

# Defining the Resource Perspective in the Development of Processes-Aware Information Systems

Luis Jesús Ramón Stroppi<sup>a</sup>, Omar Chiotti<sup>b</sup>, Pablo David Villarreal<sup>a</sup>

<sup>a</sup> *CIDISI, National Technological University Santa Fe Faculty, Lavaise 610, Santa Fe, Argentina, S3004EWB*

<sup>b</sup> *INGAR-CONICET, Avellaneda 3657, Santa Fe, Argentina, S3002GJC*

---

## Abstract

*Context.* The resource perspective has impact on the performance of business processes. However, current workflow management systems (WfMSs) provide disparate support to its implementation and business process modeling languages provide limited capabilities for its definition. Thus, it is difficult to specify requirements regarding this perspective and to select an appropriate WfMS to support them in order to obtain a technological solution aligned with the organizational needs.

*Objective.* To provide support to the definition, implementation, verification and validation of resource perspective requirements in the development of Process-Aware Information Systems (PAISs) based on WfMSs.

*Method.* The following activities were carried out: (i) identification of resource perspective aspects in executable workflow specifications, (ii) analysis of the elements provided by the BPMN modeling language to represent these aspects, (iii) development of a framework based on BPMN for defining and implementing these aspects by using the extension mechanism provided by this language, (iv) development of a model-driven development method that leverages the framework to develop PAISs, (v) demonstration of the proposed framework and method through the development of a tool supporting them, a case study, and the evaluation against the Workflow Resource Patterns.

*Results.* A framework, a method and a tool that support the definition of the resource perspective in the development of PAISs.

---

*Email addresses:* [lstroppi@frsf.utn.edu.ar](mailto:lstroppi@frsf.utn.edu.ar) (Luis Jesús Ramón Stroppi),  
[chiotti@santafe-conicet.gov.ar](mailto:chiotti@santafe-conicet.gov.ar) (Omar Chiotti), [pwillarr@frsf.utn.edu.ar](mailto:pwillarr@frsf.utn.edu.ar)  
(Pablo David Villarreal)

*Conclusion.* By using the proposed framework and method, practitioners are able to: define the resource perspective requirements in conceptual process models, select a WfMS as implementation platform, and define the implementation of these requirements maintaining the consistency between the conceptual process models and the workflow specifications.

*Keywords:* Process-aware Information System, Resource Perspective, Workflow, BPMN, Business Process

---

## 1. Introduction

Process-Aware Information Systems (PAISs) are software systems that manage and execute operational processes involving people, applications, and information sources on the basis of process models [1]. A Workflow Management System (WfMS) is a general purpose system that constitutes the implementation platform for PAISs. WfMSs manage the execution of business processes based on executable workflow specifications defining the ordering of the process activities (control flow perspective), the information passed between them (data perspective), and the assignment of responsibilities to the organization's resources for their execution (resource perspective) [2]. WfMSs were integrated over time together with Enterprise Application Integration Systems and Business Rule Management Systems into the so called Business Process Management Systems (BPMSs).

This paper is focused on the resource perspective, particularly on the involvement of human resources in the execution of business processes. The term Resource is used hereinafter to designate human resources.

The Workflow Resource Patterns [3] were defined as abstractions of recurrent requirements supported by WfMSs. These patterns have been used to evaluate the capabilities provided by multiple WfMSs to support the resource perspective [4, 5]. The WfMSs evaluated at that time showed a disparate support to this perspective. This continues to be the case in the present. Different WfMSs work with base on different organizational metamodels and provide distinct concepts to specify the distribution of work among the resources. For example, Bizagi <sup>1</sup> provides a rich organizational metamodel including areas, positions, roles, skills, groups and properties. Also, it allows defining work distribution policies with base on allocation rules, assignation

---

<sup>1</sup><http://www.bizagi.com/>

methods and preconditions. Instead, Bonita <sup>2</sup> provides a more generic organization metamodel which allows defining the membership of users to organizational groups by performing roles and handles work distribution through the assignment and filtering of actors. The Workflow Resource Patterns have also been used to evaluate high level business process modeling languages like BPMN or UML Activity Diagrams [6, 7]. These evaluations showed a poor support of these languages to the representation of the resource perspective.

The limited capabilities offered by modeling languages to represent resource perspective requirements make it difficult to define these requirements in a platform independent way. Most of these requirements are defined directly at the implementation stage, when workflow specifications are created for the selected WfMS. Choosing a WfMS before defining the resource perspective requirements increases the chances of selecting an execution platform that does not support the organizational needs [8]. Another consequence is often an oversimplified definition of the resource perspective, which results in the PAIS being either inflexible or too liberal [9, 10]. Inflexible systems have a negative impact on the process performance when the assigned resources are overloaded or unavailable. In such a case, these systems block the process until the assigned resources are able to fulfill the task even in case there are other suitable resources available to perform it. On the other hand, if the system is too liberal, the wrong people can be allowed to execute work, which may raise security issues.

The objective of this work is to provide an approach that enables organizations to define the resource perspective along the development of PAISs with base on WfMSs. For this purpose, a framework is proposed based on the well-known business process modeling language BPMN [11] in order to allow defining resource perspective requirements in both platform-independent and platform-specific models, and describing the support provided by WfMSs to the resource perspective. A method based on the principles of model-driven development that makes use of the framework is also proposed to support: the definition of resource perspective requirements in a platform-independent way, the selection of a WfMS that supports the identified requirements, the implementation of these requirements with base on the selected WfMS, the verification and validation of the resulting technological solution against the WfMS and the defined requirements, and the generation of executable work-

---

<sup>2</sup><http://www.bonitasoft.com/>

flow specifications for the target WfMS.

The remainder of this paper is structured as follows: Section 2 presents the background. Section 3 presents the framework to define the resource perspective of business processes. Section 4 describes a model-driven development method based on that framework. Section 5 presents the tool developed to support the proposed framework and method. Section 6 evaluates the framework and method through the Workflow Resource Patterns and a case study. Section 7 discusses related work. Finally, Section 8 presents the conclusions and future work.

## 2. Background

### *2.1. Resource Perspective in the Development of PAISs*

Different types of requirements regarding the resource perspective need to be supported by PAISs. These requirements were addressed in the literature through different approaches. In a previous work [12], these requirements were identified and organized in terms of three aspects of this perspective: Resource Structure, Work Distribution and Authorization. These aspects express the behavior of three general purpose components that are present in well-known reference models and architectures of WfMSs, such as the proposed by Workflow Management Coalition[13], Pesic et al. [14] or WS-HumanTask [15]. These components are: Organizational Repository, Work Distribution and Worklist Handler. These components enable WfMSs to present the work of tasks to the resources in the form of work items.

The Resource Structure aspect refers to the representation of information about the resources required to support the distribution of work items. This is defined through the characterization and classification of resources. The characterization consists in describing the attributes of the resources. The classification is the association of resources with a concept. It allows referencing groups of resources and to assign them common sets of properties. Resources can be classified using different criteria based on organizational or technological concepts [16]. Examples of such criteria include functional classification (often represented through roles), or organizational classification (represented through organizational units positions) [2]. The definition of this aspect results in a resource model or resource schema stored and managed by an Organizational Repository.

The Work Distribution aspect is concerned with the advertisement and binding of work items to the resources for their execution [3]. There are

two basic types of work distribution strategy: pull and push. Pull strategies advertise work items to one or more resources who can voluntarily undertake the responsibility for their execution. Push strategies assign work items to typically a single resource who has the responsibility to fulfill them.

A work distribution strategy includes a resource assignment or allocation definition that specifies the set of resources allowed to handle the work items of a task [17]. Resource assignments are typically defined as a query to the information defined in a resource schema stored in an Organizational Repository, or as an expression evaluating data available in the scope of the process instance. Constraints such as binding or separation of duties can also be defined for resource assignments [18, 19]. These constraints, also referred to as resolution constraints, are often used when specifying push work distribution strategies in order to ensure that the work item is assigned to a single resource for its execution. These constraints can be either hard or soft. The former are always enforced by the system, the latter define desirable restrictions that can be violated under certain conditions [20].

More advanced work distribution requirements comprise changes in the distribution of work items upon the occurrence of events or as consequence of an operation executed by a resource. The so called detour resource patterns [3] identify recurrent requirements of this type. An escalation designates a detour performed by the WfMS upon the occurrence of an event triggered in the scope of the work item. Examples of such events are the *startDeadline* and *completionDeadline* defined by WS-HumanTask [15]. Escalations are often defined in order to reduce the completion time of work items by changing their distribution when a deadline is not met [21]. Detours triggered by human resources such as delegation [22] give some degree of freedom to the resources for changing the assignment of work items.

Resources access and complete the work items distributed to them through the Worklist Handler. This component allows executing two kinds of operations. Worklist operations enable resources to organize the work items distributed to them. Work item operations allow resources to modify the state of work items by setting its outcome or changing its distribution.

The resource perspective also comprises the specification of security-related information at design time [23]. This implies defining the worklist and work item operations the resources are allowed to execute. The Authorization aspect is concerned with the definition of privileges granted to resources in order to execute these operations.

The Authorization aspect can be divided into static and dynamic [24].

The Static Authorization aspect refers to the definition at design time of the privileges granted to resources into the resource model. These privileges are referred to as Resource Privileges. They are used to grant the execution of worklist operations or work item operations. The Dynamic Authorization aspect refers to the definition of privileges granted to resources for each work item of a task at runtime. These privileges are referred to as Task Privileges. They can be only used to grant work item operations.

With the purpose to support the definition of these aspects of the resource perspective of business processes along the development of PAISs, several issues should be addressed:

- The definition of the resource perspective requirements in conceptual models of business processes.
- The selection of a WfMS that fits the resource perspective requirements defined in the conceptual models.
- The implementation of the resource perspective requirements by defining process models and workflow specifications based on the selected WfMS.
- The checking that the implementation fulfills the resource perspective requirements defined in the conceptual models.

To deal with the above issues, this work proposes a framework and a method, which are discussed in following sections.

## *2.2. Research Method*

The proposed framework and method were developed by following a design science research methodology [25]. This methodology was selected because it provides a commonly accepted framework for carrying out design science research in information systems and a suitable mental model for its presentation.

This research started with a problem-centered approach. It was triggered by the observation of the limited capabilities offered by modeling languages to represent resource perspective requirements and the disparate support provided by WfMSs to their implementation, which hinders the definition of this perspective during the development of PAIS.

First, different kinds of resource perspective requirements that may arise during the development of PAISs were classified into the aspects defined in a previous work [12] as described in Section 2.1. The problems in the definition of these aspects during the development of PAIS were identified and objectives for the framework and method were inferred from them.

Second, the elements provided by BPMN 2.0 [11] for the representation of these aspects were analyzed. This modeling language was selected as the basis for the development of the proposed artifacts because of the following reasons: (i) it is a broadly accepted language that provides a common vocabulary for practitioners, (ii) its metamodel provides basic resource perspective elements that make it easier for practitioners to understand the elements introduced by the framework, (iii) it provides an extension mechanism that allows introducing new elements to the language keeping the portability of the models across tools, (iv) existing tools supporting the language can be reused and extended to support the framework.

Third, a framework based on BPMN was designed. A Resource Perspective Implementation Metamodel (RPIMet) was developed to describe the support of WfMSs to the resource perspective. Its elements were defined with base on the aforementioned aspects and resource-related elements of the BPMN metamodel. Also, two extensions to the BPMN metamodel proposed in [12, 26] were redefined to allow representing a broader set of resource perspective requirements in conceptual models, and their implementation in platform specific models. These BPMN extensions were defined by following a method proposed in [27], which provides a UML profile called BPMN+X that allows defining BPMN extensions graphically.

Fourth, a method that makes use of the artifacts provided by the proposed framework was developed to support the definition of the resource perspective during the development of PAISs. It was proposed with base on the principles of Model-Driven Architecture [28] and Model-Driven Development [29].

The applicability and suitability of the framework and method were demonstrated through the development of a tool supporting them and a case study, which provided a basis to evaluate their effectiveness to address the stated problems. In addition, the framework was evaluated in terms of the Workflow Resource Patterns [3].

### 3. A Framework to Define the Resource Perspective of Business Processes

This section presents a framework that supports the definition of the resource perspective of business processes in both platform-independent and platform-specific modeling artifacts.

The structure of the framework is depicted in Figure 1. It provides a Resource Perspective Implementation Metamodel whose instances represent the entities provided by WfMSs to implement the resource perspective aspects. It also includes two BPMN extensions developed with base on this metamodel to enable the definition of these aspects in BPMN process models. Section 3.1 discusses the elements BPMN provides to define the resource perspective. Section 3.2 describes the Resource Perspective Implementation Metamodel. Section 3.3 describes the Resource Structure and Work Distribution extensions.

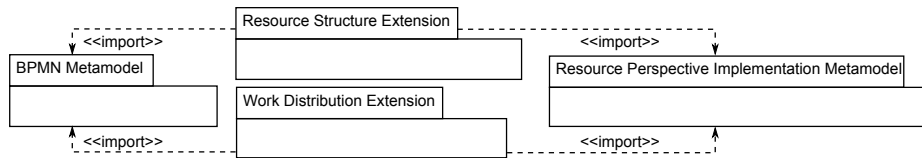


Figure 1: Framework for defining the resource perspective of business processes.

