Compliance in Resource-based Process Models

Silvano Colombo Tosatto [a,b] Yehia Elrakaiby[a] Pouyan Ziafati[a,c]

[a] *University of Luxembourg, Luxembourg*
[b] *Universitá di Torino, Italy*
[c] *Utrecht University, Netherlands*

Abstract

Execution of business processes often requires resources, the use of which is usually subject to constraints. In this paper, we study the compliance of business processes with resource usage policies. To this end, we relate the execution of a business process to its resource requirements in terms of resources consumed, produced or blocked by tasks of the business process. Policies specifying constraints on resource usage are specified in the form of obligations and the verification of whether a business process complies with a given resource usage policy is formally studied.

1 Introduction

Workflows are used for modelling, designing, executing, monitoring and optimizing business processes. Being essential parts of complex dynamic business environments, it is often necessary to annotate business processes with the resources necessary for their execution. Since the resources that are available for the execution of business processes are typically limited, there is a clear need to model the possible interactions between business processes and resources, i.e. how the resources are used when the business process is being executed, and whether the execution of the business process does not violate any resource constraints.

Previous works in literature have been mainly focused on study of the resource scheduling problem, i.e. the allocation of resources to business processes under various constraints. In contrast, we study the problem of formally verifying that a business process design model is consistent with a given set of resource usage constraints. This is a design-time problem to formally prove that the execution of a business process, as defined by its business process model, does not violate a given resource *usage policy*, specifying constraints on the availability, consumption and production of system resources.

This paper introduces a general framework for the modelling of business processes and the specification of their resource usage requirements. From the conceptual point of view, we enable modelling of the system's business processes and resources, and the specification of how tasks of business processes influence the system's resources. In particular, business process tasks may interact with resources in one of the following ways: the execution of a task could lead to the production, consumption or blockage of resources. The resource usage requirements are specified in the form of a *usage policy* defining constraints over the interaction between business processes and system resources. This is in contrast with other approaches where business process constraints are expressed as meta-data of the business process model such as in [1]. The formal verification of the compliance of a business process with a usage policy requires verifying that every possible execution of the business process satisfies the policy constraints defined in the policy.

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 presents an overview of the resource usage compliance problem. Section 4 presents our approach to tackle the problem of checking the compliance of a business process with a resource usage policy. Finally, Section 5 concludes and gives some future directions.

2 Related Work

Resource models and resource usages in business processes and workflows have been studied in the literature. One representative of this research is the work of Lerner et al [4] where a meta-model allowing a comprehensive specification of resources and their complex inter-relationships in processes and work-flows is presented. The meta-model enables the specification of classes of resources, resource hierarchies and resource instances (i.e. concrete resource objects). Resources may be annotated with attributes such as their capacity, quality and availability. The authors also describe how their meta-model can be used by a resource manager to allocate at runtime the resources needed for the execution of tasks. The focus of our work is on the verification of business process model compliance with resource usage policies. In other words, our work enables model validation given a set of resource usage constraints at design-time as opposed to the allocation of resources at runtime. Note that model validation at design-time could be the only way to ensure the success of resource allocation at runtime. In this sense, our work is complementary to theirs.

Russell et al. [7] identify around 40 patterns of interactions between resources and workflows. These patterns are mainly related to resource assignment to a task and division of workload between resources, with a focus on human resources. The ability to support the identified patterns is used as criteria to compare several work-flow management systems. In contrast, we focus on checking the compliance of a business process with a set of resource usage constraints and not on modelling of the ways work-flows and their resources interact.

A rich data model for workflow resources is introduced by Ouyang et al. in [5]. The model considers consumable and non-consumable resources and allows the annotation of resources with information such as their capacity. Their system allows the specification of some restrictions over resource usages such as restrictions on the availability of resources in time. The main difference between our work and theirs is that their focus is on modelling resources to enable the semi-automatic task/resource scheduling, whereas we study the formal verification of business process compliance with constraints over resource usage.

Business process compliance is a research area that is getting more relevant because of the need to ensure that business processes satisfy some necessary properties before their execution. However, The current research [8] mainly focuses on regulatory compliance, i.e. compliance with respect to a set of regulations specified using some normative language. The novelty of our approach is our focus on the study of compliance with resource usage policies.

