FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Formalisms and Tools to Describe and Monitor Engineering Processes

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

Alexander Maximilian Wurl
Matrikelnummer 0826264

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Priv.-Doz. Dipl.-Ing. Dr. Axel Polleres

Wien, 18. Jänner 2016

Alexander Maximilian Wurl	Axel Polleres

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Formalisms and Tools to Describe and Monitor Engineering Processes

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Alexander Maximilian Wurl
Registration Number 0826264

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Priv.-Doz. Dipl.-Ing. Dr. Axel Polleres

Vienna, 18th January, 2016

Alexander Maximilian Wurl	Axel Polleres

Erklärung zur Verfassung der Arbeit

Alexander Maximilian Wurl
Schopenhauerstrasse 29, Top 8, 1180 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. Jänner 2016

Alexander Maximilian Wurl

Acknowledgements

I would like to thank Axel Polleres, my supervisor, for supporting and guiding me through my work.
For the work of this master thesis I was lucky to be employed at Siemens AG - Corporate Technology, as junior research scientist where I had the pleasure to contribute to a scientific project. I have to forward my specials thanks to Alois Haselböck for his superior assistance within Siemens. I also have to forward my thanks to my wonderful colleagues, Giray Havur and Simon Sperl, for all their support and cooperation in realizing my work. Thanks to the whole Siemens CT RTC BAM CON-AT staff for a wonderful time at Siemens and especially to Herwig Schreiner and Wolfgang Schwaiger for their generous support.
Of course, I also want to thank my family especially my parents who always stood behind me and supported me financially.
Finally, I want to thank all my friends who supported me during my study and gave me a great time over the last years.

Kurzfassung

Technische Großprojekte im Infrastrukturbereich wie etwa Eisenbahnprojekte oder Industrieanlagen stellen komplexe Entwicklungsprozesse dar. Die zeitgerechte Ausführung und präzise Steuerung solcher Prozesse ist essentiell für das Endergebnis solcher Projekte. Das hängt einerseits von der Möglichkeit ab, einen vorgegebenen Prozessablauf durch seine Beschreibung überwachen zu können und andererseits von der Koordination des involvierten Personals, das unterschiedliche Aufgaben in einem Prozess zeitgerecht ausführen sollen.
In der vorliegenden Arbeit präsentieren wir wie mit einer ausgewählten Modellierungssprache komplexe Entwicklungsprozesse modelliert werden können. Weiters werden Tools evaluiert, die nicht nur die Ausführung modellierter Entwicklungsprozesse unterstützen, sondern auch um Methoden aus der Wissenschaft hinsichtlich automatischer Koordination von verfügbarem Personal erweitert werden können. Obwohl einige existierende Tools bereits Funktionen zur Koordinierung von Personal integriert haben, ist ein hoher manueller Aufwand notwendig, Personal zu koordinieren.
Wir präsentieren einen Ansatz, eine wissenschaftliche Methode zur automatischen Personalzuteilung basierend auf Answer Set Programming (ASP) in ein Pozess-Management-Tool zu integrieren. Das Tool, das um diesen Ansatz erweitert wird, kann somit neben der Überwachung und Steuerung komplexer Entwicklungsprozesse automatisch verfügbares Personal jederzeit unter Einhaltung von Constraints allokieren. Mit dieser Erweiterung können Zeit und Kosten gespart werden, da Fehler, die durch manuelle Koordinierung entstehen reduziert werden.

Abstract

The description and monitoring of complex engineering processes is a vital component in the overall execution of large scale engineering projects. In addition to that the just-in time coordination of (human) resources collaborating in the different tasks involved is a challenge in itself that needs to be optimized for the timely completion of the end product.
In this presented master thesis we present the modeling of engineering processes by means of a selected modeling language from Business Process Management (BPM) and we further present a proof of concept integrating an approach to automate the coordination of available resources in an optimal fashion into a selected BPM system. Although there are settings for coordinating resources in already existing systems, they are falling short of mechanisms that allow for an automated allocation of resources under constraints in one stroke. As a support for the coordination of (human) resources we employ an approach for the automated allocation of resources from academia based on Answer Set Programming (ASP) that we integrate into a BPM system.
In addition to the monitoring of complex engineering processes, the resulting extension allows to generate an automated resource allocation plan for currently available resources - a result that shall minimize errors as well as achieve time and cost savings.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Nowadays, most of the technical products result from a value-creation chain of production systems. This chain describes the sequence of activities that have to be executed within the production process. Such a production process can be described either by proprietary solutions created by means of off-the-shelve applications like MS Word, Excel or Visio or by means of specific process-oriented languages that are supported by systems - so called Business Process Management Systems (BPMS) - capable to monitor the execution of modeled processes. It is obvious that the latter process-oriented method offers to provide more overview and control since the progress of the process execution can be monitored at any time.
Activities within the modeled process can be executed manually or automatically depending on the task. Those tasks that require the involvement of manual participation of human resources, necessitate an organization of scheduled resources to complete tasks in time.
In this thesis we want to demonstrate how to describe engineering processes in order to monitor them within such a BPM system. In a following step we extend a selected system by means of a proof of concept implementation for automatic resource allocation using formalisms that enable the scheduling of resources automatically.

1.1 Motivation

In large-scale infrastructure deployments, technical infrastructure products such as industrial plants, railway automation solutions or power stations involve various business units within a company and subcontractors who are all making use of different tools in order to handle complex engineering processes. Describing and monitoring these complex engineering processes is a highly demanding task in almost all data driven business management scenarios. On the shelf of available tools is a multitude of software solutions that any company applies depending on the specific problem solution required. In case of

infrastructure providers like Siemens, some in-house programmed software solutions are in use to resolve a great variety of application challenges. But these tools are making use of internal, partially proprietary representations of engineering processes, which require strong manual efforts updating processes for any changes and all resources involved. To reduce the efforts of integrating and monitoring such processes the domain of Business Process Management (BPM) seems to be a good starting point since BPM can represent processes in a standardized way. BPM intends to manage processes using techniques for design, enactment, control, and analysis, integrating humans, organizations, applications and other sources of information [63].

1.2 Problem Statement

The development of products for infrastructure systems require efficiency in planning and execution of engineering processes, to finally ensure quality and accuracy for end products. Many proprietary solutions of managing engineering processes result in a high amount of manual effort and are difficult to comprehend. In addition to that, the sequence of timing of various components and manpower involved regularly lead to the challenge of coordination of who does what, when, how and where. In case of any misalignment this might have a negative impact on the desired end-product of the process due to errors and delays caused by the effects of manual control.
A typical example of an engineering process is located in the domain of railway automation of Siemens, that among others deals with the configuration and redeployment of various components of electronic interlocking systems. This engineering process includes several phases with many different roles and responsibilities involved. Due to the complexity of engineering processes within projects many departments and their resources and development capabilities, run-time schedules, testing facilities and many more aspects have to be observed and involved in the engineering process. The outcome needs to be a resource allocation and time schedule that satisfies various constraints. In a large-scale graphical overview the different resources required are allocated and their timing scheduled. Underneath to each of these graphical assignments the different resource allocations and schedules are planned in detail on the basis of Excel: task descriptions and their responsibilities are listed according to the phase of the engineering process. It is evident that a complex engineering process with a lot of people and facilities involved, based on planning by Excel, is prone to all kinds of inaccuracies, delays, etc. Too large Excel documents appear to be confusing. For illustration, Excel table 1.1 depicts a simplified process description.

On top of all, such kind of planning is never up to date as is almost always required. In order to tackle this ever returning challenge for each and every engineering process it is therefore the explicit goal to have available an appropriate tool that helps to circumvent these shortcomings.

Over the last decades, the evolution of Business Process Management (BPM) has made

Table 1.1: Illustration of a process description in Excel

Req.	**Task**	**Resp.**	**Output**
New Equipment needed	**Submit equipment rental request**	Site Engineer	Request
Request	**Select suitable Equipment**	Clerk	
	Check Availability	Clerk	
Available Equipment	**Review Rental Request**	Works Engineer	Request
Request	**Create PO**	Clerk	Purchase Order

significant progress providing solutions to manage business processes electronically. In this context, we consider engineering processes as one form of business processes turning a conceived product into a physical representation according to a pre-defined construction plan [41]. So, intuitively, adapting BPM to the domain of engineering processes seems to be a promising approach.
In the scope of this thesis, achieving this objective requires (i) a model and standard description language to re-model engineering processes in terms of standardization, and (ii) a tool that supports tasks such as monitoring of modeled processes. Furthermore, (iii) this tool should provide organizational features and options to extend it with approaches regarding (semi-)automatic resource allocation.

1.3 Aim of the Work

The aim of this master thesis is the evaluation of current state-of-the-art process languages, tools and their features to select an appropriate tool which should deliver the capability to manage modeled processes and to be extensible with our proof of concept to solve the involved allocation problems in order to monitor and describe safety-critical engineering processes.

Hypothesis. In this thesis we address the following research questions:

Q1 Which process language covers the required characteristics for modeling processes in the industrial context?

Q2 What of the BPM tools currently available on the market do support which kind of process language and furthermore along which criteria can they be compared against each other?

Q3 How can approaches, e.g. for resource allocation be incorporated into a BPM tool by means of inclusion of Knowledge Representation & Reasoning (KRR) methods?

Following an extensive survey on current state-of-the-art process languages, tools and their features, the outcome of this thesis should be a prototype to demonstrate how solutions for automated resource allocation (based on KRR planning and scheduling approaches) can be integrated into existing BPM tools.
To this end, we will first identify suitable BPM Suites and compare them along a set of criteria such as monitoring features, supported process description languages and extensibility (via APIs).
Next, we will identify solutions and algorithms for automatic, adaptive support of monitoring and checking conformance in engineering processes, e.g. automated resource allocation based on constraints from academia, and investigate how these can be incorporated into the tools.
Siemens will provide a use case for complex engineering processes in the railway domain, which will provide additional requirements for implementing a final prototype for managing engineering processes. We will verify our proposed solution against these requirements derived from practical use in terms of scalability applied to real-world problem instances and processes.

1.4 Impact of this Thesis

In the research project SHAPE (Safety-critical Human- and dAta-centric Process management in Engineering projects), a collaboration project[1] of the Vienna University of Economics (WU) together with Siemens AG Österreich, BPM is an important aspect that has been addressed. This thesis contributes with a technology and tool survey, leading to a selection of a tool that provides Application Programming Interfaces (APIs) in order to integrate approaches from academia.
Within Siemens, the analysis provided by this thesis should deliver a decision base for the selection of a BPM-operating language and its respective tool chain.

1.5 Structure of the Work

The presented thesis is structured as follows: In chapter 2 we begin with an introduction of Business Process Management in which we evaluate current state-of-the-art process languages with respect to their modeling capabilities for engineering processes. Then we continue in chapter 3 describing how to formalize resources under constraints for achieving resource assignment and resource allocation. In chapter 4, we evaluate a broad selection of Business Process Management Suites that are currently available on the market with respect to different criteria, e.g. process modeling, process realization, integration of systems, etc. The selection of one of these tools is the basis for further implementation. Our main contribution *Integration of Resource Allocation Capabilities into a BPMS* is presented in chapter 5. In chapter 6, we will be using the resulting implementation to

[1]Funded by the Austrian Research Promotion Agency (FFG) grant 845638 (SHAPE)

deploy this in a use case of industry. We conclude our thesis and discuss ideas for future improvements in chapter 7.

CHAPTER 2

Preliminaries in BPM

2.1 Business Process Management

There exist many similar definitions of *business process management* but probably the most cited one was presented by Gartner:

> *Business process management (BPM) is the discipline of managing processes (rather than tasks) as the means for improving business performance outcomes and operational agility. Processes span organizational boundaries, linking together people, information flows, systems and other assets to create and deliver value to customers and constituents [29].*

Thus, the management of business processes can be seen as the art and science of improving how work in an organization is performed to ensure consistent outcomes [26]. In detail, BPM is responsible for managing chains of events, activities, and decisions of different types, e.g. order-to-cash, procedure-to-pay, etc. This improvement of management results in a reduction of costs, reduction of execution times and reduction of error rates.

Business Processes encompasses a number of properties [26]:

1. **Events and activities Events.** happen atomically, meaning that they have no duration [26]. For instance, the arrival of a shipment by sea transport is an event which can then trigger the execution of series of activities [26], e.g. further transportation. The transport itself is an activity, which takes time.

2. **Activity.** An activity can be seen as one single unit of work, i.e. a task, included in an activity. If, for example, the shipment is checked on completeness, which is rather simple - we can say that this is a task.

3. **Decision Points.** Within a process there are some decision points that effects the way the process is further executed. E.g., if the shipment is not complete, the decision is that is should be returned.

4. **Actors/Resources.** Processes involve a number of actors - we also call them resources - such as human actors, organizations, or software systems acting on behalf of human actors or organizations, physical objects (equipment, materials, products, paper documents) and immaterial objects (electronic documents and electronic nodes). These resources might be required for executing custom activities [26].

5. **Outcome.** The outcome of a process should deliver some value to the actors involved in the process [26]. If, for example, the shipment is returned, no further processes can be executed - this corresponds to a *negative outcome*, as opposed to a *positive outcome* that delivers value to the actors involved [26].

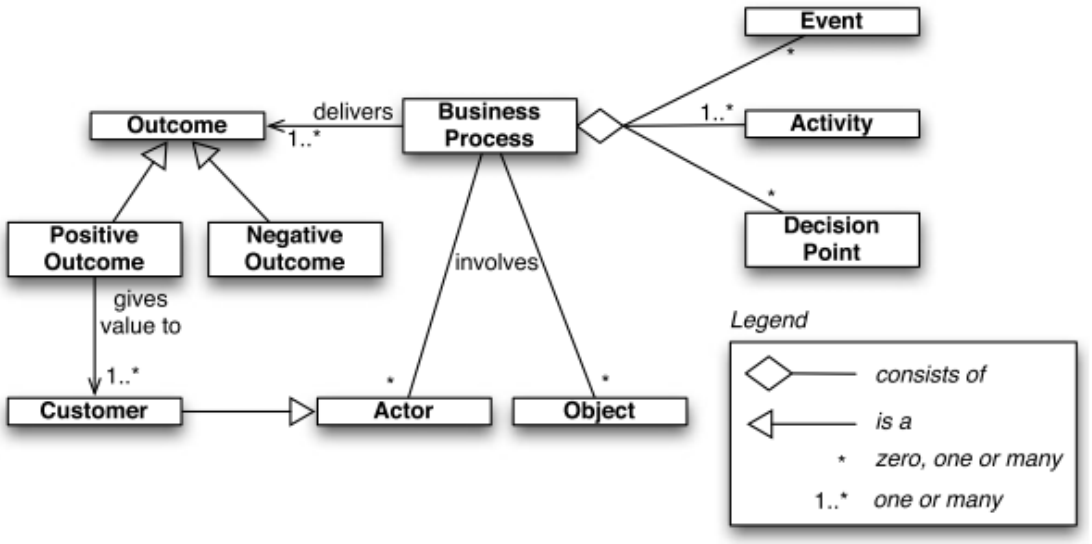


Figure 2.1: Ingredients of a business process [26]

As shown in Figure 2.1, a business process is defined by a collection of the properties (1-5). Based on the definition of business processes, BPM can be defined as a body of methods, techniques and tools to discover, analyze, redesign, execute and monitor business processes [26]. Since BPM provides this large spectrum of methodologies, the areas of application of business processes can be found in several disciplines: While business processes originate from economic business workflows, the domain of BPM is also applicable to engineering processes in industry without any changes to the concept of BPM but occasionally with some extensions. Safety-critical processes and high volumes of data require a reliable environment for processes which can perfectly be managed by

business processes within BPM.
In figure 2.2, we can view BPM as continuous life-cycle comprising the following phases:

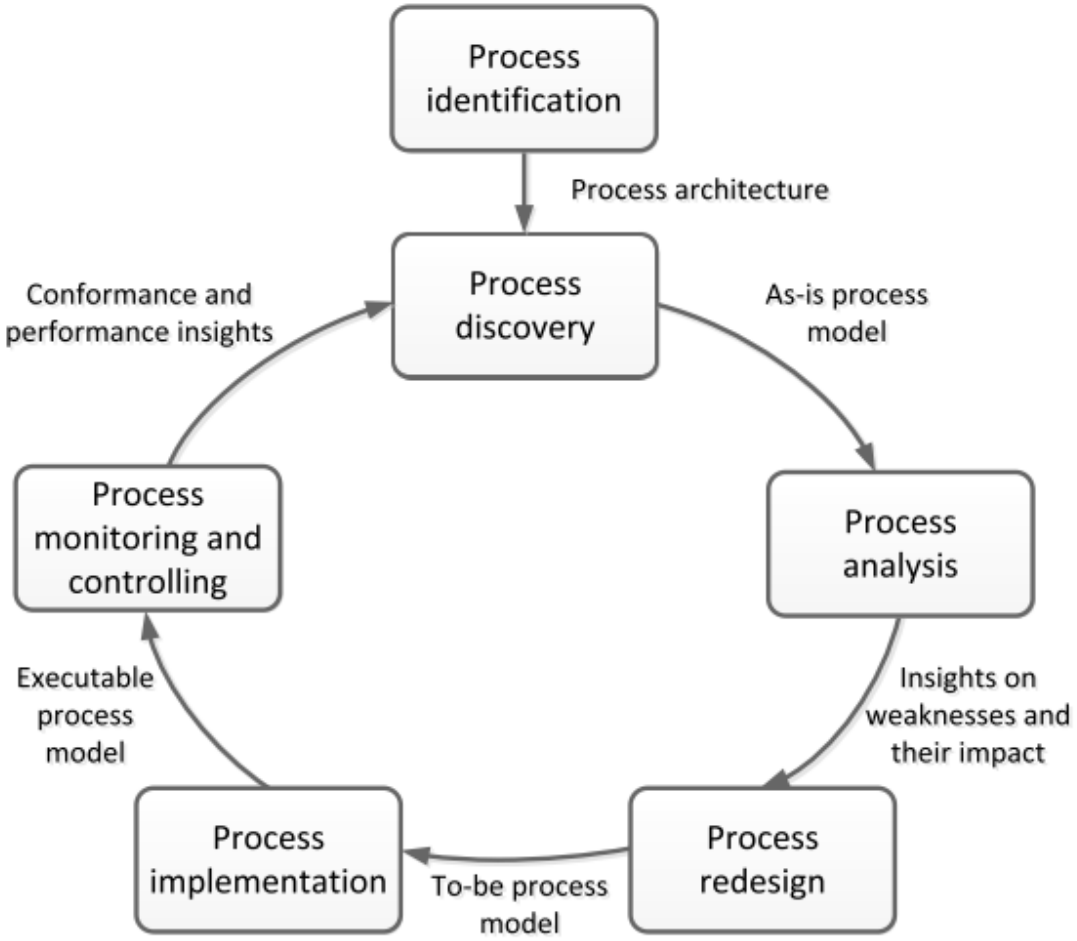


Figure 2.2: Lifecycle of BPM [26]

1. **Process identification.** In this phase, all processes relevant to the problem are identified, delimited and related to each other in order to obtain a clear overview of the processes of an organization.

2. **Process discovery.** This phase is also called process modeling which is responsible for documenting the current state of the relevant processes.

3. **Process analysis.** This phase is responsible for identifying, documenting and quantifying processes regarding prioritization of their impact in terms of estimated effort required to resolve them.

4. **Process redesign.** In this phase, the goal is to improve processes by identifying changes to the process which were identified in the previous phase. To this end, the output of this phase is a to-be process model in a suitable process language, which serves as a basis for the next phase.

5. **Process implementation.** In this phase, processes are prepared and performed regarding organizational change management and process automation. Organizational change management refers to the set of activities required to change the working of all participants involved in the process. Process automation refers to the development and deployment of IT systems.

6. **Process monitoring and controlling.** During runtime of the process, relevant data are collected and analyzed in order to determine how effective the process performs. Also bottlenecks, recurrent error or deviations are identified to undertake corrective actions in the next cycle.

In this thesis, we aim to design a business process by transforming textual descriptions into a diagram providing viewers a quick overview and easier understanding in a comprehensive way. This includes the phases from 1-4. Furthermore, we particularly focus on phase five and six by deploying the designed process to a tool environment enabling process automation.

2.2 Process Model and Description Languages

For modeling complex systems, there are different types of classification of enterprise architecture models which can be distinguished by the layer, where they are used. For instance, process models (on business process layer), information models (on information layer), technicals models (on technology layer). On the business process layer, the following technologies can be used [43]:

1. **Flowcharts.** In the basic form, *flowcharts* consist of rectangles, representing activities, and diamonds, representing points in the process where a decision is made. The specific notation that is used consists of two types of node: activity nodes and control nodes. Activity nodes describe units of work that may be performed by humans or software applications, or a combination of them. Control nodes capture the flow of execution between activities. Additionally, there is often a node that is not supported by all process modeling languages - the event node. This node provides information that something may or must happen, within the process or in the environment of the process, that requires a reaction - for example, the arrival of a message from a customer asking to cancel their purchase order. [26]
   One type of flowcharts is the language Event-driven Process Chains (EPCs), developed by ARIS [20]. The similarities with flowcharts differ in that they treat events as first-class citizens because the notation does not rigidly distinguish between output flows and control manners or between places and transitions [57].

2. **Data flow diagrams.** Data flow diagrams focus on showing relations between components instead of sequence of activities with operational information. It is often used as a preliminary step for creating an overview of a system visualizing data processing [14].

3. **IDEF0.** IDEF (Integration definition) is a set of modeling languages which involves the versions IDEF0, IDEF1X, IDEF2 etc. For modeling business processes IDEF0 and IDEF3 are used, where IDEF0 is responsible for modeling functions and IDEF3 expands IDEF0 targeting at business process modeling with the purpose to focus on object state or process [43].

4. **UML.** There exist several extensions of *flowcharts* like cross-organizational flowcharts where the flowchart is divided into so-called swimlanes that denote different organizational units like different departments in a company [26]. These flowcharts are similar to the *Unified Modeling Language (UML)* regarding the UML Activity Diagrams. The strength of UML Activity Diagrams lies in providing symbols to capture data objects, signals and parallelism among other aspects.

5. **BPMN.** UML enables to structure process models by its underlying exchange format XML (Extensible Markup Language) making them understandable for machines. This method is used by the Business Process Model and Notation (BPMN) as process description layer together with a graphical layer for process visualization. This suitable combination makes BPMN attractive for the BPM community since it unifies both business users and technical experts.

6. **BPEL and XPDL.** BPEL (Business Process Execution Language) is the first language that focuses on providing both a language for specifying business processes and a precise operational semantics for executing processes specified in this language [3]. By the definition of BPEL, the graphical representation is restricted to some situations, e.g. loops. XPDL (XML Process Definition Language) is complementing the graphical restrictions of BPEL. The evolution of the combination BPEL and XPDL leads to the first version of BPMN, described in section 2.3.

7. **RAL.** The description layer of BPMN allows to define any kind of the type *Resource* in the sense that it does not matter whether it is a person or an organization, and also no relationships can be established between them [17]. A recent extension of BPMN, RAL (Resource Assignment Language) addresses these shortcomings by providing expressions to describe relationships, e.g. Task A IS Clerk. In other words, RAL is a Domain Specific Language (DSL) aiming to ease the way resources are assigned in BPMN [18]. By this example we can describe that the human resource - in this case a clerk - is assigned to task A.

8. **Workflow languages, e.g. YAWL.** Beside the development of BPMN which is leveraged by the OMG [1], other process modeling process languages originate

[1] http://www.omg.org

from research efforts. Two of them are Workflow nets and Yet Another Workflow Language (YAWL)[62]. The former one, which is an extension of Petri nets[49] to model business processes, is simple and revolves around two elements: places and transitions. Places correspond to BPMN events, while transitions to BPMN activities [26]. YAWL[62] builds upon Workflow nets in the sense of adding specific constructs to capture the OR-join behavior, multi-instance activities, sub-processes and cancellation regions [25]. Although, it provides a much more expressive language, it retains the simplicity and intuitiveness of Workflow nets [25].
While the capabilities of YAWL gain more and more attention, it is not well established in industry yet as there is only a small amount of supporters [27].

Over the last years, a variety of languages have been developed for BPM and workflow management. BPMN is inspired by a number of previous languages, and nowadays BPMN is promoted and suggested as a standard for modeling and describing processes due to the benefit making process models understandable for humans as well as for machines [65].

2.3 Design & Modeling using BPMN

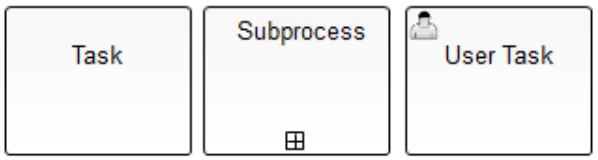
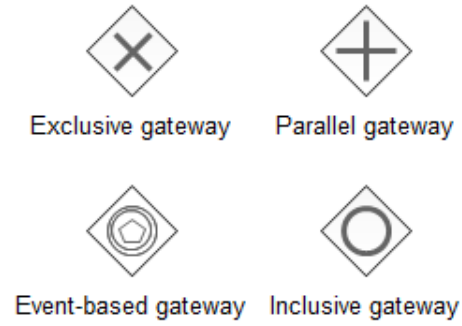
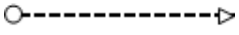
Design. The design of a process includes the identification of existing processes as well as the design of expected outcomes of processes. In detail, this means determining the representation of the workflow and also its factors within it like operational procedures, alerts and notifications, exceptions, etc.
The overall aim is to create a correct and efficient abstract design which improves the execution of processes in practice. These improvements could be in human-to-human, human-to-system or system-to-system workflows, and might target regulatory, market, or competitive challenges faced by the businesses [21].

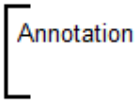
Historical evolution of modeling processes. Modeling is the process of creating models based on the theoretical design to perform a simplified representation of an empirical object, existing in the real world [43]. BPEL combines two approaches to specify business processes, namely a programmatic (or block-oriented) approach and a graph-oriented approach [26]. Concurrently, the negligence of the visual representation dimension in BPEL was the advent of BPMN 1.x since the focus lies more and more on language aspects and operational semantics aspects of business processes [26]. Trying to fill this gap, BPMN 1.x enables non-IT experts to communicate and mutually understand their models [45].
The attempt to bring both standards, namely BPEL and BPMN 1.x, under one single roof rises the following problem: “The BPEL standard is widely-spread by middleware vendors using BPEL engines for BPMN execution. To enable the execution of BPMN process models in BPEL engines, the transformation of BPMN process models to BPEL process models is needed but the metamodel underlying BPMN and the metamodel underlying BPEL are not identical - they are quite different.” [45] Furthermore, the

extension of BPEL4People [34] took five years and was not satisfying enough to meet the possibilities of the evolving BPMN [45].
Once BPMN 2.0 has been released by the Object Management Group (OMG) in 2011[50], it aims to replace BPEL [26]. This is a fundamental fact since BPMN 2.0 received for the first time a formal definition in the form of a metamodel [2]. In contrast to the previous versions, they only contained verbal descriptions of the graphic notation elements and modeling rules. "The BPMN 2.0 version's specification document has got comprehensive UML class diagrams that graphically show the features of the different BPMN constructs and their relationships." [2] Such a metamodel ensures a better accuracy and definition than strictly verbal definitions. The metamodel enables the development of an exchange format as a XML 1.0 [12] document for BPMN-models which can be then easily exchanged from one tool into another.

Modeling using BPMN. The visualization in BPMN is standardized by diagrams which consist of the following element categories: flow objects, connecting objects, swim lanes, and artifacts [43]. Table 2.1 depicts the basic constructs of BPMN.

Table 2.1: Basic BPMN constructs

Category	Elements	Notation
Flow Objects	Event	
	Activity	
	Gateway	
Connecting objects	Sequence flow	
	Message flow	
	Association	

Swim lanes	Pool and lanes	
Artifacts	Data object	
	Group	
	Annotation	

The elements listed in table 2.1 are the basic ones we are partially using for our illustration in fig. 2.3 but there are more extension elements available which can be found in the specification of BPMN [51]. In the following, we want to describe the meanings of the basic constructs.