#### 3.1. BPMN Support to the Resource Perspective

BPMN defines two types of task with human involvement: Manual Task and User Task. Manual Task represents work performed by resources without the assistance of a software application. User Task represents a workflow task scheduled through a task list manager of some sort that is performed by resources assisted by software applications [11]. This work focuses on defining the resource perspective for user tasks.

The BPMN metamodel provides elements to define the allocation of resources to process activities, which are depicted in Figure 2. A question that arises is to what extent these elements support the resource perspective aspects described in Section 2.1.

The Resource and ResourceParameter elements represent concepts of the Resource Structure aspect. Resource is an abstract representation of human and non-human resources that can be referenced by process activities. A



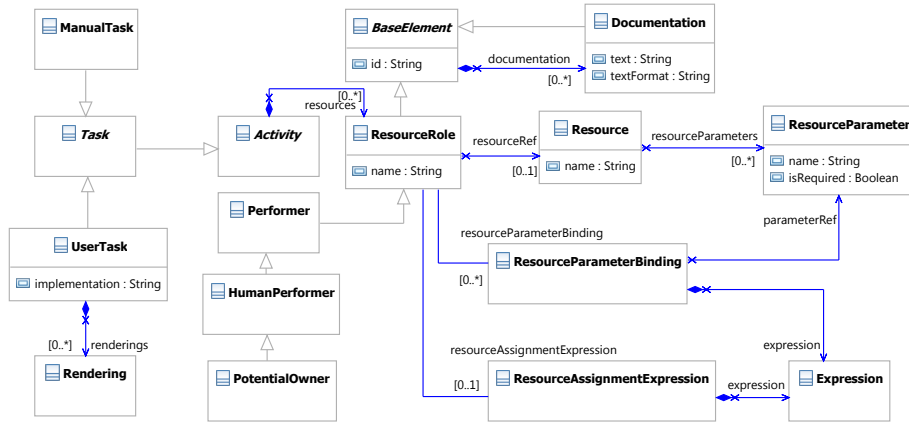


Figure 2: BPMN Metamodel for Resource Assignment.

Resource may contain zero or more ResourceParameter elements defining information about the resources that can be bound and queried at runtime to perform resource assignments [11]. However, the definition of the Resource Structure aspect is out of the scope of BPMN. It assumes this aspect to be defined in a separate model. To the best of the authors knowledge, no standard specification exists to define this aspect to be used in conjunction with BPMN.

The Work Distribution aspect is addressed in BPMN by assigning resources to roles they can play on each task. A role (ResourceRole) defines a set of interactions that a resource may perform in order to progress the work of a task by executing work item operations. The assignment of resources to roles is defined by using ResourceAssignmentExpression or ResourceParameterBinding elements. A resource assignment expression allows defining the resources to be assigned to the role by evaluating data in the scope of the task. ResourceParameterBinding in conjunction with resource references (resourceRef) allow defining a parameterized resource query against an organizational repository.

BPMN defines a hierarchy of roles that starts with ResourceRole and ends with PotentialOwner, which is the only specific resource role defined by the language. A PotentialOwner element in a user task indicates the assignment of resources who can claim the work items of the task to fulfill them. Whether the work distribution strategy defined is pull or push depends on the meaning

of the role. The Potential Owner role allows defining a pull work distribution strategy. BPMN does not support push work distribution strategies. It foresees the definition of additional roles by specializing HumanPerformer. However, this requires introducing new elements to the BPMN metamodel, which is not frequently supported by modeling tools. This often results in practitioners taking conventions such as assuming a push work distribution approach when a single resource is assigned to the Potential Owner role. These can lead to communication problems between practitioners.

The Lane element is often used in BPMN diagrams to represent resource assignments or by grouping activities by position or organizational unit. However, Lane is a generic element to organize and categorize activities with base on any criteria. Therefore, it is only useful for visualization purposes.

ResourceRole and its subclasses also allow representing the Authorization aspect. However, the work item operations that a role grants to the resources need to be inferred based on natural language descriptions included in the specification of the language. The existence of a single specific role is also a limitation, as it defines a single set of task privileges.

Thus, more artifacts and elements than the provided by BPMN are needed to define and communicate resource perspective requirements in process models.

### 3.2. Resource Perspective Implementation Metamodel

The aim of the Resource Perspective Implementation Metamodel (RPIMet) is to enable the representation of entities provided by WfMSs to implement the resource perspective aspects. The instances of this metamodel, which are called Resource Perspective Implementation Models (RPIMs), allow documenting the support provided by implementation platforms to this perspective. This enables practitioners to evaluate and compare WfMSs to select the one that better fits their requirements. This metamodel is based on the *Resource*, *ResourceParameter* and *ResourceRole* generic elements defined by BPMN. In this way, it links the concepts described in Section 2.1 to the elements of BPMN.

Figure 3 depicts the elements of the RPIMet that represent implementation entities that address the Resource Structure and Static Authorization aspects. *ResourceImpl* defines an implementation of the *Resource* BPMN element in a WfMS. There are two kinds of *ResourceImpl*. *HumanResourceImpl* represents a single resource. *ResourceClassifierImpl* represents a group of resources with common characteristics. These elements inherit three attributes

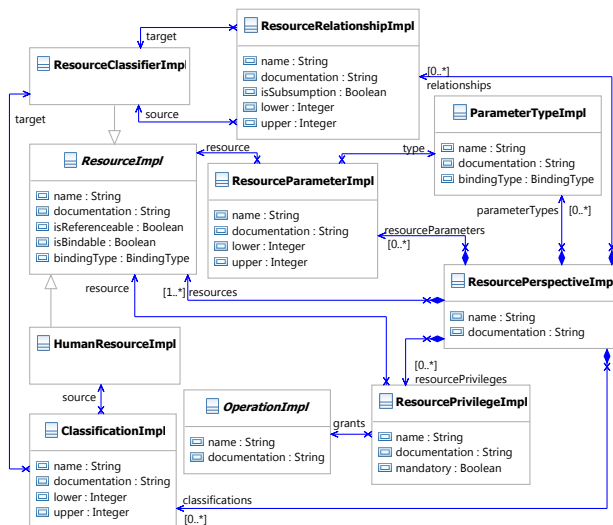


Figure 3: Resource Structure and Static Authorization RPIMet Elements.

from *ResourceImpl*. The *isReferenceable* attribute specifies whether the resource can be referenced as part of a parameterized resource query. The *isBindable* attribute defines whether the resource can be retrieved by a resource assignment expression. The *bindingType* attribute is only defined in case that *isBindable* was set to true. It specifies whether the resource element can be evaluated against a literal value, a variable in the scope of the work item, or an expression combining variables and functions.

*ResourceParameterImpl* represents an implementation of the ResourceParameter BPMN element in a WfMS. It allows specifying the types of parameter of a WfMS that can be defined for each resource entity together with lower and upper cardinalities. A parameter type is represented by a *ParameterTypeImpl* element containing a *bindingType* attribute that specifies whether the parameters of the given type can be evaluated against a literal value, a variable or an expression.

The RPIMet also provides elements to define relationships between *HumanResourceImpl* and *ResourceClassifierImpl* elements. *ClassificationImpl* is a relationship by which a human resource is included in the population of a resource classifier. *ResourceRelationshipImpl* specifies a directed relationship between two resource classifiers. It provides a boolean *isSubsumption* attribute to indicate whether the source classifier is included into the tar-

get one or not. Both kinds of relationship allow defining *lower* and *upper* cardinalities.

*ResourcePrivilegeImpl* specifies a privilege to be granted statically to resources. Two kinds of operation can be granted (see Figure 4). *WorklistOperationImpl* specifies an operation allowing resources to visualize and organize the list of work items distributed to them. *WorkItemOperationImpl* specifies an operation allowing resources to progress or fulfill work items. A work item operation may result in changes to the roles assigned to the resources. The *escalates* attribute defines the roles an operation withdraws to the resources. The *triggers* attribute defines the roles an operation allocates to the resources.

Figure 5 shows an example of RPIM, which represents the entities the Bonita WfMS provides to support the resource perspective. Figure 5a shows the entities supporting the Resource Structure and Static Authorization aspects. User and Administrator Bonita entities are human resource implementations. Role, Membership and Group entities of Bonita are resource classifier implementations. All these entities can be referenced directly in a resource assignment or bound as the result of the evaluation of an expression, as it is derived from the attributes *isReferenciable* and *isBindable* with true value.

In Bonita, Users and Administrators can only be classified by Membership, which indicates that a resource performs a Role in a Group. Users and Administrators are included in the population of Roles and Groups indirectly through the subsumption resource relationship implementations depicted by using arrows with hollow heads. Hierarchies of Groups can also be defined through the *belongsTo* subsumption relationship implementation. Bonita allows defining resource parameters only for individual resources through the customProperty or manager resource parameter implementations. This WfMS grants the Reorder worklist operation to Users and Administrators. Also, it grants to Administrators the ViewAll worklist operation and two additional operations to assign and unassign work items.

Figure 4 depicts the elements the RPIMet provides to describe the support of WfMSs to the Work Distribution and Dynamic Authorization aspects. *ResourceRoleImpl* specifies the implementation of a *ResourceRole* BPMN element in a WfMS. A resource role is defined in terms of the set of task privileges (*TaskPrivilegeImpl*) that it *grants* to the resources in order to execute work item operations (*WorkItemOperationImpl*). The *ResourceRoleImpl* element also has attributes defining the way the resource role have to be assigned

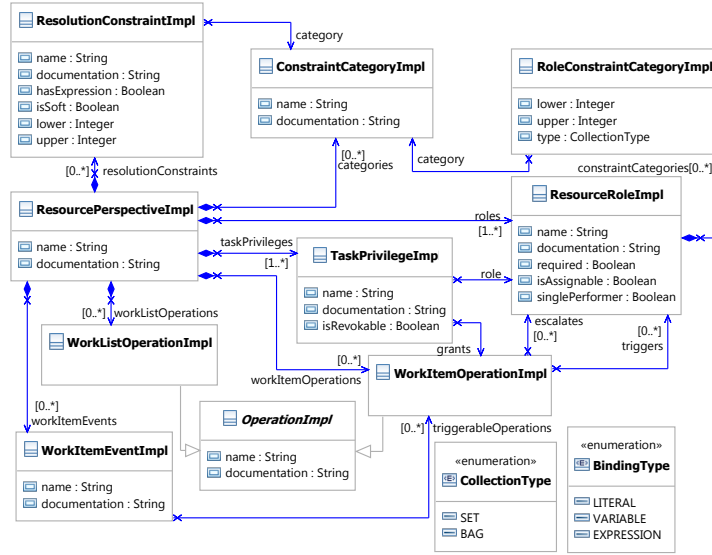


Figure 4: Work Distribution and Dynamic Authorization RPIMet elements.

to the resources. The *required* attribute specifies whether the assignment of the role is mandatory for all user tasks of a business process. The *isAssignable* attribute specifies if resource assignments can be defined or not for the role. When this attribute is false, the resource role can only be granted to resources through the execution of work item operations provided to them by another resource role. Finally, *constraintCategories* defines a set of *RoleConstraintCategoryImpl* elements specifying a group of resolution constraints to be applied during the resolution of the resources for the role, with indication of lower and upper cardinalities and if repeating constraints with the same implementation is allowed or not.

The *ResolutionConstraintImpl* element defines a strategy to restrict the resources resulting from a parameterized resource query or a resource assignment expression. It includes a *hasExpression* attribute to specify whether instances of the defined constraints have an associated expression or not. The *isSoft* attribute allows distinguishing between hard and soft resolution constraints. This element also defines *lower* and *upper* cardinalities for the resulting sets of resources and a reference to a *ConstraintCategoryImpl* element to include the constraint in a category.

Finally, *WorkItemEventImpl* defines an event triggered and handled in

the scope of a work item. It allows defining a set of work item operations that can be triggered upon the occurrence of the defined event in order to alter the resource assignments.

Bonita defines two resource role implementations: Actor and AssignedActor (Figure 5b). The Actor role is assignable and represents a pull work distribution strategy, since it grants the *take* and *do* work item operations that allow users to assume and start the execution of work items voluntarily. The assignment of this role can also be constrained by specifying ActorFiltering constraints. The AssignedActor resource role can be assigned to a single performer. It provides operations to fulfill the work items. Thus, it defines a push work distribution approach. It is assignable only by specifying an AssigningActorFilter resolution constraint, otherwise this role has to be obtained by the resources by executing the take or do work item operations granted by Actor.

