

Analysing Variability Management in BPM and SPL: A Knowledge Mapping

George Valença¹, Carina Alves¹, Vander Alves²

¹ Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE)
Recife – Brazil

² Departamento de Ciência da Computação (CIC) – Universidade de Brasília (UnB)
Brasília – Brazil

{gavs,cfa}@cin.ufpe.br, valves@cic.unb.br

***Abstract.** New government regulations, business volatility and changes on stakeholders needs are examples of process change inductors. This dynamic scenario promotes a phenomenon called business process variability. This is an emergent topic in Business Process Management (BPM) that uses Software Product Line (SPL) theoretical basis to investigate the variability of process models. This paper explores the adoption of variability concepts and theories by the BPM community, with a mapping to analyse how the variability body of knowledge has been used from SPL to the BPM field. In addition, we position current research challenges within business process variability, and discuss how concepts and theories from SPL could be used to treat these questions.*

1. Introduction

The term ‘variability’ stands for something that is capable of being varied or changed. It is a characteristic associated with diverse elements, such as species in Biology, functions in Mathematics and stars in Astronomy, which may vary due to inconstant features or to an unsteady manner. Variability is a fundamental concept for the development of techniques, tools and general instruments in Software Product Line (SPL). According to [Van Gorp et al 2001], it guides the definition of a product line, minimising the cost of creating and evolving software products that compose a product family. Variability evidences the differences between the products within a software product line and is a macro notion covering concepts such as flexibility and similarity and with techniques based on software reuse paradigm [Van Gorp et al 2001]. [Chen et al 2009] introduces terms such as configuration, change, commonality and instantiation to define the context of software variability. They introduce the notion of Variability Management (VM) as the act of managing dependencies among different variabilities and supporting their instantiations.

The underlying research and practice of SPL has improved quality and, reduced time-to-market and costs in diverse domains [Alves et al 2010]. The positive results and the maturity obtained by SPL since its advent in 1990, introducing feature-oriented paradigm, turns its concepts and techniques a relevant source for knowledge reuse.

[Sjøberg et al 2007] discusses what theories are, from which elements they are formed and how they are built. It is stated that knowledge of a specific topic or

phenomenon translated in a theory can be formed from scratch or from other disciplines. Therefore, a field may adopt a knowledge reuse perspective for constructing its theories, employing definitions from other branches of study. It is an efficient way to emerge novel principles in an inductive form, inspired from different views. A recent example of theory construction through knowledge reuse occurs in Software Ecosystems field [Barbosa and Alves 2011]. This area investigates the relationships among companies in software industry using theories from Natural and Business Ecosystems as well as concepts from SPL.

Business Process Management (BPM) also follows theories from other disciplines such as SPL, Requirements Engineering and Decision Models to treat the emergent phenomenon of process variability. The later analysis how processes change as a response to the evolution of business requirements and market dynamics. Researchers such as [Boffoli et al 2008] [Hallerbach et al 2010] [La Rosa et al 2009] [Rolland and Nurcan 2010] [Regev et al 2006] and [Vervuurt 2007] have borrowed variability management knowledge from SPL to address issues brought by this phenomenon, developing approaches inspired by concepts and theories from this area. Given this scenario, this paper is directed by the research question: *how SPL knowledge on variability management has been reused and adopted by BPM research community?*

This question is answered by analysing key literature on variability in the fields of BPM and SPL. We aim to investigate the conceptual basis of proposals in BPM with respect to variability management and examine the relation kept with SPL literature. We then summarise the available knowledge concerning business process variability.

This paper provides a conceptual reflection on the state of the art in business process variability. Drawing these conclusions we may direct efforts of BPM researchers towards a reuse of knowledge in a more conscious and coordinated form. Hence, our main contributions are: (1) a knowledge mapping between well-established notions of variability management from SPL and their correspondences in business process variability field, enabling the SPL community to become aware of how concepts and theories in VM have been adopted by BPM researchers; (2) a discussion on how the BPM community can better explore SPL knowledge already adopted and how additional VM concepts may be applied to treat research challenges within BPM.