To conclude, although resource modeling and regulatory compliance have been studied in the literature, to our knowledge, our work is the first introducing a formal model for checking the compliance of business processes with resource usage policies.

3 Motivation & Overview

Companies' architectures include various elements, among which are business processes describing the ways using which a company pursues its business objectives, resource pools describing sets of available resources in the company and policies describing regulations which the company has to follow while achieving its business objectives. The success of a company requires that its business processes and its resource usage requirements are compatible, i.e. that constraints specified on the usage of the company's resources are respected by the company's business processes.

We model the aforementioned elements by considering the architecture presented in Figure 1. The left side of Figure1 shows an abstract representation of this architecture whereas the right side gives a concrete example. The architecture includes business processes (describing the ways a company conducts its business) and resource models (describing the company's resources). Resource usage specifies how the execution of a business process affects resources of the system, i.e. whether a task execution produces, consumes or blocks some resources. Finally, usage policies define regulations that the use of resources by business processes should be compliant with.

Note that the decoupling of business and resource models offers more flexibility than the case of having them tightly combined. For example, the models could be separately designed and updated. Also, business models could be linked with different resource models and their compatibility with respect to a given set of resource usage constraints can be analysed.

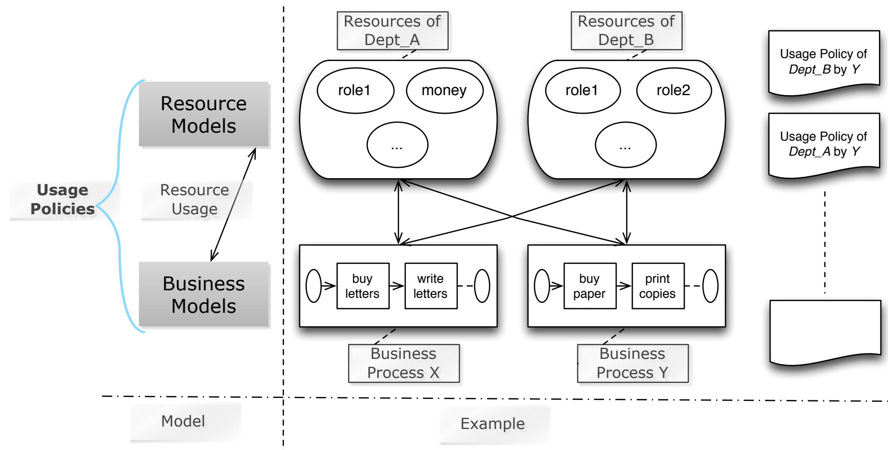


Figure 1: Exemplary Company Architecture: composed of a set of business processes, resource models (execution environments), resource usage models describing the use of resources by business processes and usage policies specifying constraints on the resource usage.

Running Example To present and explain the concepts introduced, we consider an example of a business process describing the required tasks for a family to send Christmas postcards to their friends and family. This example will be used throughout the paper to develop and illustrate our approach.

Example 1 (Christmas Letters) *A family, composed of three people (the father, the mother and a young boy), wants to send greeting letters before Christmas to their friends and relatives. In order to be able to do so, the members of the family need to write greeting letters, buy the envelopes and send the greeting letters.*

4 Resource Usage Compliance of Timed Business Processes

This section starts by introducing the modelling of temporal business processes and resources. Then, the specification of resource usage policies and the verification of whether the use of resources by business processes complies with the specified usage policies is presented.

4.1 Timed Business Process

In order to verify the compliance of a business process model with a set of usage constraints, such model must fulfill some requirements. To faithfully model resources usage, a business process needs to include the notion of time in order to specify with a finer granularity when a resource is being used. Moreover it is also necessary to specify how tasks of the business process interact with resources.

A business process is a structure representing the possible ways of achieving a business objective. Business processes are composed of a set of tasks that should be executed in an order specified by the business process model. A task represents an atomic activity which execution helps towards the achievement of the business goal associated to the business process containing it. In the present paper, we consider timed business processes. Therefore, we associate with each task in the business process a time interval representing the possible minimum and the maximum amount of time required to execute it.

