Defining and Analysing Resource Assignments in Business Processes with RAL[*]

Universidad de Sevilla, Spain
{cristinacabanillas,resinas,aruiz}@us.es

Abstract. Business process (BP) modelling notations tend to stray their attention from (human) resource management, unlike other aspects such as control flow or even data flow. They not only offer little intuitive languages to assign resources to BP activities, but neither link BPs with the structure of the organization where they are used, so BP models can easily contain errors such as the assignment of resources that do not belong to the organizational model. In this paper we address this problem and define RAL (Resource Assignment Language), a domain-specific language explicitly developed to assign resources to the activities of a BP model. RAL makes BPs aware of organizational structures. Besides, RAL semantics is based on an OWL-DL ontology, which enables the automatic analysis of resource assignment expressions, thus allowing the extraction of information from the resource assignments, and the detection of inconsistencies and assignment conflicts.

Keywords: resource-aware business process model, RAL, workflow resource pattern, organizational model, OWL, description logics.

1 Introduction

In this paper we face human-resource[1] management in BP models. Specifically, we deal with resource assignment to the activities of a BP, aiming at easing and improving the way resources can be associated with BP activities. Unfortunately, unlike other aspects such as control flow, resources have received much less attention. However, the participation of people in BPs is of utmost importance, both to supervise the execution of automatic activities and to carry out software-aided and/or manual activities, so they should be considered when designing and modelling the BPs used in an organization.

Furthermore, the alignment of the BPs of an organization with its organizational structure enables the automation of work in different directions. On the one hand, it makes it possible to infer interesting information, such as: (i) the potential performers of each BP activity, i.e., the set of people that fulfill the

[*] Partially supported by the European Commission (FEDER), Spanish Government under project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian Local Government.

[1] From now on we will use the term *resource* to refer to human resources.

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 477–486, 2011.

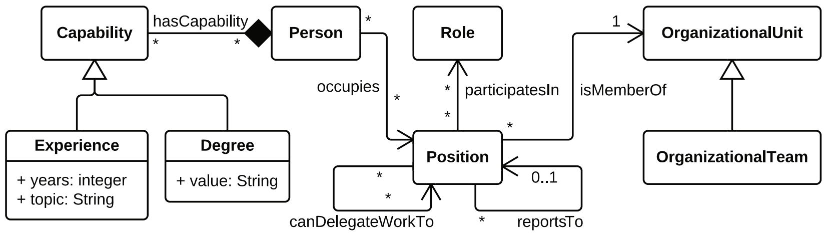


Fig. 1. Excerpt of the organizational metamodel described by Russell et al. [1]

resource-related constraints imposed in the model; or (ii) the potential set of activities each person of an organization can be assigned at runtime. This kind of information may be beneficial for an 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 and, hence, allowing reacting in time to avoid having people overburdened with work. On the other hand, it enables the detection of inconsistencies between the resource assignments associated to activities of a BP and the structure of the organization where it is used, e.g. non-existent roles or persons.

The main contribution of this paper is the description and formalization of RAL (Resource Assignment Language), a DSL (Domain Specific Language) to express resource assignments in the activities of a BP in terms of the concepts used in the organizational metamodel proposed by Russell et al. [1]. This formal description is provided by means of a semantic mapping between RAL and description logics (DLs), which is a logical formalism widely used by the semantic web community. A semantic mapping is a way to provide semantics to a model, RAL, by mapping its concepts into a target domain whose semantics has been formally defined [2]. An important advantage of our approach is that one can capitalize on existing efficient DLs algorithms for inferring the aforementioned interesting information from RAL expressions, instead of having to invent a corresponding ad-hoc algorithm for each problem. Furthermore, a prototype has been developed to show the use of RAL and the benefits of its DL-based semantics.

After introducing RAL is Section 2, we describe the semantic mapping in Section 3. Then, we detail how we can leverage DLs to analyse resource assignments in Section 4. Related work can be found in Section 5, and a set of conclusions and future work are discussed in Section 6.