The paper is divided in the following sections. Section 2 describes the research method. Section 3 details variability notions within BPM and presents the proposed knowledge mapping. Section 4 discusses the results and analyses research gaps and opportunities to mature the field of variability management in BPM. Section 5 closes the paper by providing the main conclusions and future works.

2. Research Approach

Our investigation of variability management phenomenon is based on theory building concepts [Sjøberg et al 2007]. This strategy considers that we can evolve the conceptual basis of an area by borrowing or adapting theories from other disciplines. Therefore, theories from other disciplines can be used as they are or they can be borrowed and adapted to a different context. In this paper, our goal is to examine the extent to which concepts from SPL have been used to solve specific issues in BPM.

Business process variability is a recent research topic within BPM field. As a basis for this study, we considered a systematic literature review executed by this research group [Valença, 2012] (this work was published as a Master's thesis, with its protocol and results available at <http://db.tt/nRLNcfMF>). It explores the problem of business process variability at a theoretical and empirical perspective. 80 primary studies were selected as sources of evidence to address the following issues: characteristics, mechanisms and inductors of business process variability; approaches, tool support and empirical evaluation; and current challenges within the field. As far as we are aware, [Fantinato et al 2012] is the only survey in the field of business process variability, with a mainly technical view on the subject, exploring process reuse techniques and their limitations. Besides this SLR, we also included the studies [Regev et al 2006] [Vervuurt 2007] [La Rosa et al 2009] [Rolland and Nurcan 2010] [Boffoli et al 2008] [Hallerbach et al 2010] as sources of knowledge, representing prominent works from the main contributors to business process variability analysis.

From the perspective of software product lines, to build a theoretical foundation on variability management we analysed the studies of [Svahnberg et al 2005] [Van Gurp et al 2001] [Chen et al 2009] [Mujtaba et al 2008] [Linden et al 2007] [Andersson and Bosch 2005]. [Chen et al 2009] and [Mujtaba et al 2008] are systematic reviews. Therefore, we believe these works provide a satisfactory synthesis of available approaches dealing with VM in SPL. However, we do not aim to accomplish a tertiary review, extending secondary studies in the mentioned areas. Our goal is to theorise over a set of secondary studies in order to aggregate and compare knowledge from SPL and business process variability fields.

Based on our review of relevant studies in both areas, we constructed a knowledge mapping of concepts and theories on VM originally developed by SPL community and reused by the BPM community. This mapping offers a conceptual framework to structure concepts in a succinct and precise manner. In particular, it can facilitate communication and transfer of ideas between both communities.

As a consequence from the mapping procedure, SPL knowledge not yet used by BPM researchers could be explored. We identified research challenges in BPM with respect to variability management and explored how this SPL knowledge may be useful to treat open issues in BPM. Given that SPL is becoming a mature discipline and a large body of knowledge has been accumulated, we assume that several approaches in SPL can be appropriate to solve similar variability problems in BPM. Finally, we outline directions of research to increase knowledge reuse of additional concepts from SPL.

3. Variability in BPM

Business processes have improved management activities, approximating the strategic guidance from the people who execute their work to achieve organizational goals. BPM is structured as a holistic discipline that enables the alignment between strategic and operational areas, ensuring a systematic approach to deal with the ever-changing external environment [Santos et al 2011]. In this context, managing a process lifecycle generally implies that a set of changes must be introduced. This situation is identified as business process variability, being a direct response to the dynamism of the organizational operating environment and evolving business needs. Business process

variability can be defined as the property of a process model to be changeable [Eijndhoven et al 2008].