An *Event* takes place once the trigger has been activated. The two basic types of events are start and end. Furthermore, a message event represents any kind of messaging, e.g. letter, emails or calls, and a timer event fires when its configured time expires.
An *Activity* represents the work that is executed. This can be defined as atomic, consisting of only one work as tasks or as compound with subprocesses. One specification we use in our basic example is the user task which describes the work that has to be done manually by a user.
Different *Gateways* split and join the workflow: An Exclusive (XOR gateway) gateway determines depending on the input which outgoing sequence will be executed. Parallel gateways (AND gateways) enable to execute outgoing activities in parallel. Event-based gateways are similar to the exclusive gateway but instead of routing based on the input it routes by the event that takes place next. An inclusive gateways (OR gateways) defines

that one or more of available paths will be taken. After each path has been completed they are reconnected.
Flows are responsible for connecting elements with each other: A *sequence flow* puts elements in the right order. A *message flow* represents the message exchange between different pools. *Associations* link artifacts with other elements.
A *pool* represents an organization as a whole. It separates different participants and its responsibilities from each other. *Lanes* help to separate activities of participants in one pool.
BPMN artifacts enrich diagrams with more information for the viewer of the diagram: *Data Objects* represent several types of data objects such as data inputs, data outputs and data sources. This provides more information to the viewer but does not change the process. *Groups* are used to unite similar activities in one shape. *Annotations* are used to add extra explanation information to viewers.

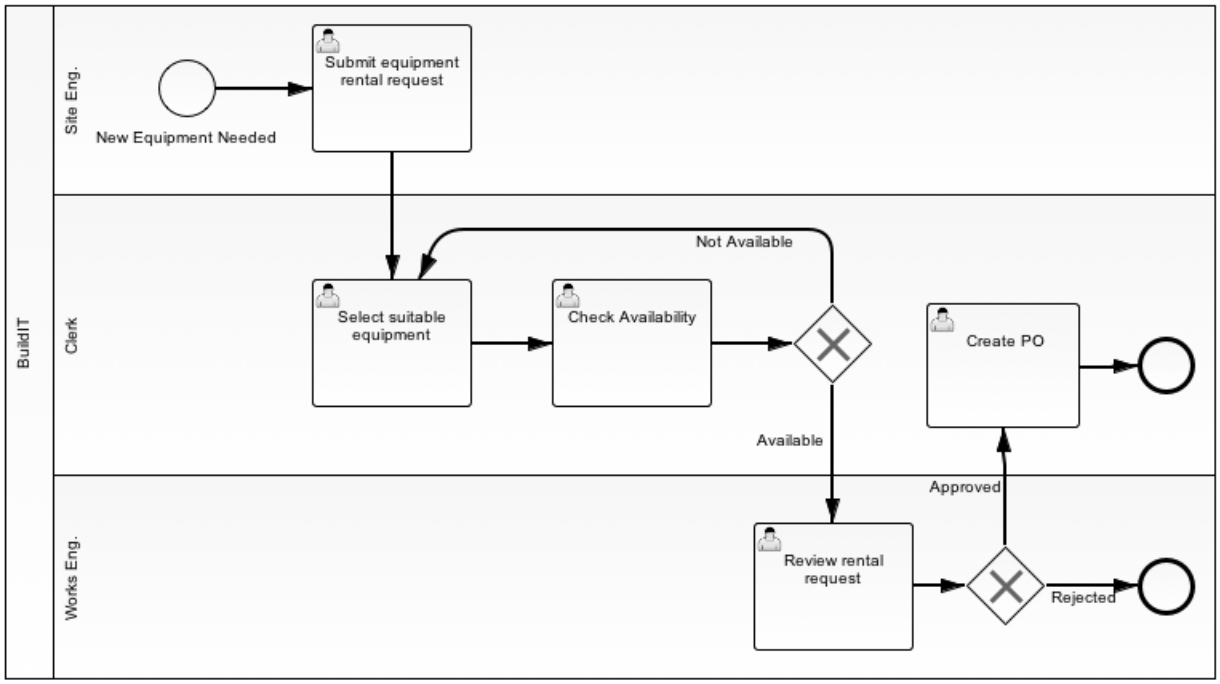


Figure 2.3: Process model of BuildIT

For illustration, we use a basic example - BuildIT - from [26], depicted in fig. 2.3, in order to demonstrate some of the functionality of BPMN 2.0. In the whole thesis, this example serves as basis for further illustration purposes since it provides a range of scenarios which are described in [26] in detail.
BuildIT represents a construction company specialized in public works such as roads, bridges, pipelines, tunnels, railroads, etc. The business process is started when a site

engineer needs to rent a piece of equipment. Furthermore, the site engineer has to fill in a form called *Equipment Rental Request*. The request is forwarded to the clerk at the depot and, after consulting the catalogs of the equipment suppliers, the most cost-effective equipment that complies with the request is selected. Afterwards, the clerk checks the availability of the selected item with the supplier by phone or e-mail. If the selected item is not available, then the clerk has to select an alternative equipment and checks its availability with the corresponding supplier again until a suitable piece of equipment is available for rental. The works engineer approves the rental request which leads to the end event by either a cancellation of the request (no equipment is rented at all) or in case the works engineer approves, a Purchase Order (PO) for renting the equipment is created by the clerk.
Listing 2.1 shows the source code according to the process model in fig. 2.3 in the form of a XML 1.0 document. XML enables the machine readability of the designed process for any tool that supports BPMN 2.0 in order to execute this process. This document is split into two parts characterized by BPMN-specific tags: The first part of the document, from line 6 to 35, refers to the process definition itself. Within this part, all tasks and its properties are defined to guarantee a sequential process execution order: From line 7 to 24, the lanes for the three roles (Site Eng., Clerk, Works Eng.) including its tasks are listed. Next, from line 25 to 34, all tasks with their outgoing sequences are defined. For example, in line 28 the start event shows the definition of its target reference - the user task *Select suitable equipment* which is set in 29 with its incoming and outgoing sequences. The user task in line 29 also shows that it is assigned to the role of a clerk. This assignment can be done at design time but once the process has started in a system changing this assignment or anything else in the diagram, this needs a complete redeployment of the process. The second part of the document, from line 36 to 55, contains the positions of each entity drawn on the graphical side which enables users to easily exchange processes between BPMN 2.0 supported tools without any loss of positioning information and errors.

Listing 2.1: BPMN 2.0 source code of the BPMN graph in Figure 2.3

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn2:definitions>
  <bpmn2:collaboration id="_Collaboration_2">
    <bpmn2:participant id="_Participant_2" name="BuildIT" processRef="Process_1"/>
  </bpmn2:collaboration>
  <bpmn2:process id="Process_1" isExecutable="false">
    <bpmn2:laneSet id="LaneSet_1" name="Lane Set 1">
      <bpmn2:lane id="Lane_1" name="Site Eng.">
        <bpmn2:flowNodeRef>StartEvent_1</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>submit_equipment_rental_request</bpmn2:flowNodeRef>
      </bpmn2:lane>
      <bpmn2:lane id="Lane_3" name="Clerk">
        <bpmn2:flowNodeRef>select_suitable_equipment</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>check_availability</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>ExclusiveGateway_1</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>create_po</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>EndEvent_1</bpmn2:flowNodeRef>
      </bpmn2:lane>
      <bpmn2:lane id="Lane_4" name="Works Eng.">
        <bpmn2:flowNodeRef>ExclusiveGateway_2</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>EndEvent_2</bpmn2:flowNodeRef>
        <bpmn2:flowNodeRef>review_rental_request</bpmn2:flowNodeRef>
      </bpmn2:lane>
    </bpmn2:laneSet>
    <bpmn2:startEvent id="StartEvent_1" name="New Equipment Needed">
      <bpmn2:outgoing>SequenceFlow_14</bpmn2:outgoing>
    </bpmn2:startEvent>
    <bpmn2:sequenceFlow id="SequenceFlow_14" name="" sourceRef="StartEvent_1"
        targetRef="submit_equipment_rental_request"/>
    <bpmn2:userTask id="select_suitable_equipment" camunda:candidateGroups="clerk"
        name="Select suitable equipment">
      <bpmn2:incoming>SequenceFlow_9</bpmn2:incoming>
      <bpmn2:incoming>SequenceFlow_13</bpmn2:incoming>
      <bpmn2:outgoing>SequenceFlow_4</bpmn2:outgoing>
    </bpmn2:userTask>
    ...
  </bpmn2:process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_1">
    <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="_Collaboration_2">
      <bpmndi:BPMNShape id="_BPMNShape_Participant_2" bpmnElement="_Participant_2"
          isHorizontal="true">
        <dc:Bounds height="461.0" width="961.0" x="156.0" y="0.0"/>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape id="_BPMNShape_Lane_2" bpmnElement="Lane_1" isHorizontal="
          true">
        <dc:Bounds height="149.0" width="931.0" x="186.0" y="0.0"/>
      </bpmndi:BPMNShape>
     <bpmndi:BPMNShape id="_BPMNShape_StartEvent_7" bpmnElement="StartEvent_1">
        <dc:Bounds height="36.0" width="36.0" x="304.0" y="50.0"/>
        <bpmndi:BPMNLabel>
          <dc:Bounds height="22.0" width="145.0" x="250.0" y="91.0"/>
        </bpmndi:BPMNLabel>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape id="_BPMNShape_UserTask_7" bpmnElement="
          submit_equipment_rental_request">
        <dc:Bounds height="80.0" width="100.0" x="408.0" y="12.0"/>
      </bpmndi:BPMNShape>
      ...
    </bpmndi:BPMNPlane>
  </bpmndi:BPMNDiagram>
```

```
56 </bpmn2:definitions>
```

CHAPTER 3

Resource Assignment and Allocation under Constraints in BPM

While a BPMN diagram allows setting predefined resources for executing tasks, this option restricts a project manager's freedom of flexibility in planning and scheduling resources[1] at runtime since any changes in the diagram require a complete redeployment of the process, i.e. the process has to be restarted. Particularly in complex process environments, scenarios of business processes can be described as a heterogeneous environment with several resources interacting and interfering with each other including several constraints. In such scenarios, a redeployment would end in misleading management with negative consequences for the end product of the process, e.g production delay or additional costs. Thus, the challenge is to plan resources for a project necessary to complete the given tasks and schedule them according to their availability over time in a flexible manner. Since similar challenges in other domains [22, 13] have been solved with Answer Set Programming (ASP), we assume using ASP in combination with an efficient solver provides suitable solutions for planning and scheduling resources in Business Process Management (BPM).

To this end, this chapter describes how resources can be planned and scheduled via resource assignment and allocation under constraints using ASP. In this context, we extract the tasks of the BPMN model using Petri Nets which enable transforming the tasks in BPMN into the semantics of ASP. Furthermore, we illustrate how these tasks together with constraints of available resources in an organizational model are solved by an ASP solver resulting in a suitable solution with optimal planned and scheduled resources.

[1] human resources are depicted as resources, e.g. project members

3.1 Answer Set Programming & Petri nets

ASP. Answer Set Programming [46, 13] is a declarative programming paradigm oriented towards difficult search problems based on the stable model semantics of Logic Programming [30]. Its search problems are declaratively defined with background knowledge in the form of rules which have the following syntax:

$$a_1 \vee \ldots \vee a_n :- \ b_1, \ldots, b_k, not\ b_{k+1}, \ldots, not\ b_m. \tag{3.1}$$

which is to be read as follows: If $b_1, \ldots, b_k$ hold and there is no evidence for $b_{k+1}, \ldots, b_m$ then at least one out of $a_1 \vee \ldots \vee a_n$ is true [53]. A rule of the form 3.1 corresponds to *Head* :– *Body* and is called a fact if the Body is $\top$, i.e. $k = m = 0$, and a constraint if the Head is $\bot$, i.e. constraints eliminate undesired solutions. Within a rule, the :– sign stands for facts and *not* for the "default negation". There are also further constructs used in order to express choices, cardinality constraints or optimization statements. For example, the cardinality constraint $1\{a, b, c\}2$ describes all possible subsets $\{a, b, c\}$ whose cardinality is between one and two.
The basic idea of ASP is representing a problem with a set of rules (a program), and employing an ASP solver for searching a satisfying assignment for the rules in the program. A programming methodology for ASP which is known as *generate - define - test* first finds all potential solutions using a choice rule, and then it tests each of them against constraints [47]. For illustration, listing 3.1 depicts an ASP program according this methodology as follows: First, two resources (P1, P2) and two tasks (T1, T2) are set as facts in the line one and two. The constrained choice rule in line 5 generates possible solutions which are limited due to the cardinality constraints $1\{\ldots\}1$: Both tasks can be executed by exactly one assigned resource at a time. Furthermore, the constraint in line eight describes that only one resource can be assigned to one task. For any reason, the resource P1 cannot execute the task T2 which is set by the constraint in line nine.

Listing 3.1: An ASP program with a Constraint

```
1 resources(p1;p2).
2 task(t1;t2).
3
4 %Generate
5 1{assign(P,T) : resources(P)}1 :- task(T).
6
7 %Test
8 :- assign(P1,T), assign(P2,T), P1!=P2.
9 :- assign(p1,t2).
```

In listing 3.1 we also make use of "grounding" which means that relating to the syntax in 3.1 b_i is a first-order predicate with variables within a rule, i.e. we obtain the set of its ground instantiations by replacing the variables with all possible constants occurring in the program.
Finally, as depicted in fig. 3.1, the program including all facts, rules and constraints are

taken by the ASP solver which computes solutions in the form of answer sets. Since the ASP solver is searching for satisfying assignments, the result of the ASP program in 3.1 consists of two satisfying answer sets, depicted in 3.2: In the first answer set, P2 is assigned to the tasks T1 and T2. In the second answer set, P1 is assigned to T1 and P2 is assigned to T2 because P1 cannot execute T2 due to its constraint.

Listing 3.2: The result of listing 3.1

```
Answer: 1
resources(p1) resources(p2) task(t1) task(t2) assign(p2,t1) assign(p2,t2)
Answer: 2
resources(p1) resources(p2) task(t1) task(t2) assign(p1,t1) assign(p2,t2)
SATISFIABLE
```

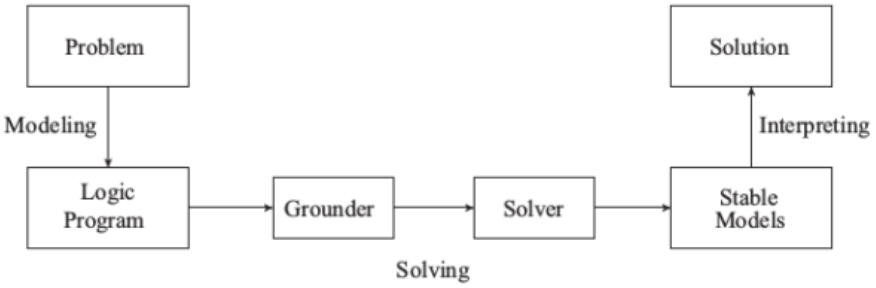


Figure 3.1: ASP solving process [30]

Petri nets. A Petri net [66] is a modeling language for the description of distributed systems. It is a directed bipartite graph consisting of two types of nodes - places (circles) and transitions (rectangles) - which are connected to each other by an arc in either one direction excluding the connections place to another place and transition to another transition.
Petri nets are used to model the behavior of systems in terms of a flow (flow of control, flow of objects or information) what makes it applicable to formally define BPMN models, since BPMN is also flow-oriented [24]. Fig. 3.2 depicts the mapping objects of BPMN to Petri net: An activity is mapped onto a transition with one input place and one output place. The transition which is labeled with the name of the activity or event, models the execution of the activity or event. A start and end event is mapped onto transitions with a silent signal when a process starts or ends.
In Petri nets, the flow is represented by tokens that flow through the system. At each place zero or multiple tokens may hold which can be represented as a state in a so-called marking function. The state of a Petri net changes when a transition fires due to an incoming token. When a transition fires it consumes all incoming tokens in places and produces tokens in each place which enables outgoing tokens.
Since we also intend to control the timing of firing tokens, we additionally adopt the concept of timed Petri nets [54] which associates a firing duration with each transition. According to the BPMN model of BuildIT in 2.3, fig. 3.3 represents the corresponding timed Petri net graphically with respect to [24, 23]. We note that for correct transforming

the semantics of BPMN to Petri nets, XOR Split in BPMN is mapped to a Petri net module with two transitions as depicted in fig. 3.2. Furthermore, in fig. 3.3 transitions have been annotated to indicate the duration in generic time units (TU).

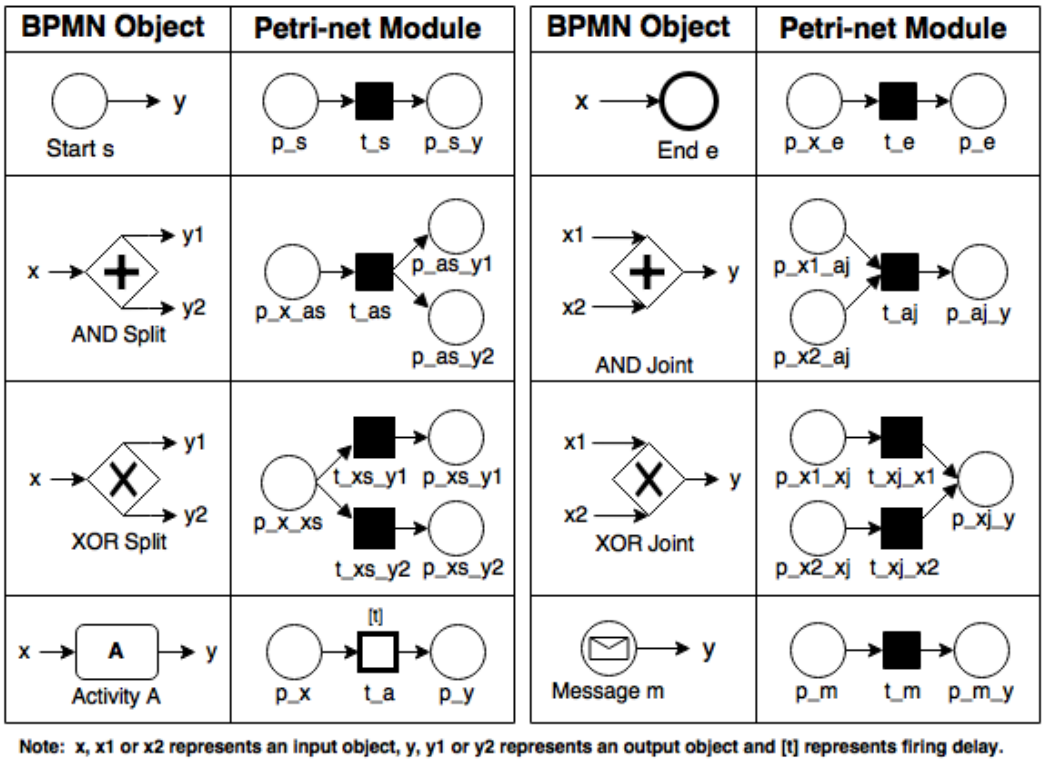


Figure 3.2: Mapping BPMN objects to Petri net objects [24]

3.2 Resource Assignment & Allocation

In this section, we describe how resources can be planned and scheduled with ASP and Petri nets in the example of BuildIT. In BPM we distinguish two steps of resource management with respect to [16]: resource assignment and resource allocation. In resource assignment, the conditions that must be fulfilled by resources in order to become candidates are defined. Candidates that meet these conditions are called potential performers which can be planned in advance. Resource allocation is the scheduling of activities and the selected potential performers while taking into consideration of both the resource availability and the project time. In addition, many temporal and precedence constraints have to be taken into account.
For assigning and allocating resources to compute suitable solutions, we apply an approach of automated resource allocation [32] which is based on an ASP program, depicted in Appendix B - An ASP Program for Resource Allocation. The input for this program

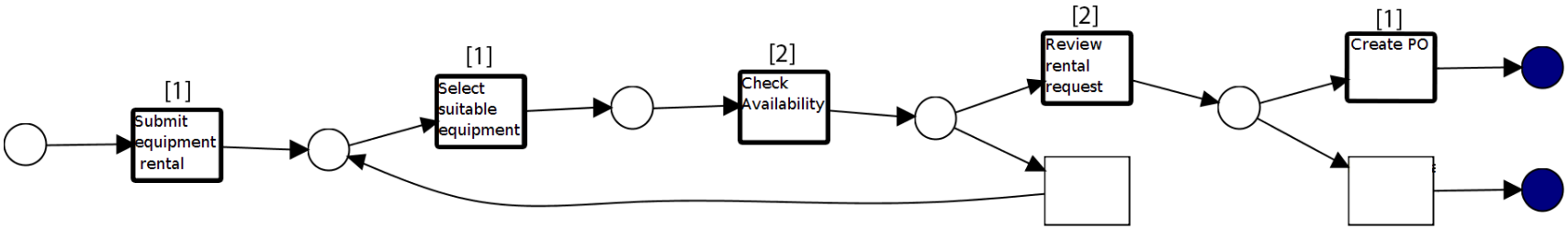


Figure 3.3: A timed Petri Net of the BPMN model of BuildIT in 2.3

is described with ASP facts enabling resource constraints in the program, therefore the input components are constraints. The following components serve as input:

1. A timed Petri net describing a set of activities and their estimated durations. Fig. 3.3 depicts the transformed timed Petri net according to the BPMN model in 2.3 with its activities and estimated durations. Listing 3.3 illustrates the encoded Petri net as input for the ASP solver. An *inPlace* is a place which points to that transition and *outPlace* is a place which is pointed to by that transition. For example, in listing 3.3 in the first line, inPlace points from the start event to the first task with the following parameter: the start *event new equipment needed*, the following task *submit equipment rental request*, and the name of the process. In line two, outPlace includes the following parameter: *p_serr_see* which can be seen as the link between *submit equipment rental request (serr)* and *submit suitable equipment (sse)*, the task *submit equipment rental request*, and the name of the process.

Listing 3.3: The Petri Net input for ASP solver

```
1 inPlace(p_new_equipment_needed, t_submit_equipmental_rental_request,bp_buildit
      ).
2 outPlace(p_serr_sse,t_submit_equipmental_rental_request,bp_buildit).
3 inPlace(p_serr_sse,t_select_suitable_equipment,bp_buildit).
4 outPlace(p_sse_ca,t_xorsplit_1,bp_buildit).
5 outPlace(p_sse_ca,t_select_suitable_equipment,bp_buildit).
6 inPlace(p_sse_ca,t_check_availability,bp_buildit).
7 outPlace(p_xorsplit_1,t_check_availability,bp_buildit).
8 inPlace(p_xorsplit_1,t_xorsplit_1,bp_buildit).
9 inPlace(p_xorsplit_1,t_review_rental_request,bp_buildit).
10 outPlace(p_xorsplit_2,t_review_rental_request,bp_buildit).
11 inPlace(p_xorsplit_2,t_create_po,bp_buildit).
12 inPlace(p_xorsplit_2,t_end,bp_buildit).
13 outPlace(p_end_po_created,t_create_po,bp_buildit).
14 outPlace(p_request_rejected,t_end,bp_buildit).
15
16 activityTransition(t_submit_equipmental_rental_request,bp_buildit).
17 activityTransition(t_select_suitable_equipment,bp_buildit).
18 activityTransition(t_check_availability,bp_buildit).
19 activityTransition(t_review_rental_request,bp_buildit).
20 activityTransition(t_create_po,bp_buildit).
```

2. An organizational model describes a set of resources and their roles in an organization, as in fig. 3.4 (1) with the corresponding ASP description in listing 3.4, from line 1 to 6. The description of this set defines the conditions for resources to execute the intended activities since particular activities are restricted to certain roles. We note that this corresponds to the execution's responsibility in BPMN where it is illustrated by lanes. Once the activities are set to a role from line 8 to 12 in listing 3.4, resources can be assigned to roles. This part of the input determines the resource assignment, that means which roles can be assigned to which activities.

Role	Resource
Site Engineer	Amy
Site Engineer	Glen
Clerk	Tom
Works Engineer	Drew
Works Engineer	Evan

(1)

Resource	Activity	TU
Glen	Submit equipment	2
Tom	Select equipment	1
Tom	Check Availability	2
Drew	Review Request	1
Evan	Review Request	3
Tom	Create PO	1

(2)

Role	Activity	TU
Works Eng.	Review Request	2

(3)

Figure 3.4: An organization model for the example of BuildIT (time in Time Units (TU))

Listing 3.4: Constraints of the organizational model

```
1 hasRole(amy,requester).
2 hasRole(amy,site_engineer).
3 hasRole(glen,site_engineer).
4 hasRole(tom,clerk).
5 hasRole(drew,works_engineer).
6 hasRole(evan,works_engineer).
7
8 canExecute(site_engineer,t_submit_equipmental_rental_request,bp_buildit).
9 canExecute(clerk,t_select_suitable_equipment,bp_buildit).
10 canExecute(clerk,t_check_availability,bp_buildit).
11 canExecute(works_engineer,t_review_rental_request,bp_buildit).
12 canExecute(clerk,t_create_po,bp_buildit).
```

3. A set of role-activity constraints. In listing 3.4, these constraints determine the ability of roles executing certain activities.

Listing 3.5: Time constraints for activities and resources

```
1 activityDuration(t_submit_equipmental_rental_request,bp_buildit,1).
2 activityDuration(t_select_suitable_equipment,bp_buildit,1).
3 activityDuration(t_check_availability,bp_buildit,2).
4 activityDuration(t_review_rental_request,bp_buildit,2).
5 activityDuration(t_create_po,bp_buildit,1).
6
7 resourceActivityDuration(glen,t_submit_equipmental_rental_request,bp_buildit
      ,2).
8 resourceActivityDuration(tom,t_select_suitable_equipment,bp_buildit,1).
9 resourceActivityDuration(tom,t_check_availability,bp_buildit,2).
10 resourceActivityDuration(tom,t_select_suitable_equipment,bp_buildit,1).
11 resourceActivityDuration(tom,t_check_availability,bp_buildit,2).
12 resourceActivityDuration(drew,t_review_rental_request,bp_buildit,1).
13 resourceActivityDuration(evan,t_review_rental_request,bp_buildit,3).
14 resourceActivityDuration(tom,t_create_po,bp_buildit,2).
```

4. A set of temporal constraints. According to fig. 3.4, in (3) we have information of how long a role is intended to execute an activity, e.g. listing 3.5 on line 4 corresponds to the constraint in fig. 3.4 (3). On the other hand, in fig. 3.4 (2) time

units illustrate how much time a resource needs to execute an activity which is encoded from line 7 to 14 in listing 3.5. For illustration of the effect of constraints, in fig. 3.4 (3) the role-activity constraint *Works Engineer - Review request* is set with 2 time units (TU). In fig. 3.4 (2), there are two resources assigned to the role of Works Engineer - Drew and Evan. Both resources can execute the activity *Review Request* but they need different TUs. We will see in the result of the ASP solver which resource is finally allocated.

After the components (1-4) are prepared - this corresponds to the planning phase, the input for the ASP solver is complete:

1. Activities and their responsibilities (roles) are set.
2. Intended resources are assigned to the roles.
3. Constraints for roles and activities are described, i.e. it is clearly defined which resource can execute certain activities.
4. Time constraints for resources and roles are set, i.e. time units of resources describe how long they need to execute an activity.

The ASP solver takes the program in Appendix B - An ASP Program for Resource Allocation and its input in order to compute an optimal solution in the form of answer sets. Listing 3.6 shows the computed result of optimal scheduled resources according to all constraints in the input. As depicted in line 4, the resource *Drew* is allocated for the execution of the activity *Review Rental Request*, since *Drew* satisfies the time constraints for this activity, i.e. the activity must be executed within 2 TUs which is possible by *Drew* who only needs one TU, *Evan* would need three TUs.

Listing 3.6: A result of the solver assigning allocated resources