2 Introduction to RAL. Definition and Application

RAL is a DSL that allows the assignment of resources to BP activities in terms of the concepts used in organizational models such as persons, roles, positions, capabilities, or organizational units. Specifically, the concepts used in RAL (cf. Figure 1) are a subset of those included in the organizational metamodel described by Russell et al. [1]. This metamodel was used by the authors as basis to

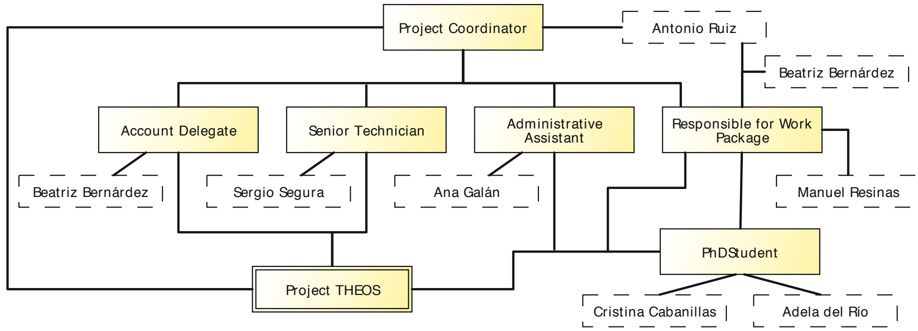
Language 1. RAL's EBNF definition

```
Expression := IS PersonConstraint
     | HAS GroupResourceType GroupResourceConstraint
     | SHARES Amount GroupResourceType WITH PersonConstraint
     | HAS CAPABILITY CapabilityConstraint
     | IS ASSIGNMENT IN ACTIVITY activityName
     | RelationshipExpression
     | CompoundExpression

RelationshipExpression := 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)

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

GroupResourceConstraint := groupResourceName
             | IN DATA FIELD dataObject.fieldName

CapabilityConstraint := capabilityName
              | CapabilityRestriction

PositionConstraint := POSITION namePosition
            | POSITION OF PersonConstraint

Amount := SOME        GroupResourceType := POSITION
     | ALL                          | ROLE
                                  | UNIT
Depth := DIRECTLY
       | λ
```

define a set of *workflow resource patterns*. These patterns have already been used by other authors as framework to extend BPMN regarding resource management [3], so we believe it is reasonable to use the same metamodel.

Building on this metamodel, RAL allows formulating expressions that define who can perform an activity in the BP. The concrete syntax of RAL is specified in Language 1, whereas its abstract representation can be found at `http://www.isa.us.es/cristal`. In short, RAL allows expressing that an activity has to be performed by, e.g.: *a*) a concrete person, the person who did another activity, or the person indicated in a data field; *b*) someone with a specific group resource[2]; *c*) a person that has a group resource in common with other person; *d*) someone with certain capability; and *e*) someone that reports to or can delegate work to a given position. The language also allows stating that an activity has the same RAL expression as another activity (no matter which

[2] We use the term *group resource* when referring to concepts that represent groups of persons, i.e., positions, roles and organizational units.



Position	Role
Project Coordinator	Responsible
	Account Administrator
	Resource Manager
	Doctoral Thesis Advisor
	Researcher
Responsible for Work Package	Responsible
	Researcher
	Doctoral Thesis Advisor

Position	Role
PhD Student	Research Assistant
Senior Technician	Responsible
Account Delegate	Account Administrator
Administrative Assistant	Clerk

Fig. 2. Excerpt of ISA group's organizational model from a project perspective

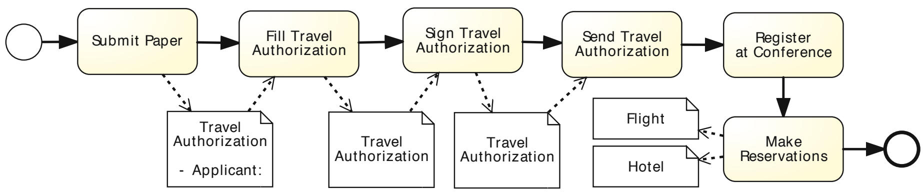
it is), and formulating negative and compound assignments with conjunctions `NOT`, `AND`, `OR` and `AND IF POSSIBLE`. The last one helps indicate preferences. For a more detailed description, we refer the reader to [4].