Definition 1 (Task) *A task $t = \langle \ell, \rho \rangle$ is a tuple, where ℓ is the label of the task, ρ is a time interval of the form $[min, max]$ where min and max are positive natural numbers including zero.*

In the following, we use the notations $t.l$ to refer to the label and $t.\rho.min$ and $t.\rho.max$ to refer to the minimum and maximum execution time of a task $t = \langle \ell, \rho \rangle$.

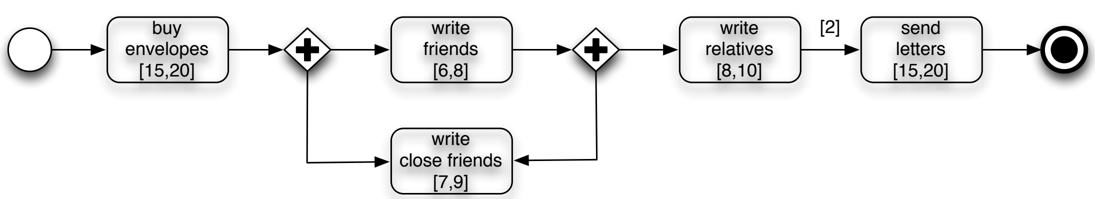
Example 1 may include the following tasks:

- Buying the envelopes: The father (or the mother) should go to buy the envelopes from the nearest post-office. This task is defined as follows: $\langle buy\_envelopes, [15, 20] \rangle$.
- Writing letters for relatives: The father (or the mother) should write letters for their relatives with the help of their son. This task is defined as follows: $\langle write\_relatives\_letters, [8, 10] \rangle$.
- Writing letters for close friends: The father (or the mother) should write letters for their close friends. This task is defined as follows: $\langle write\_close\_friends\_letters, [7, 9] \rangle$.
- Writing letters for friends: The father (or the mother) should write letters for their friends with the help of their son. This task is defined as follows: $\langle write\_friends\_letters, [6, 8] \rangle$.
- Sending the letters: The father (or the mother) should send the letters. This task is defined as follows: $\langle send, [13, 15] \rangle$.

Definition 2 (Process Block) *Let $\mathcal{T}$ be the set of all tasks. A process block B is inductively defined as follows:*

- $\forall t \in \mathcal{T}, t$ *is a process block.*
- *Let $B_1, \ldots, B_n$ be process blocks:*
  - *A sequence block:* $\mathsf{SEQ}(B_1, \ldots, B_n)$ *is a process block.*
  - *An* XOR *block:* $\mathsf{XOR}(B_1, \ldots, B_n)$ *is a process block.*
  - *An* AND *block:* $\mathsf{AND}(B_1, \ldots, B_n)$ *is a process block.*

Definition 3 (Structured Process Model) *A structured process P is a sequence block* $\mathsf{SEQ}(\mathsf{start}, B, \mathsf{end})$*, where* start *and* end *are two pseudo tasks,* $\{\mathsf{start}, \mathsf{end}\} \cap \mathcal{T} = \emptyset$*. Those pseudo-tasks are used to identify the beginning of a structured process model and its ending.*



(a) Business Process 2: write to relatives after writing to friends

Figure 4.1 shows a business process that could be considered for the achievement of the goal to send the greeting cards. This process could be textually represented as follows:

$$\mathsf{SEQ}(\mathsf{start}, buy\_envelopes, \mathsf{AND}(write\_close\_friends\_letters, write\_friends\_letters), write\_relatives\_letters, send\_letters, \mathsf{end})$$

In some cases, it can be useful to allow specifying that a task does not necessarily start after the execution of its preceding task and may start after some delay as shown in the example where a delay of two time units is specified before the task $send\_letters$. We handle a delay $[d]$ relative to the execution of a task t by considering it as a dummy task $\langle dummy_t, [0, d]\rangle$ that precedes t. Note that we omitted the time interval of each task in the textual representation of the process models for the sake of simplicity.

For timed business process models, we define the possible traces. A trace is a set representing the business process tasks and their absolute start and end time points, e.g. $\langle send\_envelopes, [17, 19]\rangle$ means that the task $send\_envelopes$ started at time point 17 and ended at time point 19. Valid traces of a structured business process is defined as follows.

