Configurable IoT-Aware Allocation in Business Processes

Kunal Suri[1,2], Walid Gaaloul[2], and Arnaud Cuccuru[1]

[1] CEA, LIST, Executable Language Engineering and Optimization Laboratory (LIDEO), P.C. 174, Gif-sur-Yvette, France
`{kunal.suri,arnaud.cuccuru}@cea.fr`
[2] TELECOM Sudparis, SAMOVAR UMR 5157, Université Paris-Saclay, France
`walid.gaaloul@telecom-sudparis.eu`

Abstract. Based on specific requirements, various Internet of Things (IoT) devices participate in multiple cross-organizational business processes. However, to achieve the desired business value, these IoT resources must be managed efficiently. Configurable Process Model (CPM) facilitates flexibility and reuse by sharing a family of process variants, which can be customized based on concrete business requirements. The classical approaches to develop CPMs focus mainly on the control-flow perspective, without providing concepts to tackle the complexity involved in the IoT domain. In this paper, we address this research gap by proposing configuration concepts for modeling IoT resource variability, which arises due to specific resource properties and behavior such as Replication and Shareability, at the CPM level. Furthermore, we validate our approach based on the results of our experimentation and demonstrate its feasibility through an implemented prototype.

Keywords: Internet of Things · Business Process Management · Configurable Process Model · Resource Management · Retail Management

1 Introduction

Recent years have witnessed a tremendous growth in the use of interconnected heterogeneous devices. These devices can be broadly classified into *Sensors*, *Actuators* and *Tags* (such as Radio Frequency Identification (RFID)). They enable sensing, actuating (or re-acting) and exchanging or collection of data through a communicating network such as the internet, thus creating the Internet of Things (IoT) [5]. Each of these devices perform a specific task to generate some value for an end user (or system). Further, they are considered as key technology enablers to bring the concept of "Smart Environments" such as smart cities, smart logistics, Industry 4.0, closer to reality. Moreover, in a real-world situation, one or more of such heterogeneous devices need to be orchestrated in a specific sequence to achieve a specific outcome. In fact, many organizations already use Process-Aware Information Systems (PAIS) [1], which efficiently manage and executes (orchestrates) various enterprise services and resources such as human-workforce and systems on the basis of process models, to provide the best return on investment [22]. Thus, it is natural to see these devices allocated to one or more business processes (BPs), which spreads across time, space (geographic location) and organizations, for being orchestrated in a specific order, to achieve some predefined business goals.

To optimally manage processes involving resources, i.e., both human and non-human (devices and systems), these PAIS need to become resource-aware [6] and evolve into *Process- and IoT-Aware Information Systems*. Thus, realizing the importance of IoT resource management in Business Process Management (BPM), there has been a growing trend on research towards integration of IoT and BPM domain [20, 23]. Even though these works focus on integrating IoT concepts at an individual process variant level, they are relevant to facilitate optimal management of IoT resources involved in BPs.

In another side, the rapidly changing business requirements, customer needs or government regulations (in context of smart ecosystems) forces these PAIS to imbibe the traits of flexibility and reuse. In other words, these systems must facilitate the "Principle of Reuse" for modeling and/or (re-)designing the processes by taking into consideration the preexisting knowledge about similar processes and/or best practices existing in an organization, rather than forcing analysts to design processes "from scratch". In order to support the flexibility and reusability for modeling BPs, our work focuses on using Configurable Process Model (**CPM**) [21], which is an active area of research for managing process variability in BPM domain [17]. A CPM consolidates various process variants (multiple process solutions) into one customizable process model via variation points called *configurable elements* (activity or gateways) [11]. In other words, a consolidated customizable model captures a family of process variants. This helps to avoid redundancy and allows improvements efforts made on one variant to benefit other variants. Moreover, the classical approaches to develop a CPM focuses mainly on configuring the *control-flow perspective* [17], without giving much consideration to the resource perspective. Additionally, the limited proposals that do consider the extension of configuration to resources [13–15], are too generic to tackle the complexity and specificity involved in the IoT domain (i.e., IoT specific features (properties), constraints, and deployment strategies). Indeed, even though the concepts of configurable process modeling being highly complementary to IoT, to the best of our knowledge there has not been any uptake in this area.

In this paper, we address this research gap by proposing configuration concepts for modeling IoT resource variability, which arises due to specific resource features, i.e., properties and behavior such as Replication and Shareability, at the CPM level. Concretely, we define a novel approach for developing CPMs with *Configurable IoT Resource Allocation* operators. This allows inclusion of explicit knowledge (options/variability) about various alternatives and constraints that exists for a typical IoT resource based on its behavior. These IoT-aware CPM can be individualized into a process variant via transformations including both, (i) the control-flow perspective, and (ii) IoT resource perspective, to meet a given set of business requirements. Further, we developed a proof of concept tool to illustrate the feasibility of our work and assist the development of conceptual models for configurable IoT-aware processes, intended for communication and analysis purposes. To demonstrate the effectiveness of our approach, we evaluate it on a CPM developed for the Retail industry.

The remainder of this paper is organized as follows: in Section 2, some basic concepts related to CPM and IoT are detailed. In Section 3, a use-case from Retail/Logistics domain is used to motivate our research. The Section 4, details the need for modeling IoT resource perspective in BPs. In Section 5, describes our approach to model configurable IoT-aware allocation. In Section 6, we detail our implemented proof of concept and evaluation results from the experimentation. In Section 7, we describe related work, and in Section 8, we conclude our work and provide a perspective on our future work.

2 Preliminaries

This section presents the preliminaries used in the remainder of this paper. In Section 2.1, we detail concepts related to configurable process modeling, and in Section 2.2, we detail key concepts from IoT domain.

