A Configurable Resource Allocation for Multi-tenant Process Development in the Cloud

Emna Hachicha[1(✉)], Nour Assy[2], Walid Gaaloul[1], and Jan Mendling[3]

[1] Telecom SudParis, UMR 5157 Samovar, Université Paris-Saclay, Evry, France
{emna.hachicha,walid.gaaloul}@telecom-sudparis.eu
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
n.assy@tue.nl
[3] Vienna University of Economics and Business Administration, Vienna, Austria
jan.mendling@wu.ac.at

Abstract. Cloud computing has become an important infrastructure for outsourcing service-based business processes in a *multi-tenancy* way. Configurable process models enable the sharing of a reference process among different tenants that can be customized according to specific needs. While concepts for specifying the control flow of such processes are well understood, there is a lack of support for cloud-specific resource configuration where *different allocation alternatives need to be explicitly defined*. In this paper, we address this research gap by extending configurable process models with the required *configurable cloud resource allocation*. Our proposal allows different tenants to customize the selection of the needed resources taking into account two important properties *elasticity* and *shareability*. Our prototypical implementation demonstrates the feasibility and the results of our experiments highlight the effectiveness of our approach.

1 Introduction

Motivated by the need of adopting agile, flexible and cost-effective business solutions, enterprises are looking for available business processes outside of their organizations to quickly adapt to new business requirements and also reduce process development and maintenance costs. Cloud Computing is recently gaining momentum due to its capability of outsourcing service-based business processes based on a scalable pay-per-use model. According to the National Institute of Standards and Technology (NIST), Cloud Computing is a model that enables providers sharing their computing resources (e.g., networks, servers, storage, applications, and services) and users accessing them in an ubiquitous, convenient and on-demand way with a minimal management effort [1]. In such a multi-tenant environment, using *configurable process models* [2] allows a cloud business process provider to deliver a customizable process that can be configured by different tenants according to their specific needs [3].

Different approaches for configurable process modeling have been proposed so far, mainly with a focus on configuring the control flow [4]. Even though the

S. Nurcan et al. (Eds.): CAiSE 2016, LNCS 9694, pp. 558–574, 2016.
DOI: 10.1007/978-3-319-39696-5_34

concept of configurable process models is highly complementary to cloud computing, there has been hardly any uptake in that area. The problem is apparently that specifics of cloud computing, specifically in how resources can be configured and integrated, are hardly considered in configurable process modeling. The few proposals on extending configuration to resources [5–7] do not cover required cloud concepts such as elasticity or multi-tenancy and focus on human resources and their dependencies [8–10].

In this paper, we address this research gap by proposing process configuration concepts for cloud computing. More specifically, we define a novel approach for modeling configurable processes with *configurable cloud resource allocation operators* that allow to explicitly model resource allocation alternatives in multi-tenant process models including elasticity and shareability. Our concepts have been prototypically implemented in the Signavio editor. We evaluate our approach using experiments, which demonstrate its effectiveness.

The paper is structured as follows. Section 2 motivates the problem with a real-world case of a Telco operator and identifies requirements. Section 3 defines our approach for configurable resource allocation. Section 4 describes our implementation and the evaluation results. Finally, Sect. 6 concludes the paper and presents an outlook on future work.

2 Preliminaries and Motivation

In this section, we describe the example from one of our industry partners. Then, we revisit essential concepts from configurable process modeling and cloud computing to identify requirements that are not yet addressed by prior research.

2.1 Motivation

Our research is motivated by a real business of the Telco operator Orange, one of our industry partners. In order to consolidate its expertise in service supervision processes, Orange affiliates share the configurable process in Fig. 1 in a common infrastructure[1]. According to its specific needs, each affiliate configures the process by taking into account the countries legislation and internal regulations. For instance, suppose that an affiliate A does not have access to the resource test management functionalities (the subprocess starting with the activity a_4) and does not have the right to neither perform manual tasks (activity a_6) nor trouble ticket escalation (activity a_{14}). Therefore, it configures the process in Fig. 1 to exclude these functionalities, resulting in a variant as illustrated in Fig. 2.

Since configurable process modeling approaches do not support the resource allocation in multi-tenant cloud environments, the affiliate defines the required resources for its derived variant in an *ad-hoc manner*. For example, for the derived variant in Fig. 2, the activity a_1 needs a *network resource* to communicate with a virtual machine via virtual networking. The network type is manual

[1] For understandability and confidentiality issues, an abstract and simplified version of the configurable process is shown in Fig. 1.

