Towards the Definition and Analysis of Resource Assignments in BPMN 2.0

Introducing RAL

Cristina Cabanillas, Manuel Resinas and Antonio Ruiz-Cortés
{cristinacabanillas, resinas, aruiz}@us.es

Applied Software Engineering Research Group
University of Seville, Spain
April 2011
This report was prepared by the

Applied Software Engineering Research Group (ISA)
Department of Computer Languages and Systems
Av/ Reina Mercedes S/N, 41012 Seville, Spain
http://www.isa.us.es/

Copyright ©2011 by ISA Research Group.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and ‘No Warranty’ statements are included with all reproductions and derivative works.

NO WARRANTY
THIS ISA RESEARCH GROUP MATERIAL IS FURNISHED ON AN ‘AS-IS’ BASIS. ISA RESEARCH GROUP MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder

Support: This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian Local Government.
Towards the Definition and Analysis of Resource Assignments in BPMN 2.0(*)(**)  

Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés  
Universidad de Sevilla, Spain  
{cristinacabanillas, resinas, aruiz}@us.es

Abstract. Resources and, specially, human resources are an important element in business process execution and should be considered also at design time (when modelling the processes). However, the resource-related support most business process modelling languages provide is quite vague, and the connection between business process models and organizational models is limited. In this paper we walk in this direction by defining the textual language RAL (Resource Assignment Language) to assign resources to the activities of a business process model, specifically as an extension of BPMN 2.0. We prove the expressiveness of this language using well-known Workflow Resource Patterns. Furthermore, we have developed an ontology-based approach to analyse resource assignments to obtain information useful for the organization that implements the process by means of description logic. Finally, the concept of consistency between a business process model and an organizational model is defined and applied.

Keywords: human-centric BPM, resource assignment, workflow resource pattern, organization model, BPMN

1 Introduction

Business processes are often analysed in terms of control flow, temporal constraints, data and resources. From all of these aspects, resources have received much less attention than other aspects, specially, control flow. For instance, even the standard business process modelling notation (BPMN) does not provide concrete definitions for the resource-related concepts, stating that “resources referenced by activities can be human resources as well as any other resource assigned to the activities during process execution time”[1]. However, the participation of people in business processes is of utmost importance, both to supervise the execution of automatic activities and to carry out software-aided and/or manual

* This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian Local Government.

** A similar version of this paper was submitted to the 9th International Conference on Business Process Management (BPM’11).
activities, and so they should be considered when designing and modelling the business processes used in an organization.

In this paper we face human-resource\(^1\) management. Specifically, we deal with resource assignment to the activities of a business process with BPMN 2.0, aiming at improving the business process modelling standard by easing the way resources of an organization can be associated with the activities of a process. This helps align the business processes of an organization with its organizational structure. Furthermore, having resource assignments included in a business process model constitutes a perfect scenario to automate work in different directions, e.g. to automatically infer information such as: (i) the potential performers of each activity of a business process, i.e., set of people that fulfills the resource-related constraints imposed in the model, or (ii) the potential set of activities each person of the organization can be assigned at runtime. This kind of information can be beneficial for the organization in several ways. For instance, in the previous case: the former benefits the person in charge of resource allocation, and the latter provides an employee-oriented vision, informing about the possible workload of each employee.

The main contribution of this paper is twofold. On the one hand, we have developed RAL (Resource Assignment Language), a textual language to express resource assignments in the activities of a business process in BPMN and based on the organizational metamodel proposed by Russell et al. [2]. The expressiveness of RAL is shown using the creation patterns of the Workflow Resource Patterns [2], which are patterns focused on expressing resource assignment constraints. On the other hand, we provide a semantic mapping of RAL into description logic (DL). This allows us to use DL reasoners to infer information such as that mentioned above (cf. Section 5 for more details). Furthermore, we define the notion of consistency between the resource assignment of a business process model and an organisational model and a mechanism to check it.