Definition 4 (Traces of a Timed Process Block) *Let B be a process block. A valid trace τ of B given an initial time point t_0, between t_0 and some end time point t_e, is defined as follows. Such a trace is said to be of length $t_e - t_0$ and is denoted by $\tau_B^{[t_0,t_e]}$.*

- *If $B = start$ then $\tau_{start}^{[t_0,t_e]} = \{\langle \mathsf{start}, [t_0, t_e]\rangle\}$ such that $t_0 = t_e = 0$,*
- *If $B = end$ then $\tau_{end}^{[t_0,t_e]} = \{\langle \mathsf{end}, [t_0, t_e]\rangle\}$ such that $t_0 = t_e$,*
- *If $B \in \mathcal{T}$ then $\tau_B^{[t_0,t_e]} = \{\langle B, [t_0, t_e]\rangle\}$ such that $B.\rho.min \leq t_e \leq t_0 + B.\rho.max$,*
- *If $B = SEQ(B_1, ...B_i, ..., B_n)$ then $\tau_{SEQ(B_1,...,B_n)}^{[t_0,t_e]} = \tau_{B_1}^{[t_{1s},t_{1e}]} \cup ... \cup \tau_{B_{i-1}}^{[t_{(i-1)s} \cup t_{(i-1)e}]} \cup \tau_{B_i}^{[t_{is},t_{ie}]} \cup ... \cup \tau_{B_n}^{[t_{ns},t_{ne}]}$ such that $t_{1s} = t_0$, $t_e = t_{ne}$ and $t_{is} = t_{(i-1)e}$ for every $1 \leq i \leq n$,*
- *If $B = XOR(B_1, ...B_i, ..., B_n)$ then $\tau_{XOR(B_1,...,B_n)}^{[t_0,t_e]} = \tau_{B_i}^{[t_{is},t_{ie}]}$ for some $1 \leq i \leq n$ such that $t_{is} = t_0$ and $t_e = t_{ie}$,*
- *If $B = AND(B_1, ...B_i, ..., B_n)$ then $\tau_{AND(B_1,...,B_n)}^{[t_{1s},t_e]} = \tau_{B_1}^{[t_{1s},t_{1e}]} \cup ... \cup \tau_{B_i}^{[t_{is},t_{ie}]} \cup ... \cup \tau_{B_n}^{[t_{ns},t_{ne}]}$ such that $t_e = max(\bigcup_{i=1}^{n}\{t_{ie}\})$ and $t_{is} = t_0$ for every $1 \leq i \leq n$.*

According to the definition above, the following are some of the valid traces of the business process:

$$\{\langle \mathsf{start}, [0, 0]\rangle, \langle \mathsf{buy\_envelopes}, [0, 17]\rangle, \langle \mathsf{write\_friends}, [17, 25]\rangle, \langle \mathsf{write\_close\_friends}, [17, 24]\rangle, \langle \mathsf{write\_relatives}, [25, 34]\rangle, \langle \mathsf{send\_letters}, [36, 54]\rangle\} \tag{1}$$

$$\{\langle \mathsf{start}, [0, 0]\rangle, \langle \mathsf{buy\_envelopes}, [0, 16]\rangle, \langle \mathsf{write\_friends}, [16, 22]\rangle, \langle \mathsf{write\_close\_friends}, [16, 24]\rangle, \langle \mathsf{write\_relatives}, [24, 33]\rangle, \langle \mathsf{send\_letters}, [33, 52]\rangle\} \tag{2}$$

4.2 Modeling Resources and Business Processes Resource Usage

The execution of a business process' tasks requires resources. These resources are not necessarily independent and there could be complex relationships among them. In this paper however, we model them as a set of independent resources since it does not directly affect our work. Executing the tasks of a business process may affect the available resources in the three following ways: (1) the execution could produce some units of some resource types, (2) it may consume some units of some resource types or (3) it could block some units of some resource types from the start until the end of the task execution.