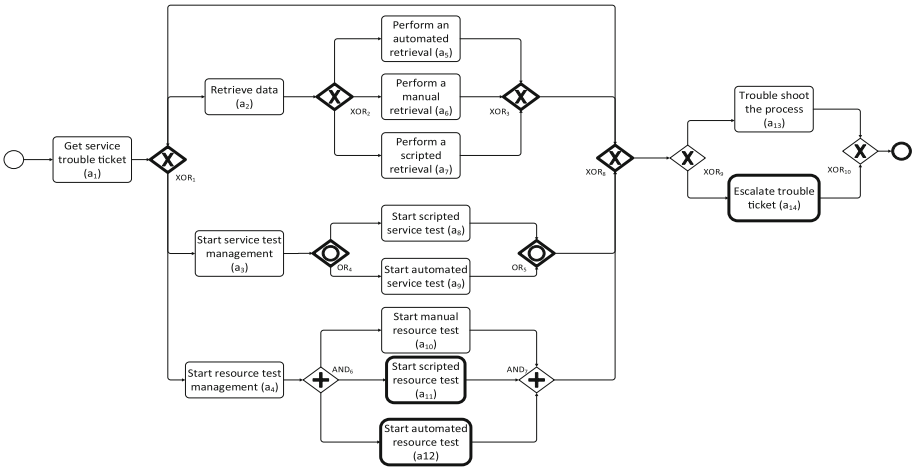

Fig. 1. A configurable service supervision process

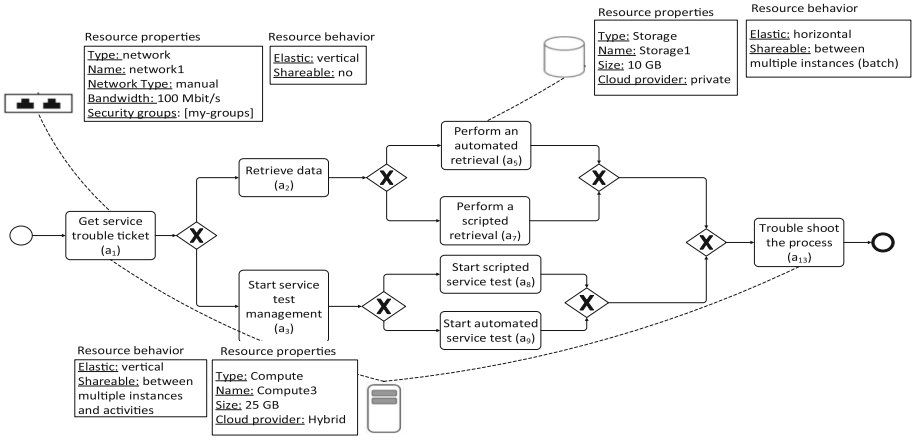

Fig. 2. Variant 1: A process variant derived from the configurable process in Fig. 1 and its allocated cloud resources

with a bandwidth of 100 Mbit/s and which is accessible for a specific security group. These parameters are identified in the "Resource properties" label in Fig. 2. Furthermore, the activity needs an elastic network resource (vertical elasticity), that for security issues is not shared with other activities or instances. These parameters are specified in the "Resource behavior" label.

Suppose that another affiliates B configures the process as shown in Fig. 3 including its required resources. Activity a_1 needs an elastic network resource (horizontally, vertically or both according to the run-time requirements). The network is dynamic with a bandwidth of 100 Gbit/s in order to support the workload from different variants' instances. We notice that the allocated resources for

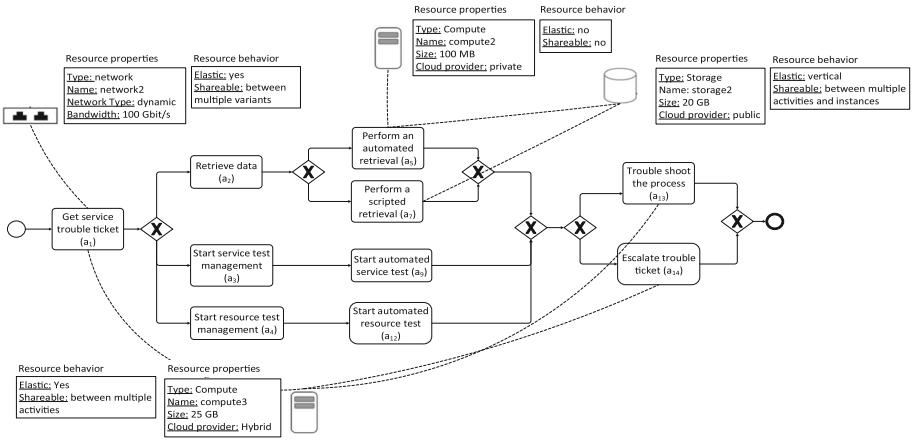

Fig. 3. Variant 2: A process variant derived from the configurable process in Fig. 1 and its allocated cloud resources

the remaining activities are similar to those allocated in the *variant 1* in Fig. 2 but with some variations.

This example shows that multi-tenant business processes do not only share commonalities between their executed tasks, but also between their allocated resources. In fact, different tenants allocate similar resources that slightly differ according to the resource properties and behavior. Up until now, these allocation parameters are hard-coded in an ad-hoc manner which is certainly undesirable in such a multi-tenant environment. Therefore, there is a need for a process configuration support at the cloud resource allocation level and which shifts the cloud resource allocation parameters from the tenant side (at the process variant level) to the cloud process provider side (at the configurable process level).