2.1 Configurable Process Modeling

A Configurable Process Model (CPM) is an integrated representation of a family of processes in a given domain [21]. It uses variation points (configurable elements) to capture the differences among the process variants (similar to techniques from Software Product Line Engineering) [17]. It maintains a clear distinction between the *commonalities* (i.e., parts shared by all process variants) and *variability* (i.e., parts specific to certain process variants) in a process family. These modeling techniques allows sharing of knowledge and best-practices, which enables analysts to develop processes based on various guidance and rules (options) provided in these models (at design-time) [17, 21]. In literature, various languages exist for modeling configurable processes such as configurable Event-driven Process Chains (C-EPCs), UML Activity Diagrams (ADs), configurable Business Process Model and Notation (**c-BPMN**) [17].

In our work we use c-BPMN as it is based on extending BPMN, which is the most popular modeling language in both academia and industry [3]. In c-BPMN, the configurable elements, i.e., activities and gateways are modeled with a thick line. These elements can be included, i.e., configured to ON or excluded, i.e., configured to OFF, depending on the specific business requirements. Likewise, a configurable gateway has a generic behavior that is restricted by its configuration. Depending on the type of the gateway, it can be configured by, (i) changing its type while preserving its behavior and/or, (ii) restricting its incoming and outgoing branches [21]. Moreover, after choosing the configurable elements, specific variants can be derived by removing the excluded nodes and edges based on algorithms such as presented in [3, 21]. For instance, Fig. 1 represents a CPM modeled using c-BPMN. In this CPM, the activity $a2$ and $a3$ are configurable, i.e., they can be configured either to ON (to keep it in the model) or to OFF (to exclude it) in the derived process variant. Similarly, the configurable OR (see ORc-2 in Fig. 1) can be configured to any type of gateway (i.e., OR, XOR, AND), while a configurable AND (AND^c) can be only configured to an AND gateway. Using this CPM, a retailer can proficiently diffuse their process expertise and knowledge with their conglomerates. Furthermore, this type of variability management technique tackles only the control-flow perspective, without dealing with the complexity and constraints involved in the resource perspective, especially from the IoT domain.

2.2 Internet of Things (IoT)

IoT comprises of connected devices such as Sensor, Actuators, Tags (e.g. RFID), which supports the creation of an smart (intelligent) environment. This intelligence when applied for making successful inferences offers a huge potential to change everyday life. Additionally, it allows decision makers to have superior transparency and value-added understanding of their complete product lifecycle. However, to efficiently consume and manage deployed IoT resources, there is an evident need to grasp the fundamental concepts in IoT such as topology of network, power usage, bandwidth, intermittent connectivity [5] along with the underlying infrastructure, i.e., Cloud, Fog or Edge computing [18], used for deployment and management of the IoT devices.

Some of these concepts are: (i) *Power Usage*: Devices consume considerable amount of power while transmitting data, particularly over long ranges. (ii) *Bandwidth*: The rate of data transmission depends on the capacity of the network, and parameters such as volume of data (raw or aggregated), number of devices, connectivity (constant stream or intermittent bursts of data), packet size of the networking protocol, to name a few. (iii) *Intermittent Connectivity*: To conserve power and bandwidth, devices connect and transmit data periodically (rather than continuously). However, other situations such as an unreliable network or issues with the quality of service (e.g., interference on a wireless network using a shared spectrum), hamper the connectivity.

Thus, the efficient use of IoT resources calls for inclusion of such information in the process models (at design-time). This will ensure proper usage, deployment and management of IoT resources during the deployment phase.

3 Motivating Example

We motivate our research through a CPM (see Fig. 1), which represents a process family from the Retail/Supply Chain Management domain. We considered this domain as various processes executing in the Retail industry effect the day-to-day life of a large number of people. Moreover, we developed this CPM using c-BPMN by adapting and merging (consolidating) a collection of model decription from the “IoT in Practice:Examples” [9] based on algorithms presented in [4, 16]. The process models in [9] focus on the application of IoT in various processes and were used in the European FP7 project, *Internet of Things Architecture* (IoT-A[3]). Overall, this CPM will assist retailers to share their process knowledge and policies (rules and constraints) in a reusable and customizable manner with their affiliates spread across the globe. The CPM in Fig. 1 represents a family of process variants (control-flow perspective) for monitoring items involved in the Retail industry that fall in two main categories: (i) *fast-moving consumer goods* (FMCG) such as vegetables, cheese, flowers, and (ii) *durable goods* such as electric appliances, cars, clothes. This process can start based on a timer event for enabling periodic monitoring of goods. After the process starts, there are two possibilities represented by two sub-processes interconnected via a configurable *XOR* gateway (*XORc-1*). The sub-process I should be employed to monitor an item from FMCG category, while the sub-process II should be used for durable goods.