Figure 2 depicts a possible instantiation of the organizational metamodel shown in Figure 1. This instance is an excerpt of the *ISA Research Group* of the University of Seville from a research project perspective. There are six positions that are members of one organizational unit (Project THEOS), and seven persons occupying these positions. Each position of the model can delegate work to any inferior position and reports work to its immediately upper position. Relationship *participatesIn* is summarized in a table. A table with relationship *hasCapability* is also required, but it is omitted here for space limitations.

Based on that organizational model, one can use RAL to assign resources to the activities of a BP. For instance, Figure 3 shows resource assignments for some activities of the example BP, along with the corresponding DL queries, which will be explained in Section 3.

3 RAL Semantics

In this section, we provide a precise definition of RAL by means of a *semantic mapping* into DLs. 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 *roles* (relations between the concepts) definitions and their relationships, while the ABox contains *assertions* about individuals (instances of concepts) using the terms from the ontology (see [5] for more details about DLs and their syntax). We have chosen DLs because of



Submit Paper: Only *Researchers* and *Research Assistants* are authorized to execute this task, and they must have a degree.
RAL: `((HAS ROLE Researcher) OR (HAS ROLE ResearchAssistant)) AND (HAS CAPABILITY Degree)`
DL: $AssignmentSubmitPaper \equiv ((\exists occupies.(\exists participatesIn.\{Researcher\})) \sqcup (\exists occupies.(\exists participatesIn.\{ResearchAssistant\}))) \sqcap (\exists hasCapability.\{Degree\})$

Sign Travel Authorization: This task must be undertaken by someone that is reported by (the position of) the person that undertook task *Submit Paper*.
RAL: `(IS REPORTED BY POSITION OF PERSON WHO DID ACTIVITY SubmitPaper)`
DL: $AssignmentSignTravelAuth \equiv \exists occupies.(\exists isExtendedReportedBy.(\exists isOccupiedBy.\{AssignmentSubmit\}))$

Make Reservations: Antonio cannot execute this task but the performer must either have some role in common with Antonio.
RAL: `(NOT (IS Antonio)) AND (SHARES SOME ROLE WITH Antonio)`
DL: $AssignmentMakeReservations \equiv (\neg\{Antonio\}) \sqcap (\exists occupies.(\exists participatesIn.(\exists developedIn.(\exists isOccupiedBy.\{Antonio\}))))$

Fig. 3. Resource assignments to activities of a process for Conference Travel

two reasons. First, DLs provide a very natural way to describe an organizational structure. Second, there is a plethora of DLs reasoners that can be used to automatically analyse RAL expressions efficiently and, hence, to automatically infer information from them.

Note that, since RAL builds on an organizational metamodel, it is necessary to provide a mapping not only to RAL expressions but also to the organizational metamodel and its instances. Next, we detail all those mappings. A full version can be found at `http://www.isa.us.es/cristal`, in which we use the W3C recommendation OWL 2 [6] to define the ontologies[3].

3.1 Mapping the Organizational Structure into DLs

Mapping the organizational metamodel into an ontology: It is quite straightforward since the elements used in both domains are quite similar. Each class of the metamodel is mapped to one *concept* and the hierarchies are mapped using the *subclassOf* relationship. The remaining relationships (i.e., *hasCapability*, *occupies*, *canDelegateWorkTo*, *reportsTo*, *participatesIn* and *isMemberOf*)

[3] Sometimes OWL terms *classes*, *properties* and *objects* will be used to refer to DL terms *concepts*, *roles* and *individuals*, respectively.

are mapped into *properties* together with their corresponding *cardinality restrictions*. In addition, properties *isOccupiedBy*, *isReportedBy*, *developedIn* and *formedBy* have been defined as the inverse properties of *occupies*, *reportsTo*, *participatesIn* and *isMemberOf*, respectively, to make it easier the formulation of some RAL expressions. Properties *hasDegree* and *hasExperience* have been added to represent the existing specific capabilities. Furthermore, in order to be able to transitively refer to upper positions in the organizational model, a *transitive* super-property *extendedReportsTo* has been added. For the same reason, property *extendedCanDelegateWorkTo* has been added as well.

