On the Support of Multi-perspective Process Models Variability for Smart Environments

Aitor Murguzur[1], Xabier de Carlos[1], Salvador Trujillo[1] and Goiuria Sagardui[2]
[1]*Information Technologies Area, IK4-Ikerlan Research Center, Arrasate-Mondragón, Spain*
[2]*Software and Web Engineering Group, Mondragon University, Arrasate-Mondragón, Spain*

Keywords: Variability Modeling, Process Configuration, Smart Buildings.

Abstract: Cloud service-based applications are to be adapted to serve multiple platforms and stakeholders. Atop of such services, Smart Green Buildings are fostering a plethora of processes within their sustainability life-cycle. This introduces a number of challenges, as how to support multiple perspectives of domain-specific variability and how to deal with large collections of related process variants. To tackle this, there is a need to handle multi-perspective variability for processes. This paper introduces an approach to manage multi-perspective process variability by means of a meta-model and a modeling methodology, representing separately people and things variability perspectives in smart environments. Initial experimental results are also described, which indicate encouraging results for managing highly complex variability models.

1 INTRODUCTION

The world is getting more smarter in this age, i.e., more instrumented, interconnected and lead to better decision making devices (e.g. the Nest[1] thermostat or its Europe's answer Tado[2]) currently are arising and pointing out a number of challenges for smart environments (such as Smart Green Buildings (SGBs), smart cities or smart healthcare). Central to this vision, agile service-based applications require to effectively be deployed into a variety of devices and different implementation platforms (Gurgen et al., 2013). Such services are globally dispersed under the control of different service providers and ready to be used whenever needed. Atop such services, a number of process models may exist which orchestrate domain-specific activities related to aforementioned services and deal with a huge data streams for analytics. In such process-intensive environments, such as SGBs, the management of domain-specific variability and inherent process models variability is becoming really challenging (Castro et al., 2012).

In an SGB, several processes are defined within its sustainability governance life-cycle (i.e. installation and commissioning processes, configuration processes, operation processes and surveying processes) to look at two major concerns, such as, improving user experience and maintaining building sustainability (Wang et al., 2012). For attainment of those purposes, these all processes may be related to different multiple dimensions - people and things respectively.

On the one hand, stakeholders (representing related roles and users) and associated operation contracts, including a number of agreement policies among stakeholders, (e.g. premium maintenance contract) will need different process models, or simply processes, to "coordinated (parallel and/or serial) set of activities that are connected in order to achieve a common goal"[3] in each life-cycle phase (e.g. energy consumption of an SGB). This gives rise to a large collection of process variants which could have common and individual parts. Since living with this complexity is not an agile practice, process variability have to be managed from the *people perspective*.

On the other hand, a similar case is found in terms of things. Beyond people perspective process variability, a plethora of process variants could also be created (e.g. energy efficiency of an SGB) for a diversity of monitored assets (e.g. building facilities - floor, room, etc., and equipment - chiller, boiler, etc.) and building types (e.g. residential, commercial). Therefore, the variability management becomes here also the quid of *things perspective* complexity.

[1]http://nest.com
[2]http://tado.com
[3]Workflow Management Coalition: `http://wfmc.org`

Murguzur A., De Carlos X., Trujillo S. and Sagardui G..
On the Support of Multi-perspective Process Models Variability for Smart Environments.
DOI: 10.5220/0004758205490554
In *Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development* (MODELSWARD-2014), pages 549-554
ISBN: 978-989-758-007-9

Regarding both perspectives, we have noticed that existing initiatives concerning process variability have not been adequate for supporting large process variants collections in process-intensive smart environments. This is due to they do not allow for multi-perspective support, but also because there is a lack of understanding and isolating both perspectives (people and things) in process models variability.

In this paper, our aim is to support process variability research by dissecting multi-perspective and variability modeling aspects. To this end, we contribute (i) a meta-model of multi-perspective process variability by a comprehensive analysis of multi-perspective properties, and (ii) a modeling methodology for representing separately people and things variability perspectives in smart environments.