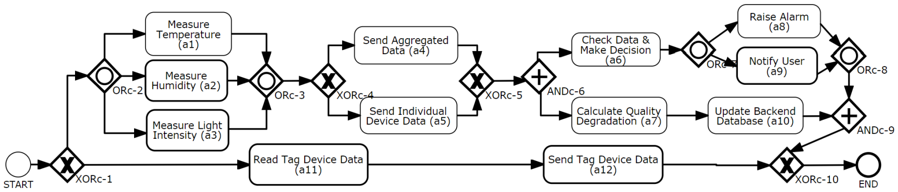

Fig. 1. Configurable Process Model from Retail management domain

To demonstrate the need of including IoT-resource perspective at CPM level, we first describe a process variant (*Variant-1*) derived from the CPM based only on the control-flow perspective. Next, we show how including the IoT resource perspective increases the complexity of this process variant. Let us assume that a French retailer (such as Carrefour) at a location A, decides to individualize the CPM to include only a temperature monitoring step for a perishable item such as *Chinese Orchids* (adapted from [23]). Thus, at the design-time an expert will customize the CPM into a process variant, i.e., Variant-1 represented via Fig. 2. The Variant-1 is configured to include activities $a1$, $a5$, $a6$, $a7$, $a8$, $a10$. Moreover, the derivation (individualization) of a process variant based on the classical control-flow perspective is done by removing the unwanted nodes (detailed in Section 2.1). Nonetheless, for efficient resource management, there is a need to capture explicit knowledge about the IoT resources (i.e., IoT properties, behavior and deployment strategies) in the process models at the variant level (detailed in our previous work [23]). For instance, based on some business needs, the activity $a1$ needs a *digital temperature sensor* having high-accuracy, i.e., Accuracy of ±0.5°C (max) from 0°C to +65°C (e.g., a TMP112[4] sensor from Texas Instruments (TI)). Additionally, during deployment this device will need a network resource, i.e., *Network-01*, which should be long range, consumes

---

[3] IoT-A project: `http://cordis.europa.eu/project/rcn/95713_en.html`

[4] TI's TMP112 - `http://www.ti.com/lit/ds/symlink/tmp112.pdf`

lower power and allows secure data transmission such as Low Power Wide Area Network (LPWAN) based LoRaWAN[5]. Further, this resource can be deployed on a public cloud infrastructure. All these parameters and information depict the IoT specific features, i.e., *Resource Properties*, which should be included in the process models.

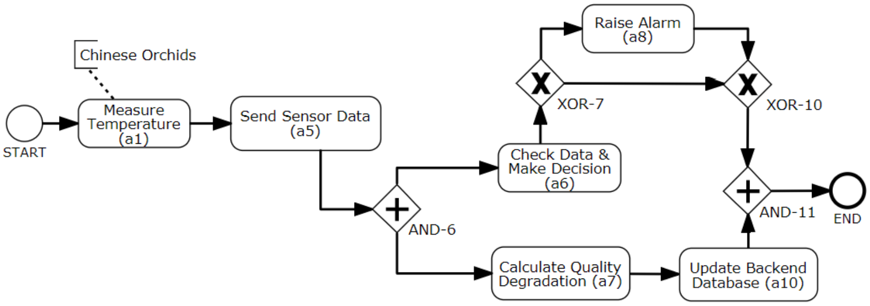

Fig. 2. Process variant 1 derived from Fig. 1 based on control-flow perspective

In Fig. 3, we use the Variant-1 and enrich it with information about the IoT resource features in form of text annotations. Likewise, IoT resource have specific *Resource Behavior* that should be included in the process models. For instance, a device and the network can be *Shareable*, i.e., it shall share its data using publish/subscribe (pub-sub) middleware, e.g., Eclipse Mosquitto[6]. Additionally, the activity $a1$ can be connected to more than one temperature sensor provided they exhibit similar capability, i.e., aggregation of a set of similar physical devices via a logical interface. This results in improvement of availability, fault-tolerance, and helps to achieve higher *Quality of Information (QoI)* [19] (detailed in Section 4). These Resource Behavior are also included as text annotation as observable in the Fig. 3.

Likewise, another Carrefour market (let say at a location B), decides to individualize the CPM (in Fig. 1) into another process variant, i.e., *Variant-2*, having same control-flow as the Variant-1, but different IoT specific requirements. For example, in Variant-2, activity $a1$ requires a low-accuracy digital temperature sensor with Accuracy of $\pm 2°$C (max) from $-40°$C to $+125°$C (e.g., TI's TMP103) and a cellular network resource (*Network-02*). Similarly, there could be another variant, i.e., *Variant-3*, having same control-flow as Variant-1 but requiring a *low-power dust resistant* sensor (e.g., TI's HDC1080). Additionally, this resource can be deployed using both cellular network or LoRA network depending on availability at deployment time. Table 1 illustrates the complexity involved in capturing the IoT resource variability while considering just a single activity ($a1$) from the CPM.

---

[5] `https://www.lora-alliance.org`

[6] `https://mosquitto.org/`

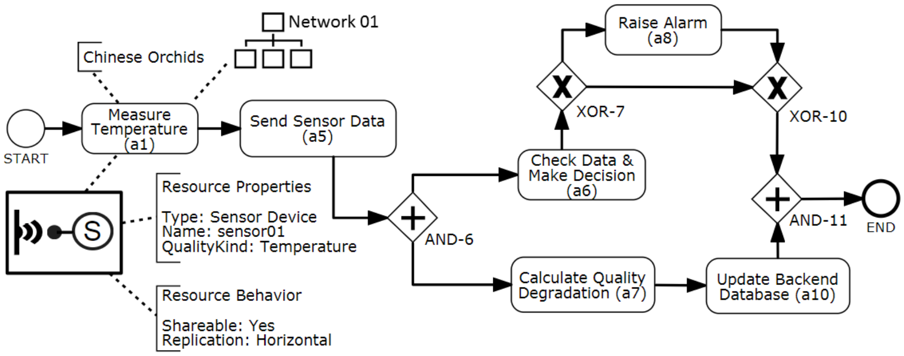

Fig. 3. Process variant 1 in Fig. 2 enriched with IoT resource features

Table 1. IoT resource variability in process variants

Variant	Control-Flow	Resources	Res. Property	Res. Behavior
Variant-1	Derived from CPM	Sensor, Network	High-Accuracy (HA)	Shareable
Variant-2	Same as *Variant-1*	Sensor, Network	Low-Accuracy (LA)	Shareable
Variant-3	Same as *Variant-1*	Sensor, Network	HA & Low-Power	Non-Shareable