Instantiating the Organizational Ontology: The structure of a concrete organization, such as that in Figure 2, is mapped as *individual assertions* in the ABox (e.g., $Role(Researcher)$) and the relationships between the individuals are stated as *property assertions* (e.g., $participatesIn(ProjectCoordinator, Researcher)$). Besides, 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 *Project Coordinator* has exactly five *participatesIn* relationships: $(= 5\ participatesIn)(ProjectCoordinator)$). This is a technical detail that is necessary to be able to express the negation included in RAL because of the open world assumption of DLs. The *open world assumption* means that DLs assume that the knowledge may be incomplete and, hence, the absence of a property assertion stating that $participatesIn(ProjectCoordinator, Clerk)$ does not mean that a *Project Coordinator* does not have role *Clerk*.

3.2 Mapping RAL Expressions into DLs

Each RAL expression can be seen as a definition of a subset of all the people in the organization who can do an activity, e.g. a RAL expression stating that certain activity can only be done by someone occupying position *Project Coordinator* reduces the set of potential owners to the persons that occupy that position. In terms of DL, a RAL expression can be seen as a new concept that characterises the individuals that belongs to it amongst all the individuals of type *Person* that there are in the ABox. Therefore, the concept that defines the resource assignment of a certain BP activity a whose RAL expression is $expr_a$ is defined as: $AssignmentA \equiv map(expr_a)$, where $map(expr)$ is a mapping from a RAL expresion into DL as summarised in Table 1. Figure 3 shows the DL queries for some activities of the BP model.

PersonConstraints provide ways to refer to a concrete person. However, in the last two cases the concrete person is unknown until runtime, in which case an approximation is made. 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 BP, in case the concrete person is defined as the person who did that activity.

IS PersonConstraint is defined as the `PersonConstraint` mapping it uses.

Table 1. Mapping of some RAL expressions into DLs *concepts*

PersonConstraints (pConst)	**DL Mapping** ($map_p(pConst)$)
personName PERSON IN DATA FIELD d.field PERSON WHO DID ACTIVITY name	$\{personName\}$ $Person$ $AssignmentActivityName$
RAL expression (expr)	**DL Mapping (**$map(expr)$**)**
IS pConst	$map_p(pConst)$
HAS POSITION posName HAS ROLE roleName HAS UNIT unitName HAS GroupResourceType IN DATA FIELD d.field	$\exists occupies.\{posName\}$ $\exists occupies.(\exists participatesIn.\{roleName\})$ $\exists occupies.(\exists isMemberOf.\{unitName\})$ $Person$
SHARES SOME POSIT WITH pConst SHARES SOME ROLE WITH pConst SHARES ALL UNIT WITH pConst	$\exists occupies.(\exists isOccupiedBy.map_p(pConst))$ $\exists occupies.(\exists participatesIn.(\exists developedIn.(\exists isOccupiedBy.map_p(pConst))$ $\exists occupies.(\exists isMemberOf.(\forall formedBy.(\exists isOccupiedBy.map_p(pConst))$
HAS CAPABILITY name HAS CAPABILITY name.attr=val	$\exists hasCapability.\{name\}$ $\exists hasCapability.(name \sqcap \forall attr.\{val\})$
IS ASSIGNMENT IN ACTIVITY name	$AssignmentActivityName$
REPORTS TO POSITION posName REPORTS TO (POSITION OF) pConst	$\exists occupies.(\exists extendedReportsTo.\{posName\})$ $\exists occupies.(\exists extendedReportsTo.(\exists isOccupiedBy.map_p(pConst))$
(expr1) AND (expr2) (expr1) OR (expr2) NOT (expr) (expr1) AND IF POSSIBLE (expr2)	$map(expr1) \sqcap map(expr2)$ $map(expr1) \sqcup map(expr2)$ $\neg map(expr)$ $map(expr1)$

HAS GroupResourceType GroupResourceConstraint is 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. When the specific resource name is given in a data field, the mapping is generalized to any person.

