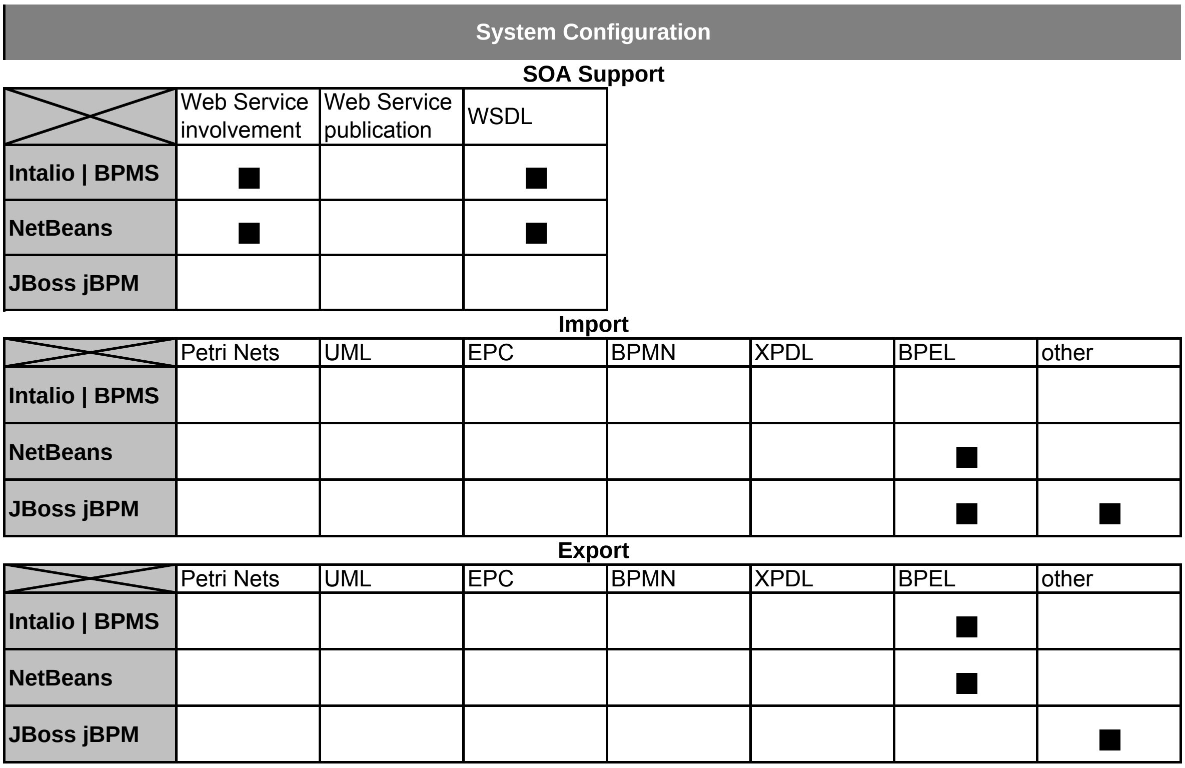
Table D.4.: JBoss jBPM - Additional Criteria

E. Comparison of the BPM Tools



Process Design							
	Petri Nets	UML	EPC	BPMN	XPDL	BPEL	other
Intalio | BPMS				■			
NetBeans						■	
JBoss jBPM							■

Table E.1.: Overall Overview of Process Design



System Configuration

SOA Support

	Web Service involvement	Web Service publication	WSDL
Intalio | BPMS	■		■
NetBeans	■		■
JBoss jBPM			

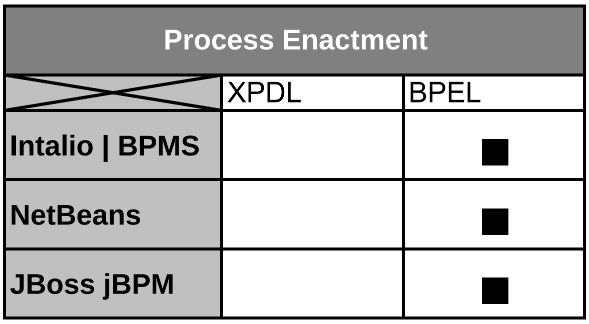
Import

	Petri Nets	UML	EPC	BPMN	XPDL	BPEL	other
Intalio | BPMS							
NetBeans						■	
JBoss jBPM						■	■