These example clearly illustrate that the process variants share commonalities not only at the *structural* and *behavioral* level (i.e., control-flow perspective) but even at the resource level. In practice, various variants have similar requirements for the allocated resources with slight changes such as choice of accuracy, network, capability, deployment strategies, or Shareability (i.e., Resource Behavior). However, not having a configuration support to model this resource variability at CPM level, causes several disadvantages: (i) the allocation parameters are hard-coded at each individual variant level in an ad-hoc manner, (ii) there is no knowledge coming from CPM level, i.e., no guidance (rules or constraints), (iii) variant creation is time-consuming and error-prone, (iv) the process enrichment (and best practices) takes place at the variant level, leading to redundancy and segregation of improvement efforts for each variant, without benefiting others. Thus, we advocate creation of configuration support for IoT resource variability at CPM level, i.e., shifting the IoT allocation parameters from the variant level to the CPM level.

4 IoT Resource Perspective in Business Processes

Even with a growing interest for integration of IoT and BPM domain for developing processes related to smart environments such as Industry 4.0 [24] and smart Retail/Logistics processes [9], overall research on IoT resource perspective in BPM has been scarce. In literature, some recent work have contributed towards modeling, allocation and management of IoT resources involved in BPs [19, 20, 23]. However, they focus on including the IoT features into individual process models (variant level) rather than developing reusable concepts at the CPM level. Besides, there are certain features associated with IoT resource perspective that are significant for incorporation at the CPM level. Broadly speaking, IoT consists of Sensor, Actuator and Tag devices, which can be battery operated or connected to main power supply. These devices need a network with a specific bandwidth, range and latency (detailed in Section 2.2). Thus, the two key features associated with IoT resources are: (i) *Replication* and (ii) *Shareability*. Further, this work assumes that IoT resources consists of a set of IoT devices and a set of network (e.g., Orange IoT networks[7]), where both can be mapped together in a process based on business needs.

Replication has been widely studied for distributed environment because it strongly impacts the following: (i) *Availability*, (ii) *Reliability*, and (iii) *Performance* [10]. Reliability and Availability have also been widely studied in context of data-centric services. For instance, Decandia et al. [8] detail their need for creating highly reliable systems, and discuss the tradeoffs between availability, consistency, cost-effectiveness and performance. Many organizations such as Amazon consider reliability as one of the most significant requirements. This is because a slightest outage can have substantial financial consequences and impacts customer trust [8]. Additionally, each IoT device has a specific *Access Cost* (*AC*) parameter, i.e., device energy consumption cost (*Processing Cost*), communication energy cost (cost for bandwidth, latency, radio range). The AC and Quality of Information (QoI) are interdependent as higher rate of sampling will increase the QoI but will also lead to higher AC [19]. Basically, in context of IoT (both centralized or distributed architecture), it is essential to explicitly detail and model these replication features (properties) to maintain optimal AC, and QoI along with high-availability, reliability and fault-tolerance, especially while dealing with time-critical systems (i.e., systems using real-time data for decision making). In this work, we consider Replication subsumes all four sub-properties, i.e., AC, QoI, Availability, and Reliability.

Likewise, Shareability subsumes two sub-properties, i.e., *Privacy* and *Security* of information, which is highly important in both IoT and BPM domain. These devices capture and transmit data that can contain sensitive or private information such as GPS location, video or audio data. Thus, the processes must be designed keeping data protection policy in mind (e.g. EU GDPR[8]). Based on such policies at both the process and resource level, the analysts can design variants having allocated resources that may or may not be shareable between multiple processes or multiple activities of the same process or even between the multiple instances of the same activity. Overall, this work focuses on modeling and including these IoT resource features at the CPM level based on the approach detailed in Section 5 without going into details about managing the sub-properties.

---

[7] `https://partner.orange.com/orange-iot-networks/`

[8] `https://www.eugdpr.org/`

5 Approach: Configurable IoT-Aware Allocation

In this section, we detail our approach for including IoT resource variability at the CPM level by taking into account two main parameters: (i) resources and their properties, and (ii) resource behavior (i.e., Replication and Shareability). We introduce three main operators: (i) *Configurable IoT Assignment* operator (A^c) (adapted from our previous work [12]) in Section 5.1, (ii) *Configurable IoT Replication* operator (R^c) in Section 5.2, and (iii) *Configurable IoT Shareability* operator (S^c) in Section 5.3.

Fig. 4 represents a process fragment taken from the CPM detailed in Fig. 1, wherein the activity *a1* is allocated with IoT resources using the above mentioned configurable resource allocation operators. This process fragment is used as a running example while detailing the operators in the following sections.