```
1 assign(amy,t_submit_equipmental_rental_request,0,1,1,bp_buildit,i1)
2 assign(tom,t_select_suitable_equipment,1,2,2,bp_buildit,i1)
3 assign(tom,t_check_availability,2,4,3,bp_buildit,i1)
4 assign(drew,t_review_rental_request,4,5,4,bp_buildit,i1)
5 assign(tom,t_create_po,5,7,5,bp_buildit,i1)
6 OPTIMUM FOUND
```

3.3 Conclusion

In this chapter we described the application of ASP aiming to achieve resource assignment and allocation under constraints in BPM. This approach has the beneficial effect that a BPMN model does not need any redeployment. At design time, only the process with its activities is modelled in BPMN. The phase of planning the resources to execute the activities is performed in the next step but is not considered at design time. As a consequence, when resources or constraints change over time, only the solver reruns and reallocates but independently of the process flow which enables flexibility, e.g. for the

project manager.
At this point, the approach is applied in a way that all constraints and inputs are directly provided. In order to integrate this into a BPM system, we have to search for a system which provides Application Programming Interfaces (APIs) enabling extensions that allows us to create this input from the running process model and process state and translate the respective input into the input format described in this approach.

CHAPTER 4

Evaluation and selection of a BPMS

This chapter presents an extensive survey of existing Business Process Management Systems (BPMS) in terms of various criteria including the ability to support resource allocation and extensibility to integrate external services, such as the automated resource allocation approach, outlined in chapter 3.
The chapter starts with the illustration of the overall architecture of a typical BPMS. It describes the structure of a BPMS. Certain criteria are selected in order to provide an expressive evaluation. In the evaluation, we analyse a handful of popular BPMS that are currently on the market. The evaluation details including support information can be found in Appendix A - The Evaluation of BPMS.
The result of the evaluation serves as a basis for selecting a BPMS which enables both monitoring engineering processes in an appropriate way and extensibility with external services, e.g. automated resource allocation as presented before.

4.1 Architecture of a BPM System

The architecture of a BPM System (figure 4.1) consists of four main internal components: (1) an Execution Engine, (2) a Process Modeling Tool, (3) a Worklist Handler, and (4) an Administration & Monitoring Tool. The connection to (5) External Services are provided via APIs. Furthermore, there can also be a Process Model Repository and Execution logs depending on the configured process environment.

These components (1-5) can be described as follows [26]:

Process Modeling Tool The overall target of a business process modeler is to create a process model with the ability to be executed from a BPM Execution Engine. Most

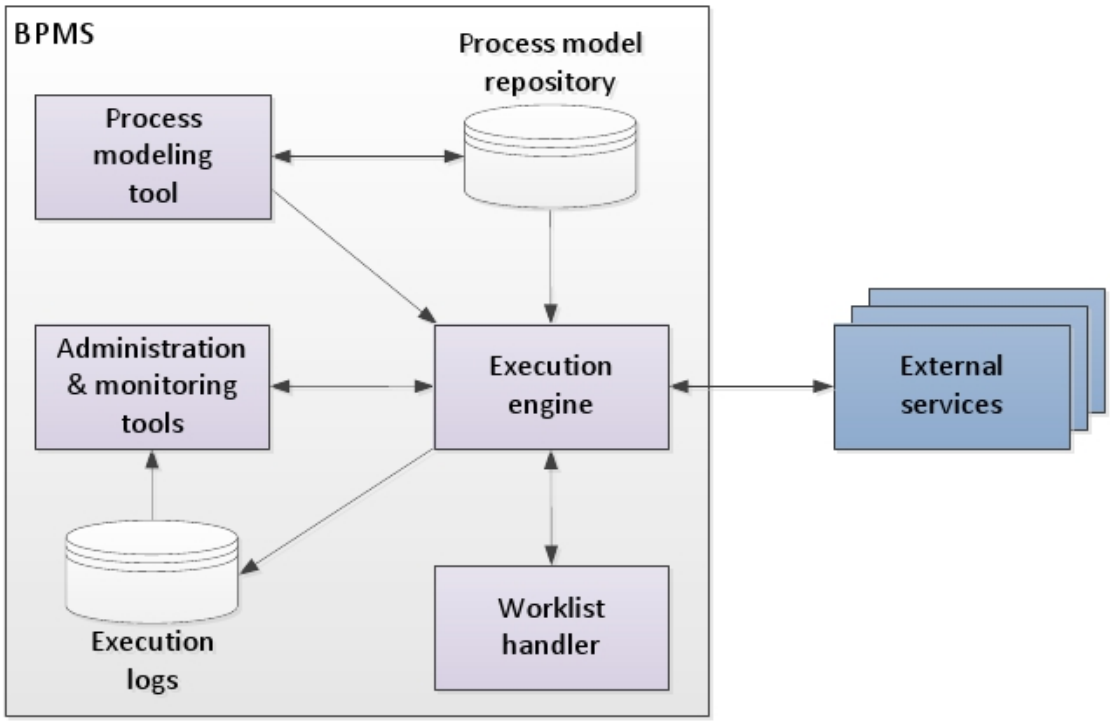


Figure 4.1: Architecture of BPMS [26]

of the advanced process modeler available on the market support the standardized Business Process Management Notation (BPMN). In its actual form, BPMN 2.0, the benefit is to provide a XML format in order to enable exchanging business processes between BPM Suites without any loss of quality.
A process modeler offers multiple facets: A modeler is implemented either in form of a Web UI or a Eclipse Plugin or as a standalone tool or some of these together. The result of the designing process, however, is a standardized business process in BPMN 2.0 notation ready to be executed.
In addition to the ability of designing business processes, process models annotate additional data, such as data input and output, participants, business rules associated with activities, or performance measures associated with a process or an activity [26].
The Process Model Repository enables storing, sharing and retrieving process models. Process models then can be deployed to the engine in order to be executed, either directly from the modeling tool or from the repository[1]. For the execution engine, the process model determines the temporal and logical order in which the activities of a business process have to be executed. Based on that, it then will be determined which work items will be generated, to whom they are allocated, and

[1]https://docs.camunda.org/manual/7.3/guides/user-guide/

what external services are called.

Execution Engine The core component of a BPM System is the execution engine. After the modeling is done, the engine creates executable process instances (also called cases). Afterwards, business processes can be executed to distribute work to process participants. The engine is also able to automatically retrieve and store data required for the execution of processes and delegate activities to software applications and services across the organization [26]. During execution, the engine is continuously monitoring the progress of different cases and coordinating which activities to progress next by generating working items. Once work items are created, they are allocated to resources which are qualified and authorized to execute them. More precisely, the engine interacts with several other components which will be discussed next.

Worklist Handler A worklist handler can be compared with a mail inbox through which process participants (i) receive a list of offered work items, and (ii) commit to these [26]. The process engine keeps track of which work items belongs to whom and make them available through the worklist handler [26]. This is represented as inbox where process participants can log into an electronic form and have a look of the current status of the process workflow and what work items are ready to be executed. Selecting and starting a work item from the worklist, this step is called check-out. The step of entering data into the form, and signal completion to the engine is called check-in [26]. When the check-in is done, the next work item has to be performed. It depends on the BPMS what features are provided but often participants can - to some extent - exert control over work items in the worklist. Additionally, the order and priority in which work items are executed can be displayed.

A worklist handler is a common feature of BPM systems to foster its efficient usage and acceptance within an organization.

Administration and monitoring tools Administration and monitoring tools are the key elements for operational matters of a BPM System. With such tools it is possible to administrate and to coordinate participants of business processes in order to guarantee a fluent progress.[2]

Administration tools are often required for staff decisions (e.g. for example, coordinating of exceptional situations like someone becomes unavailable to work because of illness or a vacation.) In such a case, administration tools can remove outdated work items from the system.

Monitoring tools are equipped with various process monitoring features. The progress as well as the performance of running business processes can be monitored. The execution of process models is recorded and stored in the execution logs. Some tools can perform analysis of historical data extracted from the execution logs and compare it with live data.

[2]https://docs.camunda.org/manual/7.3/guides/user-guide/

External Services In some cases it is useful to embed external services in the execution of a business process. Especially, to speed up the execution there are some activities that can be executed fully automatically. In this case, the execution engine calls external applications. For this purpose, the external application has to provide a service interface which the engine can interact with [26].
After completion of the request, the external service will return the outcome to the engine and signal that the work item is completed, which is also displayed in the worklist handler afterwards. This is also stored in the execution log. For activities that can be executed neither completely manual nor completely automated, such activities are performed by process participants with some form of automated support. The execution engine will invoke appropriate services with the right parameters, at the moment the participant selects a certain work item. For example, a Document Management System (DMS) would support the process participant to carry out a specific work item.

4.2 Criteria of BPM System & Suite Evaluation

The fast global market development makes companies to adapt to ever more challenges such as organizational changes regarding scheduling, human resource allocation and validation & monitoring. In this context, the importance of the usage of Business Process Management becomes more significant and therefore it is always relevant what purpose a BPM System/Suite should fulfill.
Basically, the purpose of a BPMS is to provide the ability to coordinate a business process in such a way that all work is done at the right time by the right resource. Some vendors extend a BPMS wrapping this in a completely vendor-specific environment offering a varying set of features for spanning the whole lifecycle from creating a process until its execution: a specific designing tool, a wizard enabling the execution of business processes, process intelligence functionality (e.g. advanced monitoring and process mining), complex event processing, SOA functionality, and integration with third-party applications and social networks [26] - vendors who are including these features use the naming convention of BPM Suite.
Herein, we are interested in the core features of a BPM System or a BPM Suite which resides in the execution of business processes. In our survey which is illustrated in Appendix A - The Evaluation of BPMS, different features were clustered together in groups for giving a fundamental proposition in certain areas. Both, clusters and its features are based on studies from practice [1], requirements according to Siemens, and the usage in academia. The following core clusters are taken into account to expose the BPMS's weaknesses and strengths. For more details regarding the sub criteria please have a look at Appendix A - The Evaluation of BPMS.

- General Information: General information about the product including product support.

- Process Modeling: Information about the modeling part of in a tool, i.e. which notation is used, in what format the modeled process is stored to further export or import it eventually.
- Process Realization: The step from modeling to execution - how it is represented and performed.
- Integration of Systems: Interaction with external data sources or external applications.
- Process Execution: The way how processes are executed.
- Resource Allocation/Scheduling: How resources are organized and scheduled.
- Process Controlling/Monitoring: How the control of processes and interaction during run-time is performed.

4.3 Tools

For our evaluation, we have selected a handful of tools that are currently on the market and they have also been selected in former evaluations [33, 31, 42, 58, 64].
In the following description of the tools we concentrate on the clusters which are relevant regarding the integration of resource allocation capabilities into a BPMS, described in the next chapter. We retrieved the information of the tool description from the corresponding product information website as well as from testing experiences. All details can be found in Appendix A - The Evaluation of BPMS.

4.3.1 Bizagi

General Information Bizagi[3] is a company that has its emphasis on developing BPM Suites. The software consists of three products, namely: (1) the Bizagi BPMN Modeler, (2) the Bizagi Studio, and (3) the Bizagi Engine. The Modeler is a freeware application to design processes in the BPMN standard. The Studio is also a freeware application with the aim to automate business processes and workflows. Inside the Studio, a framework (a Wizard) is responsible for organizing and building process applications, e.g. user interface, forms, business rules, etc.
The Engine is responsible for executing the previously modeled and automated processes. A great feature of Bizagi is that it provides portals by means of which end users can view processes even on mobile devices.
Bizagi software is known for its characteristics such as simple and intuitive application by the user (similar to Microsoft interface), short implementation time slots, easy adaptation to new conditions and the possibility to connect IT and Business tightly.

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

Process Modeling The company claims that they are supporting the standard notation, BPMN. As the name *BPMN Modeler* says, it should support BPMN notation. But in fact, you can graphically position BPMN entities which look like BPMN 2.0 entities but they are stored in a Bizagi internally XML file so that you can export this in the technical perspective only as a XPDL format. This creates a heavy disadvantage for exchanging created business processes to other tools. Nevertheless, it is possible to export processes to Visio and image formats and it offers a great pallet to publish processes such as Word, PDF, Wiki, Web or SharePoint.
Inside the Modeler, you have the option to define organizational structures and human resource constraints which cannot be exported due to the internal property configuration in a graphical user interface.
Before the process is going to be executed it validates the modeled process such that the execution of the workflow is guaranteed without any errors.

Process Realization In the Studio it is also possible to create proprietary Web UI forms and statistics for representing the process. Before starting a business process, a validation check is performed. The start of a business process is done by clicking on start button. The engine then follows step by step the modeled workflow which is stored in the database. Due to the encapsulated Engine, during run-time there is no programmable access to the Engine possible in the sense of retrieving information.

Integration of Systems Bizagi offers a Web service connector which is configured to consume SOAP Web services[4] or REST services[5] (i.e. as in figure 4.2, RESTful APIs, any SOAP service available through the ESB, on-premise, or available at the cloud) in an asynchronous or synchronous fashion [4].
An integration layer allows existing system, e.g. ERP, CRM, Core applications, legacy systems and ESB in general, to be integrated into Bizagi [5].
Organizational processes can use an interface wizard for Web and REST services invocations, and a feature called the Component Library for a Process-level integration (transactional) [5]. The Component Library provides extensibility options for both the business logic in the processes, and for the integration layer in the meaning of having a possibility to include specific APIs and connectors to external systems, applications or databases [5]. It acts as a middleware repository for custom-developed components which are registered in Bizagi by including its compilation file (.dll or .jar) [5].

Process Execution The execution will be performed by the Bizagi Engine. At design-time the designer creates constraints in the form of restricted access for some users at some time deriving from the organizational model that has been set up before: A security module allows to define a scheme of permissions which should guarantee the correct handling of defined process operations if living individuals are absent. The end-user can access the business process via log-in to the created Web UI.

[4] http://www.w3schools.com/webservices/
[5] http://www.w3.org/2001/sw/wiki/REST

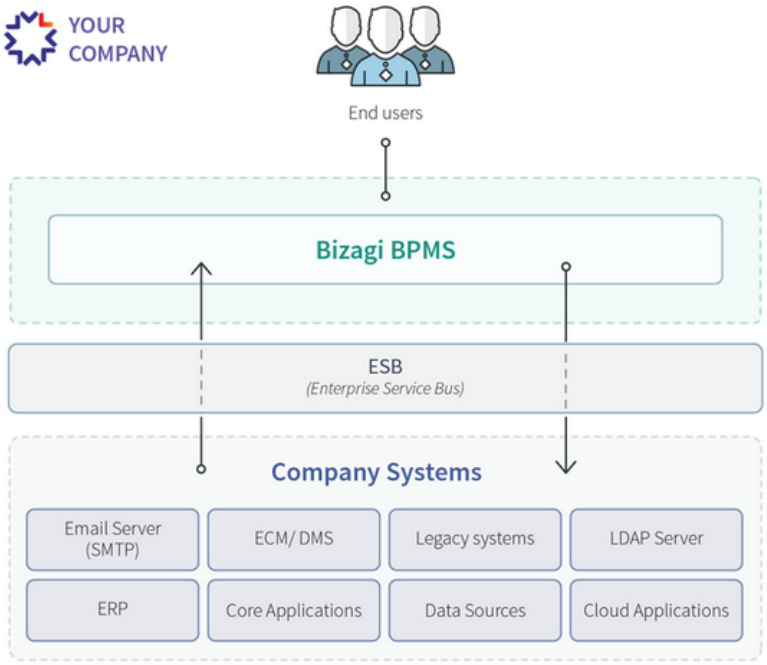


Figure 4.2: Integration and Extensibility of Bizagi [6]

Resource Allocation & Scheduling Bizagi organizes human resources internally within an organization module that defines permissions to create and administer a process and its components according to those properties.
âĂŃ Within this organizational module, "an organization stores the information related not only to the organizational structure of the members of a company and the definition of their characteristics (position, areas, groups), but also the characteristics that make them unique on a team and allow them to be active members in the processes of the application or applications (roles, skills, geographic location)." [10]
Furthermore, Bizagi provides business rules that can be implemented in various places throughout the process to perform actions, route the process, manage user interfaces and allocate users [7].
In the step of Work allocation, which is the fifth step of the process automation wizard performers are defined for each activity of the process. By means of performers, these are users which have qualities to be assigned to activities. Each user is allocated according to the rules in the organizational model. But only activities and events that interact with end users are available to be allocated. On the other hand, automatic activities like Script Tasks, Gateways or End Events cannot be selected to define allocation rules - these allocations must be defined separately for each activity and event [8].

"Once the activity or event is selected, the performers assignment window will be

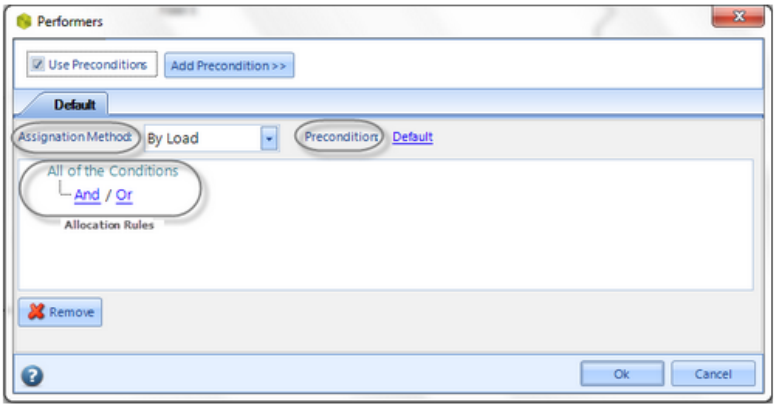


Figure 4.3: Performers assignment window

displayed (figure 4.3). In this window you can assign a resource by three types of conditions: Allocation Rules, Assignation Method and Preconditions." [9]
Next, allocation rules are built by specifying operands and operators as shown in figure 4.4. Preconditions are defined by expressions which can be configured by

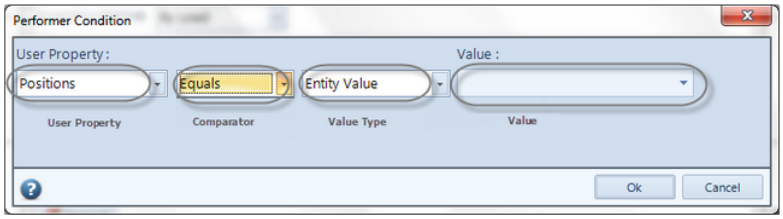


Figure 4.4: Specifying operands

clicking on the Select Expression option (figure 4.5): Boolean or scripting expressions according to the needs are defined. Further on, in an editor expressions can be defined as shown in figure 4.6.

Bizagi also provides a feature offering templates for recruitment and selection processes in order to accelerate this proceedings. These templates shall reduce the time scale in actions such as scheduling and collecting the results, e.g. technical test, assigned interviews, update of the candidate's list, etc.

Process Controlling Bizagi provides the possibility of controlling and monitoring the performance of processes through indicators which can be specifically defined by query forms and controlled by Bizagi BAM [11]. As mentioned previously, it is also possible to follow the progress of the process by mobile devices. But there is no possibility to interact with the Engine once it is started and also no reports are created. Thus no control information of concrete process instances can be retrieved.

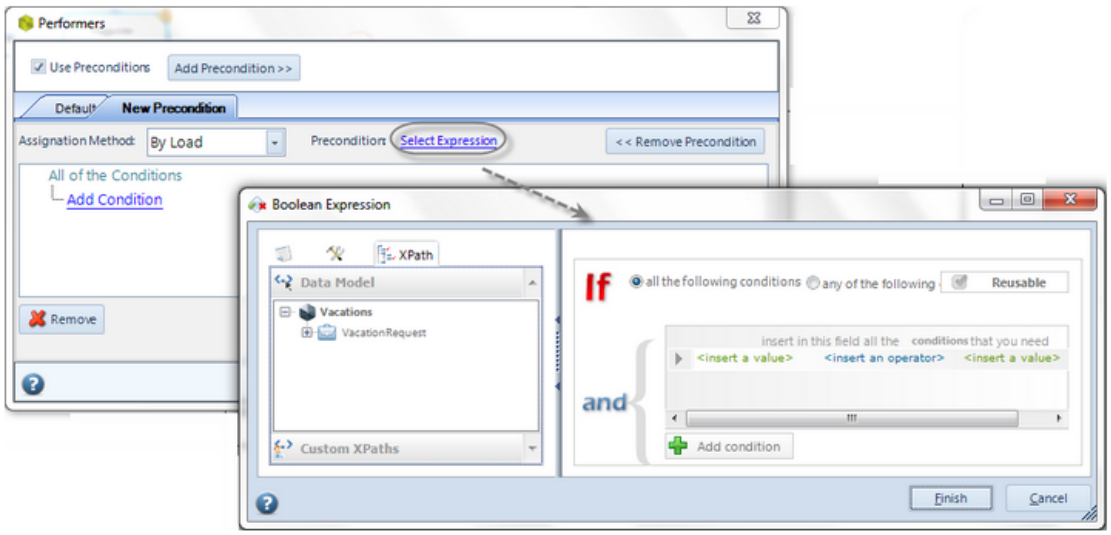


Figure 4.5: Configuring expressions (1)

Critics & Conclusion There is no doubt that Bizagi is a powerful BPMS. It has won several awards also due to the user friendliness and the performable combination of IT and business. Technically, (1) the missing support of BPMN 2.0 notation is a huge lack which makes it hard to exchange the created process with other BPMS, and (2) the created organizational structure cannot be exported.

4.3.2 Camunda

General Information Camunda[6] originates from several open-source projects and its provided BPMS software is still open-source. Camunda is following the policy to let the developer decide what environment for development is approached. Camunda provides a basic framework for BPMS which can be individually adapted to a customer's needs. The provided BPMS is based on a Java framework which can be extended in several individual ways. Due to the open-source policy Camunda is on top of researching new standards enhancing the execution of business processes.

Process Modeling Due to the fact that Camunda is an open-source tool, this also relates to the design time. The central aspect of Camunda is to fully support the BPMN 2.0 notation in the sense of extending this by individual code. The underlying XML file can be designed by all modelers that are able to export the syntax of this BPMN 2.0 file correctly. Including the accessibility to the XML code, this makes it possible to insert extensions such as HTML forms (with validation in the sense of maximal and minimum length of input strings), assignments to users or individual constraints can be set. Furthermore, the individual constraints can be

[6] http://camunda.org

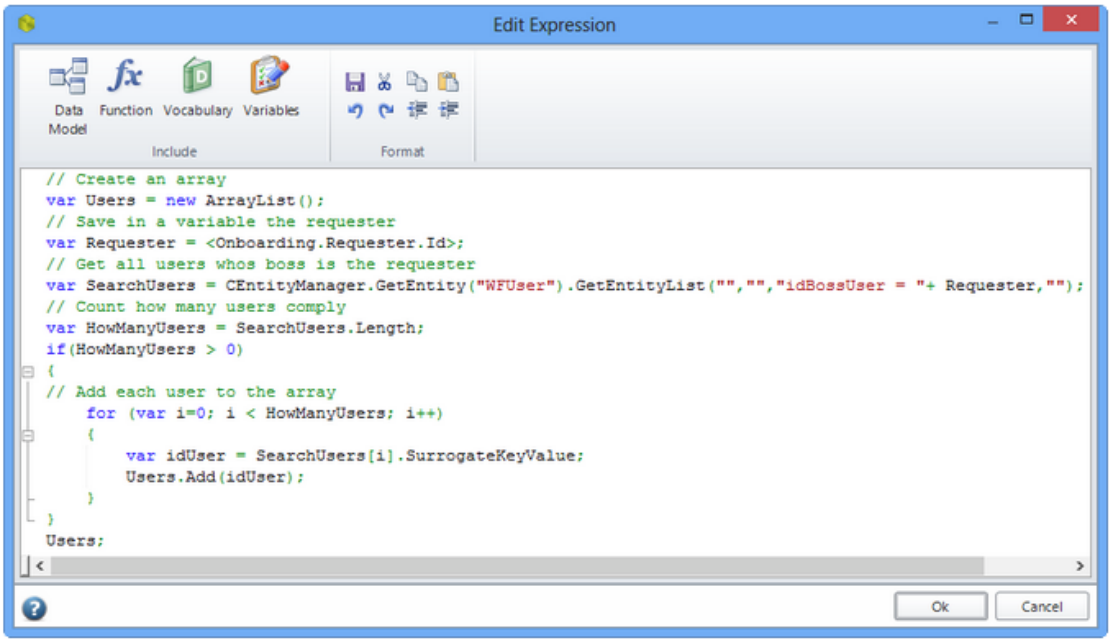


Figure 4.6: Configuring expressions (2)

used to describe organizational rules. Camunda recommends the Eclipse Plugin[7] or Signavio Modeler[8] which provide an editor for this purpose. Whereas the Eclipse Plugin is a remarkable designer and editor for designing a business process, the Signavio Modeler additionally provides features like model validation check which can become essential since the size of the process is increasing.

Process Realization Camunda is a Java-based framework which enables a totally different ways of deployment. This is possible due to the flexible architecture using Camunda for both as a standalone process engine server or embedded inside custom Java applications. The embeddability allows a tight binding of a Java Application to a custom application, and the standalone scenario is available for network solutions using the built-in REST API.
Once the process is deployed in the process engine after a build with Maven[9], for example, the deployed version can be started in several instances. When the process was changed and deployed once more it automatically turns out as a new version. A standard Web UI is given which is extensible and adaptable to a company's needs. There are three different areas that provide interacting: (1) The Tasklist Manager is comparable with a mail inbox where the tasks according to the assigned users are loaded. (2) The Cockpit is usually meant to be for the supervisor of a project to start a process, to overview and coordinate the tasks in a business process. (3)

[7] https://www.eclipse.org/bpmn2-modeler/
[8] http://www.signavio.com/de/
[9] https://maven.apache.org

The administration area includes the editing of permissions and authorizations for individual users and groups. There is no hierarchy, but a new user can be added to a group which has a specific role.

Integration of Systems Camunda offers numerous possibilities for integration: Runtime Container Integration, Spring Framework Integration, CDI (Context and Dependency Injection) and Java EE Integration.
Additionally to already available connectors for calling web services (SOAP and REST) from the processes, there are a variety of APIs to integrate applications. The entry point for all APIs provides the Process Engine as depicted in fig. 4.7. From this central API, several other APIs are inherited such as Task Service or Identity Service which allow retrieving and handling information in proprietary implementations.

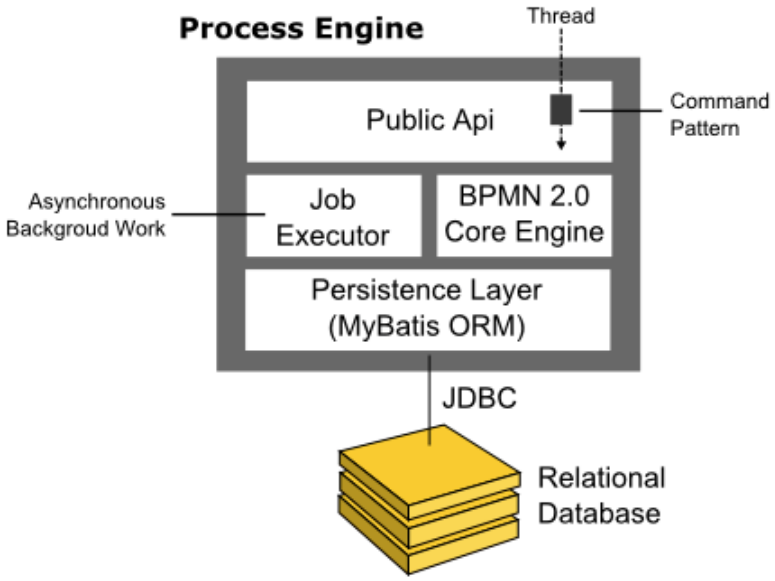


Figure 4.7: Process Engine Architecture of Camunda (source: Camunda)

Process Execution The execution starts when the designed process is deployed to the Process Engine which is a Java library responsible for executing BPMN 2.0 processes and workflows [19]. The core of the process engine is the BPMN 2.0 Core Engine [19]. It consists of a lightweight execution engine for graph structures (PVM - Process Virtual Machine), a BPMN 2.0 parser which transforms BPMN 2.0 XML files into Java Objects and a set of BPMN Behavior implementations [19]. A Job Executor is responsible for processing asynchronous background work such as Timers or asynchronous continuations in a process [19]. The persistence layer persists process instance states to a relational database. For this purpose the MyBatis[10] mapping engine for object relational mapping is used.

[10]https://mybatis.github.io/mybatis-3/

Resource Allocation & Scheduling In Camunda no hierarchical model for humans exists. Users can be created with individual permissions in the system, and they can also be added to a group.
In the BPMN 2.0 file, user and groups can already be assigned to tasks. In the property field we can define individual rule extensions. Defining such a rule can be performed via an editor, e.g. Eclipse Plugin, shown in figure 4.8. The Job

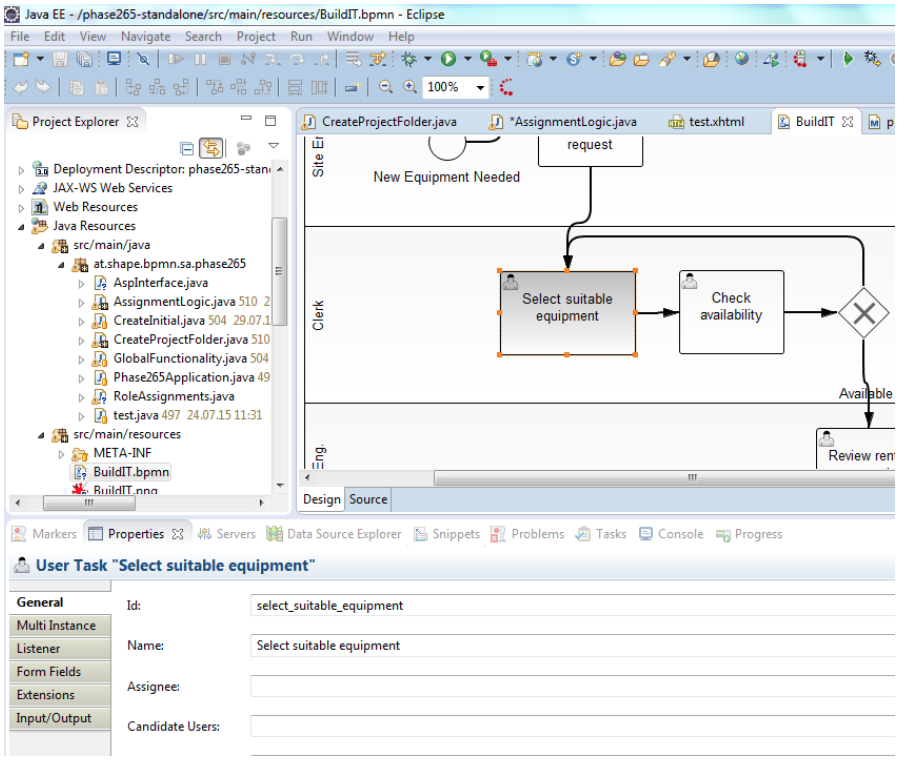


Figure 4.8: BPMN 2.0 Eclipse Plugin

Executor is one of the components of the Process Engine which is responsible for scheduling tasks in case of a timer event or a task marked for asynchronous execution is approached. In figure 4.9, two steps can be identified: while jobs are created during process execution, job acquisition and execution are the job executor's responsibility. The Job Executor is activated by default when the process engine boots. Jobs are then created for a range of purposes, namely asynchronous continuation to set transaction boundaries in the process, timer jobs for BPMN timer events, and asynchronous handling of BPMN events. During creation, jobs can receive a priority for acquisition and execution. Job Prioritization is meant for two types of use cases: (1) Anticipating priorities at Design Time, and (2) Prioritization as a Response to Runtime Conditions. The first one covers scenarios where jobs with the same business objectives have different priorities, e.g. a casual and a VIP customer. The second one deals with unforeseen conditions at runtime

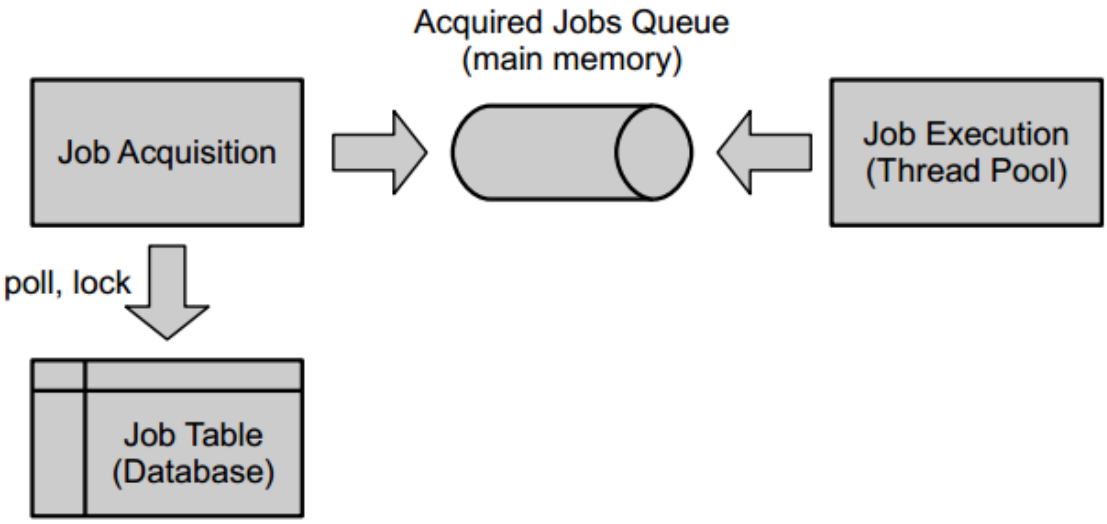


Figure 4.9: Illustration of the Job Executor

that cannot be dealt during process design, e.g. there are multiple instances of a process. To avoid occupying all the job executor's resources while waiting for the service to respond, the respective jobs' priorities can temporarily be reduced.
Job acquisition refers to the process of retrieving jobs from the database that are to be executed next. This requires a database together with properties determining whether a job can be executed. In BPMN 2.0, a job created for a timer event may not be executed before the defined time span has passed.
Due to the accessible API, the BPMS can be extended for resource allocation techniques. For example, JBoss[11] provides the service OptaPlanner[12] which returns optimal solutions for allocating resources automatically. This service will be described in subsection 4.3.3 in detail.

Process Controlling Camunda provides two dashboards to control processes and its execution. The Camunda Tasklist Manager can be seen as an inbox for each user where tasks assigned to the user will be displayed to further execute them. There is also a possibility to filter certain tasks to better overview them. This can also be used for the project manager's needs, e.g. filtering all users and the tasks according to them. This gives a certain control of every user's work.
The Camunda Cockpit gives an overview of each process and its instances: what process is running at the moment and how many instances of them are created. There is also the possibility to follow each instance in detail: what tasks are executed by whom at the moment. The interface also allows to intervene the assignments of tasks by changing them at run-time.

[11] http://www.jboss.org
[12] http://www.optaplanner.org

Camunda is open-source and therefore it also allows extending also these two dashboards. To name two interesting extensions: (1) Camunda BPM Workbench[13] and (2) Camunda Cockpit Statistics Plugin[14]. The first one is a web-application allowing to implement BPMN processes directly in the web browser. The remarkable features are a BPMN modeler, a script editing component and a debugger. These allow to design BPMN 2.0 processes and interactively explore their execution.
The second extension provides a statistics plugin for Camunda Cockpit. This plugin provides a set of charts helping to understand what is and what was going on with the engine.

Critics & Conclusion The fact that Camunda BPM is open-source should not be underestimated. This policy of Camunda BPM makes it attractive to experiment and develop new standards especially in cooperation with the OMG Group[15].
If the open-source environment might run into troubles, for this reason an enterprise solutions is available at different pricing levels. This includes issues for support, maintenance and indemnity.

4.3.3 jBPM

General Information jBPM[16] is one of the open-source projects from the KIE Group[17] (fig. 4.10) which belongs to the Red Hat JBoss Community[18] and develops open source projects for business systems automation and management. jBPM follows the paradigm of a proprietary BPM Suite, which provides a vendor-specific form of software development which is also known as "Zero-Code BPM" [61]. This BPMS tool bridges between business analysts and developers which means that on the one hand business users easily understand the implementation of business processes and on the other hand developers have alternative opportunities aside from the given jBPM tool to implement them. The BPMS tool includes the process from design to process management providing services for authoring users, deployment, process management, task management, and a dashboard.

Process Modeling The process of modeling is recommended to perform via the Eclipse Plugin[19] or the embedded Modeler within the jBPM tool. Both modelers result in a BPMN 2.0 file. The handy advantage of designing the process within the jBPM tool is the possibility to configure resource management for example. This is not possible when designing with the Eclipse Plugin. As in figure 4.11, jBPM consists of a Process Designer respectively the BPMN Designer, Data Modeler, a Form

[13] https://github.com/camunda/camunda-bpm-workbench
[14] https://github.com/camunda/camunda-cockpit-plugin-statistics
[15] http://www.omg.org
[16] http://www.jbpm.org
[17] http://www.kiegroup.org
[18] http://www.redhat.com/en
[19] https://www.eclipse.org/bpmn2-modeler/

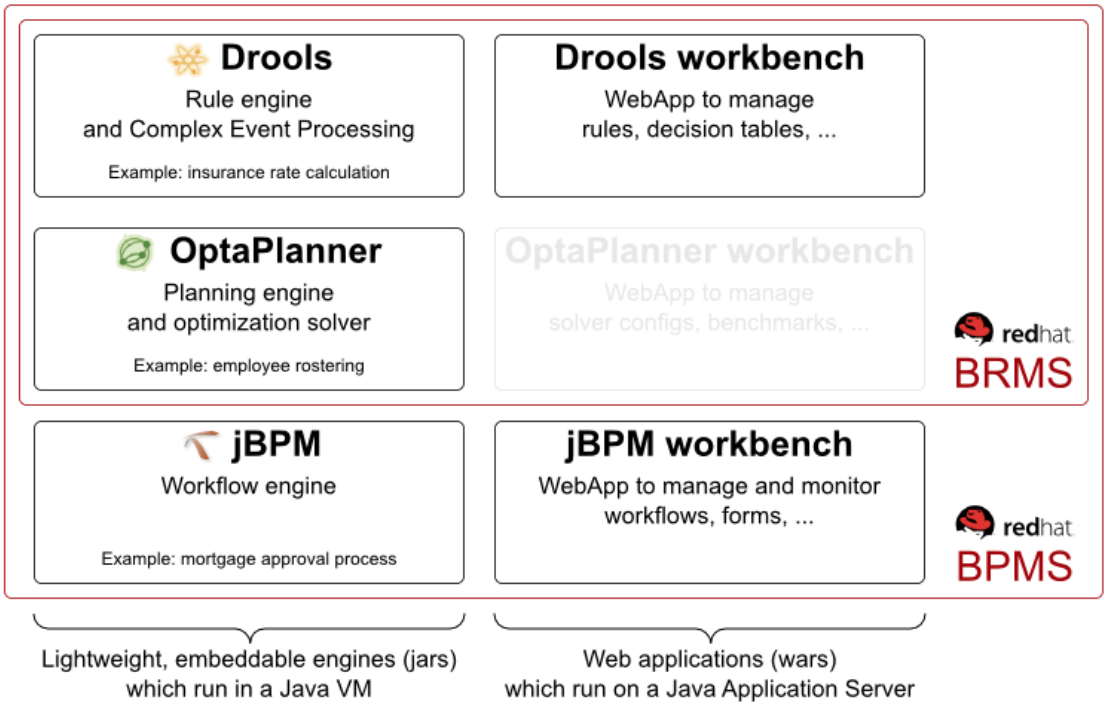


Figure 4.10: KIE functionality overview

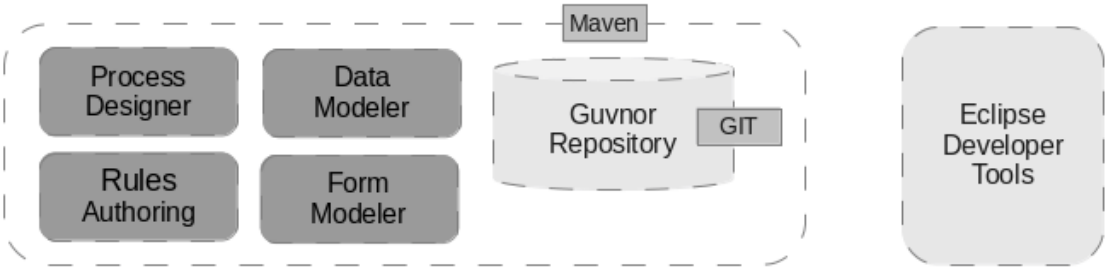


Figure 4.11: Modeling Components

Modeler, and a field for Rules Authoring: The Data Modeler includes modeling the data resources in the sense of databases, the Form Modeler enables the developer to design the presentation to interact with the running system and Rules Authoring provides the possibility of configuring rules for the business process. All assets are stored and managed on the Guvnor[20] repository and can be managed (versioning), built and deployed.

Process Realization Once the business process is designed, an integrated validation of the designed process alerts if there are some inconsistencies. After successful

[20] http://guvnor.jboss.org

deployment as a service it connects through the web-based UI or remote APIs. As in figure 4.12, the Human Task service is an optional service that will take care of the human task life cycle if human actors participate in the process. The run-time persistence will persist all states of all process instances and log audit information about everything that is happening during run-time. Also applications can connect through its Java API or as a set of CDI services, but also remotely through a REST and JMS API.

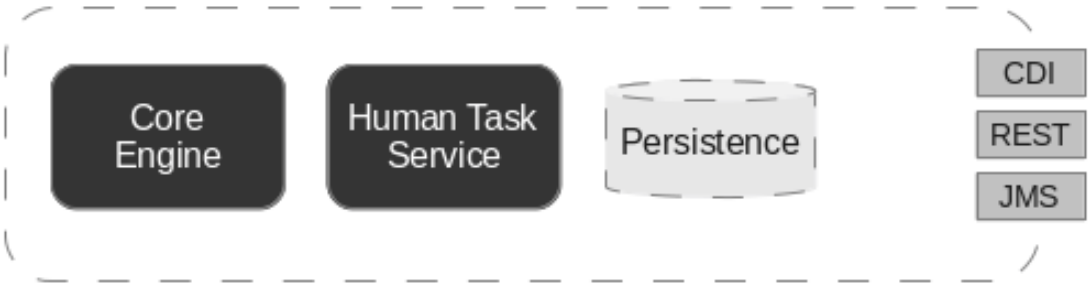


Figure 4.12: Execution Components

Integration of Systems As the KIE Group provides some extensions with proprietary functions like the management of human tasks (OptaPlanner[21]) or business rules (Drools[22]) or composing a dashboard (Dashbuilder[23]), the Java API also enables access to the core execution.

Process Execution Before the process starts, an integrated validation of the designed process alerts if there are some inconsistencies. The process instances are started in the process management which are then listed and the state of a specific process instance can then visually be inspected.
The human task management will be performed by the Tasks area. This acts like an email inbox where each user gets a list of all current assigned tasks to further complete them.

Resource Allocation & Scheduling jBPM does not provide any hierarchy model for humans. Users can be created and added to groups. Afterwards, user and groups can be assigned to tasks in the BPMN 2.0 file but normally in jBPM users are assigned to groups in the property window with some additional options, e.g. if a user task is skippable (figure 4.13). As in figure 4.14, jBPM provides an organizational structure to model departments and divisions. An organization unit can hold multiple repositories. These repositories are the place where assets are stored and each repository is organized by projects and belongs to a single

[21] http://www.optaplanner.org
[22] http://www.drools.org
[23] http://www.dashbuilder.org

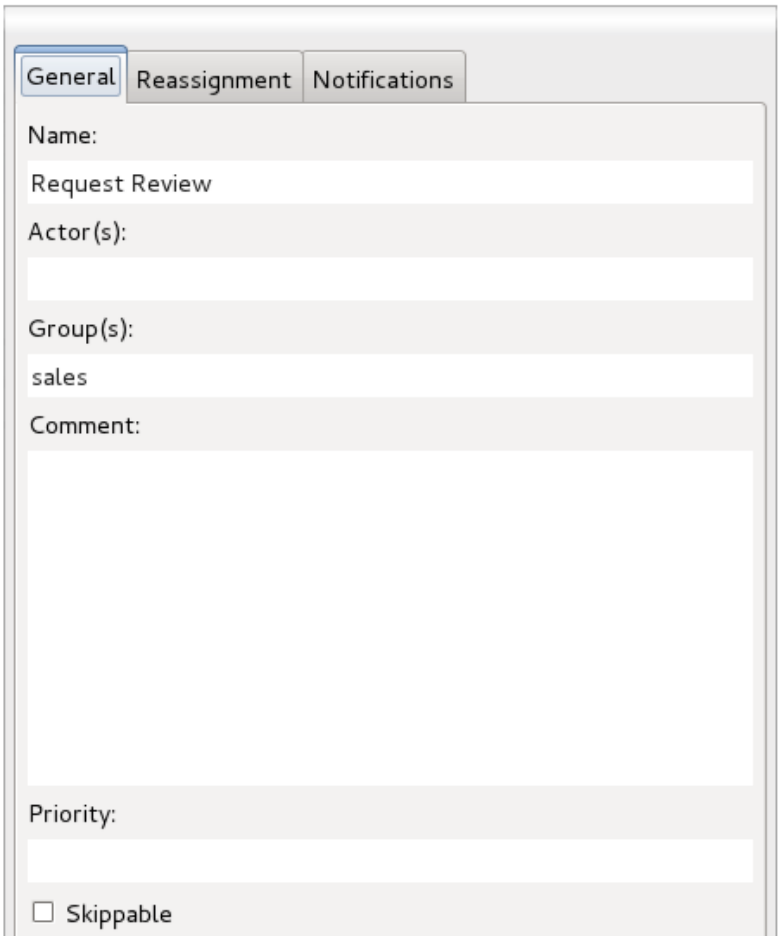


Figure 4.13: jBPM's property field of a User Task

organization unit. To administrate the system, jBPM defines the roles of admin, analyst, developer, manager, and user.
In fact, repositories are a Virtual File System based storage, that by default uses Git[24] as backend. Such setup allows the workbench to work with multiple backends and, at the same time, take full advantage of backend specifics features like in GIT case versioning, branching, and external access [37].

The KIE Group provides many services including OptaPlanner[25] which is responsible for optimizing business resource usage. It is a lightweight, embeddable constraint satisfaction engine and solves use cases such as employee shift rostering, agenda scheduling, vehicle routing, financial optimization and so on [38]. The integration of the API can be performed by Maven, Gradle, Ivy, Buildr or ANT.

[24]https://www.atlassian.com/git/
[25]http://www.optaplanner.org

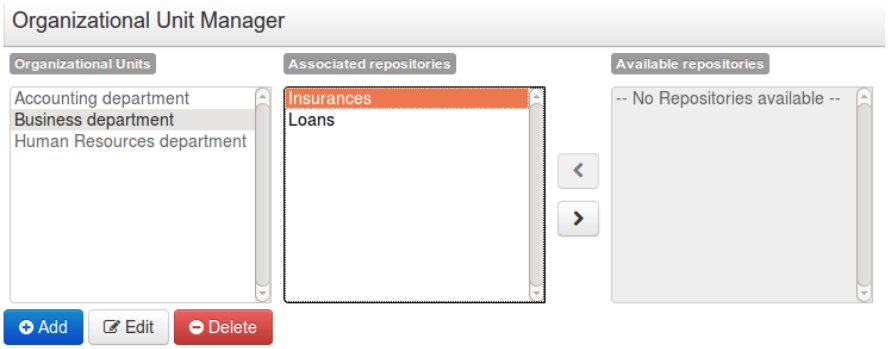


Figure 4.14: jBPM's organizational unit manager

The API of OptaPlanner provides to invoke the so-called solver factory which includes the solver configuration file (listing 4.1). This file is a XML file and consists of three parts: (1) domain model configuration, (2) score configuration, and (3) the optimization algorithms configuration. (1) is needed to make the Planner aware of the domain classes. In (2) the kind of algorithms are defined, and in (3) some details how the Planner should optimize the solution are fixed.
"The Planner supports three families of optimization algorithms: Exhaustive Search, Construction Heuristics and Metaheuristics. In practice, Metaheuristics (in combination with Construction Heuristics to initialize) are the recommended choice." [40]
To find the best solution, Drools[26] provides support in the form of a rule engine for calculating the score of a solution of a planning problem. With this it is possible "[...] to add additional soft or hard constraints such as "a teacher shouldn't teach more then 7 hours a day". It does delta based score calculation without any extra code. However, it tends not to be suitable to actually find new solutions." [39]

Process Controlling In jBPM, the areas of Process Management and Tasks include interaction with the business process. Process Management enables to start instances and provides the previously developed forms in an embedded way. While running the business process it is possible to follow the actual status with the current activities included. Tasks represents an inbox for the users of the system to perform the assigned tasks.
jBPM also provides a standard dashboard for processes and business. The Process Dashboard is a specific use case of a dashboard feed from data coming from a relational database via SQL queries related to the execution engine. These data can be exported, e.g. in Excel format.
The Business Dashboard provides a graphical presentation of running instances based on logs. In addition to that, in the KIE Group there is also the Dashbuilder[27] available which allows non-technical users to visually create business dashboards.

[26] http://www.drools.org
[27] http://www.dashbuilder.org

Listing 4.1: OptaPlanner configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<solver>
  <!-- Domain model configuration -->
  </solutionClass>
  </planningEntityClass>

  <!-- Score configuration -->
  <scoreDirectorFactory>
    <scoreDefinitionType>HARD_SOFT</scoreDefinitionType>
    <simpleScoreCalculatorClass>...</simpleScoreCalculatorClass>
  </scoreDirectorFactory>

  <!-- Optimization algorithms configuration -->
  <termination>
    <maximumSecondsSpend>120</maximumSecondsSpend>
  </termination>

  <constructionHeuristic>
    <constructionHeuristicType>FIRST_FIT_DECREASING</
        constructionHeuristicType>
      <pickEarlyType>FIRST_NON_DETERIORATING_SCORE</pickEarlyType>
  </constructionHeuristic>

  <localSearch>
    ...
  </localSearch>
</solver>
```