Definition 5 (Resource Model) *A resource model is a set $\mathcal{R}$ where each element of the set represents a type of resource available in the model.*

Definition 6 (Business Resource Usage Model) *Let $\mathcal{R}$ be the resources of a resource model R and $\mathcal{T}$ the set of tasks of a business process B. The usage model of R by B is a set of facts of the following relations:*

- *$C \subseteq \mathcal{T} \times \mathcal{R} \times \mathbb{N}$: a fact $C(t, r, n)$ means that a task t consumes n units of resource r.*

- $P \subseteq \mathcal{T} \times \mathcal{R} \times \mathbb{N}$*: a fact* $P(t, r, n)$ *means that a task* t *produces* n *units of resource* r.
- $B \subseteq \mathcal{T} \times \mathcal{R} \times \mathbb{N}$*: a fact* $B(t, r, n)$ *means that a task* t *blocks* n *units of resource* r.

For example, we identify the set of resources involved in the Christmas Letters business process as follows:

$$\{envelope, parent, child\}$$

Notice that the resources *parent* and *child* are actually roles assigned to some agents. In the present paper we use roles to determine the capabilities of an agent to accomplish the execution of a specific activity.

The usage of these resources by tasks of Christmas Letters business process may be defined as follows:

$$\begin{gathered}\{\ B(buy\_letters, parent, 1),\ P(buy\_letters, envelope, 20),\\ B(write\_relatives, parent, 1),\ C(write\_relatives, envelope, 5),\ B(write\_relatives, child, 1),\\ B(write\_close\_friends, parent, 1),\ C(write\_close\_friends, envelope, 5),\\ B(write\_friends, parent, 1),\ C(write\_friends, envelope, 10),\ B(write\_relatives, child, 1),\\ B(send\_letters, parent, 1)\ \}\end{gathered}$$

In order to study compliance of the execution of a business process with respect to a usage policy, we consider business process executions as follows. An execution is a trace composed of a sequence of states where each state is the representation of the business process execution state and the resource state at a time point. The execution state reflects the tasks being executed at the current time point whereas the resource state reflects the number of units produced, consumed and blocked by the business process up until the current time point. Thus, a state is represented as σ/δ where σ is the execution state and δ the resource state.

Definition 7 (Business Process Resource Usage Executions) *Let* τ *be a valid trace of a business process* B *of length* n *and* M_U *the usage model of some resource model* R *by* B*. The execution trace corresponding to* τ *is a sequence of states* $\sigma_s/\delta_s, \sigma_0/\delta_0, ..., \sigma_i/\delta_i, ..., \sigma_n/\delta_n, \sigma_e/\delta_e$ *where* σ_i *represent the business process execution state and* δ_i *represent the resource usage state. An execution trace is defined as follows:*

- $\sigma_s = \{start\}$, $\sigma_e = \{end\}$ *and for every* $0 \leq i \leq n$,
  - *if* $\langle t, [s, e]\rangle \in \tau$ *and* $s = i$*, then* $start(t) \in \sigma_i$
  - *if* $\langle t, [s, e]\rangle \in \tau$ *and* $e = i$*, then* $end(t) \in \sigma_i$
  - *if* $\langle t, [s, e]\rangle \in \tau$ *and* $s < i < e$*, then* $exec(t) \in \sigma_i$
- $\delta_s = \{c(R_i, 0),\ b(R_i, 0),\ p(R_i, 0)\}$ *for every resource* $R_i \in \mathcal{R}$ *(resources of R).*
- *for* $0 \leq i \leq n$*, every resource state* δ_i *depends on the previous resource state* δ_{i-1} *and the current execution state* σ_i *as follows:*

  $\delta_i = (\delta_{i-1} \cup E_i^+) \setminus E_i^-$ *where* $E_i^+ =$
  $\{c(r, m + \sum_1^{|N|} N) \,|\, N = \{n \,|\, c(r, m) \in \delta_{i-1},\ start(t) \in \sigma_i,\ C(t, r, n) \in M_U\}\} \cup$
  $\{p(r, m + \sum_1^{|N|} N) \,|\, N = \{n \,|\, p(r, m) \in \delta_{i-1},\ end(t) \in \sigma_i,\ P(t, r, n) \in M_U\}\} \cup$
  $\{b(r, m + \sum_1^{|N|} N - \sum_1^{|L|} L) \,|\, N = \{n \,|\, b(r, m) \in \delta_{i-1},\ start(t) \in \sigma_i,\ B(t, r, n) \in M_U\},$
  $L = \{n \,|\, b(r, m) \in \delta_{i-1},\ end(t) \in \sigma_i,\ B(t, r, n) \in M_U\}\}$
  $E_i^- =$
  $\{c(r, m) \,|\, c(r, m) \in \delta_{i-1},\ start(t) \in \sigma_i,\ C(t, r, n) \in M_U\} \cup$
  $\{p(r, m) \,|\, p(r, m) \in \delta_{i-1},\ end(t) \in \sigma_i,\ P(t, r, n) \in M_U\} \cup$
  $\{b(r, m) \,|\, (b(r, m) \in \delta_{i-1},\ start(t) \in \sigma_i,\ B(t, r, n) \in M_U)$ *or*
  $(b(r, m) \in \delta_{i-1},\ end(t) \in \sigma_i,\ B(t, r, n) \in M_U)\}$