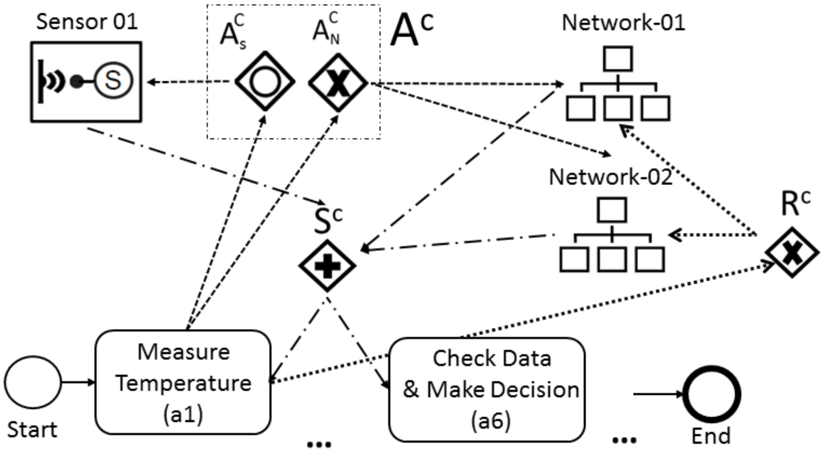

Fig. 4. Fragment from Fig. 1 illustrating configurable IoT allocation operators

5.1 Configurable IoT Assignment Operator

The configurable IoT assignment operator (A^c) allows modeling of various IoT resources allocated to a particular activity. It is the main operator for facilitating the modeling of IoT resource variability at the CPM level. It allows to define a pool of resources and set of guidelines (rules and constraints), which shall be used to derive sound process variants [25] with relevant resources allocated to the process activities. In this work, we consider the IoT resources to be divided in two main groups: (i) the choice of an IoT device (Sensor, Actuator or Tags), and (ii) the choice of a network (based on bandwidth, range, and latency). For instance, the process fragment in Fig. 4, illustrates an activity $a1$ allocated via A^c to one sensor (via OR^c represented by A^c_s) and two network resources (via XOR^c

represented by A^c_n). It represents a resource variability such as, (i) a temperature sensor and a *LoRaWAN* based network resource, or (ii) a temperature sensor and *cellular* (2G or 4G) based network resource. Using A^c, CPM can be configured to a specific variant (based on business needs) with assistance from the parameters and guidelines injected in A^c. Further, the A^c operator consists of the following three parameters (summarized in Table 2).

Configurable Type: This parameter corresponds to one of the configurable gateways, i.e. OR^c, XOR^c, AND^c. The gateways in A^c behave similar to classical configurable gateway (control-flow perceptive) and are configured in the same manner(detailed in Section 2.1). For instance, AND^c is configured to an AND, implying that all devices should be allocated. XOR^c is configured to an XOR, implying that the resource has can be allocated exclusively or cannot be allocated. Whereas OR^c can be configured to AND, OR, or XOR, depicting allocation based on the required features of the IoT resource.

Range: This parameter corresponds to the minimum and maximum number of the resources that can be allocated to an activity, i.e., *rangeD* for IoT device and *rangeN* for network. Let us assume that the activity $a1$ in Fig. 4 has guidelines to include at least one IoT device and one network resources, i.e., $min(rangeD)$=1 and $min(rangeN)$=1. This will correspond to the allocation shown in process variant in Fig. 3. The default setting for minimum range equals 0, while maximum range equals the total number of a specific resource allocated to an activity, represented by $|R_D|$ (device) and $|R_N|$ (network).

Assignment Policies: This parameter corresponds to guidelines specific to IoT resources for assisting analysts to derive semantically correct process variants. It consists of certain default policies along with advanced policies. For instance, (i) an activity should be allocated with an IoT device belonging to only one category, i.e., same activity cannot be allocated to a sensor and to an actuator, (ii) an activity can be allocated to multiple resources (e.g., multiple sensors) of the same type or hybrid type, i.e., having at least one of the needed functionality.

For example, in Fig. 4, the activity $a1$ can be allocated with a temperature sensor or a hybrid temperature-humidity sensor. Fig. 4 represents a process fragment (excerpt from Fig. 1) with an activity $a1$ allocated with one sensors and two network resource. Thus, an analyst can configure to keep one sensor and one network resource by transforming the OR^c to AND, and keeping either Network-01 or Network-02 (as represented in the derived process model in Fig. 3). The AND implies that both the sensor and network are needed. Moreover, such configuration should not violate the range defined for the resources above let us say, *rangeD* (min = 1, max = 3); *rangeN* (min = 1, max = 2).

5.2 Configurable IoT Replication Operator

For every IoT resource allocated in a process, there exists some specific requirements in terms of the QoI, AC, Availability and Reliability, i.e., Resource Behavior (detailed in Section 4). Each of these requirements will generate different process variants that behave in a different way. The Replication operator (R^c) will express these resource behavior requirements in terms of Replication

Table 2. Parameters for configurable IoT assignment operator

Parameters	**Behavior and Constraints**
Configurable Type (AND^c, OR^c, XOR^c)	Same as classical configurable gateways
Range	$0 \leqslant range_D \leqslant \|R_D\|$
	$0 \leqslant range_N \leqslant \|R_N\|$
Assignment Policies	Domain & geography specific constraints

Type. They are *Horizontal Replication* and *Vertical Replication* (inspired from *Elasticity* in Cloud computing), which are defined as follows:

Horizontal Replication (HR): The possibility to allocate and aggregate multiple resources (both device and network) with a constraint that the resources have the same AC and QoI features. Further, the total number of allocated resources should fall in the allowed *Range* (see *Range* in A^c). HR permits the system to have a higher reliability while keeping the AC lower (i.e, energy and other costs). For example, a room having four temperature sensors (all having similar AC and QoI) connected to an activity via a logical interface, allowing one or more to be active at a given time.

Vertical Replication (VR): The possibility to allocate and aggregate multiple resources (both device and network) having different AC and QoI (both higher or lower), within the allocation *Range*. VR allows to maintain high availability and high reliability, without any upper limit on AC (i.e, energy and other costs), especially for critical systems. For example, a room having four temperature sensors, with one simple sensor, one hybrid (temperature-humidity) sensor and two hybrid (temperature-humidity) dust resistant sensor (all having variable AC and QoI) mapped via a logical interface. For modeling such variability, R^c has following three parameters (summarized in Table 3):

Configurable Type: depicting the set of resources that can be replicated. The configurable type can be either an OR^c, XOR^c or AND^c (similar to A^c). For instance, AND^c is configured to an AND, implying that all devices should be replicated. XOR^c is configured to an XOR, implying that the resource has can be replicated exclusively or cannot be replicated.