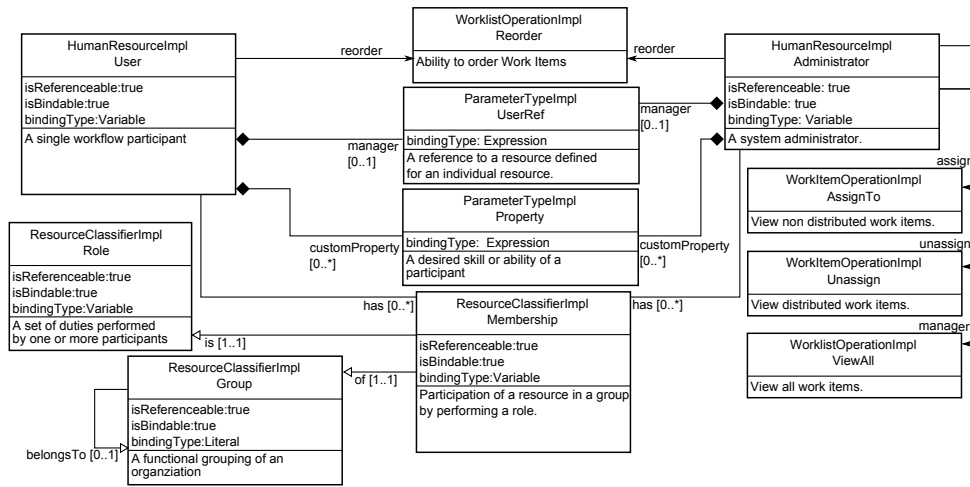
### 3.3. BPMN Extensions to Support the Resource Perspective

This section presents the Resource Structure and Work Distribution BPMN extensions developed with base on the RPIMet to define the resource perspective in BPMN process models. These extensions were developed by using the BPMN+X UML profile. See [27] for further details on the meaning of the employed stereotypes.

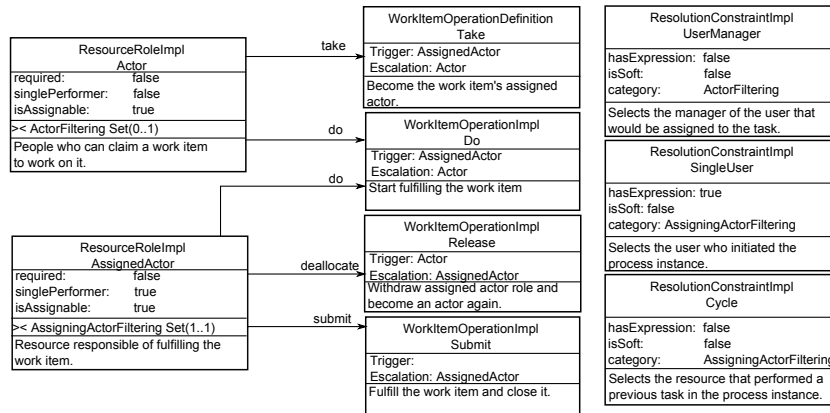
#### 3.3.1. Resource Structure Extension

The Resource Structure extension (Figure 6) allows defining the Resource Structure and Static Authorization aspects by extending the *Resource* and *ResourceParameter* BPMN elements. A model consisting of these extended elements is called Resource Structure Model (RSM). Defining an RSM by using BPMN elements makes it easier to import resource descriptions in process models in order to define resource assignments.

The Resource Structure aspect is supported by introducing the Human Resource and Resource Classifier concepts as extensions to the *Resource* BPMN element. Human Resource designates a single resource. Resource Classifier provides a means to group resources for its characterization. Any criteria can be used to classify resources, such as an organizational department, a position or a work shift. Resources can be characterized individually or in groups through the *ResourceParameter* BPMN element, which was extended to store values for parameters (ValuedParameter extension defini-



a) Resource Structure and Static Authorization Elements



b) Work Distribution and Dynamic Authorization Elements

Figure 5: Bonita Resource Perspective Implementation Model.

tion). This allows representing any information needed to distribute work items among resources like capabilities, specialties or years of experience.

The *ResourceParameter* BPMN element was also extended to define references between resources (*ResourceReferenceParameter* extension definition). A *Resource Classification* designates the membership of a Human Resource to a Resource Classifier. A *Resource Relationship* designates a directed association between resource classifiers. It provides a boolean *isSubsumption* attribute. In case this attribute is set to true, the population of the source classifier is included in the population of the target one, and the *resource*



Figure 6: Resource Structure Extension.

*parameters* and *resource privileges* of the target classifier are inherited by the source one. Subsumption relationships allow defining hierarchies of resource classifiers, for example to specify relationships between departments and sub-departments. Non-subsumption relationships (*isSubsumption* attribute value is false) represent other kinds of association, such as reporting lines. Defining resource classifications and relationships as extensions to the *ResourceParameter* BPMN element enables querying them by using *ResourceParameterBinding* BPMN elements in the same way than valued parameters.

The extension provides support to the static authorization by defining privileges (*ResourcePrivilege* extension definition) for human resources or resource classifiers.

The meaning of the extended *Resource* and *ResourceParameter* elements can be defined in conceptual models by using the *name* and *documentation* attributes that these elements inherit from the *BaseElement* BPMN element. The *implementation* of these extended elements in a WfMS is defined by referencing elements of an RPIM, which are imported as extension elements (see the elements enclosed in dashed squares in Figure 6). In this way, it is possible to define an RSM describing resource structure and static authorization requirements at both platform-independent and platform-specific levels.

Figure 7 depicts an example of RSM representing part of the resource



structure of a furniture manufacturer. At the left of the screenshot, a toolbar shows the notation provided for the elements of the Resource Structure extension. The depicted RSM defines a Manufacturing department as a resource classifier, which includes the Designer and Manufacturing Manager resource classifiers representing jobs. This inclusion is indicated by using subsumption resource relationships depicted as lines with hollow arrow head. A reporting line from Designer to Manufacturing Manager is represented by a non-subsumption resource relationship depicted as a line with solid arrow head.

The members of the Designer resource classifier are characterized by a specialty valued parameter. The membership of the human resources to the resource classifiers mentioned above is specified by using resource classification relationships depicted as solid lines.

Static authorization requirements were also defined in this model based on some workflow resource patterns [3]. The selectionAutonomy resource privilege is granted to Designer and Manufacturing Manager in order to allow resources to prioritize and organize their work sequence, as defined by the Selection Autonomy resource pattern. In addition, the viewAllocated privilege relating to the Configurable Allocated Work Item Visibility resource pattern is granted to the Manufacturing Manager resource classifier in order to allow John overseeing Designers by accessing the work items allocated to them.

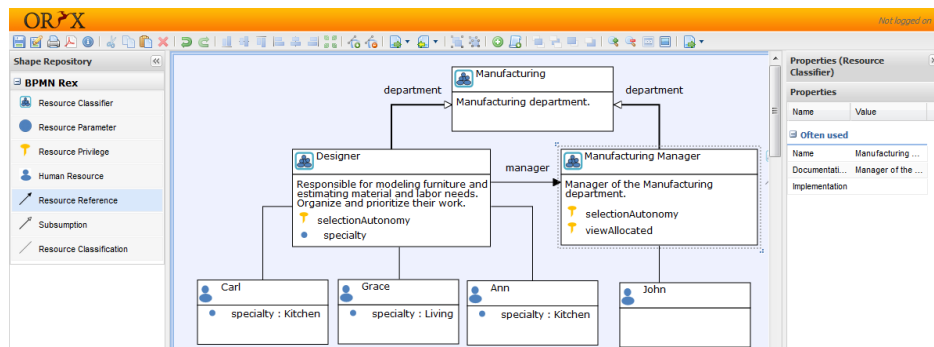


Figure 7: Sample RSM.

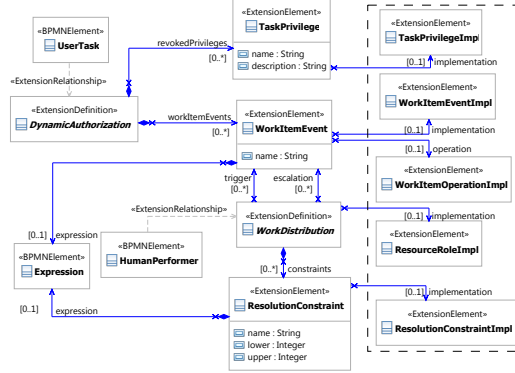


Figure 8: Work Distribution Extension.

### 3.3.2. Work Distribution Extension

The Work Distribution extension (Figure 8) allows defining the Work Distribution and Dynamic Authorization aspects through additional attributes defined by extending the *UserTask* and *HumanPerformer* BPMN elements. A BPMN process model applying this extension is referred to as Extended Process Model (EPM).

The *WorkDistribution* extension definition specifies additional attributes for the *HumanPerformer* BPMN element. It enables specifying *constraints* (ResolutionConstraint) to the resource assignments defined for the role, such as binding or separation of duties. It also defines the *trigger* and *escalation* attributes. A *trigger* allows referencing a work item event defined for the task causing the allocation of resources to the given role. This enables the definition of late distribution requirements [3]. A resource role without an associated *trigger* is assumed to be assigned when the work item is activated. An *escalation* consists in a reference to a work item event defined for the task causing the de-allocation of resources to the role. Deadlines are a typical example of escalation. The *trigger* and *escalation* attributes together enable defining the reassignment of resources without interrupting the user task.

The *DynamicAuthorization* extension definition provides two additional attributes for the *UserTask* BPMN element. The first one is *revokedPrivileges*. It enables restricting the set of privileges granted to the resources allocated to the resource roles associated with the user task. Thus, the set of task privileges granted to resources is the one resulting of subtracting the revoked privileges defined for the user task from privileges granted by as-

signed role. The second attribute provided by this extension definition is *workItemEvents*. It enables defining a set of zero or more events that may take place in the context of the work items of the task. These events can be used for the definition of triggers and escalations.

The Work Distribution and Dynamic Authorization aspects can be described in platform-independent EPMs by using the *name* and *documentation* attributes inherited by *UserTask* and *ResourceRole* from *BaseElement*. Also, platform-specific EPMs can be defined by referencing the elements of an RPIM (see the elements enclosed in dashed squares in Figure 8).

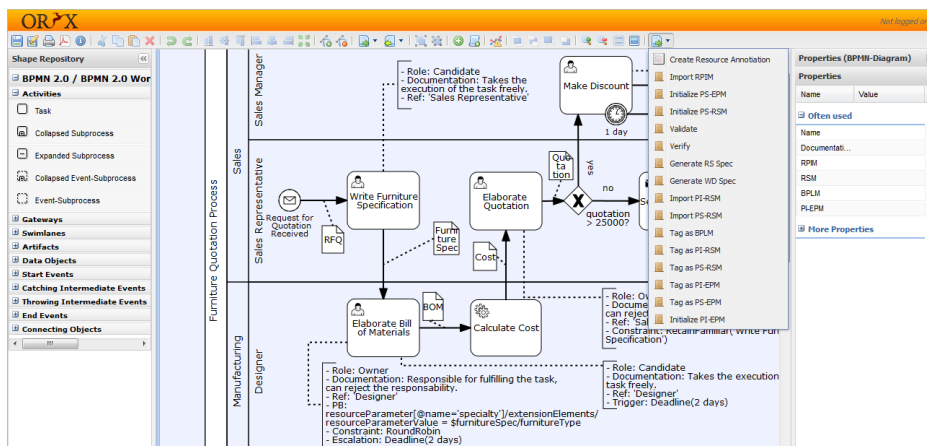


Figure 9: Sample EPM.

Figure 9 depicts an EPM representing a quotation process of a furniture manufacturer. As an example of the definition of work distribution requirements, the Elaborate Bill of Materials user task is described. Two resource roles are defined for this user task, namely Owner and Candidate. The Owner role is initially assigned to a designer whose specialty matches the kind of furniture defined in the furniture specification. This is defined through a parameterized resource query and a resolution constraint. The parameterized resource query includes a parameter binding (PB) that compares the designers' specialties with the kind of furniture specified. The RoundRobin resolution constraint, which is related to the Round Robin workflow resource pattern, allows assigning the distributed role to a single designer on a cyclic basis. The assigned role Owner specifies that the resulting designer is responsible of elaborating the bill of materials. This defines a push work distribution

strategy. This role also allows the resources to reject the responsibility for the assigned work item, as it is described in its documentation. In such a case, the work item is assigned to the next resource in the cycle. A *work item event* is also defined for this user task in order to allow changing the distribution of its work items. If the task is not completed in two days, this event causes the escalation of the Owner role and triggers the allocation of the Candidate role to all the designers without taking their specialty into account. The Candidate role specifies that the designers freely take on the responsibility for performing the redistributed task. This defines a pull work distribution strategy.

#### 4. Model-Driven Development Method to Define the Resource Perspective of Business Processes

This section presents a model-driven development method for defining the Resource Structure, Work Distribution and Authorization aspects in the development of PAISs based on WfMSs. The method is based on the proposed RPIMet metamodel and BPMN extensions. The stages of the method and their resulting artifacts are depicted in Figure 10.

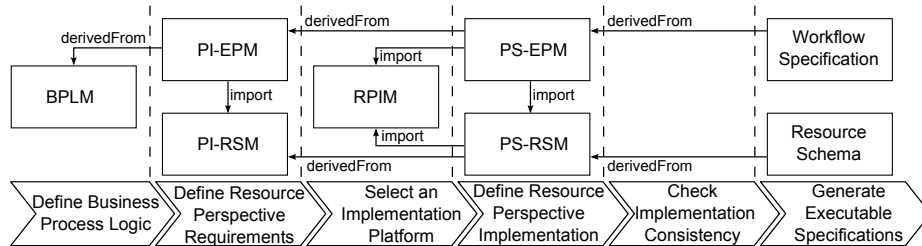


Figure 10: MDD Method to Define the Resource Perspective of Business Processes.