The executions of B *are the set of execution traces corresponding to its set of valid traces.*

In the definition above, the resource state is updated according to which tasks start/end at the current state. The function E_i^+ determines the new set of updated facts that should be inserted in δ_i whereas E_i^- identifies the set of facts that should be removed from δ_{i-1}. For example, the following is a simplified representation of a part of the execution corresponding to the trace (3) to illustrate the evolution of the execution and resource states in an execution trace.

$$\{start\}/\{..., b(child, 0), p(envelope, 0), c(envelope, 0), b(parent, 0), ...\},$$
$$\{start(buy\_envelopes)\}/\{..., b(child, 0), p(envelope, 0), c(envelope, 0), b(parent, 1), ...\}, ...,$$
$$\{end(buy\_envelopes)\}/\{..., b(child, 0), p(envelope, 20), c(envelope, 0), b(parent, 1), ...\},$$
$$\{start(write\_friends), start(write\_close\_friends)\}/$$
$$\{..., b(child, 1), p(envelope, 20), c(envelope, 15), b(parent, 2), ...\}, ...,$$
$$\{end\}/\{..., b(child, 0), p(envelope, 20), c(envelope, 20), b(parent, 0), ...\}.$$

In an execution trace, we assume that the consumption of a resource occurs at the start of the task and that the production of resources occurs at the end of a task. Although these can be extreme assumptions, they are the most reasonable given our interest in the study of business process compliance with resource usage policies since these represent the worst cases with respect to resource usage.

Definition 8 (Formula Satisfaction) *Let op be an operator from the following set* $\{<, \leq, =, \geq, >\}$*, let* X *represent symbols from the set* $\{c, p, b\}$ *and* Y *represent symbols from the set* $\{start, end, exec\}$*. The truth of a formula* f *at a state* σ_i/δ_i *of an execution trace* τ*, denoted* $\sigma_i/\delta_i \models_\tau f$*, is defined as follows:*

- $\sigma_i/\delta_i \models_\tau \top$ *and* $\sigma_i/\delta_i \not\models_\tau \bot$.
- $\sigma_i/\delta_i \models_\tau X(r, v_1)$ *iff* $X(r, v_1) \in \delta_i$, *otherwise if* $X(r, v_1) \notin \delta_i$, *then* $\sigma_i/\delta_i \models_\tau \neg X(r, v_1)$.
- $\sigma_i/\delta_i \models_\tau Y(t)$ *iff* $Y(t) \in \sigma_i$, *otherwise if* $Y(t) \notin \sigma_i$, *then* $\sigma_i/\delta_i \models_\tau \neg Y(t)$.
- $\sigma_i/\delta_i \models_\tau X(r, op, v_1)$ *iff* $\sigma_i/\delta_i \models_\tau X(r, v_2)$ *and* $v_2\, op\, v_1$ *is true.*
- *The truth of complex formulas, i.e. the conjunction, disjunction, etc are defined as usual.*

4.3 Usage Policies & Compliance

We use a subset of Process Compliance Language (PCL) [2] to specify the usage policies. In general, a PCL rule has the following form: $\Gamma \Rightarrow O$, where Γ represents the set of premises of the rule and O the obligation enforced when the premises are satisfied. Let F be a propositional literal, PCL distinguishes two types of obligations: achievement and maintenance. An achievement obligation, written $O^a(F)$, is satisfied by an execution trace if it contains a state where F is verified. A maintenance obligation, written $O^m(F)$, is satisfied by an execution trace if each of its states verifies F.