Critics & Conclusion jBPM has some relation to Camunda BPM but is still too different to call them similar products. Generally, jBPM disposes of the developers' and the business perspective. The business perspective is intended for users with less experience in development but more in business. To combine both perspectives is not trivial since business users must possess at least some technology knowledge. But at the same time this produces some restrictions for development opportunities because of the missing focus. That's why jBPM is popular as open-source option for a "Zero-Code-BPM".

4.3.4 Axon Ivy

General Information Axon Ivy[28] is one of the leading BPMS on the market [1]. Axon Ivy combines both the tool environment for developing and the BPMS for running processes - the whole BPM lifecycle is provided.

Process Modeling The Designer seems to be very similar to the Eclipse Editor (figure 4.16): Axon Ivy developed a designer that has the same appearance as Eclipse to a

[28] http://www.axonivy.com/at-de/

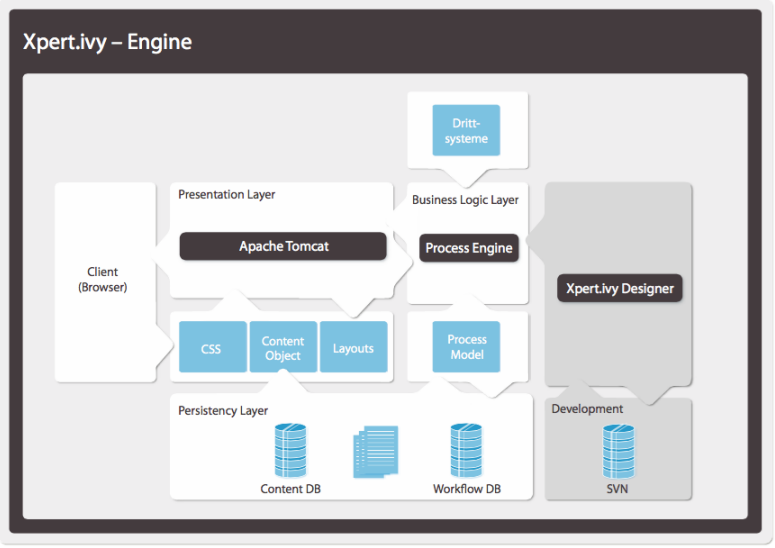


Figure 4.15: Axon Ivy's Engine

developer but it is configured just to work on business processes. Although this seemed to be focused on developers, it is possible to simply design a process to further execute it.
Axon Ivy promotes the software for a large variety of possibilities to develop a BPMS according to the customers' needs. As shown in figure 4.15, the process model is tightly connected to the persistence layer. This is also one remarkable feature to bind resources also in the process model. The following problem for this policy is that the underlying XML file is not compatible with a BPMN 2.0 standard anymore. This means that BPMN 2.0 processes can be imported into the designer but they cannot be exported as a BPMN 2.0 standard, just as an XML file.

Process Realization Before business processes are deployed, during designing phase, already designed parts of processes can be simulated and tested. The simulation is not a mandatory step but it helps for validation purposes. Afterwards business processes can be deployed.
Apart from modeling the business process, there is also an option to design individual forms of user interactions in both the HTML and GUI perspective.
Starting a business process after deployment, the designed process will be initialized by the engine. During run-time the process model can be modified, though. If modifications are performed the engine detects this and manages this intelligently leading into versioning of instances.

Integration of Systems Axon Ivy provides a variety of integration elements even in the design phase: start event element, call and wait element, database element, web service element, email element, and a script element for Java classes. Axon Ivy runs on on a Java platform and therefore every third party library for Java can be

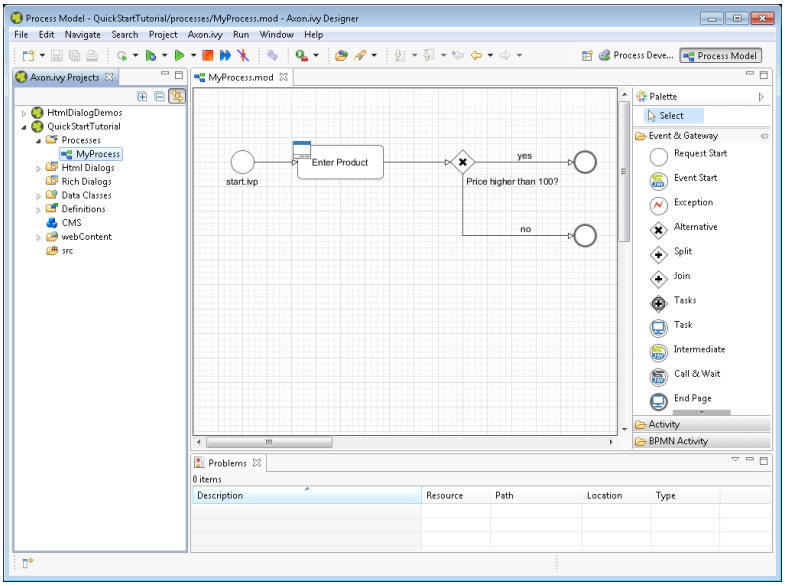


Figure 4.16: Axon Ivy's Designer

integrated.
In figure 4.17, the Enterprise Service Bus serves as one remarkable extension to the integration platform: The components include routing in the sense of transforming messages, transforming protocols, supply of connectors, high scaleable and distributed integration platform, independent and distributed deployment, support of Enterprise Integration Patterns (EIP), etc.
Additionally, Axon Ivy acts as service consumer - for this purpose the following techniques are intended to be used via ESB: Web Service, REST, Java Messaging Service (JMS), and Event Orientation.