The structure of this paper is as follows: in Section 2 we describe how we improve the resource management capabilities of BPMN 2.0 with RAL; Section 3 shows the expressiveness of RAL by applying it to some workflow resource patterns; Section 4 describes the semantic mapping between RAL and DL; Section 5 shows how DL reasoners can be used to infer information from RAL and define the notion of consistency. Finally, some related work can be found in Section 6, and a set of conclusions and possible future work are discussed in Section 7.

2 Improving Resource Management Capabilities of BPMN 2.0 with RAL

BPMN is the de-facto standard for business process modelling. It has been notably improved in its current version (2.0) to consider the relation between the activities of a business process and any kind of resources. However, the definition of resource BPMN provides and the use of this term are still very imprecise. On

\(^1\) From now on we will use the term resource to refer to human resources.
the one hand, BPMN states that “resource involves any kind of resource linked to the activities of a business process”. On the other hand, it usually uses the term to refer to human resources. Furthermore, BPMN 2.0 does not specify the way resource assignments must be done in the model. The freedom BPMN allows may cause problems when we want to insert constraints on the allocation of human resources, e.g. to state that two specific activities cannot be performed by the same role.

2.1 Aligning BPMN 2.0 and Organizational Structures

BPMN 2.0 allows assigning resources to the activities of a business process, but it does not specify exactly how to do it. Furthermore, it does not provide sufficient information about the types of resource assigned to a specific activity, i.e., it does not define the set of all possible cases to be covered. Figure 1 shows how we have aligned the BPMN 2.0 metamodel [1] with an organizational metamodel, in order to include organizational features in business processes modelled in BPMN. We have used the organizational metamodel described by Russell et al. [2]. As a result, a hierarchy of classes with the types of resources that can be assigned to activities, together with the relations between them, is now part of the BPMN 2.0 metamodel.

The organizational metamodel basically consists of persons, positions, roles and organizational units. As depicted in the figure, a resource can be an individual resource, i.e., a person, or a set of people involved in a specific group (role, position, organizational unit). Individual resources have a set of capabilities,
such as their professional experience. The metamodel is extensible to include new capabilities. Each person occupies one or more positions within an organization, which in turn can participate in several roles and which belong to an organizational unit. We will base all the resource-related explanations given in this paper on the terminology used in the metamodel.

Figure 2 depicts a possible instantiation of the organizational metamodel. Specifically, it contains an excerpt of the ISA Research Group of the University of Seville from a research project perspective. Each position of the model can delegate work to any inferior position, and can report work to its immediately upper position. The position-role associations are shown in the table in the same figure. For instance, individual Beatriz Bernárdez belongs to the positions Responsible for Work Package and Account delegate. As a responsible for work package she has three roles: Responsible, Researcher and Doctoral Thesis Advisor. On the other hand, her other position gives her the role Account Administrator. Both positions may in turn be linked to the Project THEOS, which is an organizational unit.

2.2 RAL: Resource Assignment Language

The resource associated to an activity can be the result of an assignment expression. The class ResourceAssignmentExpression of the BPMN metamodel allows
configuring resource association by means of expressions, but BPMN does not provide any definition on the kind of expressions and gives the modeller complete freedom to define them: “The Expression class is used to specify an expression using natural-language text”[1].

Note that we assign resources to activities and we do not consider the assignment of resources to pools and lanes in a business process model because “pools and lanes constitute just a way to organize and categorize activities in a business process model”[2]. However, their resource-related meaning is up to the modeller”[1]. Therefore, for now we will limit resource allocation to activities.

We have developed RAL (Resource Assignment Language), a language for resource assignment, to extend the BPMN metamodel shown in Figure 1 to include resource assignment expressions in the activities of a business process model. This language, whose EBNF notation is shown in Language 1, has been developed with the aim of easing the statement of the following expressions:

**Individual resource assignments.** RAL allows expressing that an activity must be performed by a concrete person, by the same person assigned to another activity (direct allocation), or by the same person that executed certain activity or that appears in certain data field. In the last two cases, although the assignment is configured at design-time, the potential performer(s) can not be calculated until runtime.