Based on a typical BPMN process model, the method allows defining resource perspective requirements in a platform-independent RSM (PI-RSM) and a platform-independent EPM (PI-EPM). A PI-EPM imports elements from a PI-RSM to represent the distribution of work to the resources. The support provided to these requirements by different WfMSs represented in RPIMs is then evaluated, and an RPIM is selected. The implementation of the requirements in the selected WfMS is then defined in a platform-specific RSM (PS-RSM) and a platform-specific EPM (PS-EPM) by referencing the elements of the selected RPIM. After that, the platform-specific

models are verified and validated. The verification checks the conformance of the PS-RSM and PS-EPM with the restrictions imposed by the WfMS. The validation checks the consistency of these models, i.e. it determines if the requirements defined in the platform-independent models have a correspondent solution in the platform-specific models. Finally, executable process specifications containing the defined resource perspective requirements are generated automatically from the PS-RSM and the PS-EPM for the selected WfMS. The following subsections describe the stages of the method.

#### *4.1. Define the Business Process Logic*

This stage consists in defining the logic of a business process in a BPMN process model referred to as Business Process Logic Model (BPLM). This model describes the control flow and data perspectives. The former is represented through event, activity and gateway elements connected by sequence flow elements. The latter is represented through data objects defining the data input and output of the process activities.

The atomic tasks defined in a BPLM should specify its type to distinguish user tasks requiring the definition of the resource perspective from automatic tasks such as service or script tasks. Lane elements can also be used to represent governance-related aspects such as the organizational units or departments involved in the execution of the process tasks. Further guidelines to define a BPLM can be found in existing process modeling methods, like the proposed in [30]. In addition, approaches to verify the control flow and data perspectives, such as the proposed in [31], may be applied at this stage to obtain BPLM free of errors such as deadlocks and livelocks.

#### *4.2. Define Resource Perspective Requirements*

The aim of this stage is to elicit and agree resource perspective requirements with business analysts and domain experts without taking into account restrictions imposed by any implementation technology. The purpose is to create representations of this perspective that can be readily understood by different stakeholders, and taken as input for the development of the PAIS supporting it. The output of this stage consists of two models: a PI-RSM and a PI-EPM. These models are defined by using the extensions described in Sections 3.3.1 and 3.3.2 without importing and referencing the elements of an RPIM.

A PI-RSM defines a conceptual representation of resource structure and static authorization requirements. In this model, *resource classifiers* and

*resource relationships* are used to capture the groupings of *human resources* and the organization's control structure. The characterization of human resources is defined by means of *valued parameters*. Static authorization is defined by means of *resource privileges*. Both *valued parameters* and *resource privileges* can be specified for *human resources* individually or in groups through their membership to a *resource classifier*, which is defined by *resource classification* relationships.

A PI-RSM can be defined by following two basic approaches, namely Workflow-Driven Resource Modeling or Enterprise-Driven Resource Modeling [32]. The former consists in defining the resources and their groupings with base on the BPLM. This leads to simple PI-RSMs aligned with the BPLMs. However, this approach implies the need to change the PI-RSM upon changes in the process logic and makes it more difficult to reuse the PI-RSMs in different processes. The latter consists in defining the PI-RSM with base on an existing organizational structure. This may result in more complex PI-RSMs that in turn are more likely to be reused and evolve with more independence from the BPLMs.

A PI-EPM defines Work Distribution and Dynamic Authorization requirements. In this model, resource assignments are defined by using *resource assignment expressions*, *parameterized resource queries* and *resolution constraints*. Work distribution policies are represented by describing *resource roles* for each user task defining the ways in which the assigned resources are allowed to interact with work items. The meaning of the resource roles is expressed by using the name and documentation attributes of the *Human Performer* BPMN element. *Escalations* and *triggers* may also be defined to represent changes in the distribution of work items upon the occurrence of events. Dynamic authorization is defined through revoked task privileges specified for each user task. The recurrent requirements identified in the Workflow Resource Patterns [3] can be taken as a basis for the vocabulary used in the definition of these privileges.

#### 4.3. Select an Implementation Platform

This stage consists in choosing an implementation platform to support the implementation and execution of the process under consideration. This is accomplished by evaluating the support provided by different WfMSs, which are represented by RPIMs created or taken from a model repository, to the defined resource perspective requirements. For this purpose, a set of evaluation properties was defined to enable the assessment of an RPIM against

the requirements defined in a PI-RSM and a PI-EPM. This is possible because the framework allows describing resource perspective requirements and implementation platforms with base on the same concepts.

The set of properties with true value for an element of a platform independent model (PI-RSM or PI-EPM) is called the set of *required properties*. The set of properties with true value for an element of an RPIM is called the set of *supported properties*. If the set of *required properties* of an element of a platform-independent model is included in the set of *supported properties* of an RPIM element, it is said that this element *is supported*. Therefore, it is possible to calculate a *support ratio* of an RPIM to a platform-independent model by dividing the number of *supported elements* by the total number of elements of that platform-independent model. This provides a basis for comparing different WfMSs and select one of them as implementation platform with base on the support provided to the defined resource perspective requirements.

A *support ratio* less than one does not imply the inability of a platform to implement the requirements defined in a PI-RSM and a PI-EPM. Instead, this means that the non-supported elements require to be represented in platform-specific models in a different way, by using one or more platform elements in order to fulfill the defined requirements as discussed below.

The support of an RPIM to a PI-RSM is calculated with base on a set of evaluation properties defined for the Resource elements of the PI-RSM (HumanResource or ResourceClassifier), and the ResourceImpl elements of the RPIM (HumanResourceImpl or ResourceClassifierImpl). The evaluation properties to assess the support of an RPIM to a PI-RSM are the following:

- *isResource*: true for *HumanResource* and *HumanResourceImpl* instances.
- *isClassifier*: true for *ResourceClassifier* and *ResourceClassifierImpl* instances.
- *isReferenceable*: true for *Resource* instances which are target of the *resourceRef* reference defined by *ResourceRole* and for instances of *ResourceImpl* with the *isReferenceable* attribute set to true.
- *isBindable*: true for all the *Resource* instances in a PI-RSM if there exists at least one *ResourceAssignmentExpression* in the PI-EPM; it assumes the value of the *isBindable* attribute in *ResourceImpl* instances.

- *subsumes*: true for a *ResourceClassifier* if it is target of a subsumption *ResourceRelationship* and for a *ResourceClassifierImpl* which is target of a subsumption *ResourceRelationshipImpl*.
- *isSubsumed*: true for a *ResourceClassifier* if it is source of a subsumption *ResourceRelationship* and for a *ResourceClassifierImpl* which is source of a subsumption *ResourceRelationshipImpl*.
- *isSource*: true for a *ResourceClassifier* if it is source of a non-subsumption *ResourceRelationship* and for a *ResourceClassifierImpl* which is source of a non-subsumption *ResourceRelationshipImpl*.
- *isTarget*: true for a *Resource Classifier* if it is target of a non-subsumption *ResourceRelationship* and for a *ResourceClassifierImpl* which is target of a non-subsumption *ResourceRelationshipImpl*.
- *hasPopulation*: true for a *ResourceClassifier* which is target of a *ResourceClassification* and for a *ResourceClassifierImpl* which is target of a *ResourceClassificationImpl*.
- *isMember*: true for a *HumanResource* which is source of a *ResourceClassification* and for a *HumanResourceImpl* which is source of a *ResourceClassificationImpl*.
- *hasPrivileges*: true for a *Resource* if it defines one or more *ResourcePrivilege* instances; and true for a *ResourceImpl* associated with one or more *ResourcePrivilegeImpl* instances.
- *hasParameters*: true for a *Resource* if it defines one or more *ValuedParameter* instances and for a *ResourceImpl* associated with one or more *ResourceParameterImpl* instances.
- *hasBindableParams*: true for a *Resource* if it is target of a resource parameter binding and for a *ResourceImpl* associated with at least one *ResourceParameterImpl* instance.

The support of a WfMS to the Work Distribution and Dynamic Authorization requirements defined in a PI-EPM is evaluated through properties for the *ResourceRole* BPMN elements of the input PI-EPM and the *ResourceRoleImpl* element of the candidate RPIM.



- *isAssignable*: true for every *ResourceRole* of a PI-EPM; it assumes the value of the *isAssignable* attribute for *ResourceRoleImpl* instances.
- *multiplePerformers*: true for *ResourceRole* instances not referencing *HumanResource* instances or defining *ResolutionConstraints* with the upper attribute greater than one; and for *ResourceRoleImpl* instances with the *singlePerformer* attribute set to false.
- *singlePerformer*: true for *ResourceRole* referencing *Human Resource* instances or defining *Resolution Constraints* with the upper attribute equal to one; and for *ResourceRoleImpl* instances with the *singlePerformer* attribute set to true.
- *hasConstraints*: true for *ResourceRole* instances defining one or more *Resolution Constraints*; and for *ResourceRoleImpl* instances associated with one or more *RoleConstraintCategoryImpl* instances.
- *hasTriggers*: true for *ResourceRole* instances defining one or more *triggers*; and for *ResourceRoleImpl* instances triggered by at least one *WorkItemOperationImpl*.
- *hasEscalations*: true for *ResourceRole* instances defining one or more *escalations*; and for *ResourceRoleImpl* instances escalated by at least one *WorkItemOperationImpl*.
- *revokablePrivileges*: true for *ResourceRole* defined for a *UserTask* defining one or more *revokedPrivileges*; and true for *ResourceRoleImpl* instances associated with one or more *TaskPrivilegeImpl* instances with the *isRevokable* attribute set to true.

The result of this stage consists of the selected RPIM and a summary of candidate implementation for each element of the input PI-RSM and PI-EPM (See Tables 1 and 2). The actual implementation of these elements is decided in the next stage by considering the meaning of the RPIM elements as discussed below.

#### 4.4. Define Resource Perspective Implementation

The aim of this stage is to define a technological solution based on the selected implementation platform to provide support to the resource perspective requirements defined in the PI-RSM and PI-EPM. The output of

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	hasPrivileges	hasParameters	hasBindableParams
<b>RSM Resource 1</b> <i>RPIM ResourceImpl 1</i> <i>RPIM ResourceImpl n</i>													
<b>Other ResourceImpl</b> <i>RPIM ResourceImpl m</i>													

Table 1: Candidate Resource Structure Implementation

this stage consists of two models: Platform-Specific RSM (PS-RSM) and Platform-Specific EPM (PS-EPM), which are derived from the PI-RSM and PI-EPM, respectively. These models represent the resource perspective requirements defined in their originating PI-RSM and PI-EPM in terms of the elements provided by the selected RPIM.

The PS-RSM and PS-EPM are initialized by creating a copy of their originating models and importing the selected RPIM. Then, their elements are mapped to the target WfMS by specifying their *implementation* attribute consisting in references to elements of the RPIM. This is done with base on the candidate implementation elements identified in the previous stage (Tables 1 and 2). The evaluation performed in the previous stage is based on structural properties, which does not include the semantics of the assessed elements. Therefore, the semantics of these elements specified in their *documentation* attribute should be considered in order to decide if an RPIM element is appropriate to implement a PI-RSM or PS-EPM element.

A PS-RSM is defined by mapping the *Human Resources* and *Resource Classifiers* of the PI-RSM to the *HumanResourceImpl* and *ResourceClassifierImpl* elements of the imported RPIM. For *Human Resources*, it is possible that the RPIM defines zero, one or more than one candidate elements to implement it in a straightforward way. In the two last cases, the implementation of the *Human Resource* is decided with base on their meaning expressed in their *documentation* attributes. In case than no candidate implementation

	isAssignable	multiplePerformers	singlePerformer	hasConstraints	hasTriggers	hasEscalations	revokablePrivileges
<b>EPM Role 1</b> <i>RPIM ResourceRoleImpl 1</i> <i>RPIM ResourceRoleImpl n</i>							
<b>Other ResourceRoleImpl</b> <i>RPIM ResourceRoleImpl m</i>							

Table 2: Candidate Work Distribution Implementation

elements are provided for a *Human Resource*, alternate implementations can be evaluated with base on other elements provided by the platform. These alternate implementations often consist in associating the *Human Resource* with additional *Resource Classifiers*. For *Resource Classifiers*, there is an additional possible scenario that takes place when the WfMS supports its implementation through multiple *ResourceClassifierImpl* elements. In this context, the *Resource Classifier* defined in the PI-RSM should be translated into multiple *Resource Classifiers*, each of them defining their respective implementation.

A PS-EPM is aimed to define an implementation for the work distribution and dynamic authorization requirements expressed in the PI-EPM. A PS-EPM is defined by mapping their *Resource Role* elements to the *ResourceRoleImpl* elements of the imported RPIM. This is also performed with base on the candidate implementation elements identified in the previous stage, and by taking into account the meaning of the *ResourceRoleImpl* elements. In case the RPIM does not provide a candidate implementation for triggers, escalations or constraints alternative implementations can only be defined by introducing changes in the control flow and data perspectives.

Resource assignments are defined in a PS-EPM with base on the PS-RSM. Therefore, in case the PS-RSM presents variations from the PI-RSM because of the restrictions imposed by the chosen RPIM, the assignment of resources to roles for each user task of the PS-EPM needs to be modified accordingly.