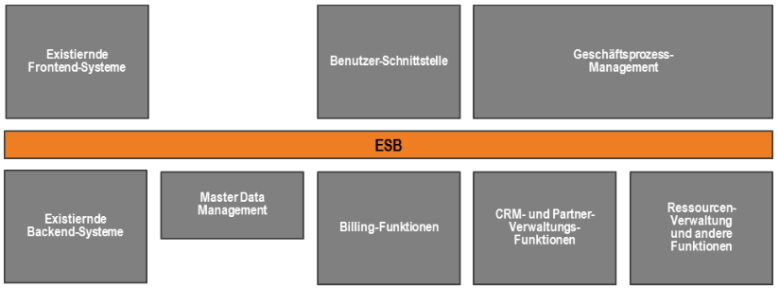


Figure 4.17: Integration via ESB

Axon Ivy also integrates the Business Rule Management System (BRMS) Drools[29] which is originally known from JBoss[30]. With this extension it is possible to

[29]http://www.drools.org
[30]http://www.jboss.org

evaluate business rules in a graphical element using the results of the engine.

Process Execution An integrated simulation helps the developer of the business process to simulate and debug the process before deploying.
Axon Ivy does not provide a user interface to manage tasks but it provides APIs to access and manipulate the tasks of an application. There are also some default HTML pages provided to adopt a user interface to the customer needs.

Resource Allocation & Scheduling During the design phase, in the task element the role or person is defined in a separate window. Additionally, as in figure 4.18, a hierarchy can be defined with permissions for each user included. Every user and also the whole hierarchy can be exported as XML file.
If the person or role is not defined in the design phase, at runtime alternatively by means of certain facts it is also possible to define the role in the process, e.g. if the order exceeds a value, the role is dynamically selected.

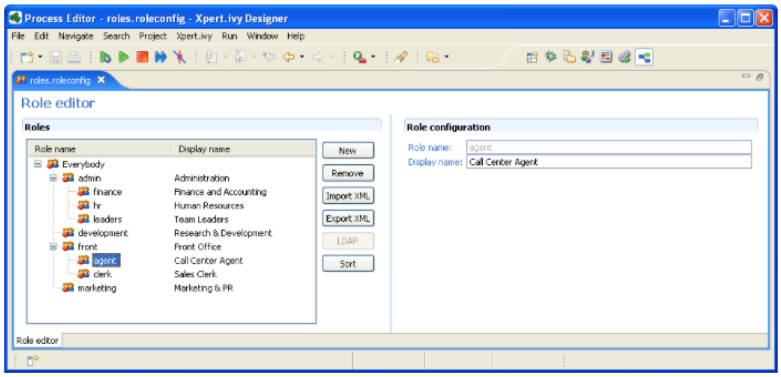


Figure 4.18: Role hierarchy in Axon Ivy

Axon Ivy has an integrated organizational tool which is responsible for mapping key roles and positions, while managing strengths, weaknesses, desires and needs of existing employees - Axon.TAM [35]. This tool supports the process of identifying employees' and candidates' potential and skills, so that key talents can efficiently recruited and leveraged [35]. By means of this module it should be possible to match employees to the growing needs of an organization.
For job scheduling, Axon Ivy uses a service called Quartz Scheduler. "This is a full-featured, open source job scheduling service that can be integrated with, or used along side virtually any Java application - from the smallest stand-alone application to the largest e-commerce system. Quartz can be used to create simple or complex schedules for executing tens, hundreds, or even tens-of-thousands of jobs; jobs whose tasks are defined as standard Java components that may execute virtually anything you may program them to do. The Quartz Scheduler includes many enterprise-class features, such as support for JTA transactions and clustering." [36]

Process Controlling The controlling of running processes can be done in two ways: The administration of roles and users can be done during runtime. There is a hierarchy, also with some permissions and this can be changed at any time. The second control is performed via a dashboard. This is started in the Axon Ivy tool and opens a port for a graphical representation of running processes with key figures included.

Critics & Conclusion Axon Ivy is the winner of the last study of the Fraunhofer Institut [1]. This relates especially to the whole picture of possibilities within the BPMS and the tool. It provides a kind of framework that is easily usable for the finance and marketing sector.
Axon Ivy obviously does not intend to cooperate for exchanging with other BPMS: The underlying format of the process design, which is just XML and not the standardized BPMN 2.0, gives a clear signal because it is not importable into other BPMS.

4.3.5 Oracle BPM

General Information Oracle provides a BPM Suite which encompasses all modules for modeling, technical implementation, execution and controlling of a company's process. All projects are build on Java EE and accessible over a web interface.

Process Modeling The design module has some similarity to the BPMN 2.0 Eclipse plugin but Oracle prefers an own developer environment. Nevertheless, with Oracle's designer it is possible to import and export BPMN 2.0 files and also validate them with the Oracle Debugger (fig. 4.19): It aims to provide debugging capability for processes, sub-processes, event sub-processes, and child sub-processes. And this debugger works like a java debugger providing debug actions like step-into, step-over, step-out, and resume.

Process Realization Before deployment of a project, the designed process might be debugged to proceed with a validating process. Afterwards, the project is built by Maven[31] for example.
Oracle provides a predefined design for the Web UI, but this is also extensible since the whole project is a Java project.

Integration of Systems Oracle BPM's strategy for integration is to leverage Oracle SOA, a best-in-class integration platform for adapters and tools: "Oracle's SOA platform is a proven, standards based approach to exposing and integrating databases and backend services using standard Java EE Connector Architecture (JCA) adapters. Some of the adapters included are: B2B, Business Activity Monitoring (BAM), database, EJB, File, FTP, HTTP, JMS, MQ and web service adapters." [15].

[31] https://maven.apache.org

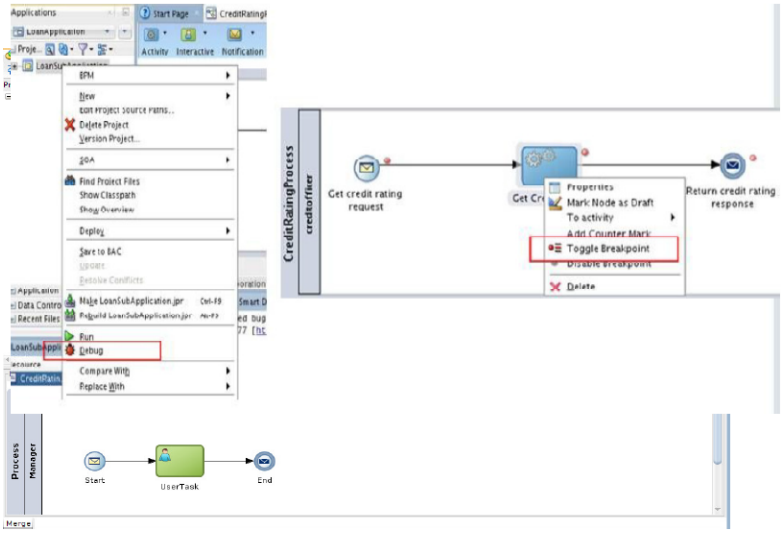


Figure 4.19: BPM debugging session

Process Execution After successful deployment, all processes are performed via the WebCenter Process Spaces which is part of the BPM Suite. Process Spaces is a web-based platform where business users only see and perform the tasks assigned to them or to their assigned groups based on the process model's design.

Resource Allocation & Scheduling In Oracle BPM Suite there is an extra area to model the needed organization, including a calendar and holidays. This model is stored in the actual project but cannot be exported. When the project is deployed and all information is handed to the engine it is possible to access the organization via the Oracle Identity Service.
To simplify the business rule authoring, Oracle BPM Suite provides a business friendly rules authoring capability. With the verbal rules editor (fig. 4.20), users can author rules using English-like sentences which are composed using as a set of user-defined business phrases. These phrases provide the business context and terminology and hide all technical constructs to make it easier to define rules for business users.

During run-time, Oracle uses process intelligence that provides sophisticated real time analytics to improve decisions. For this purpose Oracle Real-time Decisions (RTD) is used to automatically make decisions and provide recommendations.

Process Controlling The latest version of Oracle BPM provides a new Business Activity Monitoring (BAM) which gives business executives the ability not only to monitor service level agreements (SLAs) across various services and business processes in the enterprise, to correlate key performance indicators (KPIs) down to the actual business process themselves but BAM is remarkable for pinning specific risk definitions to KPIs and comprehensively visualize them in a watchlist [52].

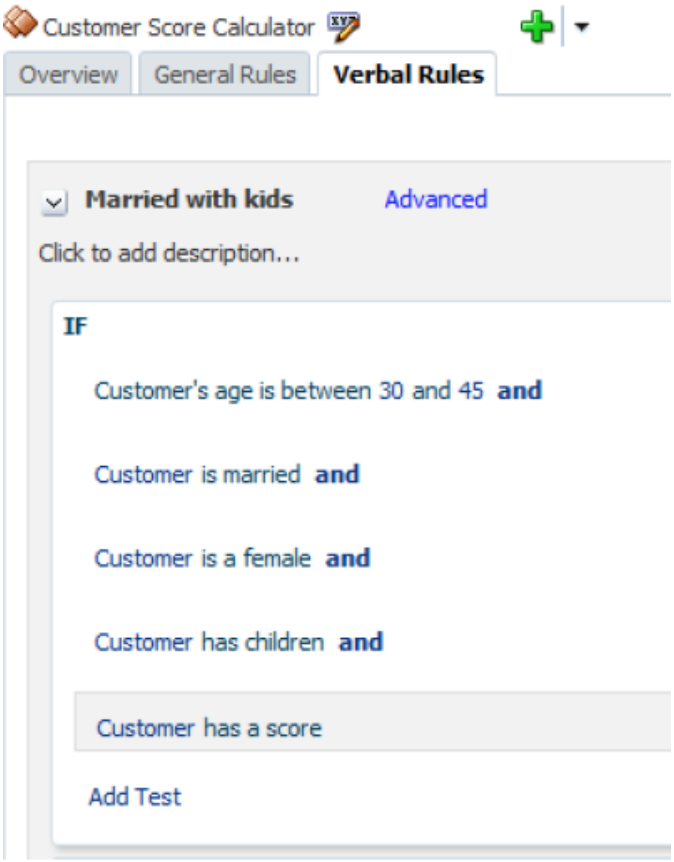


Figure 4.20: Verbal rules authoring

Critics & Conclusion Oracle's BPM Suite was ranked at the top of Fraunhofer's study 2014 [1]. Although it is one of the top BPM Suites, the massive offer of capabilities to create a BPM System only makes sense for large company structures.

4.3.6 ProcessMaker

General Information ProcessMaker[32] is a BPM Suite available in two versions: the community version which is open source and intended to be just for small companies, and the enterprise version which has different price levels depending on the features needed.
The whole BPM Suite - the process designer as well as the BPM System - is working in the internet browser window. It is especially designed for business users and process experts with no programming experience.

Process Modeling As shown in figure 4.21, the Designer is embedded within the BPM

[32]http://www.processmaker.com/

Suite. It is an intuitive drag-and-drop BPMN 2.0 process modeling tool. Apart from all functionality of BPMN 2.0, it is possible to import and export processes in the BPMN 2.0 format.

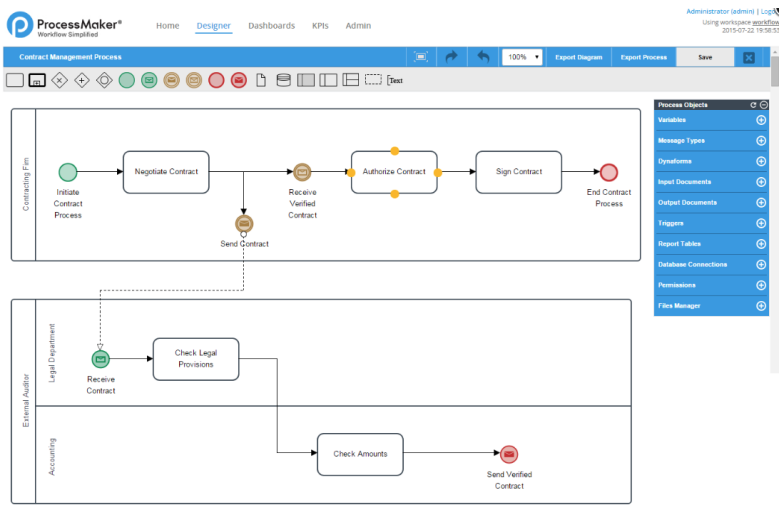


Figure 4.21: Designer within ProcessMaker

The Dynaform Designer is an intuitive and user friendly designer to create forms without writing any code. The drag-and-drop functionality enables to create a form in seconds.

Process Realization There is no deployment in the conventional sense such as deploying a created system. Before using the BPM Suite it has to be deployed to the server as a whole because this BPM Suite is web-based.
Thus, when the process design phase has been accomplished successfully in the designer, the next step after an optional debugging phase, which is only available in the enterprise version, it is to organize the user and groups with their permissions. Also departments have to be defined in a hierarchical tree. When a user is created he logs in into the system to control and further proceed the process.
Recently (June 3, 2015), Bitnami[33] cooperated with ProcessMaker[34] in the sense that now it is possible to make a one-click deployment on Amazon EC2 Cloud[35] and Google Cloud Platform[36]. With this cooperation, customers obtain a completely configured instance of ProcessMaker up and running on the cloud in just a few clicks without any cloud knowledge required.

[33] https://bitnami.com/stack/processmaker
[34] http://www.processmaker.com/press-releases/bitnami-enables-one-click-processmaker-deployment-amazon-ec2-cloud-and-google-cloud
[35] https://aws.amazon.com/de/ec2/
[36] https://cloud.google.com

Integration of Systems One of the core features is the provision of a REST API. This REST API can be used to integrate ProcessMaker with other standalone or web applications, and for administrators who want to script interactions with the ProcessMaker server [48].
Furthermore, OAuth 2.0 is used as the API Authorization framework which accepts requests and responses in JSON format.
In the enterprise version(s) features such as Business Rules Enginer, Microsoft Outlook Connector, Advanced LDAP Sync, and Simple Reporting extend and enhance ProcessMaker's performance and functionality.

Process Execution The Execution is performed by the BPM Suite itself. This means every user has to log in into the ProcessMakers' interface and depending on the permission rights the user can proceed the process. A user owning permissions to organize a group of users, this user, e.g. a project manager, can reorganize the users during run-time.

Resource Allocation & Scheduling Basically, one of the core features is the User Management, which provides functions to create roles, groups and departments to customize the process for a unique structure, reporting requirements and chain of commands in the organization. Furthermore, users can be assigned to different roles which match their functions in an organization - the head of a department cannot be part of another department. The User Management (figure 4.22) also provides to manage user vacation periods and shift case loads automatically to co-workers.
With the support of the enterprise features such as advanced LDAP and Active Directory synchronization, multi-tenant management tools, and visual backup and restore features it is possible to improve resource allocation and business outcome [55].

Process Controlling The Dashboards & Key Performance Indicators (KPIs) is one of the core features which should provide decision makers with full visibility to verify how processes and employees are performing [48]. Furthermore, there are two new KPIs in the latest version: The Process Efficiency Index (PEI) is responsible for intelligent learning from the process behavior over time and establishes optimal performance levels based on a combination of process factors including standard deviation, resource costs, and comparative rankings [56].
The second KPI, Employee Efficiency Index (EEI), is similar to PEI but measures the efficiency of each user over time and ranks the user based on efficiency and cost savings to the organization [56].

Critics & Conclusion ProcessMaker is a web-based BPM Suite which means that possibilities to interact with the engine are restricted. This BPM Suite is mostly created for business analysts and business developers with less experience in extending or developing proprietary options in a BPM System by programming.
The advantage of a web-based BPM Suite definitely lies in its mobile functionality.

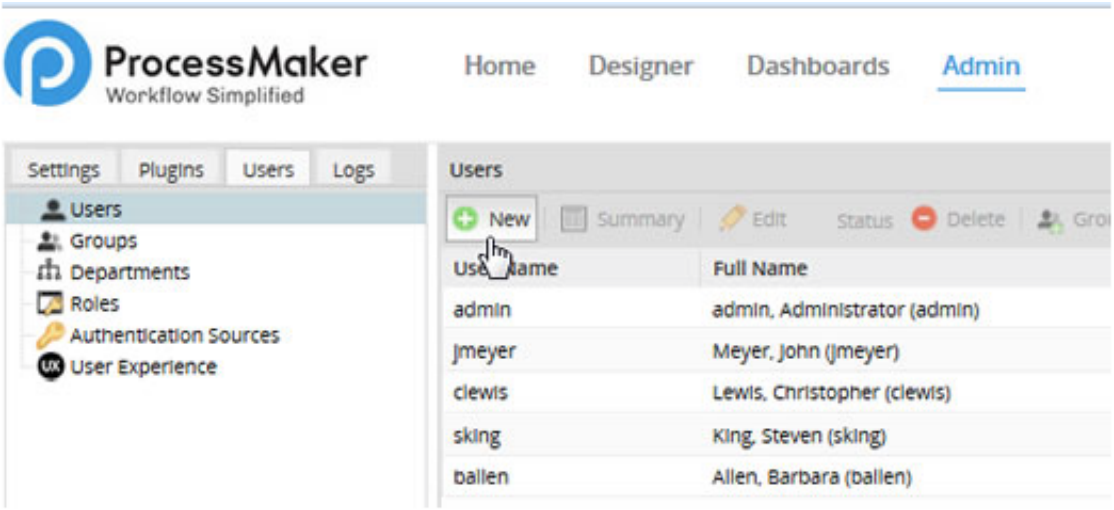


Figure 4.22: User Management

The flexible user forms in the web interface can be adapted easily to mobile devices such as tablets or smartphones.

4.3.7 Softproject

General Information The X4 BPM Suite of Softproject[37] is a comprehensive tool to automate business processes. The software encompasses all modules for modeling, technical implementation, execution and controlling of companies' processes. Prefabricated modules, adapters and process libraries allow high productivity in the implementation of the processes. Working with the Suite can be done without any programming knowledge and can begin right after installation.

Process Modeling The X4 BPM Suite includes a proprietary process modeler with all the ingredients to model a business process in the BPMN 2.0 standard. Also if process models are modeled with another conventional BPMN 2.0 modeler, e.g. Signavio[38], GBtec[39] etc., this process model can be imported into a X4 BPM Suite [60].
Apart from the graphical design of business processes, business rules are defined graphically or by rule description and executed by the integrated rules engine [60]. This can be tested without starting the whole process.

Process Realization One great feature of the X4 BPM Suite is the documentation of business processes [60]. "The X4 Designer allows modeled processes to be

[37] https://www.softproject.de
[38] http://www.signavio.com
[39] http://www.gbtec.de

documented at the push of a button. Moreover, with the supplied adapters, the documentation can be generated and published in many formats, e.g. Word, XML, etc." [60]
Once the process design is finished and the process is started, the X4 BPM Suite switches to the X4 ControlCenter which provides functions for testing, monitoring, evaluating and documenting business processes [60].

Integration of Systems As in figure 4.23, X4 BPM Suite runs on any Java EE-compatible application server and works together with the Enterprise Service Bus (ESB) which connects IT systems through the Adapter Development Kit (figure 4.24) and provides services for a service-oriented architecture (SOA) [59]. This enables synchronous and asynchronous processing, messaging, the integration of distributed resources and the transactional processing [59].

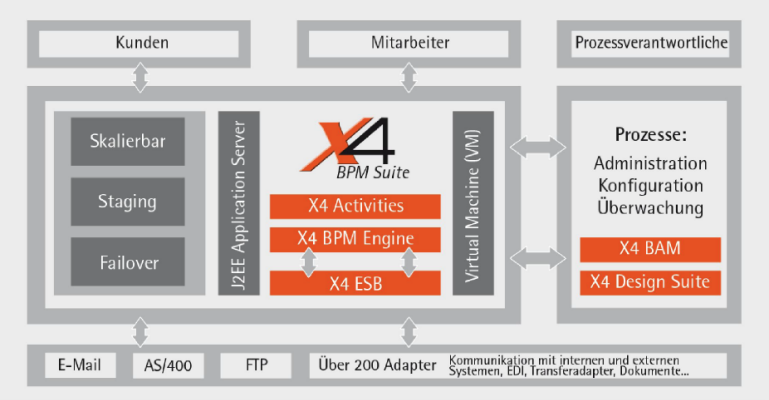


Figure 4.23: Architecture of X4 BPM Suite

Process Execution Once a process is designed it is possible to test it, also with structured data included such as invoices, contracts or other documents.
After deployment, which is done based on Java EE, the processes are administrated over the web-based X4 Control Center.

Resource Allocation & Scheduling As shown in figure 4.25, X4 BPMS provides a comprehensive user and rights management which is called Control Center. The web-based Control Center is intended to be used by administrators who want to define any users, groups, roles and rights for any activities. This tool also includes assignment and authorization functions of human resources, mail baskets including forwarding and deputy functions, specific document viewers, and calendar and map module. Additionally, there is also a connection to a LDAP server possible.
Once an activity is performed by a person, the Control Center proceeds the execution by a scheduler. Within the the BPMS the integrated scheduler automates order processing and supports advisors during essential steps of the communication

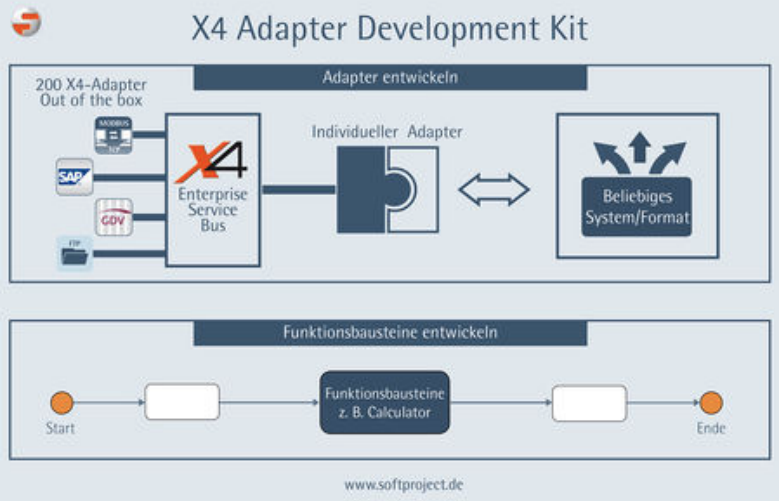


Figure 4.24: X4 Adapter Development Kit

between clients and service providers. The administrator interface (fig. 4.26) also allows to access X4 Server functions to activate or deactivate different extending server functionalities, e.g. Quartz Scheduler[40] which is an open source job scheduling library.

Process Controlling X4 Business Activity Monitoring (BAM) allows real-time analysis and display of time-relevant services which enables the execution of all services to be monitored and controled during their runtime [59]. There is also the option to retrieve messages, errors and current state as well as to implement workflows for alerts or escalations [59].
The X4 Control Center also provides functions for testing, monitoring, evaluating and documenting business processes. Additionally, service level agreements deposited within the process model can be monitored in real-time during process execution [59].

Critics & Conclusion The X4 BPM Suite of Softproject is one of the leading players in providing solutions for creating business process management. The emphasis definitely lies on the integration of services and systems and managing them with a powerful organizational concept. Especially the documentation service for activities seems to be very attractive since this can be exported in several formats which is important for further usage.
Furthermore, this BPM Suite also distinguishes itself because it does not lose sight of the process design concerning the exchangeable BPMN 2.0 standard.

[40] http://quartz-scheduler.org

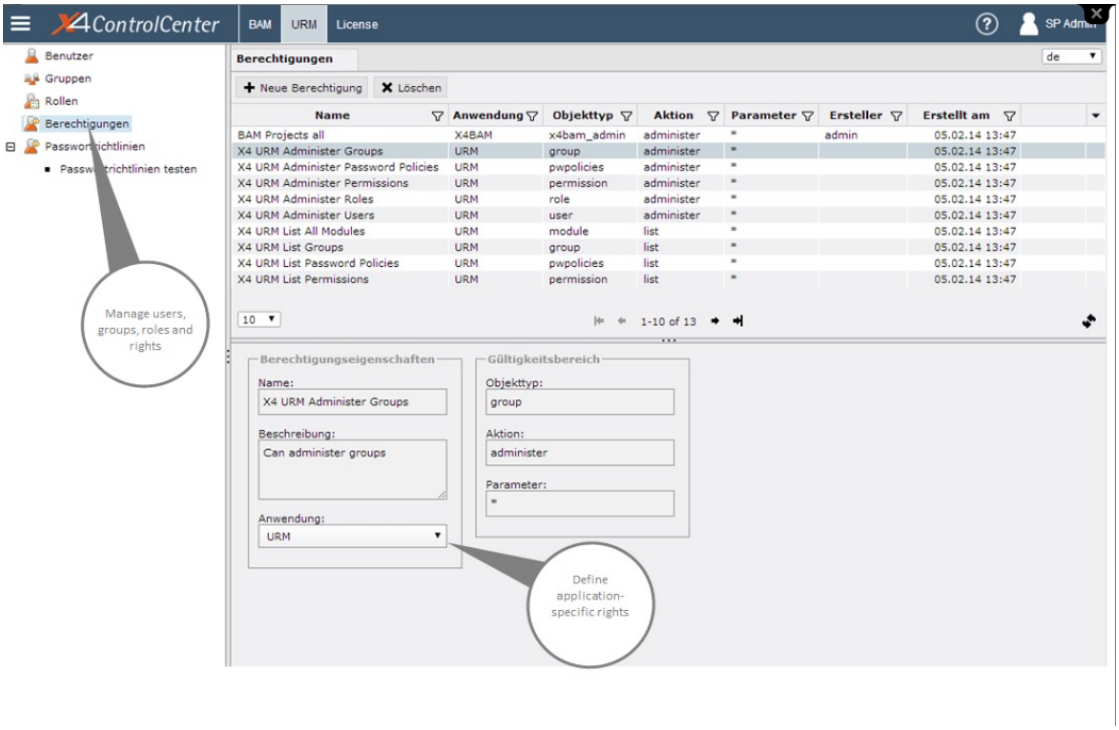


Figure 4.25: X4 Control Center

4.4 Selection and Usage of a BPMS

In this section, we summarize and argue for our choice of selecting Camunda from the seven selected BPMS that we have described in section 4.3. In the evaluation, we have analyzed each BPMS along specific criteria. The analysis shows that each BPM System/Suite has its individual strong point. Therefore, selecting an appropriate tool depends on which requirements have to be satisfied.
In the course of this thesis and in the research project SHAPE, we are highly interested in a BPMS which is state-of-the-art regarding standards and provides extensibility of external services, e.g. for automated resource allocation. It is relevant for many companies that a BPMS supports the standardized BPMN 2.0 format. Since a BPMS provides full BPMN 2.0 support, i.e. exporting and importing BPMN 2.0 files, the designed process is independent of the choice of the tool in the course of SHAPE.
These interests and requirements are considered in the criteria of the evaluation which is depicted in detail in Appendix A - The Evaluation of BPMS. As a result of the evaluation, Camunda BPMS seems to be a proper candidate for several reasons:

- Basically, Camunda is an open source BPMS based on Java. It provides a process engine architecture with a great variety of Application Programming Interfaces (APIs) that help to extend the BPMS arbitrarily. Furthermore, it is possible to redesign the whole BPMS since it provides the full source code in Java.

Figure 4.26: X4 Context Manager in the Control Center

- The company Camunda has a close cooperation with the OMG Group[41] which affects the BPMS in a way that state-of-the-art technologies and standards such as BPMN 2.0 are built-in and available for this BPMS.
- In terms of service and support, when there is no need for an open source version, Camunda offers an enterprise version with different price levels according to the needs of functionality.

Apart from Camunda, the other BPMSs also have their strengths but they do not fully meet our requirements:

- Bizagi provides designing processes in an in-built designing tool but this does not correspond to the BPMN 2.0 standard. Furthermore, within Bizagi it is possible to create a hierarchical organizational structure which cannot be exported.
- jBPM tries to find the golden mean between business analyst and process designer. Unfortunately, their focus lies on zero-coding which means that they try to avoid coding as much as possible.
- Axon Ivy provides designing processes in an in-built designing tool but without the support of BPMN 2.0
- Oracle BPM is offering a wide range of functionality with different focuses. Although they provide support of BPMN 2.0, this BPMS goes beyond the scope of the project SHAPE: Oracle provides a wide range of capabilities to create a BPMS with high initial expenditure to get a complete version.