**Assignments based on a GroupResource type.** It allows assigning an activity to a specific position, role or organization.

**Assignments based on the capabilities of individual resources.** Constraints based on the capabilities of individual resources, such as *years of experience* or *reputation* can be introduced. These constraints may consist of the existence of certain capability or of the holding of certain condition on the value of a capability. For space reasons we are not detailing the *CapabilityRestriction*, since it is based on mathematical and logical operators and its use is easily understandable.

**Equivalent assignments.** We define the expression *IS THE SAME AS* to indicate that an activity has assigned the same GroupResource of another activity or that included in a data field.

**Assignments based on relations between individual resources.** According to the concepts described by Russell et al. [2], positions can report work to positions, and positions can delegate work to positions. Using these relations we can construct contraints for resource assignment, such as “activity *Fill Travel Authorisation* must be performed by someone that reports to the *Project Coordinator*”.

**Assignments based on common group resources.** The expression *SHARE SOME ... WITH* is used to assign persons that share: (i) the position of the person who did another activity; (ii) at least one position with the set of positions that are shared by all the people who can do a given activity; or (iii) idem but depending on the content of a data (i.e., at runtime).

---

2 One pool contains exactly one process. One lane contains part of a process.
3 We can also consider issues such as *age* or *origin* as capabilities.
**Language 1 Expression assignment EBNF language definition**

Expression := IS PersonConstraint
  | HAS GroupResourceType groupResourceName
  | HAS CapabilityConstraint
  | IS THE SAME AS GroupResourceConstraint
  | RelationExpression
  | SHARE SOME GroupResourceType WITH ResourceConstraint
  | CompoundExpression

RelationExpression := ReportExpression
  | DelegateExpression

ReportExpression := REPORTS TO PositionConstraint Depth
  | IS Depth REPORTED BY PositionConstraint

DelegateExpression := CAN DELEGATE WORK TO PositionConstraint
  | CAN HAVE WORK DELEGATED BY PositionConstraint

CompoundExpression := NOT (Expression)
  | (Expression) OR (Expression)
  | (Expression) AND (Expression)
  | (Expression) AND IF POSSIBLE (Expression)

ResourceConstraint := PersonConstraint
  | GroupResourceConstraint

PersonConstraint := personName
  | PERSON IN DATA FIELD dataObject.fieldName
  | PERSON WHO DID ACTIVITY activityName

GroupResourceConstraint := RESOURCE IN ACTIVITY activityName
  | RESOURCE IN DATA FIELD dataObject.fieldName

CapabilityConstraint := capabilityName
  | CapabilityRestriction

PositionConstraint := POSITION name,position
  | POSITION OF PersonConstraint

GroupResourceType := POSITION
  | ROLE
  | UNIT

Depth := DIRECTLY
  | λ
Compound assignments. Combination and negation of the aforementioned expressions are allowed. Furthermore, the conditional expression \textit{AND IF POSSIBLE} has been included to let the modeller express preferences. In case preferences are not fulfilled, they are just ignored.

By analysing these expressions against an organizational model we can obtain useful information such as the potential performers of an activity of a business process, i.e., the set of individual resources that might be offered the activity during execution.

It is important to notice that we limit to expressions involving a single instance of a business process. The history of individual resources and past executions are not considered for now. Some specific use examples of the language are described in Section 3 with the help of a use case.

3 Application of RAL in a Business Process Model

The BPMN model shown in Figure 3 represents a process for Conference Travel Management, from the moment an author sends the Camera Ready version of his/her accepted contribution until all the bookings have been performed. We are presenting a simplified version of the procedure according to the rules of the University of Seville (Spain). The process begins when one of the authors of the paper fills up a form requesting for authorisation both to travel to the venue place and to take the fundings from some funding source. This authorisation must thus be approved by some person in charge of account management related to the author. Once the travel authorisation has been sent for revision, the travel management itself can start. The attendant must register at the conference and make the appropriate reservations. This may also be done by some person in charge of bureaucratic management (e.g., a clerk).