SHARES Amount GroupResourceType WITH PersonConstraint assigns persons that share some or all positions, roles or organizational units with the given person. Expressions with group resource types `ROLE` or `UNIT` apply the same idea but changing it accordingly for each group resource type.

HAS CAPABILITY CapabilityConstraint is defined as those persons who have the given capability and/or persons who have a capability with certain value in some of its attributes. Table 1 shows the case of *equal* operator. Other operator could be used provided it can be mapped to DLs.

IS ASSIGNMENT IN ACTIVITY activityName is defined by making it equivalent to the concept defined for the assignment of the given activity.

Table 2. Some possible analyses of RAL expressions

Question	DL operations
Who are the people that can do activity A?	*individuals(AssignmentA)*
Who are the activities that can do person p?	*realization(p)* and, then, select all those concepts that are assignments
Is there any person that can do all of the activities of the BP?	*individuals(AssignmentA* ⊓ ... ⊓ *AssignmentX)*, where *AssignmentA* ... *AssignmentX* are all of the assignments of the BP
Are the people that can do activity B a subset of those that can do activity A?	*subsumes(A,B)*
Can the same people do activities A and B?	*subsumes(A,B)* ∧ *subsumes(B,A)*

REPORTS TO PositionConstraint Depth is defined as the persons who occupy a position that has a *reportsTo* or *extendedReportsTo* relationship with a given position name depending on whether it is `DIRECTLY` reported or not, respectively. Also, the positions of a given person can be used instead of a concrete position name. The other *relationship expressions* are like this one, but changing the property accordingly, e.g., changing *extendedReportsTo* for *extendedCanDelegateWorkTo*. In *delegate expressions* no direct delegations are allowed.

CompoundExpression has a quite direct mapping except for expressions `AND IF POSSIBLE` and `NOT`. The former expresses a preference for allocation, but it is not mandatory. Thus, in order to ensure the actual potential owner of the activity is within the result, the right side of the expression is ignored in the mapping. The latter must be generalized to any person if *expr* contains runtime information, e.g. the person who did an activity.

4 DL-Based Analysis of Resource Assignments

The definition of the semantics of RAL expressions in terms of DLs makes it possible to automate their analysis 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 of these operations are:

- $satisfiability(C)$: Determine whether concept C is not contradictory.
- $subsumes(A, B)$: Determine whether concept A subsumes concept B, i.e., whether description of A is more general than description of B.
- $individuals(C)$: Find all individuals that are instances of concept C.
- $realization(i)$: Find all concepts which the individual i belongs to.

By using these operations, we can analyse the assignment of resources made to a BP model in order to extract information from it and answer questions such as "*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. These operations allow us to detect problem situations, such as that in which there is an activity that cannot be allocated to any person given the RAL expression of the activity and the organizational model of the company.

5 Related Work

The need of including organizational aspects in BP design can be seen in [7], where Künzle et al. present a set of challenges that should be addressed to make BPs both data-aware and resource-aware. Bertino et al. have defined a language to express constraints in role-based and user-based assignments to the tasks of a WF [8]. They get to check whether the configured assignments are possible at runtime and to plan possible resource allocation based on the assignments. They consider also dynamic aspects for these checkings. However, the language is more complex and hard to use than RAL because its goal is wider.

Russell et al. defined the *workflow resource patterns* with the aim of explaining the requirements for resource management in workflow (WF) environments [1]. They analysed the support provided by some WF tools, but they did not provide a specific way to assign resources to WF activities. These patterns were used by other authors as a reference framework to analyse the ability of BPMN to deal with resources and to propose solutions to improve BPMN [3]. The *creation patterns* have been used to assess the expressiveness of RAL in [4].