[41] http://www.omg.org

- ProcessMaker is a BPMS which is only web-based. The purpose is to provide an easy access to designing processes without any support in BPMN 2.0. Therefore it cannot be extended.
- Softproject is placed next to Camunda. It supports BPMN 2.0 as well as extensions in Java. The only restriction lies in the fact that the process engine is part of an enterprise version, thus it is not open-source.

To this end, these arguments are convincing to select Camunda satisfying both the requirements for the project SHAPE and the requirements of Siemens so far. In SHAPE, we are able to extend Camunda with external services developed in academia such as automated resource allocation, and within Siemens we are able to manage safety-critical engineering processes on the basis of BPMN 2.0 respectively.

CHAPTER 5

Integrating Resource Allocation Capabilities into a BPMS

Nowadays, BPMS have different fields of attention (chapter 4) corresponding to the multiple aspects of application, e.g. for business analysts or an engineer with specific expertise in programming, and they all have one thing in common: managing and optimizing the company's processes. Although there exist many solutions already, which provide excellent administration areas to manage the execution of tasks there is still a large amount of manual processing in allocating resources and as a consequence the time required is not optimal. Since we have identified Camunda as a BPMS which provides extensibility by external services, we address this problem by extending Camunda applying the method of automated resource allocation in chapter 3. Our proof of concept enables integrating a prototype to automatically allocate resources before a business process is started and at runtime.
This chapter is structured as follows: First the starting points in Camunda are discussed. Afterwards, our methodology is exposed. The chapter concludes with a conclusion and future discussion.

5.1 Extensibility of a BPMS

Since Camunda provides numerous possibilities (APIs) for extending it[1], this seems to be a proper tool to realize our work. For the implementation, we categorize the following essential points for modifying Camunda:

Development environment. Within the given framework, Camunda puts the business process model into a resource package where the deployed BPMS retrieves the information from. The model can be changed at any time but then Camunda

[1]We used Eclipse Mars for the implementation: https://projects.eclipse.org/releases/mars

automatically forwards a new version of the process after re-deployment which the user has to be aware of because otherwise he has to restart the whole process. However, we designed the business process within the Eclipse[2] BPMN2 Modeler plugin for Business Process Model Notation[3]. This enables to quickly jump from the designing to the programming tab. At any time, we can import and export this business process as BPMN file with the standardized notation of BPMN2. In addition, the modeler produces a PNG file at runtime which is quite useful for exporting to presentations.

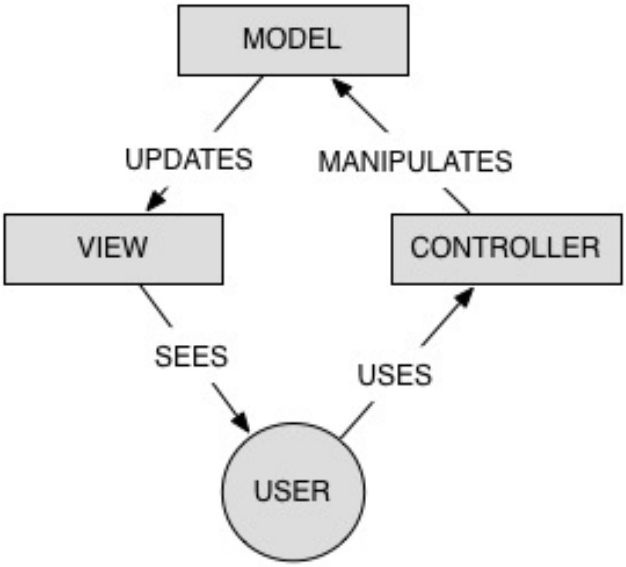


Figure 5.1: The collaboration of the MVC components.

Technical background for the extension. One of the features of Camunda is to support developing a proprietary external task form. Building upon this feature, we make use of the Model-View-Controller (MVC) pattern [44] in combination with Spring MVC[4] to develop an external service that interacts with Camunda. This pattern (fig. 5.1) allows to separate internal representations of information from the ways that information is presented to the end-user. In detail, the *Model* is responsible for storing the data into the JBoss[5] database which is preferred by Camunda. As depicted in listing 5.1, an excerpt of the *Model*, which defines the object to be stored into the database. For this procedure, we use Hibernate[6] - it provides a framework for mapping an object-oriented domain model into a relational database. The annotations labeled in red perform exactly these mappings.

[2]We used Eclipse Mars for the implementation: https://projects.eclipse.org/releases/mars
[3]https://www.eclipse.org/bpmn2-modeler/
[4]http://spring.io
[5]http://www.jboss.org
[6]http://hibernate.org

Listing 5.1: Model - Domain Class of PredictResourceAssignment

```
@Entity
@Table(name = "PredictedResourceAssignment")
public class Assignment implements Serializable {

    @Id
    @GeneratedValue
    private Long id;

    public Assignment(String processId, String taskId, String resourceId,
          Date startAssignment, Date endAssignment, Date creationDate) {
       super();
       this.processId = processId;
       this.taskId = taskId;
       this.resourceId = resourceId;
       this.startAssignment = startAssignment;
       this.endAssignment = endAssignment;
       this.creationDate = creationDate;
    }

    @NotNull
    @NotEmpty
    private String taskId;

    //more variables ...

    //Getter & Setter ...
}
```

The *Controller* is responsible for managing the communication between the *Model* and the *View* such as passing results of retrieved information from Camunda Backend, e.g. the organizational model and the BPMN diagram, to the *View* in order to visualize the results. In this context, we make use of the Spring Inversion of Control (IoC) principle [28], which enables us calling the ASP solver as in listing 5.3.

For realizing the *View*, as shown in an excerpt in fig. 5.2, we apply JavaServer Faces (JSF)[7] which is a Java specification for building graphical user interfaces for web applications.

Listing 5.2: View - a data table is built from the *Model*

```
<!DOCTYPE HTML>
<html>
<f:view>
   <h:body>
      <div class="container">
            ...
            <p:panel header="Assigned Resources">
               <p:dataTable id="userAssignmentsTable" value="#{assignmentsBean.
                   userAssignments[assignmentsBean.currentAssignment]}">
                  ...
               </p:dataTable>
               ...
            </p:panel>
   </h:body>
```

[7] https://javaserverfaces.java.net

```
14 </f:view>
15 </html>
```

External tool integration. By the Answer Set Programming (ASP) solver, we use *Clingo*[8] which has been proven as a proper tool in solving automated resource allocations [32]. Clingo is stored on the server and we call it with the input parameters as streams, as shown in listing 5.3.

Listing 5.3: Controller - Calling Clingo with parameters on a stream

```
1 @ManagedBean
2 @ViewScoped
3 public class AssignmentsBean implements Serializable {
4
5    @Inject
6    ProcessEngine engine;
7
8    //settings entered by the User
9    ArrayList<RoleAssignment> roleAssignments = null;
10
11   String currentAssignment = null;
12   Map<String,List<LockableAssignment>> userAssignments = null;
13
14   @Inject
15   PredictedAssignmentService predService;
16
17   Runtime rt = Runtime.getRuntime();
18   Process p = rt.exec("/path-on-server/clingo");
19
20   OutputStreamWriter w = new OutputStreamWriter(p.getOutputStream());
21
22   w.write(iStr);
23   w.close();
```

5.2 Methodology

This section defines the novel architecture we propose for extending Camunda with the service of automated resource allocation [32] which has been developed in academia previously. As depicted in figure 5.2, the methodology of our extension consists of two phases: First, we retrieve information regarding the business process and the organizational model. Second, we output possible allocations based on the inputs. This methodology is based on the possibilities to extend Camunda as described in the previous section.

For illustration, we assume the scenario of BuildIT for which we have described its BPMN diagram (fig. 2.3) and its relation of resources and roles (fig. 3.4) in previous chapters. We note that before starting the business process, in Camunda the administrator has to set up the organizational model by creating the roles and participants of the project that our extension can register them later on. It is also possible to assign participants to some roles manually before the process is started but this is not necessary since our

[8] http://potassco.sourceforge.net

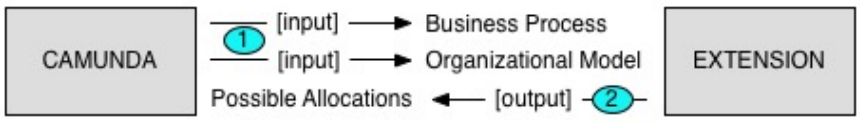


Figure 5.2: The relation of Camunda and our Extension

extension performs this automatically. However, at this point our integration is ready to start. Realizing our extension we develped a proof of concept, shown in figure 5.3, which

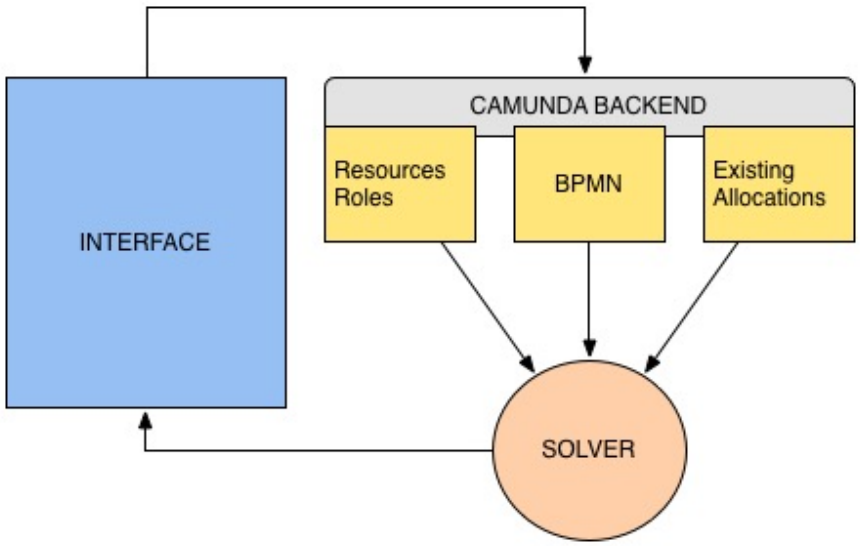


Figure 5.3: The concept of our extension

implies the following components:

- **Camunda Backend.** As described in chapter 4, Camunda provides a powerful service-oriented API allowing Java applications to interact with the process engine. This allows retrieving all necessary information we need from the Camunda BPMS in order to handle data within our methodology. For persisting data, we installed a JBoss database where all the information required by Camunda is stored.
- **BPMN.** As depicted in fig. 5.3, the BPMN diagram of BuildIT (2.3) is one of the inputs for the solver. We note, that we do not specify any roles or users at design time for being more flexible in defining them later by our extension. Since the BPMN diagram is executable it serves as guideline for the execution order of the activities in the business processes. For the translation of the BPMN diagram into a Petri Net we suggest Apromore[9] which is an open-source repository to store and

[9]http://apromore.org

disclose process models of a variety of types such as Petri Nets (fig. 3.3). However, the result in form of constraints of activities can be processed by the ASP solver.

Listing 5.4: Retrieving existing assignments

```
1 @Inject
2 ProcessEngine engine;
3
4 userCache = new HashMap<String, String>();
5 List<User> users = engine.getIdentityService().createUserQuery().list();
6    for (User u : users)
7       userCache.put(u.getId(), u.getFirstName() + " " + u.getLastName());
8
9 roleCache = new HashMap<String, String>();
10 List<Group> groups = engine.getIdentityService().createGroupQuery().list();
11
12 for (Group g : groups)
13    roleCache.put(g.getId(), g.getName());
14 if (roleCache.isEmpty())
15    roleCache.put("noRoles", "No Roles Defined");
```

- **Resources and Roles.** This component is necessary for the ASP solver to find an optimal way placing new allocations between already existing allocations. Camunda is able to manage various business processes in parallel and also several instances of them. Every allocation and its user assignments of running business processes are stored in the database. The open-source API of Camunda enables to access the process engine where we can retrieve information about current assignments, as depicted in listing 5.4.
  Assuming that there is only the organizational model according to the example of BuildIT in chapter 3, we retrieve, for instance, the information of assigned resources to roles and create strings in the form of ASP constraints. Listing 5.5 illustrates the creation of constraints which result in constraints, depicted in listing 5.7 from line 1 to 6, ready to be used by the ASP solver. In addition to that we have to convert the information regarding time management since the ASP solver cannot read continuous timespans (from the server). Therefore, we use discrete time assignments [32] that are solvable for the ASP solver, depicted in listing 5.6. These time assignments are the numbers at the end of each constraint.

Listing 5.5: Creating organizational constraints

```
1 for (RoleAssignment r : roleAssignments)
2    {
3       iStr += "hasRole(" + r.getTaskId().toLowerCase() + "," + r.getRoleId().
            toLowerCase() +").\n";
4    }
```

Listing 5.6: Time constraints for activities and resources

```
1 activityDuration(t_submit_equipmental_rental_request,bp_buildit,1).
2 activityDuration(t_select_suitable_equipment,bp_buildit,1).
3 activityDuration(t_check_availability,bp_buildit,2).
4 activityDuration(t_review_rental_request,bp_buildit,2).
5 activityDuration(t_create_po,bp_buildit,1).
```

```
6
7 resourceActivityDuration(glen,t_submit_equipmental_rental_request,bp_buildit
      ,2).
8 resourceActivityDuration(tom,t_select_suitable_equipment,bp_buildit,1).
9 resourceActivityDuration(tom,t_check_availability,bp_buildit,2).
10 resourceActivityDuration(tom,t_select_suitable_equipment,bp_buildit,1).
11 resourceActivityDuration(tom,t_check_availability,bp_buildit,2).
12 resourceActivityDuration(drew,t_review_rental_request,bp_buildit,1).
13 resourceActivityDuration(evan,t_review_rental_request,bp_buildit,3).
14 resourceActivityDuration(tom,t_create_po,bp_buildit,2).
```

- **Exiting Allocations.** In this component we retrieve information of current allocations. This procedure is analogous to retrieving role assignments. In our basic example there are no allocations yet, therefore the allocation is represented as in fig. 5.5. In contrast, one or more allocations exist, fig. 5.4 depicts the case when there are allocated tasks of one existing business process (green) and tasks of another one have to be allocated in between (blue).

Listing 5.7: Constraints of the organizational model

```
1 hasRole(amy,requester).
2 hasRole(amy,site_engineer).
3 hasRole(glen,site_engineer).
4 hasRole(tom,clerk).
5 hasRole(drew,works_engineer).
6 hasRole(evan,works_engineer).
7
8 canExecute(site_engineer,t_submit_equipmental_rental_request,bp_buildit).
9 canExecute(clerk,t_select_suitable_equipment,bp_buildit).
10 canExecute(clerk,t_check_availability,bp_buildit).
11 canExecute(works_engineer,t_review_rental_request,bp_buildit).
12 canExecute(clerk,t_create_po,bp_buildit).
```

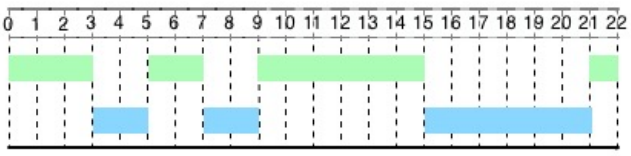


Figure 5.4: Two sample business processes (blue and green) are organized by the ASP solver.

- **Solver.** The solver is responsible for allocating resources according to the input. This means, the solver computes an efficient time optimum for allocating the activities and its participants according to the organizational model.
  For performance reasons, it is extremely important to make the range of possible timespans as small as possible: Continuous times spans, e.g. 15h:13min:7sec, must be prepared (listing 5.6) in a way the ASP solver is able to output an organized solution, as shown in fig. 5.4.
  For simplification, we assume in our example of BuildIT that there are no other

business processes in parallel and also no loops. The result of the solver, as depicted in figure 5.5, presents an allocation in listing 5.8 as follows: In line 1-5, the first parameter is the name of the resource, the second one identifies the activity that has to be executed by the first parameter, the third and fourth parameter represent the time span, the fifth parameter is the number of the activity, the sixth one displays the name of the business process followed by the number of the performed allocation. At the end in line 6, the solver responds that an optimum is found, representing that line 1-5 is the best solution of allocated resources and activities in a minimal time span.

Listing 5.8: A result of the solver assigning allocated resources

```
1 assign(amy,t_submit_equipmental_rental_request,0,1,1,bp_buildit,i1)
2 assign(tom,t_select_suitable_equipment,1,2,2,bp_buildit,i1)
3 assign(tom,t_check_availability,2,4,3,bp_buildit,i1)
4 assign(drew,t_review_rental_request,4,5,4,bp_buildit,i1)
5 assign(tom,t_create_po,5,7,5,bp_buildit,i1)
6 OPTIMUM FOUND
```

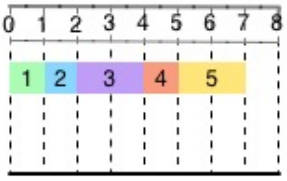


Figure 5.5: Allocations and assignments according to the result of the ASP solver in 5.8

- **Interface.** The web-interface is the representation of our extension and can be accessed through Camunda. In figure 5.6, which depicts a mockup of our extension filled with values of our example of BuildIT, the two phases according to fig. 5.2 are performed as follows: When Camunda is running and the business processes is started, we can access our extension in the next step. In the first phase, the extension identifies the business process that has been loaded in Camunda and retrieves all the information available from the organizational model together with the activities of the business processes. Also the duration regarding the resource constraints are loaded into the extension. Since phase one is completed, the next step - the resource allocation - is triggered when the form button *Role Assignment Complete* is pushed. Afterwards, (1) information in Resource Assignment Language (RAL) is provided, and (2) some solutions of assigned resources are provided with the optimum at the top.

    1. **RAL Information.** This box represents Information by the Resource Assignment Language (RAL). With RAL it is possible to construct more complex formalisms, e.g. *Submit request* IS A SITE ENGINEER OR IS A CLERK.
    2. **Assigned Resources.** Inside this box, solutions by the ASP solver are provided. Listing 5.8 shows one solution. It is possible to edit the solution

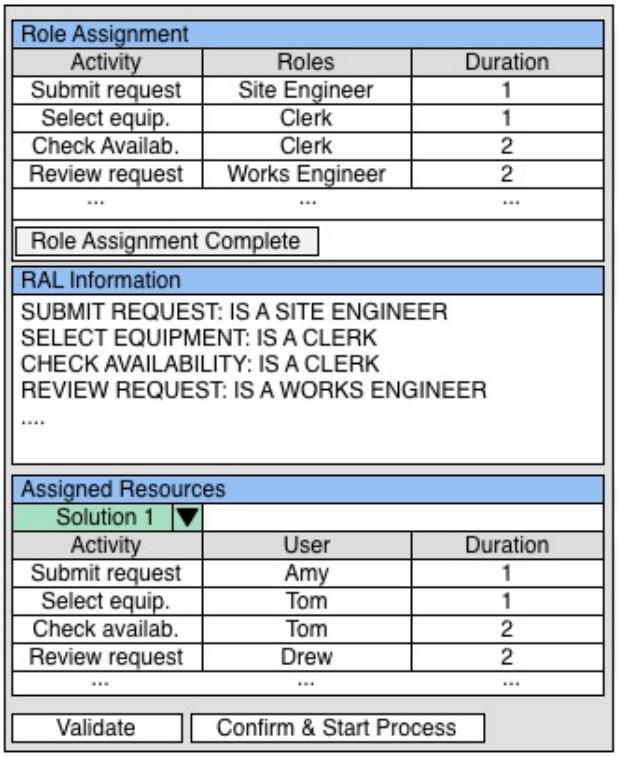


Figure 5.6: The mockup of the web interface

and validate again by means of reallocating or simply accept one solution and confirm it to continue with starting the business process with the automatically allocated resources. By confirming the solution, the computed allocations are stored to the Camunda Backend.

5.3 Conclusion

In this chapter we have demonstrated the extensibility of Camunda with our approach for resource allocation using an ASP solver by integrating an external service developed in academia, i.e. automated resource allocation. First, we described the technologies used, and second, we applied our developed methodology in the course of the example of BuildIT.
It turned out that the possibility of integrating an approach into Camunda is quite efficient since we could access all APIs easily via Java. By analysing the result, the ASP solver provides efficient results in contrast to the procedure of allocating resources manually. Since the whole framework in its actual state is a prototype to be extended, this methodology serves as basis for further functionality, e.g. other components providing input for the solver.

CHAPTER 6

A Use Case from Industry

6.1 Railway Domain

Together with Rail Automation engineering experts we collected a set of requirements on their engineering processes. A priority has been specified for each requirement to document which process improvements are most important for this particular environment.
In the course of Siemens' analysis of the strengths and options of integrating a BPM framework, the advantages had become clear enough to identify requirements for a proof of concept yielding possible automations and improvements in existing engineering processes. In this context, (1) modeling time requirements (table 6.1), and (2) runtime requirements (table 6.2) have been determined.

On our part, we focus on requirements that are in the scope of our hypothesis (chapter 1):

1. **Design Time Requirements.** One big benefit of a BPM framework in the sense of our proof of concept is the standardized way of designing any kind of process in the Business Process Management Notation (BPMN). The advantage hereby lies in the standardization itself which enables to import and export process descriptions from BPM tools without any problems of compatibility and inconsistency that satisfies D01. In addition to that, BPMN has clearly defined symbols for every scenario of a business process such that there is no need of help from professionals (D04) in case of new upcoming scenarios. BPMN also provides easy readability even for non-technicians. This means that no explicit knowledge is required to understand the whole business process. A feature that supports this understanding is the definition of roles (D06), i.e. designing a lane for each responsibility, depicted in figure 6.1. This feature also implies the potential of modeling access rights and roles (D07) since lanes are describing the operative field of a role.
   Designing a process involves one or more participants in order to guarantee explicit

Table 6.1: Design Time Requirements, Priority [1(low) - 5(high)]

No.	Requirement	Priority
D01	Use/import/export of standardized process description format, e.g. BPMN	4
D02	Automatic extraction of process models from textual description	3
D03	Automatic extraction of process models from log data (process mining)	1
D04	RA can model their processes / change their process models without help of professionals	4
D05	Automatic generation of process documentation (process handbook)	5
D06	Definition of roles	5
D07	Access rights of roles modeling	5
D08	Definition of constraints to check data integrity, e.g. file versions, tool versions	5
D09	Definition of constraints to check data completeness	5
D10	Explicit modeling of safety aspects, e.g. 4eye principe, engineer != verifier	5
D11	Safety risks analysis on processes, e.g. for support of hazard analysis	2
D12	Task duration and deadline modeling	1
D13	Escalation modeling	2
D14	Specification of templates (documents, e-mails, checklists) with placeholders for artefacts	5

modeling of safety aspects (D11). The standardized format supports this such that the stability of processes designed is not affected by various exchanges.

2. **Runtime Requirements.** Since we have selected Camunda as our Business Process Management System (BPMS) in chapter 4, this BPMS comprises several features, e.g. monitoring processes and guiding engineers at runtime (R01), or triggering notifications or requests of process steps in a web interface (R02). Furthermore, Camunda provides an option to assign individual team members to roles and tasks (R09, R10).
   One of the great benefits of Camunda is the possibility to extend this BPMS with proprietary solutions according to our needs. This means covering R11, we developed a prototypical extension for allocating resources automatically.

6.2 The prototype

Through our investigation about Business Process Languages and Business Process Management Suites (BPMS) regarding integration methods of Knowledge Representation & Reasoning (KRR) for automating resource allocation, we now have the instruments available to implement a prototpye. For confidentiality reasons, the following figures 6.1 and 6.2 have been anonymized, e.g. tasks have been simplified by removing real names. Although the engineering process already existed, we had to redesign this process in order to fit into BPMN (fig. 6.1). We note that relating to the BPMN specification, lanes illustrate roles and fields of responsibility that have no technical relevance but they

Table 6.2: Runtime Requirements, Priority [1(low) - 5(high)]

No.	Requirement	Priority
R01	Process engine monitors/guides engineers at runtime	5
R02	Notification/requests of process steps via e-mail	5
R03	Notification/requests of process steps via web interface	5
R04	Complete documentation of process steps during runtime (log file)	5
R05	Integration/examination of e-mail traffic, e.g. tags in emails cause starting/finishing of activities	3
R06	Integration/examination of SharePoint traffic, e.g. finished/changed documents start/finish activities	3
R07	Integration/examination of SVN/GIT traffic, e.g. commit automatically finishes activity	3
R08	Integration of people directory	5
R09	Assignment of actual persons to roles (people resolution)	5
R10	Assignment of actual persons to tasks based on availability/workload/custom weighting	1
R11	Access rights checking (roles based, based on policy model)	5
R12	Deadline and effort tracking by logging of activity/process durations	4
R13	Deadline and effort prediction, e.g. not enough available engineer-hours to meet deadline	4
R14	Synchronization with project planning, e.g. with Enterprise Project Management (EPM)	2
R15	Detection of process deviation, e.g. incomplete input data used	5
R16	Predictive process deviation, e.g. a predictive failed deadline causes creation of new activities like a meeting	2
R17	Rollback or safe process continuation in case of a deviation, e.g. in case of wrong input data, use correct input data	5
R18	Continuous process optimization (automatic detection of process improvements)	3
R19	Learning of task durations / variations	4
R20	Automatic generation of documents/e-mails/checklists - based on templates	5
R21	Automatic generation of a plan inventory, i.e. a document consisting of all relevant artefacts, their locations and versions	5
R22	Checking of data integrity and completeness constraints and user notification of constraint violations	5

support the readability.
Having completed design time, the administrator (i.e. project manager) of the whole process, has to set up all roles and engineers designated for this process. In the next step, we start executing the engineering process. By means of starting, we are forwarded to our extension - an external web interface, depicted in fig. 6.2. The first phase is called the Role Assignment. Within this phase, all activities designed in the BPMN are loaded as well as all roles that have been set up previously for this project. At this point, we select roles for each activity. This step is concluded by triggering the button *Role Assignment Complete*.
Phase two is initiated by the completion of role assignments. In the frame of *RAL Information*, RAL information is inserted automatically according to the role assignments. The frame of *Assigned Resources* depicts the result of the approach that allocates resources automatically. That means, all engineers of the project available (including access rights) are allocated automatically in a time-efficient way. We can switch between several solutions, edit them and validate the changes, or simply accept a solution by confirming these assignments for starting the process.

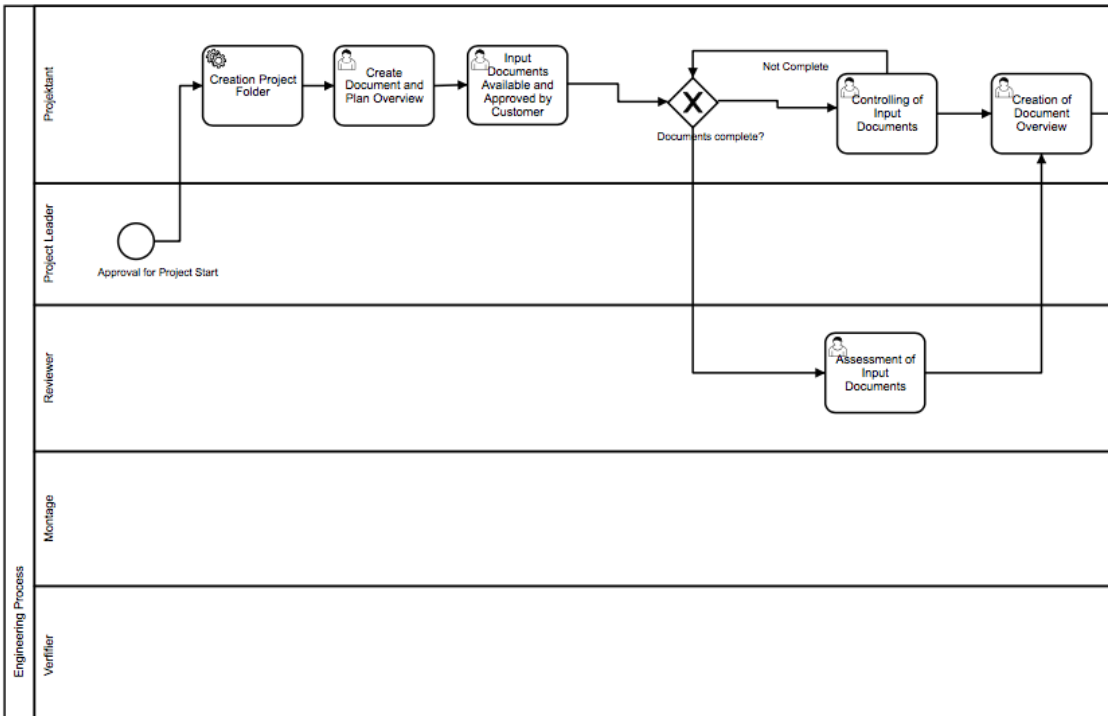


Figure 6.1: An excerpt of the designed engineering process in BPMN.