Export

	Petri Nets	UML	EPC	BPMN	XPDL	BPEL	other
Intalio | BPMS						■	
NetBeans						■	
JBoss jBPM							■

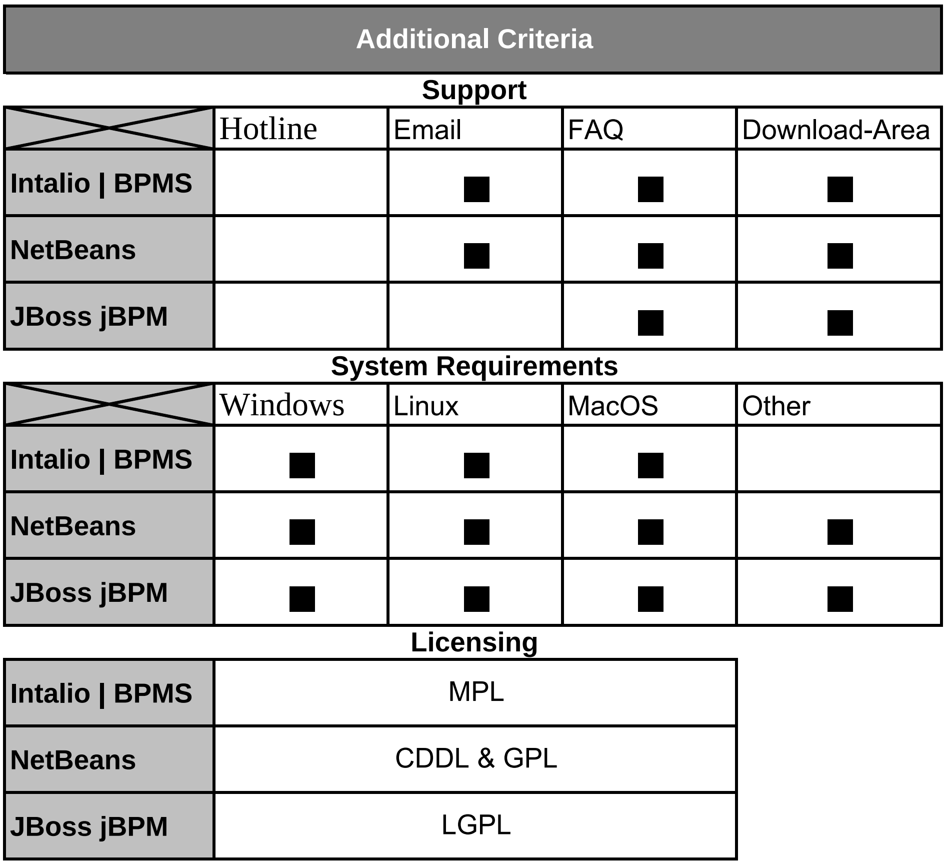
Table E.2.: Overall Overview of System Configuration



Process Enactment

	XPDL	BPEL
Intalio | BPMS		■
NetBeans		■
JBoss jBPM		■

Table E.3.: Overall Overview of Process Enactment



Additional Criteria

Support

	Hotline	Email	FAQ	Download-Area
Intalio | BPMS		■	■	■
NetBeans		■	■	■
JBoss jBPM			■	■

System Requirements

	Windows	Linux	MacOS	Other
Intalio | BPMS	■	■	■	
NetBeans	■	■	■	■
JBoss jBPM	■	■	■	■

Licensing

Intalio | BPMS	MPL
NetBeans	CDDL & GPL
JBoss jBPM	LGPL

Table E.4.: Overall Overview of Additional Criteria

Bibliography

[Allweyer, 2005] Allweyer, T. (2005). *Geschäftsprozessmanagement.* W3L-Verl.