Strembeck et al. presented a formal metamodel for process-related role-based access control models and they defined a runtime engine to enforce the different policies and constraints in a software system [9]. However, the resource assignments that can be made with their metamodel is less expressive than RAL. Besides, they have to use ad-hoc algorithms instead of reusing those already implemented for DLs. An optimal approach to allocate the most proficient set of employees for a whole BP from event logs based on Hidden Markov Models is introduced in [10] and Nakatumba et al. proposed a way to analyse and characterise resource behaviour after BP execution from event logs using process mining [11].

Finally, some extensions to enhance resource management in BP execution environments have been recently released, e.g., BPEL4People and WS-HumanTask are two extension proposals for BPEL (both can be found at `http://www.oasis-open.org/committees/bpel4people/`). However, there is limited support to express and manage resource allocation on higher level modelling languages such as BPMN.

6 Conclusions and Future Work

The result of this work lets us conclude that defining and automatically analysing new languages to describe resource assignments in BP models is possible. RAL, our proposal, allows not only precisely defining the assignments required to cover most of the *creation patterns* proposed by Russell et al. [1] and more expressive assignments, but also automatically reasoning about the resource assignments configured. 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. However, it is important to notice that RAL currently addresses only expressions involving a *single instance* of a BP, i.e., the history of individual resources and past executions are not considered for now.

We have developed a prototype that analyses RAL expressions associated to BPMN activities and returns the potential owners of a selected activity. It has been implemented as a plugin for Oryx [12] and it can be tested following the instructions described at http://www.isa.us.es/cristal.

We believe the present work settles the basis towards the spread of the use of resource assignments in BP models, something we consider vital to be able to incorporate business environments (organizations) currently limited, due to their inability to link the organizational structure with BPs, in a efficient and standardized way, and extracting information from them.

In the near future we intend to refine the mapping to obtain more precise information about the potential owners of the activities of a process, according to the execution state of a process instance. In addition, we plan to develop a visual notation for RAL and to define a specific catalogue of analysis operations for the extraction of interesting resource-related information.

References

1. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
2. Rivera, J.E., Guerra, E., de Lara, J., Vallecillo, A.: Analyzing rule-based behavioral semantics of visual modeling languages with maude. In: Gašević, D., Lämmel, R., Van Wyk, E. (eds.) SLE 2008. LNCS, vol. 5452, pp. 54–73. Springer, Heidelberg (2009)
3. Awad, A., Grosskopf, A., Meyer, A., Weske, M.: Enabling resource assignment constraints in BPMN, tech. rep., BPT at Hasso Plattner Institut (2009)
4. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: RAL: A high-level user-oriented resource assignment language for business processes. In: BPM Workshops, BPD 2011 (in press, 2011)
5. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logics Handbook: Theory, Implementations, and Applications. Cambridge University Press (2003)
6. Motik, B., Patel-Schneider, P.F., Grau, B.C.: OWL 2 Web Ontology Language Direct Semantics (2009)
7. Künzle, V., Reichert, M.: Integrating Users in Object-Aware Process Management Systems: Issues and Challenges. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 29–41. Springer, Heidelberg (2010), http://www.informatik.uni-trier.de/~ley/db/conf/bpm/bpmw2009.html
8. Bertino, E., Ferrari, E., Atluri, V.: The specification and enforcement of authorization constraints in workflow management systems. ACM Trans. Inf. Syst. Secur. 2, 65–104 (1999)
9. Strembeck, M., Mendling, J.: Modeling process-related RBAC models with extended UML activity models. Inf. Softw. Technol. 53(5), 456–483 (2011)
10. Yang, H., Wang, C., Liu, Y., Wang, J.: An Optimal Approach for Workflow Staff Assignment Based on Hidden Markov Models. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 24–26. Springer, Heidelberg (2008)
11. Nakatumba, J., van der Aalst, W.M.P.: Analyzing Resource Behavior Using Process Mining. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 69–80. Springer, Heidelberg (2010), http://www.informatik.uni-trier.de/~ley/db/conf/bpm/bpmw2009.html
12. Decker, G., Overdick, H., Weske, M.: Oryx - An Open Modeling Platform for the BPM Community. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 382–385. Springer, Heidelberg (2008)