#### 4.5. Verify and Validate Resource Perspective Implementation

The fifth stage consists in verifying and validating the resource perspective implementation defined in PS-RSM and PS-EPM. This stage is aimed to check that these models, which may present variations with respect to its originating platform-independent models, respect the restrictions imposed by the selected platform and provide support to the defined requirements consistently.

The verification of a PS-RSM consists in checking that its *resource classifications*, *resource relationships*, *valued parameters*, and *resource privileges* are in conformance with the restrictions and entities of the selected WfMS. The verification rules described below were formalized in OCL and are available at <https://code.google.com/p/bpmn-rsx/>.

- For all *resource classification* in a PS-RSM, the implementation attributes of its source *human resource* and target *resource classifier* should match with the source and target of the respective *ClassificationImpl* element in the RPIM.
- For all *resource relationship*, the implementation attributes of its source and target *resource classifiers* match with the source and target of the corresponding *ResourceRelationshipImpl* element of the RPIM.
- For all *resource relationship*, the *isSubsumption* attribute is should match with the *isSubsumption* attribute of its referenced *ResourceRelationshipImpl*.
- For all *valued parameter*, the implementation attributes of both its containing *resource* and its contained *resource parameter value* should match with the resource and type of the respective *ResourceParameterImpl* element.
- For all *resource privilege*, the implementation of its containing *resource* should match with the *resource* reference of the respective *ResourcePrivilegeImpl* element.

The verification of a PS-RSM also includes checking the lower and upper cardinalities of *resource classifications*, *resource relationships* and *resource parameter values*; and that all mandatory *resource privileges* are granted for the corresponding resources.

The verification of a PS-EPM is performed by checking the *parameterized resource queries*, *resource assignment expressions*, *work item events*, *revoked privileges* and *resolution constraints* defined for each resource role of its user tasks. The following rules were defined:

- For all *parameterized resource query*, the referenced *resource* should have the *isReferenceable* attribute set to true in the RPIM and no functions or variables should be present in bindings to parameters having *bindingType* set to literal.
- For all *resource assignment expression*, the resulting resources should have with the *isBindable* attribute set to true in the RPIM and and no functions or variables should be present in expressions in case their implementation has a literal *bindingType*.
- For all *resolution constraint*, an *expression* is only defined for it when the *hasExpression* of the respective *ResolutionConstraintImpl* is true in the RPIM.
- For all *work item event*, the *WorkItemEventImpl* implementing it can trigger the referenced *operation*, and the *WorkItemOperationImpl* implementing that *operation* allows triggering and escalating the roles referencing the *work item event* through the *trigger* and *escalates* attributes, respectively.
- For all *revoked privilege*, the respective *TaskPrivilegeImpl* defines a true value for the *isRevokable* attribute.

The verification also checks that the *lower* and *upper* limits of each constraint category is respected, and that all *required* resource roles defined in the chosen RPIM are associated with user tasks.

The validation of a PS-RSM against a PI-RSM checks that both models define the same human resources. It also checks that the resource parameters and resource privileges defined for them directly or through their membership to a resource classifier match. The suitability of the implementation for each resource role, resolution constraint and resource privilege is evaluated manually as platform-independent models define them in natural language.

The validation of a PS-EPM consists in determining that its *user tasks*, *resource roles*, *resolution constraints*, *work item events* and *resource assignments* are consistent with those defined in the PI-EPM. This is performed in

three steps. First, it is checked that the same *user tasks* are defined in these models. Then, it is checked that each user task in the PS-EPM defines an implementation for each *resource role*, *resolution constraint* and *work item event* defined by its counterpart in the PI-EPM. Finally, it is tested that the resource assignments defined for each *resource role* results in the same *human resources* in both the PI-EPM and PS-EPM. This can be checked statically with base on the information defined in these models when the resource assignments do not include resource parameter bindings or resource assignment expressions involving variables. Otherwise, values for the variables need to be supplied. Resolution constraints should be checked manually as they are defined in natural language in the PI-EPM.

#### 4.6. Generate Executable Specifications

The last stage consists in generating executable workflow specifications from the verified and validated PS-RSM and PS-EPM. This can be carried out by applying model-to-code transformations developed for each RPIM. These transformations take the elements of PS-RSM and PS-EPM as input and generate XML documents defining the process specifications for the target WfMS platform.

## 5. Tool Support

This section presents a tool that was developed in order to support the proposed framework and method with base on the model editor and repository called Oryx [33]. Three stencilsets were developed to implement the Resource Structure extension, the Work Distribution extension and the Resource Perspective Implementation Metamodel (see Figure 11).

The Resource Structure stencilset enables defining and visualizing PI-RSMs and PS-RSMs (see Figure 7). The Work Distribution stencilset was developed as an extension to the stencilset provided by Oryx to support BPMN 2.0. It enables creating PI-EPMs and PS-EPMs. No additional notation elements were defined for this stencilset. The attributes defined by the Work Distribution extension are depicted by using *Text Annotation* BPMN elements. The Resource Perspective Implementation stencilset allows defining RPIMs, which are visualized through the notation used in Figure 9.

An editor plugin was also developed to provide support to the proposed method. It enables the automatic creation of text annotations to depict the Work Distribution and Dynamic Authorization attributes defined in the user

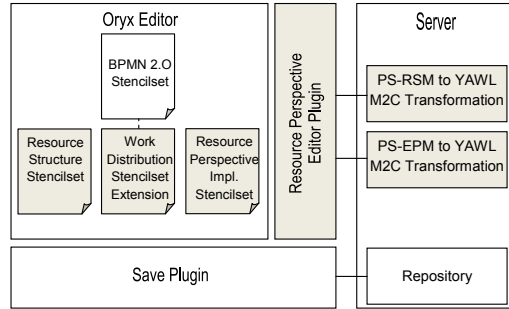


Figure 11: BPMN Oryx Architecture.

tasks of PI-EPMs and PS-EPMs. In addition, it provides a mechanism for the initialization of PI-EPMs with base on BPLMs, and for the initialization of PS-EPMs with base on PI-EPMs and RPIMs. The editor plugin also allows running the verification and static validation rules described in Section 4.5. Finally, it enables invoking model-to-code transformations developed as server-side plugins and registered in the tool in order to generate executable specifications from PS-RSMs and PS-EPMs importing an RPIM (see Figure 9). The developed stencilsets, the editor plugin and the developed model-to-code transformation are available at <https://code.google.com/p/bpmn-rex/>.

## 6. Evaluation

The aim of this section is to show the applicability and utility of the proposed framework and method in practical scenarios and to evaluate their suitability to represent recurrent resource perspective requirements. A case study from a provincial government is presented in Section 6.1. An assessment of the proposed framework against the Workflow Resource Patterns is presented in Section 6.2.

### 6.1. Case Study: Funding Process for Minor Municipal Works

Processes in bureaucratic organizations, such as governments, consist of tasks generally carried out by people. In this context, the performance of a process is highly dependent on the effectiveness and efficiency of the resource assignments in the execution of tasks, which should be performed taking into account the organizational structure, governmental regulations and laws. The proposed framework and method were applied in a provincial

government of Argentina to redesign and implement processes related to the management of funds assigned to municipalities. This section describes the application of the method to a process for the assignment of funds for minor municipal works. The goal of the process is to manage the funding requests received by the provincial government from municipal or local governments, with the purpose of executing a minor infrastructure work (refurbishment of a building, a new local healthcare center, etc.). More than two hundred of funding requests are authorized each year for 50 municipalities and 312 local governments. This requires the coordination of different technical and administrative areas of the provincial government, as well as the coordination of the assignment of tasks to people of these areas.

#### *6.1.1. Definition of the Business Process Logic*

The funding assignment process starts upon the reception of a funding request and ends with its rejection or approval along with the granted amount and terms. Figure 12 depicts the BPLM that represents this process, which was defined with the proposed tool by creating a BPMN process model and tagging it as BPLM.

A funding request received from a municipality or local government, which contains a project describing the minor work to be performed, is routed to the Department of Administrative Assistance where a clerk generates administrative records and assigns it an ID number for tracking purposes. After that, an automatic check is performed to determine if the municipality has pending accountability reports. If that is the case, the municipality is notified. It should present these reports and notify their reception within a month in order to let the process continue. Otherwise, the process ends in the state Municipality Rejected.

If the municipality does not have pending accountability reports, the request is routed to the Department of Finances in order to be assessed against national and provincial laws by a legal advisor. In case the project is in conflict with a law, it is rejected and the process ends with a notification of this result. In case the project may fit the law with minor changes, a notification of the legal issues is sent to the municipality, which should carry out the required modifications within a month in order to avoid the rejection of the request. Then the modified project is received and evaluated again.

If the project passes the legal evaluation, it is evaluated from a technical point of view by an urban planning advisor of the Department of Urban Planning. This evaluation may result in the rejection, the acceptance subject



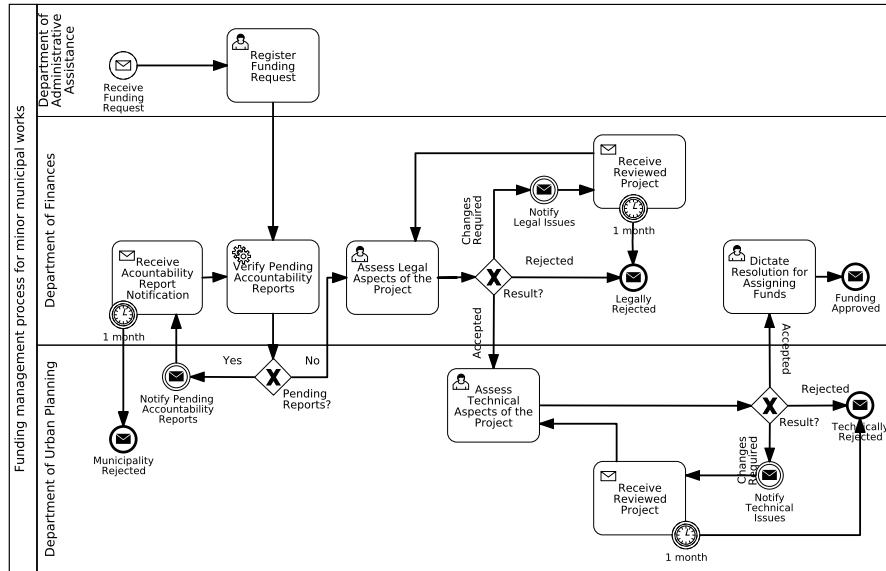


Figure 12: Funding Process Model.

to changes or the acceptance of the project. In the last case, the request is routed to the director of the Financial Department who dictates the resolution approving the assignment of the funds.

### 6.1.2. Definition of the Resource Perspective Requirements

The definition of the resource perspective requirements started with the representation in a PI-RSM of the involved resources. This model was defined with base on the government’s organizational structure, by following an enterprise-driven resource modeling approach [32]. Therefore, the resources were classified by using the concepts employed in the documentation of this organizational structure. The resulting model, which was defined to be reused in other processes, is depicted in Figure 13. It was created by using the editor provided by the proposed tool.

The government’s structure presents three hierarchical levels, namely ministries, secretaryships and departments. They were defined as *Resource Classifier* elements connected by subsumption *Resource Relationships*. The process involves the Department of Administrative Assistance and the Department of Urban Planning that belong to the Secretaryship of Municipal-

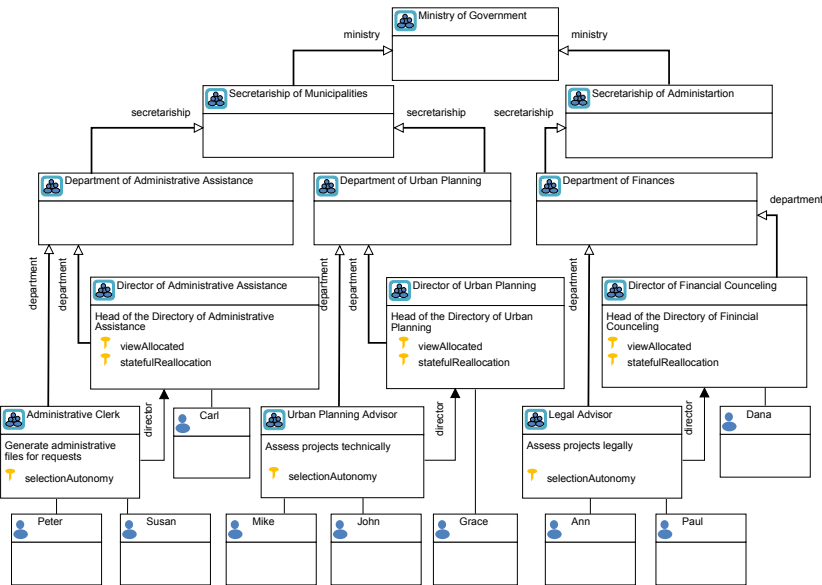


Figure 13: Funding PI-RSM.

ities; and the Department of Finances that is part of the Secretaryship of Administration. The departments define jobs or positions (such as director, clerk or advisor) which were also defined as *Resource Classifier* elements. These positions are associated through reporting lines such as the *director* non-subsumption *Resource Relationships*. These positions are performed by people represented as *Human Resources* associated with the respective positions through *Resource Classifications*. Resource Privileges were also defined on the director and job positions. These privileges represent the Selection Autonomy, Configurable Allocated Work Item Visibility and Stateful Reallocation Workflow Resource Patterns [5].