2.2 Configurable Process Models

The process in Fig. 1 has been modeled with the configurable Business Process Model and Notation (c-BPMN). A configurable process models such as in c-BPMN contain configurable elements whose configuration decision is made at design-time [2]. The configurable elements are graphically modeled with a thick line. In case of c-BPMN, *activities* and *gateways* can be configurable. A configurable activity can be included (i.e. ON) or excluded (i.e. OFF) from the process model. For example, in Fig. 1, the activity a_{11} is configurable. It can be configured either to ON in order to keep it in the process or to OFF in order to exclude it from the process. A configurable gateway has a generic behavior which is restricted by configuration. It can be configured by (1) changing its type while preserving its behavior and/or (2) restricting its incoming (respectively outgoing) branches in case of a join (respectively split) [2]. For example, the configurable OR (OR^c) can be configured to any gateway type while a configurable AND (AND^c) can be only

configured to an AND. We denote by $c \sqsubseteq c^c$ iff the behavior of c is subsumed by that of c^c. For instance, $AND \sqsubseteq OR^c$, $Seq \sqsubseteq XOR^c$ etc.

Once the configuration choices are selected for all configurable elements, algorithms such as the one presented in [2] can be used to derive the specific variant by removing the nodes and edges that have been excluded. For example, the process variant in Fig. 2 is derived as a result of the following configurations:

- XOR_1 is configured to an XOR with the two outgoing branches starting with a_2 and a_3;
- XOR_2 is configured to an XOR with the two outgoing branches starting with a_5 and a_7;
- OR_4 is configured to an XOR with the same outgoing branches;
- XOR_8 is configured to an XOR and is the join of the split XOR_1.

Various configurable modeling languages with comparable capabilities as c-BPMN have been proposed [2,4,11–13]. These works have been focused on the control flow perspective. The general benefits of integrating cloud and BPM have been stressed by different authors [14,15]. If configurable process models have the potential to be an efficient solution for modeling multi-tenant business processes [3], they need to integrate the resource perspective.

2.3 Resource Perspective in Cloud-Based Business Processes

Not only for cloud-based business processes, but also for BPM in general, the research on resources has been scarce. Specific topics of investigation in this area are human resource allocation [9,10] and scheduling [16–18]. The workflow resource patterns [19,20] are often used as a benchmark for corresponding modeling concepts such as [8]. Some works consider cloud characteristics explicitly: S. Schulte et al. in [21,22] develop a platform that allow Business Process Management Systems (BPMS) to manage resource elasticity. L. Pufahl et al. in [23] handle batch activities and allow its flexible adjustments at run-time.

Relevant for configurable processes in the cloud are the following characteristics. The cloud offers three main types of resources at the Infrastructure-as-a-Service (IaaS) model which can be spread into virtual machines (VMs). They consist of *compute*, *network*, and *storage* resources. *Compute* resources are a collection of Physical Machines (PMs) where each contains one or more processors, memory, network interface and local I/O [24]. These PMs require interconnection with a high bandwidth network using *network* resources. Last, the *storage* resources provide persistent storage services where each service have varying levels of data consistency and reliability. Two main properties are specified to the Cloud resources: *elasticity*, and *shareability*. First, the Cloud infrastructure provides two types of elasticity, vertical and horizontal, in order to account for the run-time workload. The *vertical elasticity* is the possibility to scale up and down by adding or removing resources to an existing activity in order to increase its capacity. The *horizontal elasticity* is the possibility of adding or removing

instances of activities with their consumed resources. Second, the resource shareability represents one of the important features in cloud environments. According to security, availability and scalability issues in the process, an allocated resource may or may not be shareable between multiple activities, between multiple instances of the same activity or both. A resource shared between multiple activities is referred to as *shareable* and can be consumed by more than one activity instance at the same time within the same process instance. A resource shared between multiple instances of the same activity is referred to as *batch* and can be utilized by multiple instances of the same activity within multiple process instances. An *hybrid* resource is shareable and batch. These cloud resources can be specified using the RDF-based Cloud business Process Ontology (CloudPrO) [25], which extends in turn the Business Process Modeling Ontology (BPMO) [26]. Its properties are grounded in the cloud computing API Open Cloud Computing Interface (OCCI) [27], allowing statements such as resources of type *compute* contain *speed* which corresponds to the frequency of CPU Clock in gigahertz.

Recently, works in the area of configurable process modeling have been proposed towards such a configuration of the resource perspective [5–7]. In [5], La Rosa et al. propose the configurable integrated EPC (C-iEPC) with features for capturing resource, data and physical objects. Configuration of these elements is achieved using configurable connectors borrowed from the control flow perspective to model the variable allocation of resources. Their focus is, however, on human resources and there is no direct support for cloud resources including *resource sharing* and *resource elasticity*. Moreover, A. Kumar and W. Yao in [6] propose an approach for configurable business processes that integrates resource and data needs using process templates and business rules. Resources in cloud environment are not covered in their approach and flexible resources selection is not addressed. In [7], A. Hallerbach et al. extend the *Pro*cess *v*ariants by *op*tions (Provop) framework to adequately model and manage large collections of process variants. Concepts such as resource allocation and resource selection are not considered. Table 1 summarizes these approaches and relates them to properties that are important in a cloud setting. We observe that resource variability, cloud features, and allocation are only partially covered or not at all. In the following, we aim to fill these gaps by the definition of our novel approach.