In the systematic literature review (SLR) presented in [Valença 2012], the reasoning behind process changes was investigated, describing a set of inductors of variability. The review shows that market dynamics can be considered the main source of variations, with aspects such as high competition, dynamic networked business and globalization as the key drivers for process changes. Other factors influencing changes in business processes are culture, stakeholder requirements and technology. In addition, internal organizational factors include improvement policies, organizational strategies and structure, which may also foster process variations.

To support business process variability notion, different approaches have been proposed. A total of 57 proposals to address business process variability were identified by the SLR. The approaches can be classified in five categories, representing issues related to business process variability: configuration, correctness, flexibility, similarity and variability modelling. Variability modelling was the focus of 42% (24) of the approaches investigated by the SLR. The studies focus on the graphical representation of the variation. These techniques generally extend popular modelling languages to include attributes to identify variability aspects in process models. Examples of these contributions are the introduction of features in Event-Driven Process Chain (EPC) models [Vervuurt 2007]. The former is inspired on feature diagrams, importing SPL concepts to treat process configuration issues.

Process configuration was a topic addressed by around 18% (10) of the approaches. These studies analyse how to assist the individualization of a process model to a particular context. In order to enhance this activity, we found studies that propose a decision support abstracting from modelling notations [La Rosa et al 2009] and a description of change patterns to be applied during model customization [Hallerbach et al 2010]. These approaches adapt the idea of software configuration management, when related functionality is packaged into software assets and variants are derived.

Almost 18% of the proposals explored process flexibility. Flexibility is increased when a process can be changed in a rapidly and easily. Approaches within this category employed business rules concepts to split process behaviour into a stable and a flexible part [Rittgen 2006], and also used formalisms such as Petri nets to achieve flexibility [Xiao et al 2009]. To improve flexibility features, part of them transfer to processes peculiarities of SPL, given its goal of dealing with continuous changes.

Process similarity was investigated by 12% (7) of the approaches, which shared the goal of diagnosing commonalities and variations among processes. These proposals focused on quantifying the similarity based on the sequence of activities that hold for the process model [Dongen et al 2008] or detecting similarities between process models through comparisons of the linguistic structure of their elements [Koschmider and Oberweis 2007]. Product Lines techniques inspire this context due to its ability to specify commonalities, improving comparison and adaptability efforts.

Figuring as the topic with the lowest number of contributions (around 11%; 6), process correctness aims to guarantee that process models are syntactically and semantically correct. Among the techniques included in this category, a framework for

ensuring the soundness and semantic validity of a group of process variants was proposed [Hallerbach et al 2009]. These studies indicate that their correctness-preserving approaches were inspired by SPL techniques that ensure a configured software artefact is valid [Van der Aalst et al 2010].

In addition, concerning tool availability, 45.6% of the approaches (26) provide a mechanism to support their proposals, which represents a major limitation for the wide adoption of the techniques by practitioners, since manually implementing the proposed solutions may be time-consuming and error-prone.

4. Knowledge Mapping

We present a conceptual foundation of business process variability field. Our objective is to describe the basic constructs of this phenomenon, analysing central properties and components. We define a knowledge mapping between principles from SPL to BPM. Through this structure we analyse the foundation nature of concepts and theories employed in variability management activities within BPM and highlight how this knowledge reuse was accomplished by outlining the eventual adaptations introduced.

4.1. Process Variability Foundations

Passages from the primary studies¹ in [Valença 2012] that provide a theoretical background on business process variability were integrated and evaluated in an open coding procedure. Theme categories represented the available knowledge with respect to business process variability and summarised concepts and theories within this topic. The resulting synthesis is described below, with its elements in bold in the following paragraphs. Drawing these conclusions, we may direct efforts of BPM researchers towards a more conscious and coordinated use of conceptual knowledge.

Business process variability is the capability of an artefact to be configured, customised or changed for use in a specific domain. Given its conceptual foundation on reuse-oriented development in BPM, it enables the reuse of parts of a model while adaptations to its functioning are introduced. Many authors use the term flexibility to refer to the notion of variability in the scope of business processes. Similar to business process variability, **business process flexibility** is the ability of a process to adapt to the changes in the environment or to its changing requirements. It concerns how rapid and easy a process model is modified.