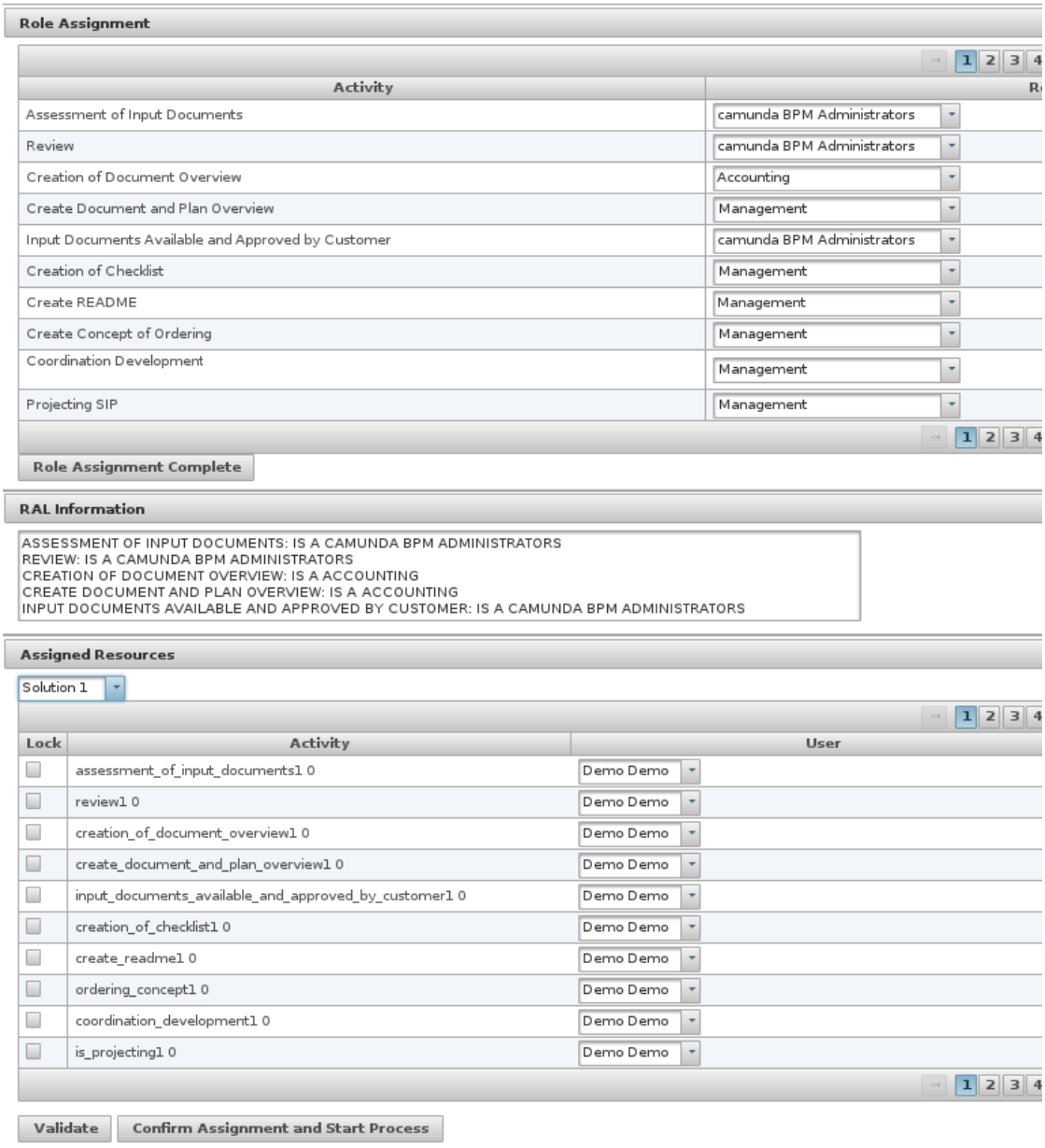


Figure 6.2: An excerpt of the resource allocation extension with the business process of fig. 6.1.

6.3 Conclusion

A prototype was implemented that produced the intended results in a first testing phase: This proof of concept satisfies the requirements formulated for the scope of the hypothesis. It is therefore already foreseeable that this proof of concept will lead to a

massive support especially for safety-critical engineering processes in terms of savings of time and resources spent and furthermore a clear-cut reduction in the amount of project coordination problems. Although this prototype currently consists of just a simplified set-up, it holds the potential to be extended by a number of additional features, e.g. logging and documentation, that will further enhance its application usability and effectiveness.

CHAPTER 7

Summary & Conclusion

Over the last decades, the evolution of Business Process Management (BPM) has made significant progress in terms of standardization mostly driven by academic research that also increasingly gained attention in the private sector. We investigated several process model and description languages with a special focus on the characteristics of the standardized Business Process Modeling Notation (BPMN) with respect to the application in the industrial context. Based on a sample process we demonstrated the core features which are applicable to complex engineering processes of technical infrastructure products.
The ability to execute processes of BPMN results became a new market niche of Business Process Management Systems (BPMS) which support modeling, execution and monitoring of business processes. In an extensive evaluation, we have analysed which tools exist and how flexible they are with regards to changes of parameters over time, how well they support modelling, execution and monitoring of complex processes. Although they provide organizational features for scheduling resources there is still a large amount of manual processing. We addressed this in a second step, were we have investigated how approaches from academia to automate resource allocation, task assignments and verification of business processes can be incorporated into selected tools.
We presented an approach from academia how resource assignment and allocation under constraints in BPM can be performed automatically. This approach applies the power of Answer Set Programming (ASP) and Petri Nets transforming the activities within the BPMN diagram and available resources to ASP constraints. In a next step, these constraints are provided as input for an ASP solver which is then responsible for computing solutions.
As a result of the evaluation we have selected the Camunda BPMS which provides a broad spectrum of Java Application Program Interfaces (APIs) to extend this BPMS with proprietary solutions. We contributed with a proof of concept integrating the approach of allocating resources automatically into Camunda. Our prototypical extension of Camunda is represented by a web-based form which is accessible at any time. This can

be applied in scenarios where resources continuously change and since our methodology allocates resources automatically in a few steps, this would improve the management of projects in terms of flexibility and time-efficiency.

7.1 Future Work

The prototype developed serves as basic framework for further work. Since our extension covers one aspect - the automated resource allocation - of managing engineering processes, there are many more challenges left for the future, among others they are:

- Currently, only the first computed solution is presented. A future option might be providing further possibilities to filter solutions by means of additional constraints.
- In our implementation of resource allocation, we have not considered in detail the priorities of human resources [16] which could be one factor to extend the resource allocations.
- Our proof of concept was executed with processes from industry scenarios but has not been tested in the all-day working environment of companies. For example, we only assumed best case conditions without any server interruptions or other infiltration into the running process. However, next challenges involve tests regarding worst case conditions in order to monitor negative impacts on constraints.
- In our example scenario we only used a small amount of complex constraints. Within a company, there are much more constraints which can influence several processes at the same time. Addressing this challenge, tests have to address a certain degree of concurrent processes.

Some more challenges are easily conceivable that will leave a lot of room for further work.

Bibliography

[1] Sebastian Adam, Matthias Koch, Fabian Neffgen, Norman Riegel, and Justine Weidenbach. *Business Process Management - Marktanalyse 2014: BPM Suites im Test.* Fraunhofer IESE, Kaiserslautern, Fraunhofer-Platz 1, 67663 Kaiserslautern, 2014.

[2] Thomas Allweyer. *BPMN 2.0: introduction to the standard for business process modeling.* BoD–Books on Demand, 2010.

[3] Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, et al. Business process execution language for web services, 2003.

[4] Bizagi. Bizagi application integration. `http://help.bizagi.com/bpmsuite/en/10.5/index.html?integrating.htm`. Accessed: 2015-08-24.

[5] Bizagi. Bizagi application programming interface. `http://help.bizagi.com/bpmsuite/en/index.html?integrating_api_custcode.htm`. Accessed: 2015-08-24.

[6] Bizagi. Bizagi bpms. `http://help.bizagi.com/bpmsuite/en/index.html?soa_layer_url_and_consideratio.htm`. Accessed: 2015-07-13.

[7] Bizagi. Bizagi busines rules. `http://help.bizagi.com/bpmsuite/en/index.html?what_to_avoid_in_allocations.htm`. Accessed: 2015-07-13.

[8] Bizagi. Bizagi work allocation. `http://help.bizagi.com/bpmsuite/en/index.html?what_to_avoid_in_allocations.htm`. Accessed: 2015-07-13.

[9] Bizagi. Bizagi work allocation components. `http://help.bizagi.com/bpmsuite/en/index.html?work_allocation_components.htm`. Accessed: 2015-07-13.

[10] Bizagi. Organization modul of bizagi. `http://help.bizagi.com/bpmsuite/en/10.5/index.html?organization.htm`. Accessed: 2015-08-24.

[11] Bizagi. Transactional Process-Construction - Bizagi Process Modeler. 2011.

[12] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, 16, 1998.

[13] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

[14] Peter David Bruza and Th P Van der Weide. *The semantics of data flow diagrams*. University of Nijmegen, Department of Informatics, Faculty of Mathematics and Informatics, 1989.

[15] IBM Business, Process Manager, and Oracle BPM October. BPM Product Analysis. (October), 2013.

[16] Cristina Cabanillas, José María García, Manuel Resinas, David Ruiz, Jan Mendling, and Antonio Ruiz-Cortés. Priority-based human resource allocation in business processes. In *Service-Oriented Computing*, pages 374–388. Springer, 2013.

[17] Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés. Ral: A high-level user-oriented resource assignment language for business processes. In *Business Process Management Workshops*, pages 50–61. Springer, 2012.

[18] Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés. RAL: A high-level user-oriented resource assignment language for business processes. *Lecture Notes in Business Information Processing*, 99 LNBIP(PART 1):50–61, 2012.

[19] Camunda. Camunda user guide. `https://docs.camunda.org/stable/guides/user-guide/`. Accessed: 2015-07-13.

[20] Aris Community. Event-driven process chain (epc). `http://www.ariscommunity.com/event-driven-process-chain`. Accessed: 2015-07-13.

[21] Joseph A De Feo. Business process management: Creating an adaptable organization. *Juran's Quality Handbook: The Complete Guide to Performance Excellence 6/e*, page 255, 2010.

[22] Marc Denecker, Joost Vennekens, Stephen Bond, Martin Gebser, and Mirosław Truszczyński. The second answer set programming competition. In *Logic Programming and Nonmonotonic Reasoning*, pages 637–654. Springer, 2009.

[23] Remco M Dijkman, Marlon Dumas, and Chun Ouyang. Formal semantics and analysis of bpmn process models using petri nets. *Queensland University of Technology, Tech. Rep*, 2007.

[24] Remco M Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and analysis of business process models in bpmn. *Information and Software Technology*, 50(12):1281–1294, 2008.

[25] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A Reijers. Essential process modeling. In *Fundamentals of Business Process Management*, pages 63–96. Springer, 2013.

[26] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A Reijers. *Fundamentals of business process management*. Springer, 2013.

[27] Yawl Foundation. Industrial uptake of yet another workflow language (yawl). `http://www.yawlfoundation.org/pages/impact/uptake.html`. Accessed: 2015-07-16.

[28] Martin Fowler. Inversion of control containers and the dependency injection pattern, 2004.

[29] Inc. Gartner. Gartner, inc. `http://www.gartner.com/it-glossary/business-process-management-bpm/`. Accessed: 2015-07-09.

[30] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(3):1–238, 2012.

[31] Stefan Gerzmann, Matthias Kostka, and Matthias Meyer. CAP (Computer Aided Planning) mit BPMN-Editoren - ein Vergleich. (168), 2012.

[32] Giray Havur, Cristina Cabanillas, Jan Mendling, and Axel Polleres. Automated Resource Allocation in Business Processes with Answer Set Programming. 845638.

[33] Fraunhofer Iese, Sebastian Adam, Norman Riegel, and Matthias Koch. *Studie - BPM Suites 2013*. Fraunhofer IESE, Kaiserslautern, 2013.

[34] Dave Ings, Luc Clément, Dieter König, Vinkesh Mehta, Ralf Mueller, Ravi Rangaswamy, Michael Rowley, and Ivana Trickovic. Ws-bpel extension for people (bpel4people) specification version 1.1. *OASIS Committee Specification (August 2010)*, 2010.

[35] Axon Ivy. Axon ivy customer information. `http://www.axonivy.com/asia/solutions/bpm-solutions-for-departments/`. Accessed: 2015-08-25.

[36] Axon Ivy. Quartz scheduler. `https://guidelines.axonivy.com/projects/guides/wiki/Quartz_Scheduler`. Accessed: 2015-08-25.

[37] JBoss. Jboss jbpm repository. `http://docs.jboss.org/drools/release/6.2.0.CR4/drools-docs/html/wb.Workbench.html#wb.VFSRepository`. Accessed: 2015-08-24.

[38] JBoss. Jboss optaplanner. https://docs.jboss.org/optaplanner/release/latest/optaplanner-docs/html/ch01.html. Accessed: 2015-08-24.

[39] JBoss. Jboss optaplanner architecture. http://docs.jboss.org/optaplanner/release/6.2.0.Final/optaplanner-docs/html_single/index.html#architectureOverview. Accessed: 2015-08-24.

[40] JBoss. Jboss optaplanner optimization algoritms. http://docs.jboss.org/optaplanner/release/6.2.0.Final/optaplanner-docs/html_single/index.html#optimizationAlgorithmsOverview. Accessed: 2015-08-24.

[41] Seyyed Khandani. Engineering design process. 2005.

[42] Matthias Koch. *Fraunhofer BPMS Studie 2014, Ergebnisbericht, Bizagi*. Fraunhofer IESE, Kaiserslautern, 2014.

[43] Artur Latifov. Dynamic enterprise architecture-from static to dynamic models. 2012.

[44] Avraham Leff and James T Rayfield. Web-application development using the model/view/controller design pattern. In *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*, pages 118–127. IEEE, 2001.

[45] Frank Leymann. Bpel vs. bpmn 2.0: Should you care? In *Business Process Modeling Notation*, pages 8–13. Springer, 2010.

[46] Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1):39–54, 2002.

[47] Vladimir Lifschitz. What is answer set programming?. In *AAAI*, volume 8, pages 1594–1597, 2008.

[48] Process Maker. Process maker product information. http://www.processmaker.com/key-features. Accessed: 2015-08-25.

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

[50] Business Process Model OMG. Notation (bpmn). *FTF Beta*, 1, 2011.

[51] Camunda OMG. Business process management notation 2.0 (bpmn 2.0). http://docs.camunda.org/7.3/api-references/bpmn20/. Accessed: 2015-07-15.

[52] Oracle White Paper. What's New in Oracle Oracle BPMS Suite 12c. (July), 2014.

[53] Michael Petritsch. Analysis of grounders and solvers for asp encodings of argumentation frameworks. *Bachelor Thesis, Vienna University of Technology*, 2010.

[54] Louchka Popova-Zeugmann. *Time Petri Nets*. Springer, 2013.

[55] ProcessMaker. Processmaker enterprise edition. http://www.processmaker.com/press-releases/processmaker-launches-its-enterprise-edition-provide-improved-management-too Accessed: 2015-08-27.

[56] ProcessMaker. Processmaker key features. http://www.processmaker.com/key-features. Accessed: 2015-08-27.

[57] August-Wilhelm Scheer, Oliver Thomas, and Otmar Adam. Process modeling using event-driven process chains. *Process-Aware Information Systems*, pages 119–146, 2005.

[58] Dieter Spath, Anette Weisbecker, Dietmar Kopperger, and Rainer Nägele. Business Process Management Tools 2011. 2011.

[59] X4 BPM Suite. X4 bpm suite esb. http://www.x4bpm.com/products/x4-esb.html. Accessed: 2015-08-25.

[60] X4 BPM Suite. X4 bpm suite product information. http://www.x4bpm.com/products/x4-bpm.html. Accessed: 2015-08-25.

[61] Service Task, User Task, Parallel Tasks, Credit Card, and Credit Card. Camunda BPM 7 compared to JBoss jBPM 6. (October), 2014.

[62] Wil MP Van Der Aalst and Arthur HM Ter Hofstede. Yawl: yet another workflow language. *Information systems*, 30(4):245–275, 2005.

[63] Wil MP Van Der Aalst, Arthur HM Ter Hofstede, and Mathias Weske. Business process management: A survey. In *Business process management*, pages 1–12. Springer, 2003.

[64] Fraunhofer Verlag. *Business Process Management - Marktanalyse 2014, BPM Suites im Test*. 2014.

[65] Jan Vom Brocke and Michael Rosemann. *Handbook on business process management 1*. Springer, 2010.

[66] Michael Weber and Ekkart Kindler. The petri net markup language. In *Petri Net Technology for Communication-Based Systems*, pages 124–144. Springer, 2003.

Appendices

Appendix A - The Evaluation of BPMS

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Product Info	Product Description	Modeler and BPM Suite (Freeware Studio + Engine)	BPM Suite	jBPM	Axon Ivy	Oracle Jdeveloper 12c	ProcessMaker	X4 BPMS
	Version	10.6, JEE (.NET and Xpress available)	7.2, final edition - Eclipse Plugin / Standalone	6.2.0 Open Source, Final released	5.1 Version	12.1.3 Designer, Engine Version	Community (Open Source), Enterprise (Trial) both is a WEB UI	4.7 Version
	Vendor	Bizagi	Camunda Inc.	jBoss, Red Hat	Axon IVY	Oracle	Process Maker	Softproject
	Website	http://www.bizagi.com	http://www.camunda.com	http://www.jbpm.org/	www.axonivy.com/	http://www.oracle.com/technetwork/developer-tools/jdev/overview/index-094652.html	http://www.processmaker.com	http://www.x4bpm.com/
Licencing	Licensing Model	BPMN Modeler and Studio is freeware, Engine not	Open Source / Enterprise Version	Open Source (distributed under Apache license v2.0)	Designer for free, Engine within cost models	Free integrated development environment but support with costs	Community for free, Enterprise and Cloud Edition with pricing	30 Days Trial
Process Modelling	Standards (BPMN, BPEL, CMMN, ..)	BPMN 2.0 (Execution not compatible with Environments), WSDL	BPMN 2.0, CMMN (not full coveraged)	BPMN 2.0, BPSim, JPA, JTA	BPMN (XML 97.1.0, BPMN2 is importable but not the standard), EPK	BPMN 2.0, BPEL	BPM workflow approach	BPMN 2.0
	Approach/Language for Workflow Description	Process Modeler with BPMN 2.0 elements	BPMN as Eclipse plugin, organized as a Java Project	One entire locally deployed Webapplication that leads to Life Cycle of BPM Process; Design Modeler for Business User, Eclipse plugin as a Java Project for Developers	XML, Eclipse Environment, additional helpful Elements	BPMN 2.0 in a graphical Tab	BPMN 2.0 in a graphical Tab	BPMN 2.0 in a graphical Tab
	Help of Research & Development is needed	In the beginning	Not as a developer	Webapplication for Business Users (deploying, managing Processes)	Online Help for Designer and Engine available	Online Help available	Online Help available	Online Help available
	Set duration of Task	Yes	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface	Yes by Property Interface

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Modelling	Model is Importable	XML (without graphical positioning), XPDL, Visio	BPMN 2.0 (graphical positioning included	BPMN 2.0 (graphical positioning included	Other versions or plain BPMN2 XML files might be imported anyway, but they are not fully supported (e.g. positions of tasks).	BPMN 2.0 (graphical positioning included	BPMN 2.0 (graphical positioning included	Through BPMN 2.0. Process models from tools such as GBtec, MID Innovator, Signavio or Aris can be read with the BPM Designer and processed further. Thus, previously static process descriptions can be technically implemented after being imported.
	Model is exportable	XML (with Bizagi-Tags), XSD, XPDL, BPMN 2.0 (with bizagi tags), Visio (loss of positioning information), Image, Publishing to Web, Sharepoint and Wiki	XML, BPMN 2.0 and project (.war, .jar, .ear)	XML, BPMN 2.0 and project (.war, .jar)	The internal version in the exported XML file is 97.1.0.	XML, BPMN 2.0 and project (.war, .jar)	BPMN 2.0	BPMN 2.0, projects, PDF
	Support for easy & quick modelling	yes, Microsoft Interface	Eclipse environment	Within jBPM or Eclipse environment	Axon.ivy is based on the Eclipse platform. So when you start Axon.ivy Designer you launch an Eclipse workbench.	Similar to Eclipse but own Environment	Quick modelling in a Webbrowser	Easy handling
	Definition of procedural Facts	BPMN 2.0 and internal property configuration	BPMN 2.0, properties are set in XML form	BPMN 2.0, properties are set in XML form	BPMN (XML 97.1.0)	BPMN 2.0, properties are set in XML form	BPM workflow definitions	BPMN 2.0, properties are set in XML form

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Modelling	Definition of organizaional Rules	Internal Property configuration in the tool	BPMN 2.0, properties are set in XML form; Constraints, e.g. RAL	BPMN 2.0, properties are set in XML form	BPMN (XML 97.1.0)	Busines Rules are an extra Category but within the BPMN	Departments can form hierachic trees. Each department has to name responsible person who can not be part of another department. ProcessMaker has a simple to use, embedded business rules engine which drives the logic behind the process. At each decision gateway, business analysts can build in logic to determine how a process should flow down one particular path instead of another in the business	Business rules are defined graphically or by rule description and executed by the integrated rules engine.
	Model is validated	through internal validation checker, making sure the process passes through all the sequence flows	Depends on the editor: Eclipse Plugin does not support a validation check, Signavio does.	Validation Button in the Designer with visible corrections, Processes can also be checked manually with the RuleFlow Process Class	No Validation in times of designing but afterwards in the simulation of the process (is not the engine!)	Automatic validation and simulation	Advanced Debugger (Enterprise Version)	The automated testing of structured data such as EDIFACT messages, invoices, contracts or other documents is possible without any difficulty. Graphical process debugging; Further testing tools to validate processes against test schemes
Process Realization	Exceptions designable	via property configuration	By BPMN 2.0 Error	By BPMN 2.0 Error	By BPMN Error Event	By BPMN 2.0 Error	BPM elements	By BPMN 2.0 Error
	Define Notification on Process Events	yes (e.g. Email)	Implementation of a Task Listener required	Yes (e.g. Email)	User mail notification can be configured on the Axon.ivy Engine in the Engine Administration UI or in the Workflow UI applications	Email, User, SMS, Voice, IM	Yes (e.g. Email)	Yes (e.g. Email)
	Illustrate Data structure and authorization	via internal property configuration	Via BPMN 2.0	Via BPMN 2.0	Via BPMN style			Via BPMN 2.0

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Realization	Support for easy/quick Programming	Programming in sense of Editing Expressions, nothing else. No Java etc. supported	Based on Java EE	Based on Java EE	In the Designer, projects may only exist in one version at a given point of time. Projects are created and organized inside an Eclipse workspace. On the Designer, the workspace corresponds to the application on the engine what makes it easy to switch for programming.	Based on Java EE	Interface for the definition of business rules but it's necessary to complete a formular (DynaForm): fields of formular are defining the application of the business (conditions). Once definced, it is to be integrated in the BP like a trigger or derivated rule of the BPs.	Based on Java EE with a easy switch between Design phase and Implementation
	Definition of Monitoring Design	WEB UI set by Bizagi Configurations	WEB UI deployed from a Java Project. Pre-defined design, but extensible	Web UI leads through BPM Life Cycle: Project explorer with Authoring, Data Modeler for including external Data, Form Modeler creating Masks for User, Dashboard for out of the box report for process instances, e.g. total amount of order that come into the system	WEB UI deployed from a Java Project. Pre-defined design, but extensible	WEB UI deployed from a Maven project. Predefined design, but extensible	WEB UI with a pre-defined design	The X4 ControlCenter provides functions for testing, monitoring, evaluating and documenting business processes. Also service level agreements deposited within the process model can be monitored in real-time during the process execution.
	Export of Process Data	XML with Bizagi-Tags	Via BPMN 2.0	Via BPMN 2.0,	XML, Axon Ivy Archive (.iar)	Export whole project with BPMN 2.0 files included	Via XPDL	Via BPMN 2.0
	Adapters to connect to External System	ERP, CRM, Legacy	Spring framework etc.		Enterprise Service Bus	File Adapter, Java API	No	SAP, MS-Sharepoint, MongoDB, ..
	Quality Assessment of Implementation	validation of processes; simulation shows if Gateways are synchronized, Messages are sychronized, Decisions probabilities are correctly assigned, Routing behaves as expected, all tokens have ended	There is no validation except from plugins, Unit Test in the Maven environment	Unit testing in the Maven environment	Testing via simulation. Test roles can be defined (they are not equal to the real roles).	Beside Validation and Simulation an Extension for JUnit Tests can be downloaded	JavaScript validates input fields	through validation and testing

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Integration of Systems	94 Interaction with external Data sources	A integration layer enables external Applications (API, Connectors) and external Systemes (ERP, CRM, Legacy). Possible for .NET and JEE. For JEE components are registered	Java Query API, REST Query API, Native Queries (own SQL), Custom Queries, SQL Queries; Distribution for Tomcat, JBoss and Glassfish. In Enterprise version: IBM Websphere AS, Oracle WLS, XSLT Template Engine.	In the Web UI; Form Modeller, Data Modeler: jar., war. etc. uploadable	Sharepoint Add-On over Web Services, access to all functions of Sharepoinot and vice versa, SAP	SQL databases can be configured	The Business Rules Enginer, Advanced Dashboards, Custom Case List Builder, Case Archive/Restore Case Archive \& Restore, SLA Manager Service Level Agreements Manager, Simple Reporting, Advanced LDAP/AD Sync, Batch Routing from Inbox, Advanced Performance Monitor Dashboards, ProcessMaker Monitor & Multi-Tenancy Management, and Microsoft Outlook Microsoft Outlook Connector	The X4 Enterprise Service Bus (X4 ESB) connects IT systems and provides services for a service-oriented architecture (SOA). It enables synchronous and asynchronous processing, messaging, the integration of distributed resources or the transactional processing. Due to its extreme scalability, the X4 Enterprise Service Bus can handle very large amounts of data and meets the highest demands in terms of availability.
	Interaction with Infrastructure Systems e.g. User administratrion, HR-Systems etc.	HR processes are offered within the tool: Offborading, Onboarding, Recruitment	Through API, IdentityService which allows access to the management of user and groups can also be used in conjunction with services such as LDAP.	Pluggable human task service based on WS-HumanTask	Defining Roles, Import and Export Roles in XML-Definition	Extra Organization Overview that also can fix holiday etc	LDAP	Human workflows with user interactions and task management with X4 Activities - also available as mobile version