3 A Configurable Cloud Resource Allocation

In this section, we present our *configurable cloud resource allocation approach* for multi-tenant business processes development. As mentioned before, cloud resources' allocation takes into account two main parameters: (1) the desired resources and their properties and (2) the desired resource behaviour (i.e. shareability and elasticity). Therefore, we identify three main operators related to the configuration of the resource properties and behavior: (i) configurable resource assignment operator denoted as A^c (Sect. 3.1), (ii) configurable resource elasticity operator denoted as E^c (Sect. 3.2) and (ii) configurable resource

Table 1. Evaluation of previous approaches

Approaches	Criteria			
	Control-flow variability	Resource variability	Cloud resources & features	Resource allocation
[2]	+	−	−	+
[7]	+	−	−	−
[5]	+	+	−	+
[6]	+	−	−	+
Our approach	+	+	+	+

sharing/batching operator denoted as $(S/B)^c$ (Sect. 3.3). Then, an excerpt of a configurable process model with the configurable resource operators is depicted in Fig. 4 and explained in the following.

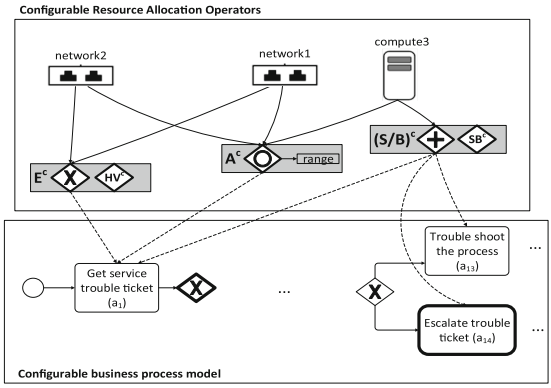

Fig. 4. Configurable resource allocation operators

3.1 Configurable Resource Assignment Operator

The configurable resource assignment operator A^c allows the modeling of *a variable number of resources allocated to a specific activity*. For instance, in our running examples in Figs. 2 and 3, the activity a_1 needs either (i) a network resource "network1" and a compute resource "compute3" or (ii) a network resource "network2" and a compute resource "compute3". Therefore, through A^c, we model a design-time choice in the configurable process that allows the tenants to select one of the available options. To do so, we define two main parameters for A^c: (i) a configurable type and (ii) a range (see A^c in Fig. 4).

The configurable type can be either a configurable OR (OR^c), a configurable AND (AND^c) or a configurable XOR (XOR^c). These connectors have the same behavior as the configurable control flow connectors. They are configured in the

same way as the configurable connectors of the control-flow perspective (see Sect. 2.2). In our example in Fig. 4, the activity a_1 is connected to the cloud resources "network2", "network1" and "compute3" through an OR^c. A tenant may configure the OR^c to an XOR associated to "network2" and "network1" in order to specify that either "network2" or "network1" can be allocated to a_1 while "compute3" is not needed. The allocation decision between "network1" and "network2" is therefore left to the run-time depending on the environment requirements, availability of the resources, etc.

The second operator parameter (i.e. range) imposes an additional constraint on the configuration choice. It is specified by the cloud process provider as a *configuration guideline* for the tenants. A range specifies the minimal and maximal number of the resources that are recommended to be allocated from each resource type ($range_C$ for compute, $range_N$ for network and $range_S$ for storage). For instance, a cloud process provider recommends that at least one compute and one network resources are allocated to the activity a_1. This corresponds to set the minimum of $range_C$ and $range_N$ to 1. By default, the range minimum is set to 0 and the range maximum is set to the total number of the resources from a specific type that are available for the activity. The configuration of the connectors in the configurable type should respect this constraints. For example, having the minimum $min_{range_C} = 1$ and $min_{range_N} = 1$, the aforementioned configuration of the OR^c to an XOR associated to the resources "network1" and "network2" is not valid.

Table 2 summarizes the configurable resource assignment parameters and their configuration constraints. The configurable type follows the configurable connectors from the control flow perspective. Its configuration constraints are the same as described in Sect. 2.2. Each of the range parameters has a minimum min (set by default to 0) and a maximum max. We denote by $|R_C|$, $|R_N|$ and $|R_S|$ the number of compute, network and storage resources respectively provided for a specific activity.

Table 2. Configurable assignment parameters and configuration constraints

Parameters		Configuration constraints
Configurable type	OR^c	Follow the description in Sect. 2.2
	AND^c	
	XOR^c	
Range	$range_C$	$min = 0$, $max = \|R_C\|$
	$range_N$	$min = 0$, $max = \|R_N\|$
	$range_S$	$min = 0$, $max = \|R_S\|$

For instance, in order to derive the resources allocated to the activity a_1 in the process variant in Fig. 2, the configurable resource assignment operator in the process in Fig. 4 is configured as following:

– the configurable type OR^c is configured to an AND associated to the resources $network_1$ and $compute_3$;
– This configuration does not violate the range that is assumed defined by the cloud process provider as follows: $range_C$ $(min = 1, max = 2)$; $range_N$ $(min = 1, max = 1)$; $range_S$ $(min = 0, max = 0)$.

The resource assignment operator is the main operator in the configurable resource allocation modeling. It allows to define the pool of resources that may be allocated to the process activities. Once it is specified, the configurable sharing/batching (see Sect. 3.3) and the configurable elasticity (see Sect. 3.2) operators can be used to model the behavior of the identified resources.