Several authors split business process variability in two perspectives: **design-time** and **runtime**. The former type essentially refers to variations of models during modelling phase, before they are implemented in a workflow management system or BPMS for execution. The latter type is associated with processes on execution, addressing runtime variability with exception handling approaches. These types are complementary to each other and can also be called design-time and runtime business process flexibility, given the tiny boundary between variability and flexibility notions.

A **reference business process model** supports the reuse paradigm by collecting and depicting proven best practices of a specific domain. It provides a starting point to

¹ This list is available at <http://db.tt/PGBXeAc0>

define process models for a particular setting (e.g. a company) and improves modelling by avoiding the construction of a model from scratch. However, the common traits captured by these models do not turn them into plug and play solutions. Adjustments must be executed, since these generally do not offer configuration facilities.

The concept of **configurable business process model** implements the notion of reference business process models. It is a step forward towards the reuse of business processes. They are defined via a **configurable process modelling language** or notation, which provides means to insert variability in a process model. Configurable models are constructed by merging several **business process variants**, which are processes achieving the same goals but slightly differing from each other in their structure due to domain specific requirements. Variants are versions of a particular process model called **business process type**, which is defined at design-time and represents a standard way of acting within an organization.

Variability is achieved by introducing special placeholders in a configurable model referred to as **variation points**, also known as configurable nodes and adjustment points. A set of options are assigned to these points by means of **variability mechanisms**, which are techniques (e.g. inheritance) that realise variability in a model.

To customise configurable models to a specific solution, an analyst selects the most suitable option for each variation point, in a procedure called **business process configuration**. It derives a process from a configurable model by restricting its behaviour and can be regarded to as **design-time business process variability**. The scheme spawned via a configurable model is a process variant, which is enacted as a **business process instance** at runtime. If changes are implemented during the execution of this process instance, this is referred to as **runtime business process variability**.

Process configuration implies decision making and it is therefore guided by **configuration decisions**. These judgements are applied over each variation point to assess its available choices, based on information from the context in which the derived model should be employed. This information may be expressed as configuration requirements (hard constraints) and configuration guidelines (recommendations), which can be bound to variation points to restrict the combination of available options.

Configuration requirements and guidelines aim to avoid undesirable configurations. This leads to the notion of **business process correctness**, which is a characteristic of the generated process models being valid in a syntactic and/or semantic form. The syntactic property is centred on the adequate use of the modelling notation through which the model structure was created. The semantic property is also called business process soundness, analysing the dynamic behaviour of the process model to ensure that no deadlocks or livelocks in the control-flow prevent a proper completion.

The collection of variants obtained by means of process configuration can be denoted as **business process line**. These models represent alternative forms of the same underlying process and share an invariant nucleus known as **core process**. This common structure expresses the compromises kept by members of a process line. The degree of commonality a process model keeps with respect to another model within a process line is known as **business process similarity**. It results from the comparison of multiple aspects of process variants to describe to which degree they share a similar structure.

5. Mapping Results

In Table I, we present knowledge correspondences between BPM and SPL theories. Each notion presented in Section 4 is directly associated with an equivalent concept in SPL. In addition, an explanation is given for each concept. We subsequently illustrate the use of the definitions in SPL and BPM, and analyse the connections identified.