The remainder of this paper is organized as follows: in Section 2, we summarize related research. Section 3 details our proposed meta-model and methodology for managing multi-perspective process variability. The proof-of-concept prototype that we have built is described in Section 4, and concluding remarks and future work towards real world implementation can be found in Section 5.

2 RELATED WORK

To best of our knowledge the work presented in this article is the first tackling the multi-perspective and the impact on process variability management in smart environments. On a more general level, this work is related to other fields of research that can be divided into two main blocks: (i) multi-perspective in process variability and (ii) multi-perspective in Software Product Line Engineering (SPLE).

2.1 Multi-perspective in Process Variability

Process variability has been studied for almost a decade including a number of approaches which utilize different process **variability realization techniques** to configure and adapt process variants, such as *single element-based process variant construction* technique (Schnieders and Puhlmann, 2006; Rosa et al., 2011), which gathers all variability aspects in a single configurable process model, representing the whole process family (i.e. the collection of process variants), and *fragment-based re-use process variant construction* technique (Hallerbach et al., 2010; Baresi et al., 2012), which employs different models for variability representation, modeling variability separately from a base model. Although process variability itself has been investigated, just a few studies have addressed the issue of multi-perspective.

For instance, the C-iEPC was presented as an extension of the Configurable Event-driven Process Chain (C-EPC) notation with the notions of roles (i.e. for resource perspective) and objects (i.e. for data perspective), enabling multi-perspective of business processes (Rosa et al., 2011). Orthogonally, the multi-perspective variants approach (Meerkamm, 2012) involves multiple types of domain stakeholders in the process configuration, offering diverse levels of abstraction and reducing the number of decisions each domain user has to make during the process modeling and/or execution. Personalized views may also be created in large process models by hiding and abstracting irrelevant information for separated user groups as well (Kolb and Reichert, 2013).

Consequently, we may recognize two different meanings for **multi-perspective** in process variability, as indicated in Table 1: the one related to *process elements*, i.e., the partition of elements found in process models themselves and the other to *process stakeholders*, i.e., the individual requirements of various users and groups for which particular process variants exist. Given definitions, with respect to the mentioned work, the main novelties of our approach are two-fold: our approach clearly separates domain-specific processes and domain-specific variability perspectives representation through a fragment-based re-use approach. Secondly, we are not just focusing on stakeholders, otherwise we consider both

Table 1: Multi-perspectives in process variability.

Variability technique/Multi-perspective	**Process elements**	**Process stakeholders**
Single element-based construction	(Rosa et al., 2011)	(Meerkamm, 2012)(Kolb and Reichert, 2013)
Fragment-based re-use construction		(Our approach)

Table 2: Multi-perspective in SPLE.

Variability modeling/Multi-perspective	**Product-line stakeholders**
Single variability model	(Czarnecki et al., 2004; Classen et al., 2009; Mendonça et al., 2008) (García-Galán et al., 2013) (Our approach)
Multiple variability models	(Rosenmüller et al., 2011; Schroeter et al., 2012) (Our approach)

people and things perspectives enabling full-support for multi-perspective variability.

2.2 Multi-perspective in SPLE

In the SPLE, several approaches have also been proposed as a means to facilitate configuration of the software product line by applying disparate variability modeling techniques to deal with multiple dimensions and configurations of a product-line.

Essentially, the **variability modeling** represents the commonality and variability of a family of systems in the domain and relationships between them by employing a particular variability modeling language (e.g. feature models, decision models, or the OMG initiative Common Variability Language (CVL)[4] (Berger et al., 2013). Such variability is often described with a feature model using, for instance, (i) a *single variability model* to handle all domain variability and leverage multi-dimensional separation of concerns in feature-based configurations (Hubaux et al., 2011), or (ii) *multiple variability models* to separate multiple dimensions such as, domain variability and implementation variability. Although the former is about visibility, the latter is easier to handle than a large single model which could avoid high complex configurations and thus enhance manageability (Rosenmüller et al., 2011; Schroeter et al., 2012).