3.2 Configurable Resource Elasticity Operator

During resource allocation, an organization may have different requirements regarding the anticipation of its activities workload, and thus may request different elasticity configurations. For instance, in a specific organization, an activity may require a network resource of at least 100 Mbit/s but may go to 600 Mbit/s during pick hours. At allocation time, a network resource of size 100 Mbit/s which can scale up to a 600 Mbit/s by vertical elasticity is selected. In a second organization, the same activity requires a network resource of at least 100 Mbit/s but may go to a maximum of 150 Mbit/s. The organization requests an horizontal elasticity that adds multiple activities' instances to acquire a 150 Mbit/s.

In order to model such variability at the elasticity level, our proposed *configurable resource elasticity* operator E^c takes into account two parameters' configurations: (i) the set of resources to be elastic and (ii) the way they scale up and down (i.e. elasticity type). Table 3 summarizes the configurable resource elasticity parameters and configuration constraints. Similarly to the *configurable resource assignment* operator, the first parameter (configurable type) can be either an OR^c, XOR^c or AND^c and is used to model the number of resources to be elastic. For instance in our example in Fig. 4, either "network1" or "network2" can be elastic (they are connected through an XOR^c). An organization may configure the XOR^c to a "sequence" associated to "network2" in order to specify that only "network2" can be elastic. The second parameter (configurable elasticity type) specifies the elasticity behavior. Four elasticity types: (i) H (i.e. horizontal), (ii) V (i.e. vertical), (iii) HV (i.e. hybrid) and (iv) HV^c (i.e. configurable hybrid) are defined from which only HV^c is configurable and can be configured to H, V or HV.

Besides the configuration constraints in Table 3, additional *configuration guidelines* can be specified by the cloud process provider regarding the configuration of the elasticity type. These guidelines assist the tenants for selecting the right configuration of the configurable HV^c. They are derived according to the maximal capacity that is provided by the cloud provider during the scale up (vertical or horizontal elasticity) which is specified in the Service Level Agreement (SLA) [28]. In case a resource has a configurable resource elasticity type HV^c, two parameters C_H and C_V are specified in the SLA which correspond to

Table 3. Configurable elasticity parameters and configuration constraints

Parameters		Configuration constraints
Configurable type	OR^c	Follow the description in Sect. 2.2
	AND^c	
	XOR^c	
Configurable elasticity type	HV^c	H, V, HV
	V	-
	H	-
	HV	-

the maximal capacities provided by the cloud provider if the configurations H and V are selected respectively. We denote by C_a the maximal capacity required by an activity "a" for a tenant specific process variant. The configuration guidelines of the configurable HV^c type are defined in Eq. (1).

$$HV^c = \begin{cases} H & if\ C_a \leq C_H\ \wedge\ C_H = min(C_H, C_V) \\ V & if\ C_a \leq C_V\ \wedge\ C_V = min(C_H, C_V) \\ HV & if\ (C_a > C_H\ \wedge\ C_a > C_V)\ \wedge\ (C_a \leq C_V + C_H) \end{cases} \quad (1)$$

where $min(C_H, C_V)$ returns the minimal capacity. The configuration H is recommended to the tenant in case (1) the maximal capacity required by its activity is less than or equal to the maximal capacity provided by the cloud provider in the horizontal elasticity and (2) the capacity of the horizontal elasticity is less than that of the vertical elasticity. The configuration V is recommended in case the same aforementioned conditions are valid for C_V. The configuration HV is recommended in case C_a is greater than C_V and C_H but is less than or equal to their sum. For example, suppose that a tenant specifies that the activity a_1 in the process in Fig. 4 requires a maximal capacity of 100 *uc* (i.e. $C_a = 100$ *unit-of-capacity* where *unit-of-capacity* can be a storage, compute or network related units). The cloud provider specifies that a maximal capacity of 200 *uc* can be provided for the vertical elasticity (i.e. $C_V = 200\,uc$) and a maximal capacity of 150 *uc* can be provided for the horizontal elasticity (i.e. $C_H = 150\,uc$). Since, $C_a \leq C_H$ and $C_a \leq C_V$ then H and V are potential configurations. However, as C_H is the minimal ensured capacity, the configuration H is recommended.

3.3 Configurable Resource Sharing/Batching Operator

As different tenants sharing the configurable process may have different requirements, the shareability of a resource should account for variability. For instance, in our running examples in Figs. 2 and 3, the resource "compute3" is shareable between multiple instances of two activities in the first process (a_1 and a_{13}) (i.e. it is shared and batch) while it is shared between three activities in the second process (a_1, a_{13} and a_{14}). Therefore, we define the *configurable resource*

sharing operator denoted as $(S/B)^c$ which allows to model the variability according to (i) the number of instances/activities that can share the corresponding resource and (ii) the way the activities share this resource (i.e. in a shareable, batch or hybrid manner) (see $(S/B)^c$ operator in Fig. 4).