Given this use case, and considering the organizational model in Figure 2, we are going to show examples of resource assignments to the activities of the business process in Figure 3 using RAL language. We are using as example some \textit{Workflow Resource Patterns} introduced by Russell et al. \cite{2}. As our interest is
Towards the Definition and Analysis of Resource Assignments in BPMN 2.0

focused on expressing resource assignment constraints in a BPMN model (at design time), in principle we are only interested in the creation patterns.

**Direct Allocation:** The ability to specify at design time the identity of the resource that will execute a task. For instance: the *Sign Travel Authorisation* task must only be undertaken by *Antonio*:

Sign Travel Authorisation: IS Antonio

**Role-Based Allocation:** The ability to specify at design time that a task can only be executed by resources which correspond to a given role. For instance, instances of the *Fill Travel Authorisation* task must be executed by a *Research Assistant*:

Fill Travel Authorisation: HAS ROLE ResearchAssistant

**Deferred Allocation:** The ability to defer specifying the identity of the resource that will execute a task until runtime. For instance, during execution of the process, instances of the *Send Travel Authorisation* task will be executed by the person named in the resource field *Applicant* of data object *Authorisation*:

Send Travel Authorisation:
IS PERSON IN DATA FIELD Authorisation.Applicant

**Authorisation:** The ability to specify the range of resources that are authorised to execute a task. For instance, only a Researcher and a Research Assistant are authorised to execute instances of the *Submit Paper* task:

Submit Paper:
(HAS ROLE Researcher) OR (HAS ROLE ResearchAssistant)

**Separation of Duties:** The ability to specify that two tasks must be allocated to different resources in a given workflow case. For instance, instances of the *Sign Travel Authorisation* task must be allocated to a different person to that who executed the *Fill Travel Authorisation* task:

Fill Travel Authorisation:
NOT (IS PERSON WHO DID ACTIVITY SignTravelAuthorisation)

Sign Travel Authorisation:
NOT (IS PERSON WHO DID ACTIVITY FillTravelAuthorisation)

In this case, we assume at design time we do not know the real execution order of the activities and, thus, we set the contraint in both resource assignments.

**Case Handling:** The ability to allocate the activities within a given workflow case to the same resource. For instance, all tasks assigned to position *PhD-Student* are allocated to the same person.

Assigned to some activities: (HAS POSITION PhDStudent) AND (IS PERSON WHO DID ACTIVITY FillTravelAuthorisation).
The second part of the composition is not necessary for the first task that has been assigned the position PhDStudent. Please, note that the example exposed is this case is fictitious and will not be considered later in this paper.

**Retain Familiar**: Where several resources are available to undertake an activity, the ability to allocate an activity within a given workflow case to the same resource that undertook a preceding activity. For instance, any PhD-Student available can undertake the Make Reservations task, but it should be allocated to the same person that undertook the Submit Paper task.

Make Reservations: (HAS POSITION PhDStudent) AND IF POSSIBLE (IS PERSON WHO DID SubmitPaper)

**Capability-based Allocation**: The ability to offer or allocate instances of a task to resources based on specific capabilities that they possess. For instance, instances of the Submit Paper task must be allocated to someone with a degree:

Submit Paper: HAS CAPABILITY degree

**Organisational Allocation**: The ability to offer or allocate instances of a task to resources based on their position within the organization and their relationship with other resources. For instance, the Sign Travel Authorisation task must be allocated to someone that is reported by (the position of) the person that undertook the Fill Travel Authorisation task:

Sign Travel Authorisation: IS REPORTED BY POSITION OF PERSON WHO DID ACTIVITY FillTravelAuthorisation

Note that we have not included history-based allocation because we focus on a single business process instance and disregard previous executions and automatic execution because no resource assignments is required in this case.