[Alves et al., 2007] Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guızar, A., Kartha, N., et al. (2007). Web Services Business Process Execution Language Version 2.0. *Committee Draft*, 25.

[Andrews et al., 2003] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., et al. (2003). Business Process Execution Language for Web Services, Version 1.1. *Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation.*

[Arroyo et al., 2004] Arroyo, S., Bussler, C., Kopecky, J., Lara, R., Polleres, A., and Zaremba, M. (2004). Web Service Capabilities and Constraints in WSMO. *W3C Workshop on Constraints and Capabilities for Web Services.*

[Barros et al., 2005] Barros, A., Dumas, M., and Oaks, P. (2005). A Critical Overview of the Web Services Choreography Description Language (WS-CDL). *BPTrends Newsletter*, 3(3).

[Bartonitz, 2005] Bartonitz, M. (2005). BPMS, BPML, BPEL, BPMN, WMS, XPDL,...

[Binner, 2004] Binner, H. (2004). *Handbuch der prozessorientierten Arbeitsorganisation.* Hanser.

[Blekxtoon and van Ballegooij, 2005] Blekxtoon, R. and van Ballegooij, W. (2005). *Handbook on the European Arrest Warrant.* Asser Press.

[Bortz and Döring, 2002] Bortz, J. and Döring, N. (2002). *Forschungsmethoden und Evaluation: Für Human-und Sozialwissenschaftler.* Springer.

[Brown and Wallnau, 1996] Brown, A. and Wallnau, K. (1996). A framework for systematic evaluation of software technologies. *IEEE Software*, 13(5):39–49.

[Chang, 2005] Chang, J. (2005). *Business Process Management Systems: Strategy and Implementation.* CRC Press.

[Chappell, 2004] Chappell, D. (2004). *Enterprise Service Bus.* O'Reilly Media, Inc.

[Chinnici et al., 2004] Chinnici, R., Moreau, J., Ryman, A., and Weerawarana, S. (2004). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C Working Draft*, 3.

[Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web services description language (wsdl) 1.1. Technical report, World Wide Web Consortium.

[Coalition, 1999] Coalition, W. (1999). Workflow Management Coalition Terminology and Glossary. Technical report, Workflow Management Coalition.

[Decker et al., 2006] Decker, G., Puhlmann, F., and Weske, M. (2006). Formalizing Service Interactions. *LECTURE NOTES IN COMPUTER SCIENCE*, 4102:414.

[Di Maio, 2007a] Di Maio, A. (2007a). Open-Source Software Grows in the Public Sector. *Gartner Research*, (G00149692).

[Di Maio, 2007b] Di Maio, A. (2007b). Public-Sector Open-Source Survey, Worldwide: Getting There. *Gartner Research*, (G00147639).

[Dumas and Hofstede, 2001] Dumas, M. and Hofstede, A. (2001). UML Activity Diagrams as a Workflow Specification Language. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 76–90.

[Durvasula et al., 2006] Durvasula, S., Guttmann, M., Kumar, A., Lamb, J., Mitchell, T., Oral, B., Pai, Y., Sedlack, T., Sharma, H., and Sundaresan, S. (2006). SOA Practitioners' Guide, Part 1: Why Services-Oriented Architecture.

[Dusollier et al., 2004] Dusollier, S., Laurent, P., and Schmitz, P. (2004). Open Source Licensing of software developed by The European Commission (applied to the CIRCA solution). *European Commission DG ENTR. Available online at http://europa. eu. int/idabc/servlets/Doc.*

[Fensel et al., 2006] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., and Domingue, J. (2006). *Enabling Semantic Web Services: The Web Service Modeling Ontology.* Springer-Verlag New York, Inc. Secaucus, NJ, USA.

[Fischer, 2005] Fischer, L. (2005). *Workflow Handbook 2005.* Future Strategies Inc.

[Fowler, 2004] Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley Professional.