Configurable Replication Type: depicting the type of replication allowed. R^c can model various resources that can be replicated based on replication type that specifies the replication behavior (RB^c), which can be of two types, i.e., HR and VR. Thus, the RB^c can be configured to one of the HR or VR.

Replication Policies: depicting specific guidelines related to QoI and AC. The replication policy parameter comprises of guidelines, rules and constrained specified by domain expert for configuration of the replication type. These guidelines assist in deriving variants conforming to domain requirements and Service Level Agreements (SLAs).

For instance, the Fig. 4 illustrates that both $Network01$ or $Network02$ can be replicated (lets say with HV), however as only one of them can be configured at a time, thus they are connected via XOR^c.

Table 3. Parameters for configurable IoT replication operator

Parameters	**Behavior and Constraints**
Configurable Type (AND^c, OR^c, XOR^c)	Same as classical configurable gateways
Replication Type (RB^c)	HR, VR
Replication Policies	Access Cost & QoI related constraints

5.3 Configurable IoT Shareability Operator

Some BP activities may share various IoT resources (and their data). These BPs include stakeholders from within the same organization or different organizations. Thus, various constraints related to sharing of the resources and data (based on privacy and security concerns) should account for another layer of variability. For managing this type of variability, we define the configurable IoT Shareability operator, represented as S^c. It permits modeling the variability based on: (i) the way the activities share the IoT resources (and data) within the process, and (ii) the number of process instances or activities that can share the corresponding resource. This operator comprises of the following three parameters (summarized in Table 4).

Configurable Type: it is similar to other configurable IoT operators, i.e., OR^c, AND^c or XOR^c. It allows to model the Shareability feature. *Shareability Type*: the Shareability type ST^c comprises of two sub-types: (i) Shareable (S), and (ii) Non-Sharable (NS). Thus, the ST^c can be configure to one of the them. *Shareability Policies*: the policies contains guidelines and rules specific to a domain or geographic needs.

For instance, to derive a process such as variant-01 (see Fig. 3) having Shareability, the configurable IoT Shareability operator in Fig. 4 must be configured as follows: (i) S^c operator (having AND^c gateway) associated with $a1$, $a6$, along with the sensor and network resources, and (ii) ST^c is configured to a S, to depict data Shareability between multiple activities. Further, it is important to note that the Replication and Shareability operators are semantically dependent on assignment operator. This is because a device needs to be first assigned before it can exhibit Replication or Shareability behavior. This makes the formal verification for resource allocation an essential work, however it is out of the scope of this paper.

Table 4. Parameters for configurable IoT Shareability operator

Parameters	**Behavior and Constraints**
Configurable Type (AND^c, OR^c, XOR^c)	Same as classical configurable gateways
Configurable Shareability Type (ST^c)	S, NS
Shareability Policies	Privacy & Security constraints

6 Validation

In this section, we illustrate the feasibility of our approach by implementing a proof of concept as detailed in Section 6.1. In Section 6.2, we detail the experimentation performed on datasets developed using three different approaches on the same CPM. The experimentation result illustrates that our approach reduces the complexity involved in modeling IoT specific features at the CPM level.

6.1 Proof of Concept

We implemented a proof of concept by extending the Signavio[9] process editor (open-source version). Signavio provides a web-based graphical environment for developing process models in BPMN (serializable as BPMN.xml). This extension supports the development of configurable IoT-aware BPs, detailed in our university web-page[10]. As illustrated in Fig. 5, our prototype supports the following functionality for managing process variability at design-time:

IoT Resource Modeling: We extended the BPMN 2.0 semantics to include concepts from IoT domain, i.e, Sensor, Actuator, RFID and the Network, along with their properties (based on IoT-A framework). These specifications integrated within the Signavio extension allows users to drag and drop IoT resources during process modeling.

Configurable IoT Allocation Operators: These operators assist modeling and integrating the IoT resource perspective at the CPM level by allocating configurable operators to activities based on the approach presented in Section 5. These three configurable IoT resource operators, i.e., Assignment (A^c), Replication (R^c) and Shareability (S^c) are used to link the process activities to their allocated IoT resources (e.g., Fig. 4). These operators consist of various configurable parameters such as configurable type, configurable replication type, and policies, which will assist the users during development of process variants.

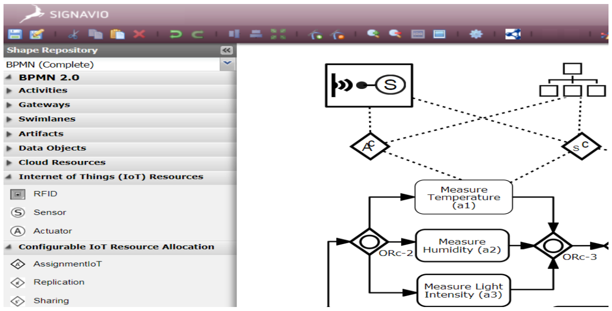

Fig. 5. Screenshot illustrating the implemented proof of concept

---

[9] https://code.google.com/archive/p/signavio-core-components/source

[10] `http://www-inf.it-sudparis.eu/SIMBAD/tools/ConfigurableIoTBPM`

6.2 Experimentation

In this section, we illustrate the effectiveness of our approach by performing experiments on a CPM from the Retail domain (see Fig.1). This CPM was developed by integrating process variants[11] adapted from [9]. Our work consolidates both, the control-flow perspective and the IoT resource perspective, along with their allocation strategies for developing configurable IoT-aware process models. Thus, to compare our approach with the current state-of-the-art, we developed the same *IoT-aware* CPM using three different approaches, detailed as follows:

First, we develop an IoT-aware CPM using the classical control-flow perspective, which does not consider any variability at the resource level. To do so, an activity is duplicated in the model in a choice block to express the existence of different resource allocation possibilities. This CPM (see Process Fragment-1 in Fig. 6) represents the IoT resource variability, and can be individualized based on business requirements. However, it leads to an increase in process complexity. Second, we develop an IoT-aware CPM based on the approach from La Rosa et al. [15]. Their approach supports basic resource configuration without considering the complex IoT features such as resource behavior. Thus, the activities need to be duplicated to depict these features. For instance, an activity may have different Shareability requirements in different process variants, which is depicted by duplicating activities and including these features (see Process Fragment-2 in Fig. 6), leading to increase in model complexity. Third, we use our approach to develop the IoT-aware CPM, which represents the variability considering both the control-flow perspective and the IoT resource allocation (see Process Fragment-03 in Fig. 6). Fig. 6 represent three process fragments taken from three separate IoT-aware CPM developed as explained above. For simplicity reasons, the fragments in Fig. 6 represent an activity *a1*, assigned to a *Sensor-01* and a *Network-01*, wherein both resources are *Shareable*.

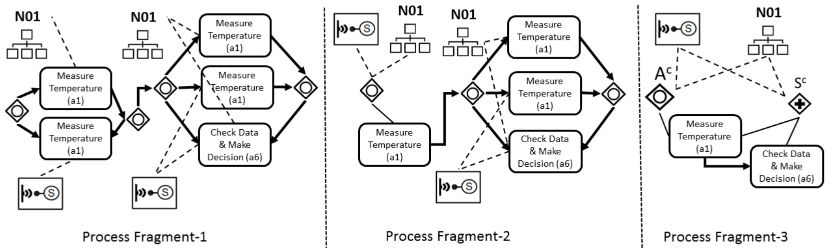

Fig. 6. Process fragments illustrating three different approaches

In the classical approach represented via Process Fragment-01, activity *a1* has been duplicated multiple time to represent the configurable resource assignment concept. One *a1* is linked to the network resource *N01* and another *a1* to

---

[11] `https://github.com/kunalsuri/process-models`

Sensor01, both $a1$ are connected via a configurable OR. Likewise, to represent the concept of configurable resource shareability between *a1* and *a6*, the activities *a1* is duplicated and linked to the IoT resources and connected to *a6* via a configurable OR. Likewise, following the approach in [15] (see Process Fragment-2), the allocation of two IoT resources is done using a OR^c. However, to represent the concept of configurable resource shareability, *a1* is duplicated and connected with *a6*. Further, based on our approach (see Process Fragment-3), the concept of configurable resource allocation is depicted by linking the IoT resources with activity *a1* via a configurable IoT assignment operator (A^c). While the resources Shareability is represented by linking the resources to activities $a1$ and $a6$ via a configurable IoT shareability operator (S^c).

To evaluate the quality of these three IoT-aware CPMs, we calculate and compare a well-known complexity metric, i.e., Control-Flow Complexity (CFC) [7]. The CFC_c evaluates the process complexity in terms of the classical gateways and is used to better understand and examine process models before their actual implementation [7]. As the resource allocation operators are based on the control-flow gateways, we also apply this metric to them. However, we distinguish it by calling it CFC_r. Further, we developed three datasets, i.e., one for each approach, wherein each dataset has five IoT-aware CPM (using the same CPM from Fig. 1), developed by allocating IoT resource features (with varying complexity).

The results are summarized in Table 5. The results illustrate that our approach has lower aggregated CFC values than other two approaches. As compared to [15], our resource-flow complexity is higher since we need to duplicate the control-flows (subsuming the resource-flow operators (see Section 5.3)) for assignment and behavior, both falling under CFC_r, to model the resource behavior such as Shareability.

Table 5. Complexity metrics comparing different approaches

Complexity Metric	**DataSet 1** **Classical Approach**	**DataSet 2** **La Rosa Approach**	**DataSet 3** **Our Approach**
Average CFC_c	37	30	**15**
Average CFC_r	N.A.	6	16
Total CFC	37	36	**31**

7 Related Work

In literature, various existing work on CPM focus mostly on the control-flow perspective [17, 21] . Though, some works such as [13–15] extended the configuration to include the resource perspective [17], but then again they are not sufficient to handle the complexity of IoT domain. La Rosa et al. [15] proposed the configurable integrated EPC (C-iEPC), which included features for capturing resource, data and physical objects via configurable connectors (based on control-flow pespective) to model the variable allocation of resources. However, they focus on human resources and generic non-human resources without any support for IoT specific features. Kumar et al. [14] proposed an approach based on templates and rules for creating configurable processes allowing some integration of both resources and data. However, their approach does not cover flexible resources selection and is not suitable for IoT resources. Hallerbach et al. [13] extend the process variants by options (Provop framework) to model and manage large collections of process variants without going in depth for considering concepts related to resource selection and allocation. Overall, these works consider the resources perspective in a generic way without considering the intricacy of the IoT domain, which leads to creation of multiple IoT-specific process variants. Recently, in literature there has been a growing focus on including resource perspective to PAIS, i.e., developing Process- and Resource-Aware Information Systems (PRAIS) [6]. Thus, some researchers have contributed on including Cloud computing concepts at CPM level [12] and its simulation for finding optimal variants [2]. However, there has been no uptake on integrating IoT resource perspective to CPMs.