Table 4. Configurable sharing/batching parameters and configuration constraints

Parameters		Configuration constraints
Configurable type	OR^c	Follow the description in Sect. 2.2
	AND^c	
	XOR^c	
Configurable shareability type	SB^c	S, B, SB
	S	-
	B	-
	SB	-

Table 4 summarizes the configurable resource sharing/batching parameters and their configuration constraints. The first parameter (configurable type) is similar to the configurable type in the configurable resource assignment and configurable elasticity operators. It can be either an OR^c, AND^c or XOR^c and allows to model the behaviour of the resource shareability. Referring to our example in Fig. 4, an AND^c is used to connect the resource "compute3" to the activities a_1, a_{13} and a_{14}. Since an AND^c can be only configured to an AND with possible restricted branches, one can configure the type by selecting only a subset of the activities to share the corresponding resource. For example, the AND^c can be configured to an AND associated to a_1 and a_{13} in order to specify that only a_1 and a_{13} may share "compute3". The second parameter (configurable shareability type) allows to define the way the activities share the resource. Four shareability types: (i) SB (i.e. hybrid), (ii) S (i.e. shareable), (iii) B (i.e. batch) and (iv) SB^c (configurable hybrid) are defined from which only SB^c is configurable and can be configured to S, B or SB.

For instance, in order to derive the shareability configuration of the resource "compute3" in the process variant in Fig. 2, the configurable shareability operator in Fig. 4 is configured as follows:

- AND^c is configured to an AND associated to a_1 and a_{13};
- The configurable shareability type SB^c is configured to a SB (as "compute3" in Fig. 2 is hybrid, i.e. it is shared between multiple instances and activities)

4 Proof of Concept

In this section, we first outline our extension of *Signavio Process Editor*[2] as a proof of concept to validate our approach. Signavio is an open source web application for

[2] Source Code at: https://code.google.com/p/signavio-core-components/.

developing process models in BPMN. Thus, it can support our context of cloud-based processes. More details on our application can be found at http://www-inf.it-sudparis.eu/SIMBAD/tools/Configurable-RA-BPM. We have added two main functionalities to *Signavio* as follows:

1. *Cloud resources' modeling:* We have extended BPMN 2.0 in order to allow for cloud resources' description and integrated it within Signavio (Area 1 in Fig. 5). The user can drag and drop the cloud resources needed for different activities in the process. He can also specify the different attributes and properties as defined in the OCCI standard.
2. *Configurable resource allocation operators:* This functionality allows to allocate the cloud resources to activities using the configurable operators presented in Sect. 3. The three configurable operators (i) assignment for A^c, (ii) elasticity for E^c and (iii) sharing/Batching for $(S/B)^c$ (Area 2 in Fig. 5) can be used to link the process activities to their allocated cloud resources. Their different configurable parameters (e.g. configurable type, configurable elasticity type, etc.) and configuration choices can be also specified (Area 3).

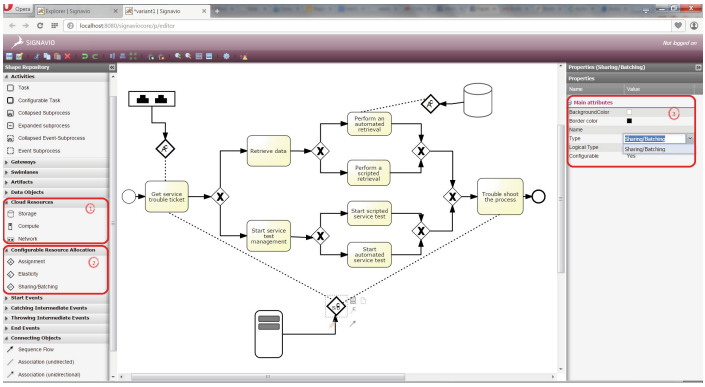

Fig. 5. Application screenshot

5 Experimentation

In order to evaluate the usefulness and effectiveness of our approach, we performed experiments using a real dataset of business processes from Orange, a french telecom industrial partner. Different variants of business processes for VoIP assurance in France are defined and used by Orange. These variants and their allocated resources are manually and separately described. In total, there are 28 variants of the same process using about 30 different resources. Some activities have the same allocated resources in multiple variants, while others have different allocated resources and different needs for shareability and elasticity. In order to consolidate their expertise in telecommunication domain,

Orange experts were interested in constructing one consolidate configurable model that also depicts the different resource allocation strategies.

To construct the configurable model, we proceeded in three different ways. First, using our approach, we designed a configurable process model that depicts the variability both at the control flow and resource flow levels. Second, we modelled a configurable process model with a basic approach that does not consider the variability at the resource level. Thus, whenever an activity has different resource allocation possibilities, it is duplicated in the model in a choice block to express that there exist different resource allocation possibilities and so one should be selected. Third, we designed the same configurable process model using the approach introduced by La Rosa et al. in [5] which is close to ours but does not consider the variability at the shareability and elasticity levels. Therefore, when such a variability occurs (e.g., an activity has the same allocated resources in different variants but with different needs for elasticity and shareability) the same strategy as in the second approach (i.e. duplication of activities) is used. Figure 6 shows three process fragments from Model 1, Model 2 and Model 3. An activity a_1 is assigned to a variable number of resources (network and compute). The compute resource can be shared between a variable number of the activities (instances) a_1 and a_2. According to our approach (represented by Model 1), a_1 is linked to the compute and network resources with a configurable OR via the configurable assignment operator. The compute resource is shared and therefore is linked to a_1 and a_2 with a configurable AND via the configurable Sharing/Batching operator. In the basic approach (represented by Model 2), there are two duplications. The first one is to model the configurable allocation. It is represented in the model by an activity a_1 assigned to the compute resource, and another activity a_1 assigned to the network resource which are connected by a configurable OR. The second one is to model the configurable shareability. It is represented in the model by the activities a_1 and a_2 assigned to the compute resource and connected through a configurable AND that represents the configurable shareability choice. In the approach of La Rosa et al. (represented by Model 3), there is only one duplication to model the configurable shraeability. The configurable allocation is supported and can be modelled as in our approach.