[Freiheit et al., 2007] Freiheit, J., Matheis, T., and Ziemann, J. (2007). Definition of static and dynamic models of collaborative workflow interoperability. R4eGov.

[Georgiev et al., 2004] Georgiev, E., Haber, G., Reifensteiner, J., and Sallmann, R. (2004). *Open Source Software-Einsatz in der öffentlichen Verwaltung.* Österr. Städtebund.

[Ghosh et al., 2004] Ghosh, R., Glott, R., Robles-Martinez, G., and Schmitz, P. (2004). Guideline For Public Administrations On Partnering With Free Software Developers. *European Commission DG ENTR. Available online at http://europa. eu. int/idabc/servlets/Doc.*

[Gudgin et al., 2007] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., Nielsen, H. F., Karmarkar, A., and Lafon, Y. (2007). SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).

[Gulledge Jr and Sommer, 2002] Gulledge Jr, T. and Sommer, R. (2002). Business process management: public sector implications. *Journal, Vol*, 8(4).

[Haas and Brown, 2003] Haas, H. and Brown, A. (2003). Web Services Glossary. *World Wide Web Consortium, August.*

[Hamadi and Benatallah, 2003] Hamadi, R. and Benatallah, B. (2003). A Petri net-based model for web service composition. In *Proceedings of the 14th Australasian database conference-Volume 17*, pages 191–200. Australian Computer Society, Inc. Darlinghurst, Australia, Australia.

[Hegner, 2003] Hegner, M. (2003). *Methoden zur Evaluation von Software.* IZ, InformationsZentrum Sozialwiss.

[Hill et al., 2006a] Hill, J., Sinur, J., Flint, D., and Melenovsky, M. (2006a). Gartner's Position on Business Process Management, 2006. *Gartner Research*, (G00136533).

[Hill et al., 2006b] Hill, J., Sinur, J., Flint, D., and Melenovsky, M. (2006b). Selection Criteria Details for Business Process Management Suites, 2006. *Gartner Research.*

[Hörbst et al., 2005] Hörbst, A., Fink, K., and Goebel, G. (2005). The ISO/IEC 9126-1 as a Supporting Means for the System Development Process of a Patient Information Web Service. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, 116:967.

[Huys, 2007] Huys, D. (2007). Detailed technical requirements of the eurojust/ europol - fedict case scenario. R4eGov.

[ISO, 1999a] ISO (1999a). International standard ISO/IEC 14598-1. information technology: Software product evaluation-part 1: General overview.

[ISO, 1999b] ISO (1999b). International standard ISO/IEC 14598-2. information technology: Software product evaluation-part 2: Planning and management.

[ISO, 1999c] ISO (1999c). International standard ISO/IEC 14598-3. information technology: Software product evaluation-part 3: Process for developers.

[ISO, 1999d] ISO (1999d). International standard ISO/IEC 14598-5. information technology: Software product evaluation-part 5: Process for evaluation.

[ISO, 1999e] ISO (1999e). International standard ISO/IEC 14598-6. information technology: Software product evaluation-part 6: Evaluation modules.

[Jeckle et al., 2004] Jeckle, M., Rupp, C., Hahn, J., Zengler, B., and Queins, S. (2004). *UML2 glasklar.* Carl Hanser Verlag Munchen Wien.

[Jost and Kruppke, 2004] Jost, W. and Kruppke, H. (2004). BPM-Der Aris Value Engineering Ansatz. *Scheer, A.-W.; Abolhassan, F.; Kruppke, H.: Innovation durch GPM. Jahrbuch Business Process Excellence*, 2005.

[Jung et al., 2004] Jung, H., Kim, S., and Chung, C. (2004). Measuring software product quality: a survey of ISO/IEC 9126. *Software, IEEE*, 21(5):88–92.

[Juric et al., 2004] Juric, M., Mathew, B., and Sarang, P. (2004). *Business Process Execution Language For Web Services.* Packt Publishing Ltd.

[Keller and Partner, 1999] Keller, G. and Partner (1999). *SAP R/3 prozeßorientiert anwenden iteratives Prozeß-prototyping mit ereignisgesteuerten Prozeßketten und Knowledge Maps.* Addison-Wesley.