Table 1. Knowledge mapping between BPM and SLP concerning VM

BPM	SPL	
<i>Concept</i>	<i>Concept</i>	<i>Explanation</i>
Configurable process modelling language	Variability modelling language	Variability modelling language allows describing commonality and variability in SPL products.
Configurable business process model	Product family architecture	A product family architecture is the unification of all system architectures in a SPF, capturing the architectural commonalities and variabilities in a product family and guiding developers when new application instances are derived.
Configuration decision	Configuration Decision	Decision when binding a variation point.
Core business process	Core asset	A core asset is reusable software artefact across at least two SPL instances. It can be as coarse-grained as the product family architecture or as fine-grained as small components, e.g. classes.
Business process configuration	Product instantiation	Product instantiation means creating a specific software product using a software product line. It removes from the SPL architecture all unneeded functionality and therefore selects pre-implemented variants.
Business process correctness	Well-formed SPLs	Well-formed SPLs are obtained when a set of constraints are captured during the configuration of software artefacts, determining if a configuration is valid.
Business process line	Product family	A product family captures commonalities between software products for the product family, promoting reuse of core components across related software products.
Business process flexibility	Flexibility	Flexibility is a key concern in SPL development. Core assets must easily change to accommodate the requirements of different products.
Business process instance	SPL instance	An instance is a running product of the SPL.
Business process reference model	Reference architecture	Setting up and describing a reference architecture for a product line provides a basis for member instantiation.
Business process similarity	Commonality	A software product line captures commonalities between software products of a product family.
Business process type	Artefact type	Artefacts comply with well-formed rules: types.
Business process variability	Variability	For a SPL, variability describes the characteristics that vary from application to application within the product family.
Business process variant	Variant	A variant in SPL stands for a software component, being design alternatives to resolve the variability in a product family.
Variability mechanism	Variability mechanism	A variability mechanism is a variability realization technique for a product line.

Variation point	Variation point	Variation points are available for re-users to build variants in a SPL.
Design / build-time and runtime	Binding time (pre-compilation, compilation, load time, runtime) and binding mode (static, dynamic)	Development stage when variability is bound.

Table 1 summarises concepts and theories from Software Product Line literature, defining a knowledge mapping with VM notions in BPM. The concept of process line transfers to BPM the peculiarities typical of SPL paradigm. It adapts the notion of SPL asset to that of a process model in BPM [Boffoli et al 2008] and moves away from managing individual processes to coordinating a set of similar processes considered as a whole [Rolland and Nurcan 2010]. A process line is based on the notions of process similarity and process variability. The former indicates the degree of commonality between a set of process models, while the later analyses variable aspects within models.

Process variability is generally modelled via a configurable model, which introduces variation points in a process type. Process variants are obtained from this artefact through process configuration, which takes place at design-time. This procedure is supported by variability mechanisms and guided by configuration decisions. A process variant can also be defined with a customization of a process reference model. During runtime the variant then is executed as a process instance. In SPL this scenario corresponds to obtaining a software component, a variant in SPL, by adjusting variation points with the support of variability realization techniques or mechanisms. Hence, configuring a software system implies defining actions over a set of parameters, features or choices to obtain a customised system from a generic one.

Process configuration highlights that typical operations of configuration and specialization were borrowed from SPL field [Boffoli et al 2008]. Configuring products through asset integration in SPL, giving rise to a product family architecture, is equivalent to the creation of a configurable process model via a configurable modelling language. The specialization of the assets through the specification of their parametric parts stands for product instantiation and corresponds to business process configuration.

A general concern during process configuration is to obtain correct models. This involves the analysis of the configured process to verify whether it keeps a correct syntax and/or an executable behaviour. Some process correctness approaches were inspired by techniques from SPL field that enable the configuration of software artefacts with models that relate these to domain concepts [Van der Aalst et al 2010]. Within SPL, well-formed product lines result from the application of constraints while configuring software artefacts.

Process flexibility aims at exploring capabilities to modify process models quickly and adequately in face of internal and external changes. In the SPL context, flexibility refer to incremental and iterative development of products within a reacting adoption strategy, whereby short time-to-market is achieved.

Approaches to treat process variability must reflect the stage when variation is treated. Design-time variability encompasses concepts and techniques focused on VM in a type level, managing a process variant before its introduction in a workflow system. For SPL practices, this early binding can happen at pre-compilation, compilation, or

linking time during product derivation. It also reflects the static linking of systems, applied before product delivery [Van Gurp et al 2001]. The adaptation of a process instance during its execution with a flexible behaviour [Schonenberg et al 2008] is equivalent to the runtime binding time, with a dynamic binding mode in SPL. With dynamic binding, the system is able to use new components at runtime, offering flexibility to users [Van Gurp et al 2001].