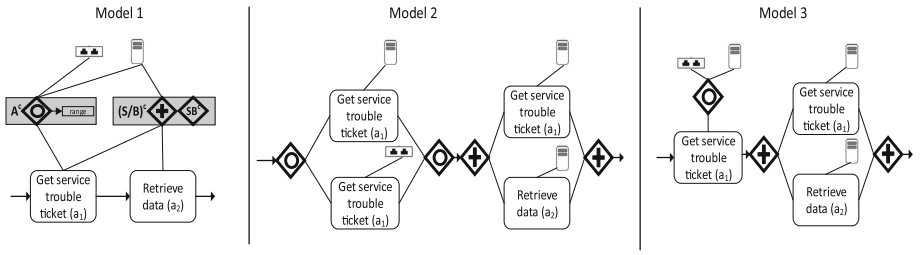

Fig. 6. Fragments of configurable processes of the three models

Thereafter, we assessed the quality of the three models in terms of their structural complexity. We computed the well known complexity metrics proposed in the literature: CFC (Control Flow Complexity), ACD (Average Connector Degree), CNC (Coefficient of Network Connectivity) and density. The CFC [29] metric evaluates the complexity of the process with respect to the presence of gateways OR, AND and XOR. The ACD [30] metric generates the number of nodes that a connector has as an average. The CNC [31] gives the ratio of edges to nodes. Whereas the density [32] metric relates the number of edges to the number of maximum edges that can exist among nodes.

The above metrics have been proposed to assess the complexity of the control flow perspective in business processes. We also use these metrics to compute the complexity of the resource flow perspective since we are using control-flow like operators (i.e. XOR, OR and AND). The obtained values for the three configurable process models are summarized in Table 5. Model 1 refers to the configurable process model constructed with our approach; Model 2 is the configurable process model constructed with a basic approach that does not take the variability at the resource level; and Model 3 is the configurable process model constructed using the approach in [5]. For Model 1 and Model 3, we separately compute the complexity metrics at the control flow (referred to as $[\text{metric}]_c$) and resource flow (referred to as $[\text{metric}]_r$) perspectives. This is a logical choice since the resource and control-flow perspectives are separately modeled.

The results show that the metrics of Model 1 have noticeably low values compared to values of Model 2. For instance, even by summing the CFC_c (28) and CFC_r (25) of Model 1, the result remains smaller than the CFC_c (128) of Model 2. Hence, separately modelling the control-flow and resource-flow variability decreases the complexity of the model. We also notice that the density $density_c$ (0.02), and $density_r$ (0.04) of Model 1 are greater than the density $density_c$ (0.01) of Model 2. However, as stated in [33], the density metric is negatively correlated with the complexity of the model.

By comparing the metrics' values of Model 1 and 3, we notice that Model 1 has better complexity values for the control-flow while Model 3 has better

Table 5. Structural Complexity metrics for different approaches

Complexity metric		Model 1	Model 2	Model 3
CFC	CFC_c	28	128	39
	CFC_r	25	-	14
ACD	ACD_c	3.30	7.37	5.61
	ACD_r	2.11	-	1.64
CNC	CNC_c	0.56	1.18	0.61
	CNC_r	0.51	-	0.78
Density	$Density_c$	0.02	0.01	0.01
	$Density_r$	0.04	-	0.03

complexity values for the resource-flow. This can be explained by the fact that Model 1 fully supports the resource variability modelling (i.e. allocation, shareability and elasticity). Therefore, we do not need to do duplications in the control flow and hence we obtained better complexity values for the control flow. Whereas, Model 3 is less expressive and only supports the resource variability modelling (i.e. allocation). So, it has better complexity values for the resource flow. Since we did duplications in the control flow to model the variability in the shareability and elasticity, we obtained worst complexity values for the control flow.

6 Conclusion

In this paper, we proposed an approach for configurable cloud resource allocation in multi-tenant business processes. Our aim is to shift the cloud resource allocation from the tenant side to the cloud process provider side for a centralized resource allocation management. Through configuration, different tenants can easily derive their allocated resources. The approach has been described through a real example from an industrial partner and implemented in Signavio process editor. Further, we conduct experiments that validate our proposal.

Some potential threats to validity exist in our study. First, we have been interested in IaaS resources since they are the raw resources on which all others (i.e. PaaS and SaaS resources) are built. Our approach can be easily extended to consider the PaaS and SaaS resources. Second, we have shown the feasibility of our approach through real examples from an industrial partner. The work requests a larger dataset to further evaluate the effectiveness of our approach. Third, the proposed configurable resource operators as well as dependencies among cloud resources should formally be described, which are of a high importance in a multi-tenant environment. We aim, in future work, at extending our previous work [25] so that we define these dependencies in a flexible way so that tenants can customize them depending on their needs. In fact, research on cloud resources' management in BPM still at its beginning stage.