[Keller et al., 1992] Keller, G., Scheer, A., and Nüttgens, M. (1992). *Semantische Prozeßmodellierung auf der Grundlage" Ereignisgesteuerter Prozeßketten (EPK)".* Inst. für Wirtschaftsinformatik.

[Krafzig et al., 2004] Krafzig, D., Banke, K., and Slama, D. (2004). *Enterprise SOA: Service-Oriented Architecture Best Practices.* Prentice Hall Ptr.

[Leymann et al., 2002] Leymann, F., Roller, D., and Schmidt, M. (2002). Web services and business process management. *IBM Systems Journal*, 41(2):198–211.

[Mendling and Nüttgens, 2004] Mendling, J. and Nüttgens, M. (2004). Exchanging EPC Business Process Models with EPML. In *Proc. of the 1st GI-Workshop XML4BPM-XML Interchange Formats for Business Process Management, Marburg, Germany*, pages 61–79.

[Mendling and Nüttgens, 2006] Mendling, J. and Nüttgens, M. (2006). EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). *Information Systems and E-Business Management*, 4(3):245–263.

[Miserez, 2006] Miserez, P. (2006). Wegbereiter der nächsten Industrialisierungswelle. *Ehlers, S.: BPM-Business Prozessmanagement in Praxis und Anwendung. Books on Demand, München*, pages 12–27.

[Murata, 1989] Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

[Myatt, 2007] Myatt, A. (2007). *Pro NetBeans IDE 5.5 Enterprise Edition.* Apress.

[Nissen et al., 2008] Nissen, V., Petsch, M., and Schorcht, H. (2008). *Service-orientierte Architekturen: Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen*. Deutscher Universitätsvlg.

[Noel, 2005] Noel, J. (2005). BPM and SOA: Better Together. *Whitepaper, IBM Corporation*.

[Nüttgens, 2002] Nüttgens, M. (2002). Rahmenkonzept zur Evaluierung von Modellierungswerkzeugen zum Geschäftsprozessmanagement. *Gesellschaft für Informatik (GI) eV): Informationssytem-Architekturen, Wirtschaftsinformatik Rundbrief der GI Fachgruppe WI-MobIS*.

[Nüttgens et al., 1998] Nüttgens, M., Feld, T., and Zimmermann, V. (1998). Business Process Modeling with EPC and UML: Transformation or Integration. *The Unified Modeling Language-Technical Aspects and Applications*, pages 250–261.

[OASIS, 2004] OASIS (2004). Introduction to uddi: Important features and functional concepts.

[OpenBRR.org, 2005] OpenBRR.org (2005). A proposed open standard to facilitate assessment and adoption of open source software. *Business Readiness Rating for Open Source*, BRR 2005 - RFC 1.

[Ouyang et al., 2007] Ouyang, C., Verbeek, E., van der Aalst, W., Breutel, S., Dumas, M., and ter Hofstede, A. (2007). Formal semantics and analysis of control flow in WS-BPEL. *Science of Computer Programming*, 67(2-3):162–198.

[Owen and Raj, 2003] Owen, M. and Raj, J. (2003). BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard. *Available http://whitepaper. techweb. com/cmptechweb/search/viewabstract/71*, 885.

[Papazoglou and Georgakopoulos, 2003] Papazoglou, M. and Georgakopoulos, D. (2003). Service-Oriented Computing. *Communications of the ACM*, 46(10):25–28.

[Peltz, 2003] Peltz, C. (2003). Web Services Orchestration and Choreography.

[Peterson, 1981] Peterson, J. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR Upper Saddle River, NJ, USA.

[Peterson, 1977] Peterson, J. L. (1977). Petri Nets. *ACM Computing Surveys*, 9(3):223–252.

[Plummer et al., 2004] Plummer, D., Andrews, W., and Valdes, R. (2004). Consider WSDL a Critical Standard. *Gartner Research*.