6. Discussion

We provide here an overview of the main research challenges in BPM field with respect to VM. From the review presented in [Valença 2012], a set of issues associated to process variability were identified. As a starting point to treat these research challenges, we discuss how contributions from SPL not yet adopted by BPM community may be useful. This is a relevant input considering the grounded knowledge SPL field holds.

The lack of expressiveness with respect to variability aspects in most process modelling standards is a major issue in the field. On the other hand, approaches that extend modelling languages with variability features generate a complexity explosion within a process model, introducing an excessive description of business aspects. This decreases the degree of understandability of a model, with the creation of complex structures. In SPL, VM approaches can be classified as annotative, compositional, transformational and parametric [Schaefer et al 2010]. These might have to be combined to address different kinds of variability, considering its degree of scattering, tangling, and granularity, in order to achieve sufficient expressiveness and acceptable levels of internal and external quality. Such specificity is yet to be leveraged in the BPM context.

Given that the adaptation of process models due to changing conditions must be rapid and precise, it is fundamental to offer guidance and decision support for analysts during process configuration. Some techniques for process configuration have focused on domain aspects for variability elicitation. However, the core issue is to provide analysts with the impact of each configuration decision on the process model. Among other benefits, this would reduce the need for a domain expert analyst in the configuration of process models. In the SPL context, process configuration is normally aided by a decision modelling approach [Gacek and Anastasopoulos 2001]. Recent approaches allow specifying QoS constraints on the configurations and then run heuristics to prune the rapidly increasing configuration space to identify complying configurations [Roos-Frantz et al 2010]. This can reduce the burden on the domain expert. These techniques could be applied in the BPM context as well to provide automation support and informed decisions in the configuration process.

Issues associated to the correct configuration of a whole process line are only at a preliminary stage. There are considerable works on process correctness topic guaranteeing structural and behavioural soundness of a single model. However, characterizing the correctness of a family of processes for a particular configurable process model is difficult and time-consuming. Additionally, the use of manual methods for configuring process models does not ensure the resultant models are correct. Syntactical and semantic correctness can benefit from specific tool support, with process modellers free of the burden of ensuring process correctness or manually fixing errors.

The SPL community is currently exploring this issue in almost all VM mechanisms categories: annotative, compositional, and transformational approaches. However, the target languages are either simple (e.g., Feature Feather Weight Java [Apel et al 2008]) or specific (Delta-oriented programming [Schaefer et al 2010]). In addition, the extent to which this issue is explored is still mostly type safety, i.e., syntactic. Semantic issues are only beginning to be tackled automatically by the use of model checkers [Classen et al 2010]. Analytic reasoning on semantic correctness is still unaddressed. A similar path could be followed in the BPM community, by initially handling syntactic correctness in different VM approaches, then moving to the use model checkers and then analytic reasoning for semantic correctness. This is crucial for scalable family-aware quality assurance and definition of certification procedures.

Another important aspect is the implementation of process systems that consider a myriad of business requirements linked to flexible processes. Also, modelling languages should consider the achievement of process flexibility in a controlled manner, not only capturing the reactive part of flexibility but also investigating the stimulus for change. Likewise, in the SPL context, flexibility is normally constrained by a domain scope and more recently partially supported by change impact analysis techniques and family agility. These lines of research could be leveraged in the BPM context.

7. Conclusion and Future Work

The design of software product lines or families demonstrated to BPM the need to elicit commonalities and variable parts in a family, stressing the relevance of the variability notion. The management of commonality and variability within a product line promotes reuse and demands specific adaptations of products to different contexts. We considered this scenario to analyse concepts employed in VM within BPM, defining knowledge correspondences in SPL field. This mapping provided a structured comparison about how notions from SPL paradigm were adopted by BPM researchers. We hope this study can foster insights with respect to the manner this knowledge reuse occurs.