8 Conclusion and Future Work

In this paper, we propose configuration concepts for handling IoT resource variability to be integrated at a Configurable Process Model (CPM) level. Concretely, we defined a set of configurable IoT-aware allocation operators, which will enable the inclusion of explicit information (options/variability) about various alternatives and constraints for IoT resources based on their features (properties) and behavior such as Replication and Shareability. These IoT-aware CPMs can be individualized into a specific process variant via transformations that includes both, (i) the control-flow perspective, and (ii) IoT resource perspective, to meet a given set of business requirements. Our research is motivated and illustrated through a CPM from the Retail management domain. Furthermore, we implement a proof of concept using Signavio process editor, and validate our proposal through our experimentation results.

As a perspective, we plan to work with a larger dataset for further evaluation of our approach. We intend to formalize the operators (and its constraints). We also plan to do implementation of the configuration step and the formal verification to obtain a sound variant.

References

1. Van der Aalst, W.M.: Process-aware information systems: Lessons to be learned from process mining. T. on Petri Nets and Other Models of Conc. II **2**, 1–26 (2009)
2. Ahmed-Nacer, M., Suri, K., Sellami, M., Gaaloul, W.: Simulation of configurable resource allocation for cloud-based business processes. In: Services Computing (SCC), 2017 IEEE International Conference on. pp. 305–313. IEEE (2017)
3. Assy, N., Chan, N.N., Gaaloul, W.: An automated approach for assisting the design of configurable process models. IEEE T. on Service Comp. **8**(6), 874–888 (2015)

4. Assy, N., Gaaloul, W., Defude, B.: Mining configurable process fragments for business process design. In: Intl. Conf. on Design Science Research in Inf. Syst. pp. 209–224. Springer (2014)
5. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks **54**(15), 2787–2805 (2010)
6. Cabanillas, C.: Process-and resource-aware information systems. In: Enterprise Dist. Obj. Comp. Conf. (EDOC), 2016 IEEE 20th Intl. pp. 1–10. IEEE (2016)
7. Cardoso, J.: Evaluating the process control-flow complexity measure. In: Web Services, 2005. ICWS 2005. Proc. 2005 IEEE Intl. Conf. on. IEEE (2005)
8. DeCandia, G., Hastorun, D., et al.: Dynamo: amazon's highly available key-value store. In: ACM SIGOPS Operating Sys. Review. vol. 41, pp. 205–220. ACM (2007)
9. Fiedler, M., Meissner, S.: Iot in practice: Examples: Iot in logistics and health. In: Enabling Things to Talk, pp. 27–36. Springer (2013)
10. Goel, S., Buyya, R.: Data replication strategies in wide-area distributed systems. In: Ent. Serv. Comp.:From Concept to Deployment, pp. 211–241. IGI Global (2007)
11. Gottschalk, F., Van der Aalst, W.M., Jansen-Vullers, M.H.: Configurable process modelsa foundational approach. In: Reference Modeling, pp. 59–77. Springer (2007)
12. Hachicha, E., Assy, N., Gaaloul, W., Mendling, J.: A configurable resource allocation for multi-tenant process development in the cloud. In: Adv. Inf. Systems Eng. (CAiSE). pp. 558–574. Springer (2016)
13. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the provop approach. J. of Soft.: Evol. and Process **22**(6-7), 519–546 (2010)
14. Kumar, A., Yao, W.: Design and management of flexible process variants using templates and rules. Computers in Industry **63**(2), 112–130 (2012)
15. La Rosa, M., Dumas, M., Ter Hofstede, A.H., Mendling, J.: Configurable multi-perspective business process models. Information Systems **36**(2), 313–340 (2011)
16. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Business process model merging: An approach to business process consolidation. ACM T. on Soft. Eng. and Methodology (TOSEM) **22**(2), 11 (2013)
17. La Rosa, M., Van Der Aalst, W.M., Dumas, M., Milani, F.P.: Business process variability modeling: A survey. ACM Computing Surveys **50**(1), 2 (2017)
18. Mahmud, R., Kotagiri, R., Buyya, R.: Fog computing: A taxonomy, survey and future directions. In: Internet of Everything, pp. 103–130. Springer (2018)
19. Martinho, R., Domingos, D.: Quality of information and access cost of iot resources in bpmn processes. Procedia Technology **16**, 737–744 (2014)
20. Meyer, S., Ruppen, A., Hilty, L.: The things of the internet of things in bpmn. In: Adv. Inf. Systems Eng. (CAiSE). pp. 285–297. Springer (2015)
21. Rosemann, M., Van der Aalst, W.M.: A configurable reference modelling language. Information Systems **32**(1), 1–23 (2007)
22. Schrepfer, M., Kunze, M., Obst, G., Siegeris, J.: Why are process variants important in process monitoring? the case of zalando se. In: Business Process Management Cases, pp. 431–448. Springer (2018)
23. Suri, K., Gaaloul, W., Cuccuru, A., Gerard, S.: Semantic framework for internet of things-aware business process development. In: Enabling Tech.: Infra. for Collab. Ent. (WETICE), 2017 IEEE 26th Intl. Conf. on. pp. 214–219. IEEE (2017)
24. Suri, K., Cadavid, J., et al.: Modeling business motivation and underlying processes for rami 4.0-aligned cyber-physical production systems. In: Emerging Tech. And Fact. Automation (ETFA), 2017 22nd IEEE Intl. Conf. on. IEEE (2017)
25. Van Der Aalst, W.M., Van Hee, K.M., Ter Hofstede, A.H., Sidorova, N., Verbeek, H., Voorhoeve, M., Wynn, M.T.: Soundness of workflow nets: classification, decidability, and analysis. Formal Aspects of Computing **23**(3), 333–363 (2011)