References

1. Mell, P.M., et al.: The nist definition of cloud computing (Technical report) (2011)
2. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. Inf. Syst. **32**(1), 1–23 (2007)
3. Aalst, W.: Business process configuration in the cloud: how to support and analyze multi-tenant processes? In: ECOWS, pp. 3–10. IEEE (2011)
4. Rosa, M.L., van der Aalst, W.M., Dumas, M., Milani, F.P.: Business process variability modeling: a survey (2013)
5. Rosa, M.L., et al.: Configurable multi-perspective business process models. Inf. Syst. **36**, 313–340 (2011)
6. Kumar, A., Yao, W.: Design and management of flexible process variants using templates and rules. Comput. Ind. **63**, 112–130 (2012)

7. Hallerbach, A., et al.: Capturing variability in business process models: the provop approach. J. Softw. Maintenance Evolution Res. Pract. **22**, 519–546 (2010)
8. Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., Ruiz-Cortés, A.: Ralph: a graphical notation for resource assignments in business processes. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 53–68. Springer, Heidelberg (2015)
9. Cabanillas, C., Norta, A., Resinas, M., Mendling, J., Ruiz-Cortés, A.: Towards process-aware cross-organizational human resource management. In: Bider, I., Gaaloul, K., Krogstie, J., Nurcan, S., Proper, H.A., Schmidt, R., Soffer, P. (eds.) BPMDS 2014 and EMMSAD 2014. LNBIP, vol. 175, pp. 79–93. Springer, Heidelberg (2014)
10. Kajan, E., et al.: The network-based business process. IEEE IC **18**, 63–69 (2014)
11. Mietzner, R., Leymann, F.: Generation of BPEL customization processes for SaaS applications from variability descriptors. IEEE Conf. SC **2**, 359–366 (2008)
12. Gottschalk, F., et al.: Configurable workflow models. IJCIS **17**(2), 177 (2008)
13. Ciuksys, D., et al.: Reusing ontological knowledge about business processes in is engineering: process configuration problem. Informatica **18**, 585–602 (2007)
14. Duipmans, E.: Business process management in the cloud: Bpaas (2012)
15. Wang, M., Bandara, K.Y., Pahl, C.: Process as a service. In: SCC, pp. 578–585 (2010)
16. Candra, M.Z.C., Truong, H.-L., Dustdar, S.: Provisioning quality-aware social compute units in the cloud. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 313–327. Springer, Heidelberg (2013)
17. Sengupta, B., Jain, A., Bhattacharya, K., Truong, H.-L., Dustdar, S.: Who do you call? problem resolution through social compute units. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) Service Oriented Computing. LNCS, vol. 7636, pp. 48–62. Springer, Heidelberg (2012)
18. Hoenisch, P., et al.: Workflow scheduling and resource allocation for cloud-based execution of elastic processes. In: IEEE 6th International Conference on SOCA, USA (2013)
19. Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**, 5–51 (2003)
20. 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)
21. Schulte, S., et al.: Costdriven optimization of cloud resource allocation for elastic processes. Int. J. Cloud Comput. **1**(2), 1–14 (2013)
22. Schulte, S., Hoenisch, P., Venugopal, S., Dustdar, S.: Realizing elastic processes with ViePEP. In: Ghose, A., Zhu, H., Yu, Q., Delis, A., Sheng, Q.Z., Perrin, O., Wang, J., Wang, Y. (eds.) ICSOC 2012. LNCS, vol. 7759, pp. 439–442. Springer, Heidelberg (2013)
23. Pufahl, L., Herzberg, N., Meyer, A., Weske, M.: Flexible batch configuration in business processes based on events. In: Franch, X., Ghose, A.K., Lewis, G.A., Bhiri, S. (eds.) ICSOC 2014. LNCS, vol. 8831, pp. 63–78. Springer, Heidelberg (2014)
24. Jennings, B., Stadler, R.: Resource management in clouds: survey and research challenges. J. Netw. Syst. Manag. **23**, 567–619 (2015)
25. Hachicha, E., Gaaloul, W.: Towards resource-aware business process development in the cloud. In: 29th IEEE International Conference AINA, South Korea, pp. 761–768 (2015)

26. Cabral, L., et al.: The business process modeling ontology. In: SBPM, pp. 9–16 (2009)
27. Edmonds, A., et al.: Toward an open cloud standard. IEEE IC **16**, 15–25 (2012)
28. Rady, M.: Parameters for service level agreements generation in cloud computing. In: Castano, S., Vassiliadis, P., Lakshmanan, L.V.S., Lee, M.L. (eds.) ER 2012 Workshops 2012. LNCS, vol. 7518, pp. 13–22. Springer, Heidelberg (2012)
29. Cardoso, J.: Evaluating the process control-flow complexity. In: ICWS (2005)
30. Cardoso, J., Mendling, J., Neumann, G., Reijers, H.A.: A discourse on complexity of process models. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 117–128. Springer, Heidelberg (2006)
31. List, B., et al.: Evaluation of conceptual bp modelling languages. In: SAC (2006)
32. Reijers, H.A., Vanderfeesten, I.T.P.: Cohesion and coupling metrics for workflow process design. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 290–305. Springer, Heidelberg (2004)
33. Vogelaar, J., et al.: Comparing business processes to determine the feasibility of configurable models: a case study. In: BPM Workshops, France, pp. 50–61 (2011)