We also explored how to solve research gaps in process variability with the support of SPL knowledge and highlighted a set of potential research opportunities. This can therefore be the basis for introducing and extending novel VM concepts by the BPM community. We believe that discussing this shared theoretical foundation brings advances and contributes to the evolution of both research fields.

A potential limitation of this study is associated to the procedure of gathering the conceptual knowledge in SPL. Since we have not found a formal taxonomy, the definition of a terminology to be the basis for the conceptual mapping was a result of filtering tasks. We examined relevant literature to extract main concepts, removed overlaps and identified individual terms or theories.

Concerning the BPM field, these findings shall trigger research efforts to define a taxonomy for process variability. As a future work, we plan to perform expert reviews to assess the completeness and correctness of the results obtained, considering researchers' and practitioners' perspectives in both SPL and BPM fields. Additionally, from SPL perspective, the results provided in this study are a motivating initiative for researchers to analyse how knowledge has been reused and adopted in a different area. It is also possible to examine improvements made by BPM researchers.

References

- Alves, V., Niu, N., Alves, C. and Valença, G. (2010) “Requirements Engineering for Software Product Lines: A Systematic Literature Review”. *Information and Software Technology* 52, pp. 806–820.
- Andersson, J. and Bosch, J. (2005) “Development and Use of Dynamic Product-line Architectures”. *IEE Proceedings Software*, 152(1), pp. 13-26.
- Apel, S., Kästner, C. and Lengauer, C. (2008) “Feature Featherweight Java: a Calculus for Feature-oriented Programming and Stepwise Refinement”. In *Proceedings of the Int’l Conference on Generative Programming and Component Engineering*.
- Barbosa, O. and Alves, C. (2011) “A Systematic Mapping Study on Software Ecosystems”. In *Proceedings of Int’l Workshop on Software Ecosystems*. p.12.
- Boffoli, N., Caivano, D., Castelluccia, D., Maria Maggi, F. and Visaggio, G. (2008) “Business Process Lines to Develop Service-Oriented Architectures Through the Software Product Lines Paradigm”. In *Proceedings of SPLC*, pp. 143-147.
- Chen, L., Ali Babar, M. and Ali, N. (2009) “Variability Management in Software Product Lines: A Systematic Review”. In *Proceeding of Software Product Line Conference*, pp. 81–90. San Francisco, CA, USA.
- Classen, A., Heymans, P., Schobbens, P.Y. and Legay, A. (2010) “Symbolic Model Checking of Software Product Lines”. In *Proceedings of Int’l Conference on Software Engineering*, pp. 321-330.
- Dongen, B., Dijkman, R. and Mendling, J. (2008) “Measuring Similarity Between Business Process Models”. In *Proceedings of Int’l Conference on Advanced Information Systems Engineering*.
- Eijndhoven, T., Iacob, M. and Ponisio, M. (2008) “Achieving Business Process Flexibility with Business Rules”. In *Proceedings of Int’l IEEE Enterprise Distributed Object Computing Conference*.
- Fantinato, M., Toledo, M. B. F., Thom, L. H., Gimenes, I. M. S., Rocha, R. S., Garcia, D. Z. G. (2012) “A Survey on Reuse in the Business Process Management Domain”. *Int’l Journal of Business Process Integration and Management* 6, pp. 52-76.
- Gacek, C. and Anastasopoulos, M. (2001) “Implementing Product Line Variabilities”. In *Proceedings of the Symposium on Software Reusability*, pp. 109-117.
- Hallerbach, A., Bauer, T. and Reichert, M. (2009) “Guaranteeing Soundness of Configurable Process Variants in Provop”. In *Proceedings of IEEE Conference on Commerce and Enterprise Computing*, pp. 98-105.
- Hallerbach, A., Bauer, T. and Reichert, M. (2010) “Capturing Variability in Business Process Models: the Provop Approach”. *Software Maintenance* 22, pp. 519-546.
- Santos, HM., Santana, AF, Alves, C. (2011) “Análise de Fatores Críticos de Sucesso da Gestão de Processos de Negócio em Organizações Públicas”. In *Proceedings of VII Simpósio Brasileiro de Sistemas de Informação*.
- Koschmider, A. and Oberweis, A. (2007) “How To Detect Semantic Business Process Model Variants?”. In *Proceedings of ACM Symposium on Applied Computing SAC*.