Most of the variability configurations techniques are concerned with multi-user support for a particular variability modeling language. Feature models staged-configuration was proposed with the aim of providing step-by-step configurations in which each user refines previous user decisions (Czarnecki et al., 2004; Classen et al., 2009). Multi-user collaboration product-line configuration improves staged-configuration by allowing users to make decisions in a concrete space (i.e. limiting the space for each user) at the same time without conflicts (Mendonça et al., 2008). General approaches, without space limitation and enabling user satisfaction, have also been studied in the literature (García-Galán et al., 2013). Therefore, **multi-perspective** in SPLE is related to multiple stakeholders' configurations support, providing them more accurate views.

Bearing in mind the aforementioned work (see Table 2), our approach differs from traditional approaches adopting a single or multiple models for tackling multi-perspective configurations in process variability. We provide an approach by means of a meta-model and three modeling strategies for supporting multi-perspective in process variability.

[4]http://variabilitymodeling.org

3 MULTI-PERSPECTIVE IN PROCESS VARIABILITY

3.1 Formalizing Multi-perspective Process Variability

As it was remarked, the succeeding conceptualization pursues a *fragment-based re-use variant construction approach* which employs different models for multi-perspective process variability representation in smart environments (e.g. SGBs), modeling variability separately from a base model. Especially for that purpose, we make use of a slightly altered Base-Variation-Resolution (BVR) method (including additional concepts as described below) from the Software Product Line Engineering (SPLE) (Bayer et al., 2006) to define a total of three models for supporting multi-perspective process models variability: a **base model** (BM) - representing commonalities, a **variation model** (VM) - representing individualities, and a **resolution model** (RM) - representing process variant configurations, as illustrated in Figure 1.

3.1.1 Base Model

The *base model* describes, employing a business process modeling language (e.g. Business Process Model and Notation (BPMN), Business Process Execution Language (BPEL)), the commonalities shared by all process variants and placeholder activities that are subjected to vary. Placeholder activities (i.e. variation points) identify concrete parts in a BM in which variant binding occurs. A *base model element* stands for any kind of model asset in a BM (i.e. a common or placeholder activity). This relationship has a zero-to-one cardinality, since not all model elements (e.g. common activities among related process models) are affected with variability (i.e. some activities are not placed as variation points).

A *variation point* represents a configurable activity element, in which a configuration may be selected, after which it is no longer possible to change the selection. This selection is called binding, and the stage in the life-cycle at which binding occurs is called binding time. We distinguish three kind of variation points - static, partial and dynamic:

- *Static* is resolved at build-time, resolving variation points by selecting suitable alternatives from a set of available ones. Therefore, process variants are fully specified at configuration-time and deployed into a process engine for later enactment.
- *Partial* extracts a pattern with a small number of fragments, simplifying the spectrum of fragment