Putting together the constraints we have mentioned as examples for all the creation patterns, the final resource assignment of every activity of the process in Figure 3 are those depicted in Figure 4. Note that the last assignment does not belong to the previous examples and has been specified here to show how Language 1 allows expressing quite complex constraints.

4 Mapping Assignment Expressions into DLs

In the previous sections, we have provided a description of a language to express resource assignment expressions. However, its semantics has been defined in an intuitive way. In this section, we provide a precise definition of their by means of a semantic mapping. A semantic mapping is a way to provide semantics to a model, in this case the RAL language, by mapping the concepts into a semantic domain, i.e., a target domain whose semantics has been formally defined [3]. The advantage of defining such semantic mapping is that it allows one to use the techniques specific to the target semantic domain for analysing the source models [3] and obtaining information from them.
Towards the Definition and Analysis of Resource Assignments in BPMN 2.0

Submit Paper:
((HAS ROLE Researcher) OR (HAS ROLE ResearchAssistant))
AND (HAS CAPABILITY degree)

Fill Travel Authorisation:
(HAS ROLE ResearchAssistant) AND
(NOT (IS PERSON WHO DID ACTIVITY SignTravelAuthorisation))

Sign Travel Authorisation:
(IS Antonio AND (NOT(IS PERSON WHO DID ACTIVITY FillTravelAuthorisation)))AND
(IS REPORTED BY POSITION OF PERSON WHO DID ACTIVITY FillTravelAuthorisation)

Send Travel Authorisation:
IS PERSON IN DATA FIELD Authorisation.Applicant

Register:
(HAS POSITION PhDStudent) AND IF POSSIBLE (IS PERSON WHO DID Submit_Paper)

Make Reservations:
(NOT (IS Antonio)) AND ((SHARE SOME ROLE WITH RESOURCE IN ACTIVITY
SignTravelAuthorisation) OR (HAS ROLE TeacherAssistant))

Fig. 4: Resource assignment of activities of the process

We have chosen Description Logics (DLs) [4] as the semantic domain for our mapping. DLs are logics that serve primarily for formal description of concepts, roles (relations between the concepts, also known as properties) and individuals (instances of the concepts). Specifically, knowledge representation systems based on DLs consist of two components: TBox and ABox. The TBox describes terminology, i.e., the ontology in the form of concepts and role definitions and their relationships, while the ABox contains assertions about individuals using the terms from the ontology. DLs have been chosen because of two reasons. On the one hand, the definition of an organizational structure fits nicely into the way DLs expresses its concepts and, hence, DLs provide a very natural way to describe the problem. On the other hand, DL is best known for providing a logical formalism for ontologies and the semantic web. As a matter of fact, the semantics of the W3C recommendation OWL 2 [5] to express ontologies for the semantic web is defined in DL. Consequently, there is a plethora of DLs reasoners available that can be used to automatically analyse the RAL expressions efficiently.

There are three elements that need to be mapped from RAL expressions into DLs. Next, we describe each mapping in detail. A full version of this mapping expressed in OWL 2 and applied to the example in Section 3 can be found at http://www.isa.us.es/cristal.
4.1 Metamodel of the organizational structure

The metamodel of the organizational structure depicted in Figure 1 is mapped into the TBox, i.e., classes of the metamodel are mapped into concepts of the TBox and relations in the metamodel are mapped into roles of the TBox. This mapping is quite straightforward since the elements used to express them are quite similar. Thus, on the one hand, classes Resource, IndividualResource, GroupResource, Capability, Person, Role, Position, OrganizationalUnit and OrganizationalTeam are directly mapped into concepts of the TBox as well as their subclassOf relationship.