- La Rosa, M., van der Aalst, W., Dumas, M. and ter Hofstede, A. (2009) "Questionnaire-Based Variability Modeling for System Configuration". *Software and Systems Modeling* 8, pp. 251-274.
- Linden, F. J. v. d., Schmid, K. And Rommes, E. (2007) "Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering". Springer-Verlag.
- Mujtaba, S., Petersen, K., Feldt, R. and M. Mattsson. (2008) "Software Product Line Variability: A Systematic Mapping Study". In *Proceeding of Asia-Pacific Software Engineering Conference*.
- Regev, G., Soffer, P. and Schmidt. R. (2006) "Taxonomy of Flexibility in Business Processes". In *Proceedings of Workshop on Business Process Modelling, Development, and Support*.
- Rittgen, P. (2006) "Supporting Planned and Ad-Hoc Changes of Business Processes". In *Proceedings of CAISE Workshops*.
- Rolland, C. and Nurcan, S. (2010) "Business Process Lines to Deal with the Variability". In *Proceeding of Hawaii Int'l Conference on System Sciences*, pp. 1-10.
- Roos-Frantz, F., Benavides, D. and Ruiz Cortés, A. (2010) "Automated Analysis of Orthogonal Variability Models using Constraint Programming". In *Proceedings of Jornadas de Ingeniería del Software y Bases de Datos*, pp. 269-280.
- Schaefer, I., Bettini, L., Bono, V., Damiani, F. and Tanzarella, N. (2010) "Delta-oriented Programming of Software Product Lines". In *Proceedings of Soft. Product Line Conference*, pp. 77-91.
- Schonenberg, M. H., Mans, R. S., Russell, N. C. and Mulyar, N. A. (2008) "Process Flexibility: a Survey of Contemporary Approaches". *Int'l Workshop CIAO!*.
- Sjøberg, D.I.K., Dybå, T., Anda, B.C.D and Hannay, J.E. (2007) "Building Theories in Software Engineering". *Advanced Topics in Empirical Softw. Eng.* Springer Verlag.
- Svahnberg, M., van Gorp, J. and Bosch, J. (2005) "A Taxonomy of Variability Realization Techniques". *Softw., Pract. Exper.*, Vol. 35(8), pp. 705-754.
- Valença. G. (2012) "Business Process Variability: A Systematic Literature Review". Master thesis. Center of Informatics, Federal University of Pernambuco, Brazil.
- Van der Aalst, W., Dumas, M., Gottschalk, F., ter Hofstede, A., La Rosa, M. and Mendling, J. (2010) "Preserving Correctness During business Process Model Configuration". *Formal Aspects of Computing* 22, pp. 459-482.
- Van Gorp, J., Bosch, J. and M. Svahnberg. (2001) "On The Notion of Variability In Software Product Lines". In *Proceedings of the 2nd Working IEEE/IFIP Conference on Software Architecture*, pp. 45-54. IEEE Computer Society.
- Vervuurt, M. (2007) "Modeling Business Process Variability". Master Thesis. University of Twente.
- Xiao, L., Koo, B.H.Y. and Zheng, L. (2009) "Achieving Flexibility in Business Process Modeling Using an Algebraic Language". In *Proceedings of Int'l Conference on Model-Based Systems Engineering*.