Developing a Context Model of Process Variants for Business Process Integration

Jorge E. Giraldo[1,2(✉)], Demetrio A. Ovalle[1], and Flavia M. Santoro[3]

[1] Universidad Nacional de Colombia, Medellín, Colombia
{jegiraldp,dovalle}@unal.edu.co
[2] Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia
jegiraldo@elpoli.edu.co
[3] Universidad Federal del Estado de Rio de Janeiro, Rio de Janeiro, Brazil
flavia.santoro@uniriotec.br

Abstract. Since the integration of business process variants can be affected by contextual information the aim of this paper is to propose the development of a context model for Business Process Integration that considers the propagation of process changes towards process variants of the reference model. The method used to develop such a model focuses on following stages: (1) conceptualization of process variants, by building a related ontology, (2) identification of contextual elements, (3) definition of operational and organizational constraints; (4) formalization of contextual situations in the variants, and (5) construction of a reasoning mechanism for searching and retrieval of process variants. In order to validate the context model proposed we implemented both a prototype and a case study that consider each of the five stages proposed. The obtained results demonstrate the effectiveness of using this kind of approaches for business process integration.

Keywords: HCI in business · Context process model · Business process integration

1 Introduction

A business process is defined as the logical description of a specific business activities sequence being modeled through process models which allow new value to be added to their products or services [1]. Due to the rise of using business processes and their automation, process models generated can reach an amount of hundreds or even thousands. This fact allows them to be considered as model collections for searching, retrieval, and reuse [2, 3].

When a business process is executed, there may be domain-dependent situations, implying the need for configuration of reference process model variations. These variations are known as variants and constitute an adjustment of the reference model under specific requirements [4].

Updating a process reference model requires propagation of changes to the associated variants. Variant management can be individually performed or through a group mechanism by means of an integration process [5]. However, not all variants can

support the propagation of changes due to following issues: (a) their close relationship with the domain, (b) the execution in a specific context, and (c) the associated events that generated the variant [6].

Since the integration of business process variants can be affected by contextual information [7], it is important to define a decision mechanism that determines the set of variants to be integrated wherein their execution will not be affected from variants updates that have been previously performed.

The aim of this paper is to propose the development of a context model for Business Process Integration that considers the propagation of reference model process changes towards process variants based on following stages: (1) conceptualization of process variants, by building an ontology, (2) identification of contextual elements, (3) definition of operational and organizational constraints; (4) formalization of contextual situations in the variants, and (5) construction of a reasoning mechanism for searching and retrieval of process variants.

The rest of the paper is organized as follows: Sect. 2 presents the conceptual framework of this research. Section 3 reviews some related works analysis. Section 4 describes the context model of process variants proposed for business process integration. In order to validate the proposed context model, a case study is described in Sect. 5, and finally Sect. 6 presents the conclusions and future work.

2 Conceptual Framework

Following are the main concepts related to business process integration and model process variants.

2.1 Definition of Business Process Integration

The Business Process Integration -BPI- is defined as "*the consolidation of a set of process variants partially or permanently, in order to analyze and propagate changes from its reference model, related to their structural, logical and organizational information*".

The result of BPI is represented by the description of a process model, which should allow the propagation of changes and ensuring a minimum amount of change, which is considered as the similarity measure between the process involved and the variants generated.

Figure 1 presents an example of business processes integration wherein starting from two processes (Process A and Process B) a third process (Process C) is generated to be able to replace both A (solid lines) and B (dashed lines). The first is composed of two sequential activities and the second by two separate mandatory activities.

Therefore the process generated is composed of an activity 'A', a condition and an activity of the process B. In this case the activities 'A1' and 'B2' are selected because are appropriate for integration. Additionally, the process B contains a conditional C1 is selected because it enriches the description of the process model generated.
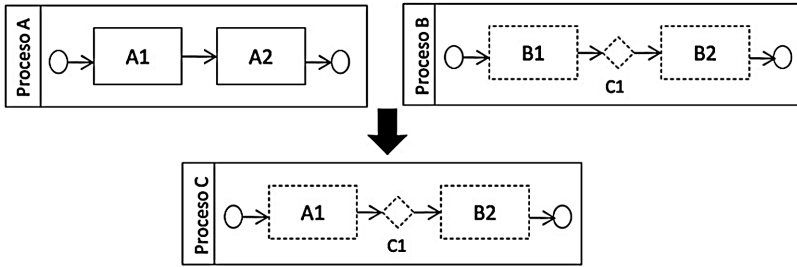
Fig. 1. Business process integration