On the other hand, relationships capabilities, occupies, can delegate work to, reports to, participates in and is member of are mapped into roles of the TBox together with their corresponding cardinality restrictions. For instance, the cardinality restriction: “each Position is member of exactly one OrganizationalUnit” is mapped into the following restriction: CatPos = 1 isMemberOf OrganizationalUnit.

In addition, role canDelegateWorkTo is defined as transitive and role extendsReportTo is defined as a transitive superrole of reportsTo.

Finally, for some of the roles, their inverse role has been defined because they are very convenient for the definition of RAL expressions. Specifically, the roles isOccupiedBy, canHaveWorkDelegatedBy, isReportedBy, isExtendedReportedBy, developedIn and formedBy have been defined as the inverse roles of occupies, canDelegateWorkTo, reportsTo, extendsReportTo, participatesIn and isMemberOf, respectively.

4.2 Structure of a concrete organization

The structure of a concrete organization, such as the organizational structure described in Section 3 is mapped into the ABox, according to the terms used in the TBox. Like the previous one, this mapping is also quite straightforward. The instances of the metamodel are defined as individual assertions in the ABox (e.g., Researcher is a Role: Role(Researcher)) and the relationships between the individuals are stated as property assertions (e.g., Position Lecturer participates in Role Researcher: participatesIn(Lecturer, Researcher)).

Finally, an additional individual assertion has been made for each individual to state that each individual has exactly the properties stated and no more (e.g. Position Lecturer has exactly three participatesIn relationships: (= 3 participatesIn)(Lecturer)). This is technical detail that is necessary to be able to express the negation included in RAL language because of the open world assumption of DL. The open world assumption means that DL assumes that the knowledge may be incomplete and, hence, the absence of a property assertion stating that participatesIn(Lecturer, Clerk) does not mean that a Lecturer does not have role Clerk, it may simply mean that the knowledge is incomplete and it has not been asserted yet.
4.3 RAL expressions

The RAL expressions depicted in Language 1 are mapped into concepts of the TBox. The idea is that each RAL expression can be seen as a definition of a subset of all the people in the organization who can do a certain activity. In terms of DL, this can be seen as a concept that characterizes the individuals that belong to it amongst all the individuals of type Person that there are in the ABox. Unlike the previous mappings, this mapping is not as straightforward and involves expressing in DL the restrictions that RAL language imposes on the people that can do a certain activity of the business process. Let $map(expr)$ be a mapping from a RAL expression into DL statements, the concept that defines the people that can do a certain activity $a$ of the business process whose RAL expression is $expr_a$ is defined as:

$$AssignmentA \equiv map(expr_a)$$

This mapping $map(expr)$ is summarised in Table 1 and detailed as follows:

- Person expressions provide ways to refer to a concrete person. However, in the last two cases, the concrete person is unknown until runtime. Therefore, the expression is mapped to a concept containing the concrete person if it is known at design-time or an approximation, otherwise. The approximation is either all persons in the organization in case the concrete person is defined in a data field because we cannot figure out who might be, or all the persons who can do a certain activity in the business process in case the concrete person is defined as the person who did that activity.
- Trivially, expressions of type IS personExpr are defined directly as the person expression mapping it uses.
- Expressions of type HAS GroupResource are defined either as the persons that occupy a given position, or as the persons that occupy a given position that participatesIn or isMemberOf a certain roleName or unitName, respectively.
- Expressions of type HAS CAPABILITY are defined as those persons who have the given capability.
- Expression of type THE SAME AS is defined by making it equivalent to the concept defined for the given activity.
- Expressions of type REPORTS TO are defined as the persons who occupy a position that has an extendedReportsTo relationship with a given position name or with the positions occupied by a given person. Note that since extendedReportsTo is defined as transitive, it will apply not only to the position to which it reports directly, but to all of its ancestors as well.
- Expressions of type IS REPORTED BY, CAN DELEGATE WORK TO and CAN HAVE WORK DELEGATED BY are like the previous one, but changing the relationship extendedReportsTo with the adequate relationship

---

4 The DL mapping is expressed using a syntax commonly used for DL. The interested reader can consult details about the syntax in [4].
Table 1: Mapping from RAL expressions to DL concepts.