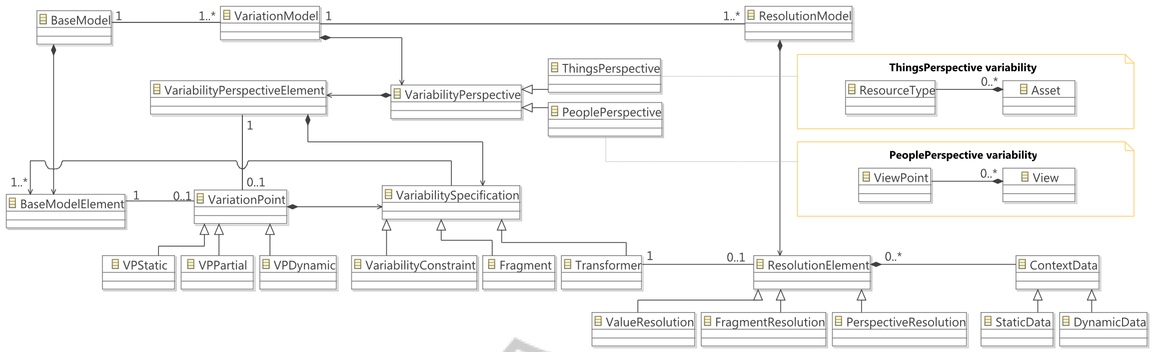

Figure 1: The Ecore meta-model for multi-perspective process variability.

choices for runtime, helpful when a certain threshold of variation points or fragments are reached.

- *Dynamic* is bound at runtime since we cannot pre-specified which of the applicable fragments will be suitable considering monitoring dynamic context data (e.g. floor internal temperature), as in Dynamic Software Product Lines (DSPLs) (Baresi et al., 2012).

3.1.2 Variation Model

The *variation model* gathers all the individualities introduced by each process model variant characterizing individualities and constraints, i.e., relationships among particular domain features and variables.

The *variability perspective* describes a pre-configuration of a reference BM, i.e., it takes the aggregation of a particular variability perspective (e.g. one stakeholder view) under consideration. Beyond people's perspective, process variability could also be resolved interpreting context data from embedded and smart devices (e.g. Heating, Ventilation and Air Conditioning (HVAC), etc.). Hence, two variability perspectives are interwoven - *people perspective* and *things perspective* respectively, see Figure 1:

- **People Perspective:** Following the terminology of (Schroeter et al., 2012) two main items are outlined regarding people perspective. On one hand, a *view* shows only process variants that could be configured for a specific stakeholder or role. According to the ISO/IEC/IEEE 42010:2011, Systems and software engineering standard[5], "a view is a representation of the whole system from the perspective of a related set of concerns". On the other hand, a *viewpoint* represents a set of views, each restricting a reference BM to a subset of configuration, e.g. a particular operation contract relevant for a group of stakeholders.

- **Things Perspective:** In terms of things, process variants could also be created for the same purpose (e.g. energy consumption or chiller optimization) but considering resource types and asset specifications. A *resource type* describes each different source of information from which context data is gathered (e.g. residential, commercial buildings). Similarly, an *asset* represents something that is built, installed, or established to serve a particular purpose which might include a number of facilities and monitored equipment to perform different actions.

The *variability specification* represents the actual variability of a variation point which is directly related to a *variability perspective element* (i.e. view, viewpoint, resource type or asset specification). Just like in domain feature modeling, it stands for a family of systems in the domain and relationships between them (i.e. mandatory, optional, alternative, and or relations), their corresponding domain and constraints. Here, we identify three different types of elements which are fundamental in our approach:

- *Variability Constraints* represent constrains on valid resolutions (i.e. requires, excludes and cross-tree constraints). Constraint sub-model could include constrains on valid process variant configurations that may be defined at three levels: complex feature-feature, feature-variable and variable-variable cross-tree relationships.

- *Fragment* describes a single realization option for each variation point placed in the BM specification. Depending on variability specification including people and things perspectives, and considering context data, applicable fragments are selected and executed to resolve multi-perspective process models variability.

- *Transformers* have concrete transformations associated to a variability specification. When context values are bound to a concrete transformer,

[5] http://www.iso-architecture.org/ieee-1471/

it executes the corresponding resolution. In other words, it is the responsible for deciding which resolutions should be adopted by the current data values (value, fragment or perspective).