The semantics of PCL are defined with respect to the satisfaction of propositional literals in the states of execution traces. To reuse PCL and keep its semantics, we simply substitute the satisfaction of propositional literals over execution traces by formula satisfaction as in Definition 8. For example, consider Example 1, we specify some policies on the usage of resources using maintenance and achievement obligations as follows.

- $\top \Rightarrow O^m(b(\mathsf{parent}, \leq, 2))$
- $\top \Rightarrow O^a(p(\mathsf{envelope}, \geq, 20))$

The first policy specifies that the amount of parents blocked at any moment should be less or equal to two. The second rule specifies that more than twenty envelopes should be produced during the execution of the business process. Notice that we used $\top$ as the set of premises to represent that these policies are active for the whole duration of the business process.

Compliance The compliance of a business process with a usage policy is defined as follows: A business process is said to be fully compliant with a usage policy if none of its execution traces violates any of the obligations specified in the usage policy, it is said to be partially compliant if at least one (but not all) of its executions violates at least one of the obligations in the usage policy. The analysis of business process compliance with a usage policy allows to identify the risks of failures due to non-compliance with resource usage requirements.

5 Conclusion

In this paper, we study the modelling of resources and their usage requirements in timed BPMs. Our extension allows a high level specification of resources, how they are used by the tasks of a business process and the specification of constraints over resource usage in the form of well-structured usage policies. The main advantage of the approach is that it enables the validation of business processes at design-time and to ensure that a business model is consistent with the resource usage requirements specified by the company.

This work can be extended in several ways. For example, we may investigate ways to improve the design of a business process to guarantee satisfaction of usage constraints or, reciprocally, the relaxation of usage constraints to make a business process compliant with the resource usage policy. We may also extend the work by allowing more sophisticated resource models and resource usage policies. One further direction of research is the study of runtime optimizations of resource allocation and scheduling and the improvement of the computational properties of compliance verification.

References

[1] Carlo Combi, Matteo Gozzi, Roberto Posenato, and Giuseppe Pozzi. Conceptual modeling of flexible temporal workflows. *ACM Trans. Auton. Adapt. Syst.*, 7(2):19:1–19:29, July 2012.

[2] Guido Governatori and Antonino Rotolo. Norm compliance in business process modeling. In *Proceedings of the 4th International Web Rule Symposium: Research Based and Industry Focused (RuleML 2010)*, volume 6403 of *LNCS*, pages 194–209. Springer, 2010.

[3] Bartek Kiepuszewski, Arthur H. M. ter Hofstede, and Christoph Bussler. On structured workflow modelling. In CAiSE '00, pages 431–445, London, UK, 2000. Springer-Verlag.

[4] Barbara Staudt. Lerner, Anoop George. Ninan, Leon J. Osterweil, and Rodion M. Podorozhny. Modeling and managing resource utilization for process, workflow and activity coordination. Technical report, University of Massachusetts, Amherst, MA, USA, 2000.

[5] Chun Ouyang, Moe Thandar Wynn, Colin Fidge, Arthur H.M. ter Hofstede, and Jan-Christian Kuhr. Modelling complex resource requirements in business process management systems. In Michael Rosemann, Peter Green, and Fiona Rohde, editors, *21st Australasian Conference on Information Systems : Defining and Establishing a High Impact Discipline (ACIS 2010)*, Queensland University of Technology, Brisbane, December 2010. ACIS.

[6] Artem Polyvyanyy, Luciano García-Bañuelos, and Marlon Dumas. Structuring acyclic process models. *Inf. Syst.*, 37(6):518–538, September 2012.

[7] Nick Russell, WilM.P. Aalst, Arthur H.M. ter Hofstede, and David Edmond. Workflow resource patterns: Identification, representation and tool support. In Oscar Pastor and Joo Falco e Cunha, editors, *Advanced Information Systems Engineering*, volume 3520 of *Lecture Notes in Computer Science*, pages 216–232. Springer Berlin Heidelberg, 2005.

[8] Shazia Sadiq and Guido Governatori. Managing regulatory compliance in business processes. In Jan van Brocke and Michael Rosemann, editors, *Handbook of Business Process Management*, volume 2, chapter 8, pages 157–173. Springer, Berlin, 2010.