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Integration of Systems	Interaction with external Applications	SOA layer is implemented to provide the processes funtionality for the ESB (Enterprise Server Bus)	Java API; From the ProcessEngine, you can obtain the various services that contain the workflow/BPM methods. ProcessEngine and the services objects are thread safe. So you can keep a reference to 1 of those for a whole server. Runtime Container Integration; an approach offers access to the Process Engine Service. Available connector for calling web services (SOAP and REST) from the process, languages (XML, JSON, etc.), combination with ESB.	Data Modeller (Drools is a Business Rules Management System (BRMS) solution) is responsible for including applications, Java Clients with remote URL, WS, REST, jBPM Executor allows executing jobs	Enterprise Service Bus provides Routing of messages, transformations of protocols, connections scalability of platforms, (in)dependend deployment, Enterprise Integration Patterns, etc. Call element is waiting for external Events. Interfaces allow access to Java Bean. DROOLS, ESB, REST	Either via Java API or BPMN Data Sources	Via REST API, OAuth 2.0 for authorization	Via Enterprise Service Bus. Also to Web 2.0 (Facebook, Twitter)
Process Execution	Correct Execution in Person's abscence / presence	Before launching, User Interface is defined. A security module allows to define a schema of permissions. Correct handling of defined process operations if individual people are not available.	Since the project is deployed and then started it runs along the defined process.	Since the project is deployed and then started it runs along the defined task-todos.	Since the project was tested through the simulation it then will be deployed and started.	Since the project was tested through the simulation it then will be deployed and started.	Since the project was started it runs on the server	Since the project was tested through the simulation it then will be deployed and started.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Execution	96 Information and Help for End-User	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	Web UI guided by RuntimeManager that involves KieSession and TaskService, manages easily Session managment strategies (Singleton session, Session per request, Session per process instance), Deployment, REST, JMS, Java Client	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help	As far as designed before launching, the user interface should be self-explanatory, no exclusive help
	Organisation's changes during Runtime	as far as concerning changes for in the WEB UI, they are pre-defined	Changable in Web UI during Runtime, e.g. as Admin	Changable in Web UI during Runtime, e.g. as Admin	In the Role Editor	Changable in Web UI during Runtime, e.g. as Admin	In terms of delegation of tasks	Changable in Web UI during Runtime, e.g. as Admin
	Correct Execution of defined Processes	assessment through validation	Testing process definition, process application, your application with other deployments of services, End-to-end integration test including all external systems; Unit Testing, using Mocks, using Arquillian	Parallelism, Symmetric (Load balancing, High Availability), Timer Service using Quartz, Inject services into your Context Dependency Injection environment	The simulation of the process gurantees for the correct execution since there are no errors detected	The Validation and Simulation gurantees a correct execution	ProcessMaker provides DynaForms, or "Dynamic Forms", which are the familiar forms, that can be designed in ProcessMaker to interface with the user while running a case	The automated testing of structured data such as EDIFACT messages, invoices, contracts or other documents is possible without any difficulty.
	Possibilities to start Process Instances	One Run-Button to execute	The process engine is a java library responsible for executing BPMN 2.0 processes and workflows. It is deployed, it is accessible via WEB UI where the process can be startet.	The RunTime Manager will start through Deployment of the REST UI	The program start element allows to start a process by a trigger from embedded external Java code. This opens a possibility to integrate an Axon.ivy application into other applications and systems.	Deploy the Maven Project, run .jar	Entry Task is fixed on Entry form	The RunTime Manager will start through Deployment of the Web UI

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Execution	Interdependency of Processes	Constraints in security module allow a schema of permissions.	process engine supports multi-tenancy: Users should choose the model which fits their data separation needs.	JAXB + JSON is responsible for passing important custom data back and forward to the execution server.	The simulation should notify processes that may be erroneous to each other.	The simulation should notify processes that may be erroneous to each other.		
	Possbilities to access from different channels/devices	Yes (e.g. Login, also on mobile device)	Any Webbrowser	Any Webbrowser	Any Webbrowser, CMS	Any Webbrowser, CMS	Any Webbrowser except Internet Explorer, mobile devices	Any Webbrowser, CMS
Resource Allocation / Scheduling	How human resources are represented	Allocations are done based on user properties which are defined int the Organization Module. Likewise permissions to create and administrate a process and its components are granted according those properties.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the TaskManager User profiles can be created within an Administration account who owns access rights	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the Administration field User profiles can be created including the assignment to groups.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. External Systems like LDAP AD, Novell map roles.	Organization Editor archives human resources	In the User Management. Inside: create roles, groups and departments.	Allocations are done based on properties which are defined in the property window but in the BPMN 2.0 notation as well. In the ControlCenter human resources are listed

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	86 Internal representation of organizational Models	An organization in Bizagi stores the information related not only to the organizational structure of the members of a company and the definition of their characteristics (position, areas,groups), but also the characteristics that make them unique on a team and allow them to be active members in the Processes of the application or applications (roles, skills, geographic location).	There is no hierachy. In the admin area User and Groups can be defined. User can be added to Groups which can also be seen as Role. In the Admin area permissions on each User and Group can be defined. RAL (Resource Assignment Language) can be used to define organizational constraints.	In the BPMN2 Modeler, the Diagram Wizard helps you decide how to document your process-driven organizational models. Organizational model includes naming of company and assigning a repository to it. There is no hierarchy for humans within.	Organizations are represented in through the BPMN notation via the Pool with its lanes. DROOLS BRMS is already integrated.	Organization Editor enables to create and edit the components within an organization. Organizations are composed of roles, organizational chart, holidays, calendars. Organizations are defined at the project level. You can export organizational information to be used within other projects. You cannot create organizational charts, calendars, or holidays using Business Process Composer. You can define roles and assign them to swimlanes. Organization information is not carried over when a project is deployed to runtime.	Departments can be structured in a hierarchical tree. It cannot be exported.	The user, groups, roles and rights is administrated by the Control Center by the administrator.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Administration and Execution of individual Tasks in the System	Work allocation; Bizagi automatically evaluates the allocation rules defined for each Task and selects one or more users that meet the given conditions from the user's list. Guided by a wizard, allocation tool includes load optimization algorithms and deals with delegates and working calendars.	User and Groups can be assigned to tasks. In the administration, authorization of User and Groups can be defined which may play a role on executing a tasks. Spring integration has a special feature for deploying resources. In the process engine configuration, you can specify a set of resources. When the process engine is created, all those resources will be scanned and deployed. There is filtering in place that prevents duplicate deployments. Only in case the resources have actually changed, new deployments will be deployed to the engine database.	Web UI allow instance based view of Tasks. Each Node has a RuntimManager with ist KieSession & TaskService. No matter how many request are sent to the server, they are all processed in parallel. User and Groups can be assigned to tasks. In the administration, authorization of User and Groups can be defined which may play a role on executing a tasks.	Web UI allow instance based view of Tasks. It is possible to administrate tasks and its executin roles.	The implementation of user tasks requires you to define a Human Task. You can use an existing Human Task or define a new one. If your project contains Human Tasks, then they automatically appear in the business catalog under the HumanTasks predefined module. Individual Tasks with ist functions will be set in the BPMN model. They can be controlled in the WEB UI. Human tasks are independent from BPMN processes. If you terminate a BPMN process while it runs a user task, the associated human tasks keeps running independently. If the process instance leaves the user task before the human tasks is completed, the human task continues running and can you can still access it. This is	Roles are used to collect permissions that define a particular function within ProcessMaker, according to a particular scope. A roles is basically just a collection of permissions that defines a function; basically ProcessMaker manage 3 types of roles PROCESSMAKER_ADMIN, PROCESSMAKER_MANAGER and PROCESSMAKER_OPERATOR, these roles have the permissions to: access to the ADMIN menu, access to the Users tab, amongst others.	Business processes frequently involv interaction with users. With X4 Activities, web-based workflow solutions can be implemented within a very short time. Reusable solution components for typical use cases such as operation lists or release workflows allow a quick and easy adjustment to customer-specific requirements. The process logic is defined in X4 processes and provided by X4 ReST as a stateless resource.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	100 User can be assigned to a group	Definition of User Groups in assignations. Bizagi helps to organize the modeling process by categoriz ing each of the rules according to its use. This feature helps the user w hen associating each of the rules in a specific situation by only listing the rules that correspond to the category being used.	It is clear that user and group assignments are quite cumbersome for use cases w here the assignment is more complicated. To avoid these complexities, custom extensions on the user task are possible. assignee attribute: this custom extension allow s direct assignment of a user task to a given user. candidateUsers attribute: this custom extension allow s you to make a user a candidate for a task. candidateGroups attribute: this custom extension allow s you to make a group a candidate for a task.	The Class org.jbpm.task.Organiz ationalEntity contains the subclasses Group and User w hich enables assigning Users to a Group in Java. In the Web UI User can be added to groups.	Roles and users are alw ays configured per application.	In the Organization Editor Users can be grouped	Individual users, groups, departments, roles. Tasks can only be assigned to users or groups. Users can represent assigned roles.	In the ControlCenter

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Scheduling mechansim is given	Bizagi offers a new template Recruitment and Selection Process to assist companies in this process. The Recruitment and Selection Process automates and reduces the time scale in actions such as scheduling and collecting the results of psych technical tests, assigns interviews, updating the list of candidates, etc. Finally provides the possibility of controlling and monitoring the performance of the process, through indicators that can be created using Query Forms and Bizagi's tools like BAM.	Not for humans but for events; The Camunda process engine includes a component named the Job Executor. The Job Executor is a scheduling component, responsible for performing asynchronous background work. Consider the example of a Timer Event: whenever the process engine reaches the timer event, it will stop execution, persist the current state to the database and create a job to resume execution in the future. A job has a due date which is calculated using the timer expression provided in the BPMN XML.	With the integration of OptaPlanner (http://www.kiegroup.org/, http://www.optaplanner.org/): OptaPlanner optimizes business resource usage.	AXON.TAM supports the process of identifying employees' and candidates' potential and skills, so you can recruit and leverage key talents efficiently. This module also helps match employees to the growing needs of your organization.	Oracle Enterprise Scheduler Jobs; Definition of the constraints within which jobs may run, based on factors such as system resources.		ControlCenter schedules process execution with X4 Scheduler. ControlCenter archives human resources as well. Additional: http://quartz-scheduler.org can be activated in the admin area.

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	102 Delegation of Tasks is supported	Work allocation is the fifth step of the Process automation Wizard where Performers are defined for each Activity of your Process. Performers are the users that have the qualities to be assigned to activities. Each Task created for end user interaction requires definition that will allow Bizagi to allocate the correct users within your organization. Bizagi automatically evaluates the allocation rules defined for each Task and selects one or more users that meet the given conditions from the user's list. Only these users will have access to work on the Activity allocated to them.	camunda Tasklist is an end-user friendly web app to manage and complete your user tasks. It supports both direct assignment of tasks to concrete users as well as role based task lists containing tasks that users can claim, delegate and unclaim.	Tasks can be assigned to individuals or to groups, instances can be clustered;	Tasks can be assigned to individuals or to groups.	Tasks can be assigned and delegated in the WEB UI	It allows to delegate tasks by Ad-hoc decision (User/assigned group can re-delegate)	Tasks can be assigned and delegated in the WEB UI The integrated user and rights management allows users, groups, roles, and rights for any Activities application to be defined. The web-based management interface is intended for administrators who want to manage users and permissions for their X4 Activities applications. A connection to an LDAP server is also possible.
	Changing and Controlling of Process Instances during Runtime	No, Before starting the process	RunTimeService is a service which provides access to Deployments, ProcessDefinitions and ProcessInstances.	Context Dependency Injection environment enables injection in an application via the RuntimeManager	Via Ivy Script	In the predifined possibilities it is possible to control in the WEB UI	Roles can be assigned by different Users	In the predifined possibilities it is possible to control in the WEB UI
	Repository	Integrated Repository Management tools to better support the complex mishmash of legacy systems and BPMS components.	It is also possible to access the BPMN model instance by the process definition id using the Repository Service (API).	Organization - Repository - Project, Source Control (Git for version control), Deployment (kjar (knowledge jar = all related artifacts), Maven), Clustered environment	BRMS based repository, Deployment spezialised databases, Sharepoint, ..	SQL Server	MySql, PostgreSQL	Own X4 Repositories

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Resource Allocation / Scheduling	Ability to extend Human Resource e.g. via API	No, Just fixing special attributes in the property section	If you connect the Camunda BPM platform with the LDAP identity service you have read-only access to the users and groups. Create new users and groups via the LDAP system, but not in the admin application. The OptPlanner (Human Resource Planner) provided by JBoss can also be embedded.	jBPM also includes a so-called human task service, a back-end service that manages the life cycle of these tasks at runtime. The jBPM implementation is based on the WS-HumanTask specification. Note however that this implementation is fully pluggable, meaning that users can integrate their own human task solution if necessary.	Rules Engine: DROOLS, ESB (http://www.mulesoft.org) and REST	User specific properties can be defined either in an LDAP or defined outside of LDAP inside BPM Workspace. In most situations, it might not be possible to extend LDAP to specify process specific user extension attributes.	LDAP, REST API	LDAP or using Adapter Development Kit (ADK)
	Export of Organizational Structure	It is possible to share the Extended Attributes that you created in one model and use then in other models. This allows you to maintain a uniform standard in your documented processes by always utilizing the information in the same manner.	The BPMN 2.0 file does not provide full information about the organizational structure. There is no given export for the organizational structure.	The BPMN 2.0 file contains parts of the organizational information	The BPMN 2.0 file contains the organizational information	Organizations are defined at the project level. You can export organizational information to be used within other projects. Not exportable out of the system but during runtime: access vie Identity Service	No	The BPMN 2.0 file contains the organizational information
	Changing of tasks and processes in Case of Exceptions	Only in pre-defined terms	They can be suspended, changed or deleted by the ProcessEngine API	Availability: In case one Node is down, the Session will be restored	As defined before with the exception handlling (partly through BPMN), also controlable in the Engine	The status of a process can be changed in the deployed WEB UI	The status of a process can be changed in the deployed WEB UI	The status of a process can be changed in the deployed WEB UI
Process Controlling	Examination of individual Tasks	No	With the enterprise version the actual status of tasks is traceable	Through instance based Monitoring	Through instance based Monitoring	Through instance based Monitoring		Through instance based Monitoring

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Controlling	Examination of concrete Process Instances	No	With the enterprise version the actual status of processes is traceable	Through instance based Monitoring	Through instance based Monitoring	Through instance based Monitoring		Through instance based Monitoring
	Detection of divergent and problematic Process Instances	Covered through validation	A lab project, the camunda BPM workbench, a debugging tool that allows inspection of the runtime state of processes alongside the process model, allowing breakpoints to be set in the process model (rather than in code). A console interface allows for interrogation and updating of the process variables as the developer steps through the process. The process model is displayed using the bpmn.io viewer.	In case of problems the UI will monitor this. At design time they will also be displayed.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.	In case of problems the UI will monitor this.
	Traceability of Procedure of concrete Process Instances	No reports of any tasks	Java Logging and history view in the Web App	Logging as an option in the connection section	Forum, frequently serviced, Chat, Blog	History is visible to all processes and tasks also of individuals	The Simple Reports plug-in allows ProcessMaker administrators to create Report Tables based on any data involved in a process and grant permission for select users to view and export the report directly from their ProcessMaker inbox.	Automated documentation of business processes, services, adapter configurations and transformations

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Process Controlling	Definition and Analyse of relevant Key Data	No	A business dashboard by modifying Cockpit, the camunda IT operations dashboard, to remove all of the “dangerous” operations while exposing the work in progress, and adding some additional functions such as searching by a business key	From the data provider perspective there exists 3 data providers in charge of retrieving the data needed by all the key performance indicators of the jBPM Process Dashboard. These data provides are all defined in the Dashbuilder tooling data provider management screen.	The definition of key performance indicators (KPI) is essential for the proper implementation of possible measures.	Once you have your application's services in place, you can use JDeveloper to create data controls that provide the information needed to declaratively bind UI components to those services.	The Dashboards & Key Performance Indicators (KPIs)	The X4 ControlCenter provides functions for testing, monitoring, evaluating and documenting business processes. Also service level agreements deposited within the process model can be monitored in real-time during the process execution.
BPM-Governance	Representation of company-specific BPM-Management	Partly; It is not possible to draw the BPM-Cycle in the Suite.	Through BPMN 2.0	Through BPMN 2.0	Through BPMN, but without full support of BPMN 2.0	Through BPMN 2.0	Via BPM forms	Through BPMN 2.0
	Definition and Administration of BPM-specific Roles and Rights	Yes, but there is no possibility to restrict the modelling language but just elements	Yes, BPMN 2.0	Yes, BPMN 2.0	BPMN, XML	Yes, BPMN 2.0 and a extra role definition Category	BPM forms	Yes, BPMN 2.0 and a extra role definition Category
Non Functional Requirements	Internationality through different languages is	Yes, different languages	Yes, different languages (also in the	Yes, different languages (also in the	Yes, different languages (also in the Web UI)	Yes, different languages (also in the	Yes, different languages	
	Usability for different Process types (data centered, document centered, sequential, parallelized, ..)	Yes, BPMN 2.0	Yes, BPMN 2.0	Yes, BPMN 2.0	BPMN and extended elements	Yes, BPMN 2.0	BPM elements	Yes, BPMN 2.0
	Robust for stable System Operation	Yes, covered through internal automatic validation	Yes, covered through testing component	Yes, covered through testing component; Junit-Tests	Yes, covered through a simulation scenario	Yes, covered through a simulation scenario	No Pre-Simulation included	
	Scalability at different Quantity Structure	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API	Yes, by API
Administration	Administration of BPMS Users	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions, Role Administration at Design Time	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions	Yes, at run-time; Admin-Role defines permissions

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Administration	Administration of Processes and Deployment	Yes, at run-time; table of activities	Yes, at run-time; table of activities. The Cockpit also shows in design pattern where the actual token is	Everthing guided by the REST Web UI	Guided by the Engine, when started	Everything guided in the WEB UI	Everything guided in the WEB UI	Everything guided in the WEB UI
	Self-Administration and Administration of End-User	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile	Yes, every End-User can log in to its profile
Setup	Instalability / Complex Installation	Difficult Installation of Server Environment	Simple to deploy with Maven	Seam, Spring, OSGi,	Designer and Engine serperate; Designer uses Eclipse environtment. Engine is a single or clustered application server based on Java	JDeveloper is not difficult to install but needs many computational resources for Windows.	Designer and Engine as Web UI	One-click installation – Less effort for new installation and update. Easy start and less development effort thanks to extensive examples and integration and process patterns
	Client; Adjustment of End-User Workplace	Yes, automatically through deployment	Yes, automatically through deployment	Yes, automatically through deployment	Windows 8, 9 interface	Yes, automatically through deployment	Nothing locally, just in the WEB UI	Yes, automatically through deployment
	Portal; Specified Organization Configuration for visible End-User's Process Start	Depends on configuration: .Net, JEE, Xpress (smal usage)	After Deployment, access via Web App	Sharepoint add-On	Sharepoint add-On	After Deployment, access via Web App	No	After Deployment, access via Web App
	Platform	.Net, JEE	Java EE	Jboss, Java EE	independent	JEE	independet (Web UI)	JEE
	Server; Adjustment of BPMS-Backend	Microsoft SQL Server and Oracle.	MySQL, Oracle, MariaDB, IBM DB2, PostgreSql, MS Server, H2	db2, h2, mysql, postgresql, sqlserver	Microsoft, Oracle, DB2, PostregreSQL	Oracle SQL	MySQL, PostgreSQL	Databases; SQL, MongoDB
Support	Frequently serviced	Yes	Yes		Yes	Yes	Yes	Yes
	Handbook is available	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Quality of Handbook is up	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook	Online Handbook
	Online Developer Community is set	Forum, frequently serviced	Forum, frequently serviced	Forum, frequently serviced, Chat, Blog	Forum, frequently serviced, Chat, Blog	Forum, frequently serviced	Forum, frequently serviced	Forum, frequently serviced
	Further Online Resources (Handbook, Docu, Tutorial,	Workshops	Workshops, Online Demo of Cockpit	Workshops, Demo version	Workshops, Demo version	Workshops, Demo version	Wiki	Workshops, Demo version

Cluster	Feature / Aspect	Bizagi	Camunda	jBPM	Axon IVY	Oracle BPM	ProcessMaker	Softproject
Maturity	Product Maturity and Support	ranked at #1 BPMS Fraunhofer Institute Study 2013, although it is rather based on comfort than on programming	Powerful and flexible developing framework for business applications. Especially for developer it offers innovative, powerful features. The pricing model is flexible and allows a low cost entry.	jBPM strives for the "Zero-Code-BPM"-ideal. Powerful features are missing.	Powerful and easy-to-use BPMS. BPMN2 is not fully supported - for companies not very important, big Clients, one of the Best of Fraunhofer study 2014	Very powerful Application Development Framework with the ability to develop BPM systems. User has to get used to the maturity of possibilities. Ranked at the top of Fraunhofer study 2014	Quick and simple illustration of BPM Workflows in a WEB UI. But not very powerful since there is no great compatibility with other systems.	One of the leading players. Emphasize lies on integration of services and managing them with a powerful organization concept. Also documentation service with export in several formats (Word, Excel) included. Designer based on BPMN 2.0
Usage	Used within Siemens	No	No	No	No	No	Yes	Yes

Appendix B - An ASP Program for Resource Allocation

Listing 1: ASP Program which sets constraints for the input

```
% Author: Giray Havur

#const istop  = "SAT".
#const imin   = 1.
#const imax   = 100.
#const iquery = 1.

#program base.

% default duration of a transition
firingDelay(A,B,0) :- not activityDuration(A,B,_), activityTransition(A,B).
firingDelay(A,B,D) :- activityDuration(A,B,D).

% default resource-activity duration preference handling for assignment
defaultRAD(R,A,B,D) :- resourceActivityDuration(R,A,B,D).
defaultRAD(R,A,B,D) :- roleActivityDuration(L,A,B,D), hasRole(R,L), canExecute(L,A,
    B), not resourceActivityDuration(R,A,B,_).
defaultRAD(R,A,B,D) :- firingDelay(A,B,D), hasRole(R,L), canExecute(L,A,B), not
    resourceActivityDuration(R,A,B,_), not roleActivityDuration(L,A,B,_).

#program cumulative(s).

%%% PETRI NET DYNAMINCS %%%

% Generate action fire
{fire(T,s,B,I) : inPlace(P,T,B), bpInstance(B,I)}.

% fire precondition: if no token at preceding P, then can't fire
:- fire(T,s,B,I), bpInstance(B,I), inPlace(P,T,B), not tokenAt(P,s,B,I).

% fire effect: if fire at s-1, token at next place
tokenAt(P,s,B,I) :- fire(T,s-1,B,I), outPlace(P,T,B), bpInstance(B,I).

% fire constraint only 1 succeeding transition can fire
:- inPlace(P,T1,B), inPlace(P,T2,B), T1!=T2, fire(T1,s,B,I), fire(T2,s,B,I),
    bpInstance(B,I).

% inertia: tokenAt(P,s): if not fire token remains at its place (TOKENAT FRAME)
consumeToken(P,s,B,I) :- inPlace(P,T,B), fire(T,s,B,I), bpInstance(B,I).
tokenAt(P,s,B,I) :- tokenAt(P,s-1,B,I), not consumeToken(P,s-1,B,I).

%%% TIME MANAGEMENT %%%

% Max time
maxTimeAtInPlace(P,T,s,B,I) :- inPlace(P,T,B), not greaterTimeExistsAtInPlace(P,T,s
    ,B,I), fire(T,s,B,I), bpInstance(B,I).
greaterTimeExistsAtInPlace(P1,T,s,B,I) :- inPlace(P1,T,B), inPlace(P2,T,B), fire(T,
    s,B,I), timeAt(P1,C1,s,B,I), timeAt(P2,C2,s,B,I), P1!=P2, C1<C2, bpInstance(B,I
    ).

% fire effect on time: NOT ACTIVITY: time doesn't change
%timeAt(P2,X,s,I) :- not activity(T), fire(T,s-1,I), X = #max{C: timeAt(P1,C,s-1,I)
    , inPlace(P1,T)}, outPlace(P2,T), bpInstance(I).
timeAt(P2,X,s,B,I) :- not activityTransition(T,B), fire(T,s-1,B,I),
    maxTimeAtInPlace(P,T,s-1,B,I), timeAt(P,X,s-1,B,I), outPlace(P2,T,B),
    bpInstance(B,I).

```

```
49 %timeAt(P2,X+N,s,I) :- not activity(T), fire(T,s-1,I), X = #max{C: timeAt(P1,C,s-1,
       I), inPlace(P1,T)}, N = #min{C: firingDelay(T,C)}, outPlace(P2,T), bpInstance(I
       ).
50 % fire effect on time: ACTIVITY: time is C2 : time summed up
51 timeAt(P2,C2,s,B,I) :- activityTransition(T,B), assign(R,T,C1,C2,s-1,B,I), fire(T,s
       -1,B,I), outPlace(P2,T,B), bpInstance(B,I).
52 %alternatively
53 %timeAt(P2,C+D,s,I) :- fire(T,s-1,I), activity(T), firingDelay(T,D), inPlace(P1,T),
        outPlace(P2,T), timeAt(P1,C,s-1,I), bpInstance(I).
54
55 % not fire: only transition not activity : no relaxation but can wait  (TIMEAT
       FRAME)
56 timeAt(P,C,s,B,I) :- timeAt(P,C,s-1,B,I), inPlace(P,T,B), not activityTransition(T,
       B), not consumeToken(P,s-1,B,I), bpInstance(B,I).
57
58 % not fire: time relaxation: if activity : relaxation is possible
59 timeAt(P,C+1,s,B,I) :- timeAt(P,C,s-1,B,I), inPlace(P,T,B), activityTransition(T,B)
       , not consumeToken(P,s-1,B,I), bpInstance(B,I).
60 %timeAt(P,C,s,I) :- relaxationAt(P,C,s,I).
61
62
63 %%% RESOURCE ASSIGNMENT %%%
64
65 % assign each activity to a person
66 {assign(R,A,C,C+D,s,B,I): defaultRAD(R,A,B,D)} :- inPlace(P1,A,B), timeAt(P1,C,s,B,
       I), activityTransition(A,B), bpInstance(B,I).
67
68 %:- binded(R,A1,S,I), assign(R1,A1,_,_,S,I), R1!=R.
69
70
71 % an activity can not be fired before assigned
72 %:- not assigned(T,s,I), fire(T,s,I), activity(T), bpInstance(I).
73 :- not assign(_,A,_,_,s,B,I), fire(A,s,B,I), activityTransition(A,B), bpInstance(B,
       I).
74 % an activity can not be assigned if not fired
75 %:- assigned(T,s,I), not fire(T,s,I), activity(T), bpInstance(I).
76
77 % can not assign same task to another person (task identifier: T-I-S)
78 :- assign(R,A,C1,C2,S1,B,I), assign(R1,A,C1,C3,S2,B,I), R!=R1.
79
80 % can not assign same person to another activity at the same time
81 :- assign(R,A1,C1,C2,S1,B,I), assign(R,A2,C1,C3,S2,B,I), C1<C2, C1<C3, A1!=A2. % in
        same bp&i
82 :- assign(R,A1,C1,C2,S1,B,I1), assign(R,A2,C1,C3,S2,B,I2), C1<C2, C1<C3, A1!=A2, I1
       !=I2. % in same bp but different i
83 :- assign(R,A1,C1,C2,S1,B1,I1), assign(R,A2,C1,C3,S2,B2,I2), C1<C2, C1<C3, B1!=B2.
       % in different bp
84
85 % can not assign same person to same activity at the same time in another instance
86 :- assign(R,A,C1,C2,S1,B,I1), assign(R,A,C1,C3,S2,B,I2), C1<C2, C1<C3, I1!=I2. % in
        same bp but different i
87 :- assign(R,A,C1,C2,S1,B1,I1), assign(R,A,C1,C3,S2,B2,I2), C1<C2, C1<C3, B1!=B2. %
       in different bp
88
89 % when assignments overlap
90 :- assign(R,T,Y1,Y2,S1,B1,I1), assign(R,T2,X1,X2,S2,B2,I2), X1>Y1, X1<Y2.
91 :- assign(R,T,Y1,Y2,S1,B1,I1), assign(R,T2,X1,X2,S2,B2,I2), X2<Y2, X2>Y1.
```