The work distribution requirements for the user tasks of the funding assignment process were defined in the PI-EPM of Figure 14. This model was derived from the BPLM of Figure 12 by executing the *Initialize PI-EPM* and *Import PI-RSM* commands provided by the tool. The text annotations depicting the defined requirements were generated automatically from the model through the *Create Resource Annotation* command.

The Register Funding Request user task is carried out by clerks of the Department of Administrative Assistance. A requirement for this task was

to fairly distribute its work items among them. To fulfill this requirement, a push work distribution strategy was defined to assign each work item to a clerk on a cyclic basis. This was represented by the Performer *Resource Role* that assigns the responsibility for executing the work items of the task. It defines a *resource reference* (ref) that indicates that clerks perform this role on the task. The *Cycle resolution constraint* indicates that the work items of this task are distributed among clerks on a cyclic basis.

A different work distribution strategy was required for the Assess Legal Aspects of the Project and Assess Technical Aspects of the Project user tasks. Given that the effort required to assess different projects may vary considerably, a push work distribution strategy may have a negative impact on the process performance. Therefore, a pull work distribution approach was taken in order to let advisors decide which projects assess and in what order. Another requirement for these tasks was to let the ability to the director of each department to undertake, oversee and change the assignment of their work items. Two resource roles were defined for this purpose. The first one is Candidate, which allow advisors, to claim and execute the work items. The second one is TaskManager that allows the directors of the departments of Urban Planning and Finances to undertake and assign the work items of these tasks.

Finally, a push work distribution strategy was defined for the Dictate Resolution for Assigning Funds, which has to be carried out by the Director of Finances. For this purpose, the Performer resource role was assigned.

### 6.1.3. Selection of an Implementation Platform

The provincial government has the policy of using open source software to support its activities. Given that the Bonita WfMS had previously been used by the Department of Information Technology for other initiatives, it was required to use it as implementation platform. A structural evaluation was performed to evaluate support provided by Bonita to the work distribution requirements described above. The result of evaluating the Bonita RPIM (see Figure 5 of section 3.2) against the platform-independent models of Figures 13 and 14 is summarized in Tables 3 and 4.

The hierarchical decomposition of the Ministry of Government into secretaryships and departments is supported from a structural point of view by the Group resource classifier implementation provided by Bonita. None of the resource classifier implementations provided by this WfMS support the *isSource* and *isTarget* structural properties of the Clerk, Advisor and Direc-

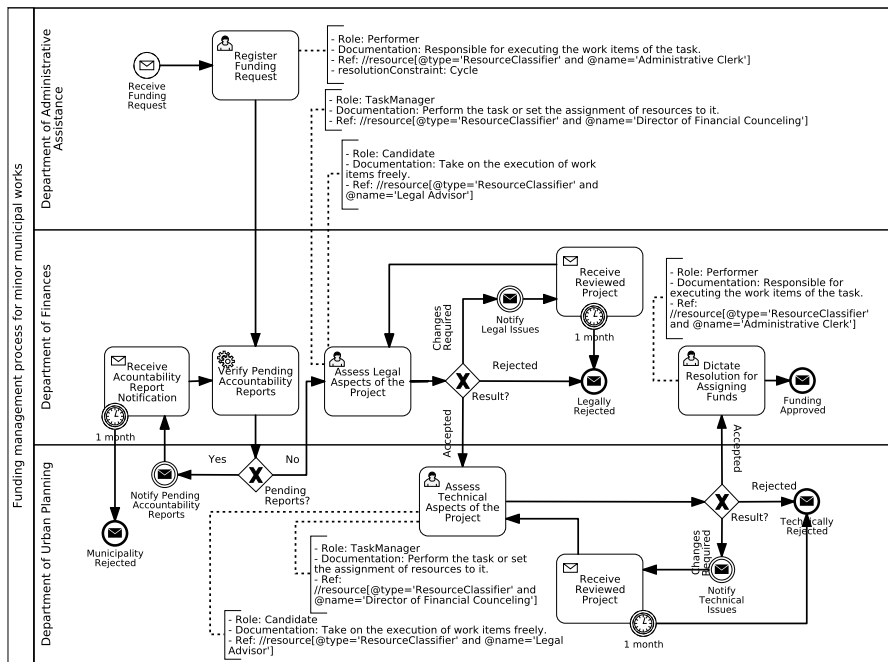


Figure 14: Funding PI-EPM.

	isResource	isClassifier	isReferenceable	isBindable	subsumes	isSubsumed	isSource	isTarget	hasPopulation	isMember	hasPrivileges	hasParameters	hasBindableParams
<b>Ministry</b>	0	1	0	0	1	0	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<b>Secretariship</b>	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<b>Department</b>	0	1	0	0	1	1	0	0	0	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<b>Director</b>	0	1	1	0	0	1	0	1	1	0	1	0	0
<i>Membership</i>	0	1	1	0	0	1	0	0	1	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	0	1	0	0	0	0	0	0	0	0
<b>Clerk / Advisor</b>	0	1	1	0	0	1	1	0	1	0	1	0	0
<i>Membership</i>	0	1	1	0	0	1	0	0	1	0	0	0	0
<i>Group</i>	0	1	1	0	1	1	0	0	0	0	0	0	0
<i>Role</i>	0	1	1	0	1	0	0	0	0	0	0	0	0
<b>Carl / Peter</b>	1	0	0	0	0	0	0	0	0	0	0	0	0
<i>User</i>	1	0	1	1	0	0	1	1	0	1	1	1	1
<i>Administrator</i>	1	0	1	1	0	0	1	1	0	1	1	1	1

Table 3: Bonita Candidate Resource Structure Implementation

	isAssignable	multiplePerformers	singlePerformer	hasConstraints	hasTriggers	hasEscalations	revokablePrivileges
<b>Candidate</b>	1	1	0	0	0	0	0
<i>Actor</i>	1	1	0	1	0	0	0
<b>TaskManager</b>	1	1	0	0	0	0	0
<i>Actor</i>	1	1	0	1	0	0	0
<b>Performer</b>	1	0	1	1	0	0	0
<i>AssignedActor</i>	1	0	1	1	0	0	0

Table 4: Bonita Candidate Work Distribution Implementation

tor resource classifiers. This means that the organizational entities provided by Bonita do not support the definition of resource relationships such as reporting lines at the resource classifier level. Finally, the user and administrator human resource implementations of Bonita provide support to the human resource elements defined in the PI-RSM. The resulting support ratio was therefore of 0.71.

With regard to the support provided by Bonita to the work distribution requirements defined in the PI-EPM, the Candidate and Task Manager resource roles are supported by the Actor role implementation. Also, the Performer role is supported by the Assigned Actor role implementation. Therefore the support ratio was of 1.

#### 6.1.4. Definition of the Resource Perspective Implementation

The implementation of the requirements defined in stage two was performed with base on the candidate implementation elements resulting of the previous stage and the meaning of the elements described in the documentation attribute of the elements defined in the PI-RSM, PI-EPM and RPIM input models.

The resulting PS-RSM is depicted in Figure 15. It was created by executing the *Initialize PS-RSM* and *Import RPIM* commands provided by the tool, and by referencing the respective implementation elements. The hierarchy of ministry, secretaryships and departments was implemented through

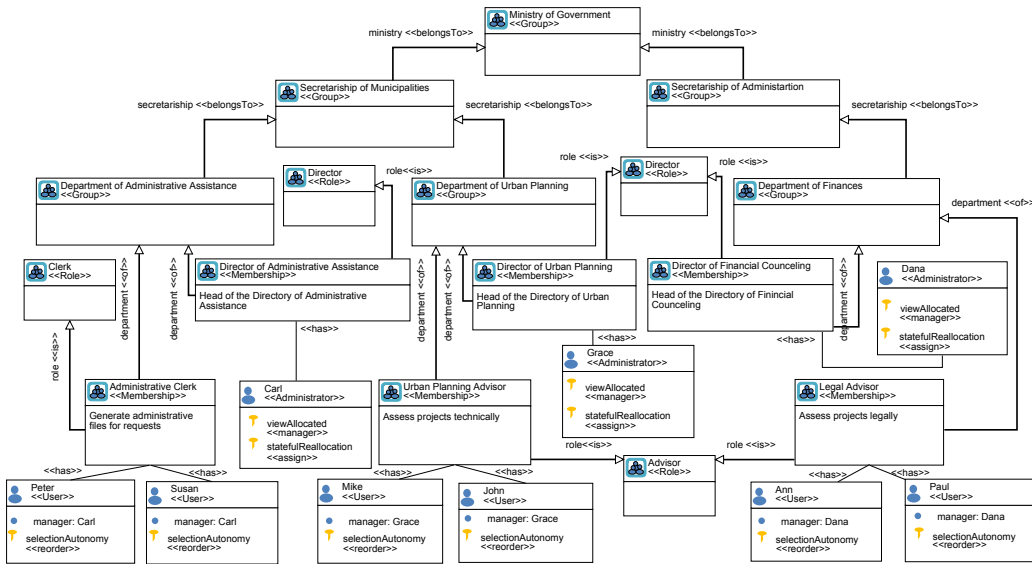


Figure 15: Funding PS-RSM.

the *Group* resource classifier implementation and *belongsTo* resource relationship implementation. The resource classifiers defining jobs were implemented by referencing the *Membership* resource classifier implementation, which is the only one defined by Bonita that may have an associated population. Additional resource classifiers implemented as *Role* needed to be added in order to fulfill the *is* subsumption relationship between *Role* and *Membership* that is mandatory. Human resource elements classified as Clerk, Urban Planning Advisor and Legal Advisor were implemented by using *User* Bonita elements. A resource parameter called manager was also added to these elements in order to implement the reporting lines to their directors, who were represented in the implementation as *Administrator* elements in order to allow them to change the distribution of work items assigned to them.

The resulting PS-EPM is shown in Figure 16. It was created by executing the *Initialize PS-EPM*, *Import PS-RSM* and *Import RPIM* commands provided by the tool (see Figure 9). The Performer resource role of the Register Funding request and Dictate Resolution for Assigning funds user tasks were implemented through the Assigned Actor resource role implementation defined in the RPIM of Figure 5. The resource assignments for the former task were defined by referencing the Administrative Clerk membership and

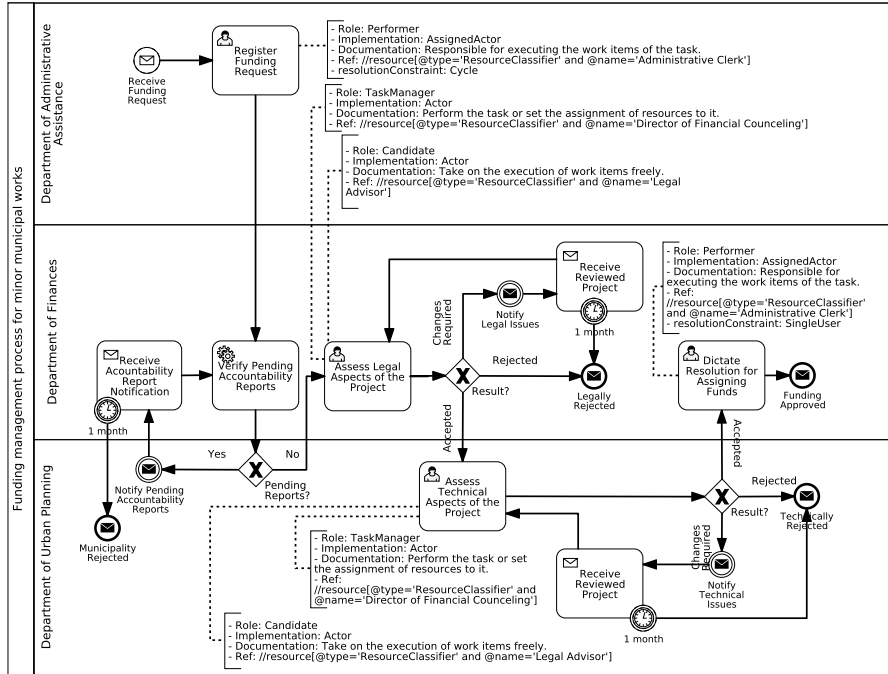


Figure 16: Funding PS-EPM.

by implementing the Cycle resolution constraint with a custom Cycle assigning actor filter defined in the Bonita RPIM. This resolution constraint implementation is not provided by default by Bonita. Instead, it was developed by using an API provided by the WfMS to define custom resolution constraints. The use of this kind of mechanism can also be considered when defining an RPIM. The resource assignments for the latter user task were defined by referencing the Director of Finances membership and by specifying the *singleUser* resolution constraint required by the AssignedActor resource role implementation.

The Candidate and Task Manager roles defined for the Assess Legal Aspects of the Project and Assess Technical Aspects of the Project were implemented through the Actor resource role implementation. The additional operations required by task managers are granted to individual human resources by defining them as administrators, since Bonita relies on a static authorization approach for this purpose.