<table>
<thead>
<tr>
<th>RAL Expression</th>
<th>DL Expression</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{map}($expr$)$</td>
<td>$\text{expr}$</td>
<td></td>
</tr>
<tr>
<td>$\text{not}$ $\text{map}($expr$)$</td>
<td>¬$\text{map}($expr$)$</td>
<td></td>
</tr>
<tr>
<td>$\text{map}($expr$) \cup \text{map}($expr$)$</td>
<td>$\text{map}($expr$) \cup \text{map}($expr$)$</td>
<td></td>
</tr>
<tr>
<td>$\text{map}($expr$) \cap \text{map}($expr$)$</td>
<td>$\text{map}($expr$) \cap \text{map}($expr$)$</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- The mapping from RAL expressions to DL concepts is shown in the table. Each RAL expression is mapped to a corresponding DL expression and mapping.
- The table includes mappings for logical operations such as `not`, `or`, `and`, and `not`.
- The DL expressions use a combination of `map`, `not`, and logical operators to represent the RAL expressions.
- The mappings are designed to maintain the semantic equivalence between the RAL and DL representations.

**Example:**
- RAL expression: `map(person)`
- DL expression: `map(person)`

**Explanation:**
- The RAL expression `map(person)` represents a mapping to a person in the DL concept.
- The DL expression `map(person)` is used to express the same concept in the DL framework.
Towards the Definition and Analysis of Resource Assignments in BPMN 2.0

in each case. This mapping is not depicted in the table for the sake of simplicity.

- Expressions of type \textit{SHARE SOME POSITION WITH} are defined as the persons that (1) occupy at least one common position with a given person, (2) occupy at least one common position with the set of positions that are shared by all the people who can do a given activity, or (3) all people, because, again, we cannot determine the content of a data until runtime.

- Expressions of type \textit{SHARE SOME ROLE/UNIT WITH} are like the previous one, but changing it accordingly for each of the group resource types. It applies the same idea than in the \textit{HAS GroupResource} expressions.

- Compound expressions have a rather direct mapping, the only exception being expression \textit{AND IF POSSIBLE}. This expression just expresses a preference about the assignment, but its fulfillment is not mandatory. Therefore, it is ignored in the mapping.

5 Querying Assignment Expressions

The definition of the semantics of the RAL expressions in terms of DLs makes it possible to automate the analysis of such expressions by means of a DL reasoner. DL reasoners are software tools that implement several operations on the ontologies in an efficient manner by using several heuristics and techniques. Some examples of such operations are:

- \textit{satisfiability}(C): Determine whether a description of the concept C is not contradictory

- \textit{subsumes}(A, B): Determine whether concept A subsumes concept B, i.e., whether description of A is more general than description of B.

- \textit{individuals}(C): Find all individuals that are instances of concept C.

- \textit{realization}(i): Find all concepts which the individual i belongs to.

By using these operations, we can analyse the assignment of resources we have made to a given business process in order to extract information from it and answer questions such as \textit{Who are the people that can do activity A?} or \textit{Who are the activities that can do person P?}. Table 2 depicts some of these questions and how they can be written on the basis of DL operations.

Furthermore, these operations allow us to define the consistency of a resource assignment of a business process. Specifically, let A be the set of activities of a business process, we can say a resource assignment of a business process is consistent if for all the activities in the business process the resource assignment is consistent and there is at least one person that can do the activity:

\[
\forall a \in A (\text{\textit{satisfiability}}(\text{Assignment} A) \land |\text{\textit{individuals}}(\text{Assignment} A)| \geq 1)
\]

6 Related work

Most existing approaches for business process modelling and analysis focus on structural and control flow issues. For instance, Awad et al. deal with structural
Who are the people that can do activity A?