3.1.3 Resolution Model

A *resolution model* represents a binding of variability specifications, which may be used to derive a new process variant considering context data. It specifies which alternatives described in the VM are valid for settled placeholder activities in the BM considering current conditions. To that end, a set of resolution elements define how a VM is bound. A *resolution element* represents a binding of variability. It could either do a complete binding in which all variability from a variability specification is resolved, or a partial resolution in which some variability is still present. It has three types of subtypes:

- *Value resolution* determines a value for the variability where context values are mapped to variables' or features' resolutions in the variability specification.
- *Fragment resolution* represents resolutions for variation points in the variability specification as well as the corresponding mappings in the BM.
- *Perspective resolution* considers context data related to people and/or things perspective resolution. It is used to minimize the process variant configuration options reducing alternatives. Once the multi-perspective variability specification is fully described, our system starts to collect context data in order to start a perspective variability resolution. This separation of concerns allows to perform different strategies for and automated staged-configuration within our toolkit - people-first, things-first and in-row.
  - `People-first`: an initial pre-configuration on people perspective variability is performed, reducing process variants alternatives for a latter things perspective resolution.
  - `Things-first`: In contrast, this strategy enables to be things related data first recognized in order to perform a staged-configuration of process variability.
  - `In-row`: The perspective resolution is just done by interpreting context data as it is collected to resolve multi-perspective process variability.

To perform the aforementioned types of resolutions, we classify two types of context data: *static* context data variable-value pairs (e.g. temperature sensor type or stakeholder) representing static preferences of a particular domain that are known just before the process instantiation and which rarely change over time, and *dynamic* context data pairs (e.g. specific temperature of a room) which change frequently over time and thus only available when the selection of a placeholder activity occurs. Once a set of context pairs is fulfilled (i.e. assigning a value to context variables from context data), value, fragment and perspective resolutions are executed.

3.2 Modeling Multi-perspective Process Variability

One of the advantage of separating commonality and variability specification is that there may be more than one VM for each BM, as illustrated in Figure 1. Process modelers may associate as many variation models as they require to an unique BM even representing multiple perspectives. Ideally, each smart environment solution (e.g. a concrete SGB solution) would require a separate VM, so we will build a different VM for each domain solution (specifying all particularities by multiple perspectives support) where a BM is applied. Therefore, this set of variation models will represent different variability scenarios at different levels regarding a common shared BM.

The cornerstone of our approach is multi-perspective support, which may be modeled in a single VM or also taking advantage of separated models for each perspective description. However, from our previous work (Murguzur et al., 2013), we have noticed that to avoid highly complex models, we should try to separate different variability dimensions such as people variability and things variability. Thus, our approach now enables the process variability modeling of people perspective (PP) and things perspective (TP) in disjointed models as follows:

- `In-PP`: people variability perspective may only contain variation points.
- `In-TP`: things variability perspective may only contain variation points.
- `SharedBy`: both variability perspectives may contain elements directly related to variation points.

4 PROTOTYPE AND TOOLKIT

To implement our multi-perspective process variability we utilize Activiti Designer[6] for base model and fragments specification and Clafer[7] (CLAss, FEature,

[6]http://activiti.org
[7]http://clafer.org

Reference) for modeling people and things perspective variability. The Activiti Designer is an open-source and light-weight Business Process Management (BPM) platform which offers among other features a powerful designer and a solid BPMN2 process engine. Clafer is also a lightweight modeling language which enables to express variability among products and systems.

Following the presented approach, we model the variability of an elevators remote predictive maintenance and monitoring process in an SGB domain using two separated variation models and techniques (`In-PP`, `In-TP` and `SharedBy`): one for people perspective (e.g. maintenance contracts) and the other for things perspective (e.g. elevator's modules - door and pulley) variability modeling.[8]

Overall, our prototype shows encouraging results to cope with complex large models, by adopting base models, variation models (representing things and people perspective separately) and process fragments for representing multi-perspective process variability. However, it is restricted in the sense that it purely focuses on the control-flow.