#### 6.1.5. *Verification and Validation of the Resource Perspective Implementation*

The implementation of the resource perspective requirements in the previous stage required introducing variations to the platform-specific models from the platform-independent models, since Bonita did not provide straightforward support to all the requirements defined in the PI-RSM. The verification rules were applied in order to check that the PS-RSM and PS-EPM met the restrictions imposed by the Bonita RPIM. The obtained evaluation result was correct. The validation rules were also applied in order to check the consistency of the PS-RSM and PS-EPM against the requirements defined in the PI-RSM and PI-EPM. The validation warned about an inconsistency originated in the implementation of the *director* resource relationships by defining *manager* resource parameters. The automation of these rules by the tool allowed performing the checks during the definition of the platform-specific models, reducing the time required to be in conditions of generating the executable specifications.

#### 6.1.6. *Generate Executable Specifications*

The PS-RSM and PS-EPM of Figures 15 and 16 were transformed into three XML documents to be imported into the Bonita BPMS: a BPMN document, a Bonita organization schema, and an actor mapping document defining the resource assignments. This was accomplished through three model-to-code transformations developed for the Bonita RPIM, which were developed and incorporated into the proposed tool.

Figure 17 shows a screenshot of the generated resource schema imported into the Bonita organization administration and the resulting Bonita process specification for the funding assignment process with the work distribution strategy of the Register Funding Request user task.

#### 6.1.7. *Case Study Results*

Several benefits can be highlighted from the application of the proposed framework and method to this process. The defined platform-independent models enabled government authorities, business analysts and developers to communicate and agree on resource perspective requirements. The defined PI-RSM allowed characterizing and classifying resources with base on the government structure, in a way that also enabled the definition of resource assignments based on them. The definition of the work distribution in a PI-EPM allowed stakeholders to analyze and define its requirements in an early

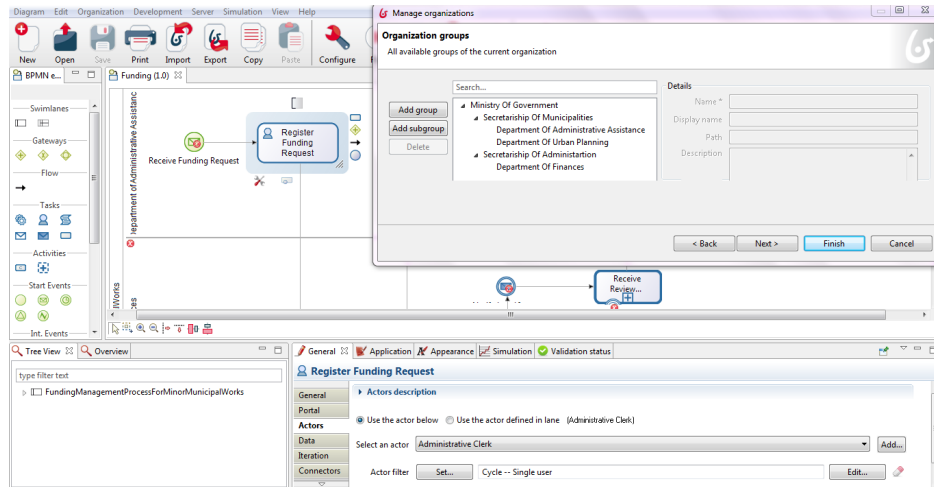


Figure 17: Funding Implementation.

stage of the development of the PAIS. Moreover, this enabled the definition of advanced work distribution policies that were not considered in previous versions of this process, such as push distribution strategies defining cyclic assignments or pull work distribution strategies involving multiple resource roles.

Several benefits for the developers can also be highlighted. The evaluation of the support provided by Bonita to the defined resource perspective requirements enabled them to deal with the limitations of this platform in an early stage. The verification and validation of the PS-RSM and PS-EPM allowed determining that the defined implementation based on Bonita fulfilled the requirements expressed in the platform-independent models. Finally, the provided model-to-code transformations contributed to reduce the time and errors in the definition of the artifacts that were deployed in Bonita to implement the PAIS that supports this process.

The application of the method and framework also presented some difficulties. The more important one consisted in the definition of the RPIM. Although the BPMN extensions were easily understood and used by stakeholders, the definition of the Bonita RPIM was complex for them. This requires to have a good comprehension of the platform and also of resource perspective aspects. Hence, the Bonita RPIM was defined by us and stored in a model repository with the aim to allow its reuse.

## 6.2. Workflow Resource Patterns Support

The expressiveness of the proposed framework was also evaluated in terms of their suitability to represent resource perspective requirements captured in the Workflow Resource Patterns (WRPs) [3].

The WRPs are organized in eight categories defined with base on the states of a reference life cycle of work items. Creation Patterns (WRP-1 to WRP-11), Push Patterns (WRP-12 to WRP-20), Escalation (WRP-28) and two of the Auto-Start Patterns (WRP-36 and WRP-37) describe recurrent requirements that may take place during the creation, assignment and advertising of work items to resources. Therefore, these patterns fall within the scope of the Work Distribution aspect. Pull Patterns (WRP-21 to WRP-26), user-triggered Detour Patterns (WRP-27 and WRP-29 to WRP-35), Visibility Patterns (WRP-41 and WRP-42) and Additional Resource Patterns (WRP-42 and WRP-43) describe different operations that may be performed by the resources in order to organize and affect the state of work items. Therefore, they fall within the scope of the Authorization aspect.

Table 5 summarizes the patterns that can be represented by each modeling element provided by the proposed framework to define the Work Distribution and Authorization aspects. Given that the framework is based on generic concepts, the provided modeling elements can be used to define multiple WRPs. The meaning of these elements can be defined by practitioners in order to represent the different WRPs in platform-independent models. In the case of a platform-specific model, the patterns that can be supported are constrained by the implementation elements provided by the selected WfMS. For example, the WRP-10 can be supported in a PS-EPM if the imported RPIM defines resource classifier implementations that allow representing organizational positions and reporting lines, which can be employed in the definition of resource parameter bindings.

In addition, the proposed framework supports but is not constrained to the representation of the WRPs. Further requirements which are not considered by the WRPs can be represented by using the modeling elements provided. For example, it allows defining additional resolution constraints to support a new resource assignment approach, or additional resource privileges or task privileges in order to allow granting new operations for resources to see and progress the work items assigned to them.

Finally, four of the WRPs are not covered by the proposed framework. These patterns are WRP-06 Case Handling, WRP-18 Early Distribution,

WRP-34 Redo, WRP-35 Pre-Do given that they contradict restrictions specified by BPMN. Basically that a work item cannot be distributed or completed before its creation upon the instantiation of the respective user task (for WRP-06, WRP-18 and WRP-35) and that a task instance cannot be reopened after its finalization (for WRP-34).

## 7. Related Work

There exist previous work on the evaluation of WfMSs regarding the support provided to the resource perspective. The approach proposed in [34] enables the assessment of different WfMSs with base on metamodels providing elements to support the resource structure aspect. The artifacts proposed in this work, specifically the RPIMet, enable the comparison of different WfMSs by using this approach. In addition, our proposal also allows taking into account the specific process requirements during the selection of an implementation platform in the development of PAISs. Another evaluation approach [14] proposes a reference model for WfMSs based on Colored Petri Nets. Different WfMSs were represented as upgrades of this reference model and compared with base on their differences to it. However, its main purpose is not to serve as an aid in the selection of a WfMS but to reach a better understanding of how work distribution mechanisms are implemented in WfMSs.

There also exist proposals of extensions to BPMN to represent the resource perspective. An extension of BPMN 1 to represent resource assignment constraints by using OCL was proposed in [17]. According to the authors, it supports nine resource patterns. A similar work [35] presents an extended resource layer to BPMN via the Business Process Definition Metamodel (BPDM) providing support to thirteen resource patterns. Both proposals are centered on the work distribution aspect. They partially cover the authorization aspect and do not provide notations for the definition of resource structure models. As it was discussed in section 6.2, the BPMN extensions proposed in this paper support 39 of 43 Workflow Resource Patterns.

An extension to BPMN 1 was also proposed in [36] for the definition of task-based authorization constraints. It allows defining the assignment of tasks to roles representing groups of people and the formal specification of requirements such as binding of duties, separation of duties or history based allocation, which are visualized in BPMN artifacts. This approach can be

	<b>Framework Elements</b>	<b>Covered Patterns</b>
<b>Work Distribution Aspect</b>	Resource Parameter Bindings	WRP-01 Direct Allocation, WRP-02 Role-Based Allocation, WRP-08 Capability Based Distribution, WRP-10 Organizational Distribution
	Resource Assignment Expression	WRP-03 Deferred Allocation, WRP-09 History Based Distribution
	Resolution Constraint	WRP-05 Separation of Duties, WRP-07 Retain Familiar, WRP-09 History Based Distribution, WRP-15 Random Allocation, WRP-16 Round Robin Allocation WRP-17 Shortest Queue
	WorkItem Event	WRP-19 Distribution on Enablement, WRP-20 Late Distribution, WRP-28 Escalation
	Automatic Task	WRP-11 Automatic Execution
	Resource Role	WRP-12 Distribution by Offer (Single Resource), WRP-13 Distribution by Offer (Multiple Resources), WRP-14 Distribution by Allocation (Single Resource), WRP-36 Commencement on Creation, WRP-37 Commencement on Allocation
<b>Authorization Aspect</b>	Resource Privileges or Task Privileges	WRP-04 Authorization, WRP-21 Resource Initiated Allocation, WRP-22 Resource Initiated Execution (Allocated Work Item), WRP-23 Resource Initiated Execution (Offered Work Item), WRP-24 System Determined Work Queue, WRP-25 Resource Determined Work Queue, WRP-26 Selection Autonomy, WRP-27 Delegation, WRP-29 Deallocation, WRP-30 Statefull Reallocation, WRP-31 Stateless Reallocation, WRP-32 Suspension-Resumption, WRP-33 Skip, WRP-43 Additional Resources
	Resource Privileges	WRP-38 Piled Execution, WRP-39 Chained Execution, WRP-40 Configurable Unallocated Work Item Visibility, WRP-41 Configurable Allocated Work Item Visibility WRP-42 Simultaneous Execution

Table 5: Supported Workflow Resource Patterns

leveraged in our framework by representing its roles as *resource classifiers* and by changing the expression language of *resolution constraints* by the proposed formalism. Also, our approach is broader in the sense that enables grouping resources by using different concepts.

An extension to BPMN 1 was also proposed in [36] for the definition of task-based authorization constraints. It allows defining the assignment of tasks to roles representing groups of people and the formal specification of requirements such as binding of duties, separation of duties or history based allocation, which are visualized in BPMN artifacts. Our approach is broader in the sense that enables grouping resources by using different concepts. The approach defined in [36] can be leveraged in our framework by representing its roles as resource classifiers and by changing the expression language of resolution constraints in order to use the proposed formalism.

A domain-specific language called RAL for the definition of resource assignments embedded in BPMN resource assignment expressions with base on the metamodel adopted in the definition of the Workflow Resource Patterns was proposed in [37] and formalized in an ontology called OWL-DL in [38]. It allows defining resource assignments on a high level of abstraction and to analyze them using descriptive logic reasoners. However, its use is restricted to platforms providing support to the organizational metamodel it is based on. RAL can be used in the context of our framework as an expression language to define resource assignments in platform-specific models implementing resource elements with base the elements defined by its adopted organizational metamodel.

There also exist model-driven approaches to define and implement resource perspective requirements of business processes. A work based on an MDA method for generating WS-HumanTask task definitions was proposed in [39]. It addresses the definition of the resource perspective of business processes by means of a UML Profile called Human Task Profile, which is based on WS-HumanTask. Hence, the method and UML profile are constrained to the resource perspective requirements supported by this platform. In contrast, the framework and method proposed in this paper allow representing the support of different WfMS platforms and implement the processes by using the most suitable platform according to the resource perspective requirement defined in them.

In a previous work [26], we defined a model-driven approach for generating BPEL4People/WS-HumanTask specifications from BPMN process models extended to support the resource perspective requirements described by the

Workflow Resource Patterns. The BPMN extensions defined in that work were redefined and completed in the framework proposed in this paper in order to also define platform-specific details in separate models. This allows defining resource perspective requirements supported by different platforms and also requirements that are not considered by the Workflow Resource Patterns.

Another work [40] proposed a model-driven approach for defining human aspects of business processes by representing process perspectives as views of a core metamodel and implementation platforms as extensions of the defined views. Taking BPMN as the core metamodel, our approach also provides a model-driven method addressing these aspects reusing the well known conceptual framework, notation and tools developed for this language.

Other related proposals exist which are particularly oriented to support the authorization aspect of the resource perspective. An approach for modeling and enforcing access control requirements called SecureBPMN [41] provides a means for the definition of role based access control, separation of duties, binding of duties by introducing them as first class citizens in the BPMN metamodel. It also proposes a model-driven approach for generating executable specifications of these requirements. The extensions defined in our framework provide a means to define such requirements by keeping the interchangeability of the resulting models as they were defined by using the extension mechanism provided by the language.

Another work [42] provides a means for modeling security requirements extensively by extending BPMN 1. Our framework allows defining security requirements associated with the involvement of human resources in the execution of business processes. However, more security-specific requirements such as Attack Harm Detection are out of the scope of our work.

An alignment between the concepts of the domain of information systems security risk management with the BPMN elements was proposed in [43]. It uses BPMN to represent the behavior of information systems and represents intended and non-intended users by using pools called User and Violator. Our framework uses BPMN to represent the behavior of business processes including automatic and user tasks. In this context, intended and non-intended users should be represented through Resource Role elements called user and violator.