Who are the activities that can do person p?

Is there any person that can do all of the activities of the business process?

Are the people that can do activity B a subset of those that can do activity A?

Can the same people do activity A and B?

<table>
<thead>
<tr>
<th>Question</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who are the people that can do activity A?</td>
<td>(\text{individuals}(\text{AssignmentA}))</td>
</tr>
<tr>
<td>Who are the activities that can do person p?</td>
<td>(\text{realization}(p)) and, then, select all those concepts that are assignments</td>
</tr>
<tr>
<td>Is there any person that can do all of the activities of the business process?</td>
<td>(\text{individuals}(\text{AssignmentA} \cap \ldots \cap \text{AssignmentX})), where (\text{AssignmentA} \ldots \text{AssignmentX}) are all of the assignments of the business process</td>
</tr>
<tr>
<td>Are the people that can do activity B a subset of those that can do activity A?</td>
<td>(\text{subsumes}(A, B))</td>
</tr>
<tr>
<td>Can the same people do activity A and B?</td>
<td>(\text{subsumes}(A, B) \land \text{subsumes}(B, A))</td>
</tr>
</tbody>
</table>

Table 2: Questions that can be answered regarding RAL expressions

detection of deadlocks by performing graphical queries with BPMN-Q [6]. In [7] Vanhatalo et al. propose a technique to speed up control-flow analysis of business process models. In addition, handling the timing between activities of a business process is also interesting, specially in processes with critical temporal constraints. For instance, Dai et al. present an approach to specify and verify timing constraints in Web service compositions based on BPEL4WS [8].

Besides control flow and timing, considering artifacts such as data objects in business processes is of utmost importance. Sadiq et al. present a catalogue of data anomalies in [9], and Awad et al. deal with the diagnosis and verification of data anomalies in [10]. Cabanillas et al. [11] introduce an approach to automatically generate a data-centered view from a business process model.

As far as resources are concerned, Russell et al. have described a set of Workflow Resource Patterns important to manage resources in workflow environments, and have analysed the support provided by some workflow tools, but they do not provide a way to assign resources to business process activities [2]. We have used the patterns related to the modelling of resource allocations to prove the expressiveness of language RAL. La Rosa et al. present a rich meta-model for capturing role-task and object-task associations embodied in the EPC notation, which can be transposed to other notations. However, they do not provide a way to obtain organizational and business information. Nakatumba et al. analyse and characterise resource behaviour after business process execution from event logs using process mining [12]. Finally, an optimal approach to allocate the most proficient set of employees for a whole business process from event logs based on Hidden Markov Models is introduced in [13].

7 Conclusions and future work

The results obtained during the development of this work let us conclude that defining formal languages to describe resource assignments in BPMN 2.0 is possible. RAL, the formal language we propose in this paper, does not only allow
Towards the Definition and Analysis of Resource Assignments in BPMN 2.0

precisely defining the assignments required to cover most of the creation patterns proposed by Russell et al. [2] and more complex assignments, but it allows the reasoning about the resource assignments configured as well. To this end, RAL semantics has been described in an OWL-DL ontology and we have shown how DL reasoners can be used to extract information from them. Further details about the ontology and several additional examples can be found at http://www.isa.us.es/cristal.

It is important to highlight the expressiveness of RAL that outperforms the capabilities not only of BPMN but also other commercial systems to express resource assignments, according to the analysis carried out in [2].

We believe the present work settles the basis towards the spread of the use of resource assignments in BPMN 2.0, something we consider vital to be able to incorporate BPMN in business environments (organizations) currently vetoed due to their inability to link the organizational structure with business processes in an efficient and standardized way.

In the future we intend to address resource assignments at runtime (also known as resource allocation) as well as to develop a visual notation for RAL and to define a catalogue of analysis operations that can be done to extract information from RAL.

References