5 CONCLUSIONS

Likewise in SPLE, process variability needs to consider the multiple dimensions coming from smart environments, i.e., it needs to support the multiplicity of stakeholders, devices and data derived from process-intensive applications. This premise is the underlying idea of our paper that presents a novel approach for supporting multi-perspective process variability by using a fragment-based re-use approach. After discussing the impact of multiple dimensions in SGBs with respect to multi-perspective process variability, we pointed out the related work and also described a methodology by means of a meta-model for supporting multi-perspective process variability adopting a fragment-based re-use approach.

In our future work we plan to consider an automated resolution of multi-perspective process variability by processing over context data streams, as well as evaluating the presented perspective resolution strategies against different modeling methods.

REFERENCES

Baresi, L., Guinea, S., and Pasquale, L. (2012). Service-Oriented Dynamic Software Product Lines. *IEEE Computer*, 45(10):42–48.

Bayer, J., Gérard, S., Haugen, O., Mansell, J., Moller-Pedersen, B., Oldevik, J., Tessier, P., Thibault, J., and Widen, T. (2006). Consolidated product line variability modeling. In *SPLC*.

Berger, T., Rublack, R., Nair, D., Atlee, J. M., Becker, M., Czarnecki, K., and Wasowski, A. (2013). A Survey of Variability Modeling in Industrial Practice. In *VaMoS*.

Castro, M., Jara, A. J., and Skarmeta, A. F. (2012). An analysis of m2m platforms: Challenges and opportunities for the internet of things. In *IMIS*.

Classen, A., Hubaux, A., and Heymans, P. (2009). A formal semantics for multi-level staged configuration. In *VaMoS*.

Czarnecki, K., Helsen, S., and Eisenecker, U. (2004). Staged configuration using feature models. In *SPLC*.

García-Galán, J., Trinidad, P., and Ruiz-Cortés, A. (2013). Multi-user variability configuration: A game theoretic approach. In *ASE*.

Gurgen, L., Gunalp, O., Benazzouz, Y., and Gallissot, M. (2013). Self-aware cyber-physical systems and applications in smart buildings and cities. In *DATE*.

Hallerbach, A., Bauer, T., and Reichert, M. (2010). Capturing Variability in Business Process Models : The Provop Approach. *Software Maintenance and Evolution*, 22(6-7):519–546.

Hubaux, A., Heymans, P., Schobbens, P.-Y., Deridder, D., and Abbasi, E. (2011). Supporting multiple perspectives in feature-based configuration. *Software and Systems Modeling*, pages 1–23.

Kolb, J. and Reichert, M. (2013). A flexible approach for abstracting and personalizing large business process models. *Applied Computing Review*, 13(1):6–17.

Meerkamm, S. (2012). Staged configuration of multi-perspectives variants based on a generic data model. In *BPM Workshops*.

Mendonça, M., Cowan, D., Malyk, W., and Oliveira, T. (2008). Collaborative Product Configuration: Formalization and Efficient Algorithms for Dependency Analysis. *Journal of Software*, 3(2):69–82.

Murguzur, A., Sagardui, G., Intxausti, K., and Trujillo, S. (2013). Process variability through automated late selection of fragments. In *CAiSE Workshops*.

Rosa, M. L., Dumas, M., ter Hofstede, A. H., and Mendling, J. (2011). Configurable multi-perspective business process models. *Information Systems : Databases : Their Creation, Management and Utilization*, 36(2):313–340.

Rosenmüller, M., Siegmund, N., Thüm, T., and Saake, G. (2011). Multi-dimensional variability modeling. In *VaMoS*.

Schnieders, A. and Puhlmann, F. (2006). Variability mechanisms in e-business process families. In *BIS*.

Schroeter, J., Lochau, M., and Winkelmann, T. (2012). Multi-perspectives on feature models. In *MODELS*.

Wang, K.-Y., Lin, G., Chou, P., and Chou, A. (2012). Leverage smart system services technology for smart green building management. In *EGG*.

[8] Detailed information at: http://aitormurguzur.com/projects/lateva/multi-perspective