[Reisig, 1985] Reisig, W. (1985). *Petri nets: an introduction*. Springer-Verlag New York, Inc. New York, NY, USA.

[Richards and Griffith, 2005] Richards, N. and Griffith, S. (2005). *JBoss: A Developer's Notebook*. O'Reilly Media, Inc.

[Richter et al., 2005] Richter, J., Haller, H., and Schrey, P. (2005). Serviceorientierte Architektur. *Informatik-Spektrum*, 28(5):413–416.

[Rossi et al., 2004] Rossi, P., Lipsey, M., and Freeman, H. (2004). *Evaluation: A Systematic Approach*. Sage Publications Inc.

[Rumbaugh et al., 1996] Rumbaugh, J., Jacobson, I., and Booch, G. (1996). *The unified modeling language*. University Video Communications.

[Russell et al., 2006] Russell, N., van der Aalst, W., ter Hofstede, A., and Wohed, P. (2006). On the suitability of UML 2.0 activity diagrams for business process modelling. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling-Volume 53*, pages 95–104. Australian Computer Society, Inc. Darlinghurst, Australia, Australia.

[Schlimmer et al., 2002] Schlimmer, J. et al. (2002). Web Services Description Requirements. *World Wide Web Consortium (W3C) working draft. Oct*, 28.

[Schulte, 2008] Schulte, R. W. (2008). Tutorial for EDA and How It Relates to SOA. *Gartner Research*, (G00155163).

[Schulte and Abrams, 2007] Schulte, R. W. and Abrams, C. (2007). The Business Impact of Service-Oriented Architecture. *Gartner Research*, (G00153828).

[Shapiro, 2005] Shapiro, R. (2005). Workflow Process Definition Interface–XML Process Definition Language. *Lighthouse Point (Fl): Workflow Management Coalition*, (WFMC-TC-1025).

[Smith and Fingar, 2002] Smith, H. and Fingar, P. (2002). *Business process management*. Meghan-Kiffer Press Tampa, FL.

[Snell et al., 2002] Snell, J., Tidwell, D., and Kulchenko, P. (2002). *Programming Web Services with SOAP*. O'Reilly Media, Inc.

[Stamelos et al., 2000] Stamelos, I., Vlahavas, I., Refanidis, I., and Tsoukiŕs, A. (2000). Knowledge based evaluation of software systems: a case study. *Information and Software Technology*, 42(5):333–345.

[van der Aalst, 1999] van der Aalst, W. (1999). Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639–650.

[van der Aalst, 2003] van der Aalst, W. (2003). Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. *Department of Technology Management Eindhoven University of Technology, The Nether-lands.*

[van der Aalst, 2004] van der Aalst, W. (2004). Business Process Management: A personal view. *Business Process Management Journal*, 10(2):135–9.

[van der Aalst et al., 1998] van der Aalst, W. et al. (1998). The Application of Petri Nets to Workflow Management. *JOURNAL OF CIRCUITS SYSTEMS AND COMPUTERS*, 8:21–66.

[White, 2004a] White, S. (2004a). Business Process Modeling Notation (BPMN) Version 1.0. *Business Process Management Initiative, BPMI. org, May*, 1.

[White, 2004b] White, S. A. (2004b). Introduction to BPMN. *IBM Cooperation.*

[Wimmer et al., 2006] Wimmer, M., Schmidt, D., and Diedrich, E. (2006). A Three Dimensional Framework to Realize Interoperability in Public Administrations. *Semantic Web for eGovernment.*

[Wohed et al., 2006] Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede, A., and Russell, N. (2006). On the Suitability of BPMN for Business Process Modelling. *Proceedings 4th International Conference on Business Process Management (BPM 2006), LNCS, Vienna, Austria.*

[Woods and Mattern, 2006] Woods, D. and Mattern, T. (2006). *Enterprise SOA: Designing IT for Business Innovation.* O'Reilly Media, Inc.

[Wottawa and Thierau, 1990] Wottawa, H. and Thierau, H. (1990). *Lehrbuch Evaluation.* Huber.