The authorization aspect is also addressed in [44]. It proposes an extension to the WfMS architecture with a separate component for the management of security policies. It supports the definition of security requirements

based on the concepts of responsibility, permission and role. Compared to our framework, the concept of role matches the concept of *resource classifier*, the concept of permission bundle matches the concept of *resource role* and the concept of responsibility matches the concept of *user task*.

Finally, in comparison with the work discussed in this section, the framework and method proposed in this work is broader as it provides a single approach to define the resource perspective aspects that can be used to integrate other proposals addressing specific aspects.

## 8. Conclusions

### 8.1. Contributions

This work proposed a framework and a model-driven development method for defining the resource perspective of business processes in the development of PAISs based on WfMSs. The aim is to allow practitioners to make decisions on resource perspective requirements at both business and technological levels. The framework extends the capabilities of BPMN to define the Resource Structure, Work Distribution and Authorization aspects of this perspective in process models. The method, which makes use of the framework, guides the development of this perspective from its definition in conceptual models to the automatic generation of executable workflow specifications.

The framework provides the Resource Perspective Implementation Meta-model, which allows defining models referred to as RPIMs that represent the entities provided by existing or future WfMSs to implement this perspective. In addition, two BPMN extensions were developed to support the definition of resource perspective requirements in both platform-independent and platform-specific process models.

The Resource Structure extension supports the definition of the Resource Structure and Static Authorization aspects in Resource Structure Models (RSMs). In order to overcome the lack of a standard specification for the definition of these aspects to be used in conjunction with BPMN, this extension provides an approach for their representation by reusing Resource and Resource Parameter BPMN elements. This makes it easy to import RSMs into BPMN process models in order to define resource assignments. This extension allows representing a broad range of organizational structures in terms of two generic concepts: Human Resource and Resource Classifiers. The elements representing these concepts can be mapped into platform-specific elements by referencing their implementation in an imported RPIM. Given



that the definition of these aspects is not in the scope of BPMN, a notation was also proposed in order to visualize RSMs.

The Work Distribution extension allows defining the Work Distribution and Dynamic Authorization aspects in Extended Process Models (EPMs). It provides a means to describe the resource roles performed by human resources on user tasks by using the Human Performer BPMN element. This extension allows representing different work distribution strategies without modifying the BPMN metamodel. It also provides attributes that allow defining dynamic authorizations, resolution constraints, triggers and escalations in BPMN process models. No additional notation artifacts were defined for this extension. The extension attributes are depicted by using BPMN text annotations associated with user tasks.

This work also presented a model-driven development method for defining this perspective by leveraging the proposed framework. Starting from a typical BPMN process model defining the control flow and data perspectives (BPLM), it provides a systematic approach for: (1) the definition of resource perspective requirements in a Platform-Independent Resource Structure Model (PI-RSM) and a Platform Independent Extended Process Model (PI-EPM); (2) the evaluation and selection of a WfMS or implementation platform represented by an RPIM; (3) the implementation of the requirements in a Platform-Specific Resource Structure Model (PS-RSM) and a Platform-Specific EPM (PS-EPM) based on the selected platform; (4) the verification and validation of the resource perspective requirements defined in the platform-specific models; and (5) the automatic generation of executable workflow specifications implementing the resource perspective in the selected WfMS, which is supported by model-to-code transformations defined for each platform. The verification and validation allows checking the consistency of the resulting PS-RSM and PS-EPM with the selected RPIM and with the requirements defined in the PI-RSM and the PI-EPM. This is an important aid when there is not a direct or full support of the selected WfMS to the requirements defined in a PI-RSM and a PI-EPM.

In brief, the proposed framework allows representing a broad range of resource perspective requirements and the proposed method allows addressing the stages required to implement this perspective in the development of PAISs.

A tool that supports the framework and stages of the method was developed with base on Oryx model editor and repository. Its purpose was to validate the feasibility of implementing our approach in existing BPMN

tools, as well as to have a tool that allows its application of real cases.

The evaluation of the framework and method was performed through a case study and the Workow Resource Patterns. The case study showed the applicability and utility of the framework and method to a process of a provincial government for the assignment of funds to its municipalities and local governments. Benefits and limitations were highlighted. As final result, the method allowed defining and implementing detailed resource perspective requirements and reduced the effort required for the development of the PAIS. In addition, the expressiveness of the proposed BPMN extensions was validated in terms of their suitability to represent resource perspective requirements captured in the Workflow Resource Patterns (WRPs). As result, most of the WRPs are supported by these extensions. However, these extensions are not limited to support WRPs. They provide generic elements that allow expressing other requirements that are not considered by the WRPs and are present in other reference models or WfMSs.

### *8.2. Limitations and Future Work*

Although the framework and method support defining a broad range of resource perspective requirements in conceptual models, their implementation can be restricted by the features provided by the existing WfMSs. To deal with this issue, an ongoing work is the development of extensible components for managing this perspective in the execution of processes. These components are intended to be integrated into different types of WfMS to replace their original components.

Another limitation of the framework and method is the lack of mechanisms to verify the correctness of the resource perspective requirements defined in platform-independent and platform-specific process models. Desired or expected properties of this perspective should be identified and checked, as it is verified soundness for the control flow perspective. In addition, analysis of the performance implications of the resource perspective requirements could be appropriate to validate them. Proposals to deal with these issues are considered as future work.

## **References**

- [1] M. Dumas, W. M. Van der Aalst, A. H. Ter Hofstede, Process-aware information systems: bridging people and software through process technology, Wiley-Interscience, 2005.

- [2] K. M. van Hee, et al., Workflow management: models, methods, and systems, The MIT press, 2004.
- [3] N. Russell, W. M. van der Aalst, A. H. ter Hofstede, D. Edmond, Workflow resource patterns: Identification, representation and tool support, in: Advanced Information Systems Engineering, Springer, 2005.
- [4] P. Wohed, N. Russell, A. H. Ter Hofstede, B. Andersson, W. M. van der Aalst, Patterns-based evaluation of open source bpm systems: The cases of jbpms, openwfe, and enhydra shark, Information and Software Technology 51 (2009).
- [5] N. Russell, W. M. van der Aalst, Evaluation of the bpel4people and ws-humantask extensions to ws-bpel 2.0 using the workflow resource patterns, Bpm center report, Department of Technology Management, Eindhoven University of Technology GPO Box 513 (2007).
- [6] P. Wohed, W. M. van der Aalst, M. Dumas, A. H. ter Hofstede, N. Russell, On the suitability of bpmn for business process modelling, in: Business Process Management, Springer, 2006.
- [7] N. Russell, W. M. van der Aalst, A. H. Ter Hofstede, P. Wohed, On the suitability of uml 2.0 activity diagrams for business process modelling, in: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling-Volume 53, Australian Computer Society, Inc., 2006.
- [8] M. Weske, Business process management: concepts, languages, architectures, Springer, 2012.
- [9] M. Pesic, W. M. van der Aalst, Towards a reference model for work distribution in workflow management systems, Business Process Reference Models (2005) 30.
- [10] A. Kumar, W. M. Van Der Aalst, E. M. Verbeek, Dynamic work distribution in workflow management systems: How to balance quality and performance, Journal of Management Information Systems 18 (2002).
- [11] OMG, Business Process Model and Notation (BPMN), V.2.0, <http://www.omg.org/spec/BPMN/2.0>, 2011.

- [12] L. J. R. Stroppi, O. Chiotti, P. D. Villarreal, A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models, in: Proceedings of the XIV Iberoamerican Conference on Software Engineering, 2011.
- [13] D. Hollingsworth, U. Hampshire, Workflow management coalition the workflow reference model, Workflow Management Coalition (1993).
- [14] M. Pesic, W. M. van der Aalst, Modelling work distribution mechanisms using colored petri nets, International Journal on Software Tools for Technology Transfer 9 (2007).
- [15] A. Agrawal, M. Amend, M. Das, M. Ford, M. Keller, C. and Kloppmann, D. Knig, F. Leymann, R. Mller, G. Pfau, K. Plsser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, M. Zeller, Web Services HumanTask (WS-HumanTask), Version 1.1, 2010.
- [16] M. Zur Muehlen, Resource modeling in workflow applications, in: Proceedings of the 1999 Workflow Management Conference, volume 70, 1999.
- [17] A. Awad, A. Grosskopf, A. Meyer, M. Weske, Enabling resource assignment constraints in BPMN, Technical Report, Technical report, Business Process Technology-Hasso Plattner Institute, 2009.
- [18] E. Bertino, E. Ferrari, V. Atluri, The specification and enforcement of authorization constraints in workflow management systems, ACM Transactions on Information and System Security (TISSEC) 2 (1999).
- [19] W. Hummer, P. Gaubatz, M. Strembeck, U. Zdun, S. Dustdar, Enforcement of entailment constraints in distributed service-based business processes, Information and Software Technology 55 (2013) 1884 – 1903.
- [20] C. Stefansen, S. Rajamani, P. Seshan, Softalloc: A work allocation language with soft constraints, in: Web Services, 2008. ICWS '08. IEEE International Conference on, 2008, pp. 441–448. doi:10.1109/ICWS.2008.41.
- [21] W. M. van der Aalst, M. Rosemann, M. Dumas, Deadline-based escalation in process-aware information systems, Decision Support Systems 43 (2007). Emerging Issues in Collaborative Commerce.

- [22] S. Schefer-Wenzl, M. Strembeck, A. Baumgrass, An approach for consistent delegation in process-aware information systems, in: *Business Information Systems*, Springer, 2012, pp. 60–71.
- [23] M. Leitner, S. Rinderle-Ma, A systematic review on security in process-aware information systems constitution, challenges, and future directions, *Information and Software Technology* 56 (2014) 273 – 293.
- [24] C.-x. Zhang, Y.-x. Hu, G. Zahng, Task-role based dual system access control model, *International Journal of Computer Science an Network Scurity* 6 (2006) 211–215.
- [25] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, *Journal of management information systems* 24 (2007) 45–77.
- [26] L. J. R. Stroppi, O. Chiotti, P. D. Villarreal, Extended resource perspective support for bpmn and bpel., in: *Proceedings of the XV Iberoamerican Conference on Software Engineering*, 2012, pp. 56–69.
- [27] L. J. R. Stroppi, O. Chiotti, P. D. Villarreal, Extending BPMN 2.0: Method and Tool Support, *Lecture Notes in Business Information Processing* 95 (2011).
- [28] O. MDA, *Mda guide. version 1.0. 1*, 2003.
- [29] B. Selic, The pragmatics of model-driven development, *IEEE software* 20 (2003) 19–25.
- [30] B. R. Silver, *BPMN Method and Style: With BPMN Implementer’s Guide*, Cody-Cassidy Press Aptos, 2011.
- [31] R. M. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in bpmn, *Information and Software Technology* 50 (2008) 1281–1294.
- [32] M. Zur Muehlen, *Organizational Management in Workflow Applications Issues and Perspectives*, *Information Technology and Management* 5 (2004).

- [33] G. Decker, H. Overdick, M. Weske, Oryx—an open modeling platform for the bpm community, in: Business process management, Springer, 2008.
- [34] M. Rosemann, M. Zur Muehlen, Evaluation of workflow management systems—a meta model approach, Australian Journal of Information Systems 6 (1998).
- [35] A. Großkopf, An extended resource information layer for bpmn, Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam (2007).
- [36] C. Wolter, A. Schaad, Modeling of task-based authorization constraints in bpmn, in: Business Process Management, Springer, 2007, pp. 64–79.
- [37] C. Cabanillas, M. Resinas, A. Ruiz-Cortés, Defining and analysing resource assignments in business processes with ral, in: Service-Oriented Computing, Springer, 2011, pp. 477–486.
- [38] C. Cabanillas, M. Resinas, A. Ruiz-Cortés, Ral: A high-level user-oriented resource assignment language for business processes, in: Business Process Management Workshops, Springer, 2012, pp. 50–61.
- [39] S. Link, P. Hoyer, T. Schuster, S. Abeck, Model-driven development of human tasks for workflows, in: Software Engineering Advances, 2008. ICSEA’08. The Third International Conference on, IEEE, 2008.
- [40] H. Tran, U. Zdun, S. Dustdar, et al., Modeling human aspects of business processes—a view-based, model-driven approach, in: Model Driven Architecture—Foundations and Applications, Springer, 2008.
- [41] A. D. Brucker, I. Hang, G. Lückemeyer, R. Ruparel, Securebpmn: Modeling and enforcing access control requirements in business processes, in: Proceedings of the 17th ACM symposium on Access Control Models and Technologies, ACM, 2012, pp. 123–126.
- [42] A. Rodríguez, E. Fernández-Medina, M. Piattini, A bpmn extension for the modeling of security requirements in business processes, IEICE transactions on information and systems 90 (2007) 745–752.

- [43] O. Altuhhova, R. Matulevičius, N. Ahmed, Towards definition of secure business processes, in: *Advanced Information Systems Engineering Workshops*, Springer, 2012, pp. 1–15.
- [44] M. Leitner, J. Mangler, S. Rinderle-Ma, Sprint-responsibilities: design and development of security policies in process-aware information systems, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 2 (2011) 4–26.