2.2   Model Process Variants

A variant is considered as the change of the structure of a business process, specifically in instances of a reference model. Variants are classified into the following types:

Evolutions variants: Once a business process is in production, adjustments can be made in the structure, due to the recollected information during execution leading to perform redesign or optimization. Also changes may occur from the execution domain, involving an upgrade process, for example, during the merger of two companies, it may require two different versions of the same process.

Request variants: During process execution, situations can be generated by special customer requests, for which adjustments are necessary to make. For example the case making a sale to a corporate client, for which the company is prepared to do it only with small and medium enterprise, therefore requires modifying the structure of the process.

The main use of the variants is to allow the propagation of changes from a reference model. The propagation can be of two types as well: Single and Multi-modal [8], the first taken separately each of the variants and propagate the changes, while the multimodal approach obtains a single process model from consolidation variants. Each type has its advantages and disadvantages.

According to [9], change management involves three steps, (1) detection of differences, (2) analysis of their relations, and (3) resolution of differences.

3   Related Works

The propagation of changes from a reference model to its instances focuses its efforts in allowing the identification and alignment, either syntactic or semantic, among business process models [10].

Weidlich et al. present in [11] a mechanism to support the change propagation between models processes related. Basically from a given change, an abstract behavior is generated in order to identify the region of change in process instances. Thus alignment processes in inconsistent behavior is ensured.

Weber et al. define in [12] a process instance adaptation wherein common high-level changes have been classified as change patterns. Authors discuss how changes can be applied in a net system and in this way they introduce the notion of a change region.

For business process alignment, similarity metrics are often used [13] to measure the degree of difference between two processes. The main focus thereby consists in identifying their differences to attempt to minimize them.

Once processes are aligned then it is proceed to perform the integration. La Rosa et al. consider in [14] the construction of sets of multiple models (called merged models) as well as intersections. Merged models are intended for analysts who wish to create a model that subsumes a collection of process models, typically representing variants of the same underlying process.

Process variants exhibit a major challenge in story generation, search and retrieval of information from events. Ruopeng et al. present in [15] a tool for discovering of preferred variants through effective search and retrieval based on the notion of process similarity, where multiple aspects of the process variants are compared according to specific query requirements.

Ploesser et al. propose in [16] a model for the design of a context-sensitive process which highlight relevant issues such as context modeling, context learning, taxonomies of context and process operations associated with the business context.

Saidani and Nurcan propose in [17] a context model to delegate tasks based on skills and experience wherein an operation can be executed or not by a role, depending on changes in the requirements that have been previously expressed by the customers of the process.

Bernal et al. present in [18] a constraint-based model for implementation and invocation of activities in order to perform decomposition of objectives and thus to obtain rules related to own model activities. Tavares et al. propose in [19] an architectural approach for managing the flexibility of business processes considering contextual information being gathered from the environment and thus attempt to improve and automate adaptation mechanisms.

Mattos et al. show in [20] a formal background of business process models based on conceptual models approach. This formalization seeks to identify the status of an activity in order to support decision-making during the execution of a process.

It is important to develop solutions that provide the kind of contextual dependence when integrating process variants, since the spread of non-relevant changes to the structure and semantics of the model process integration could affect its performance and representation.

4   Model Proposed

The context model for Business Process Integration proposed considers the propagation of process changes from reference model towards process variants based on five stages that are described as follows. However, before developing all stages some useful definitions will firstly be presented.

4.1 Definitions

Some useful definitions for context model development such as reference model, activities, transition, domain, among others, appear as follow.

- **Reference Model.** Giving MR(A, Pr, Gw, T, Ev, Rv, D) the reference model is composed of A: Activities, Pr: Participants, Gw: Gateways, T: Transitions, Ev: Evolution Variants, Rv: Request Variant, and D: Domain.
- **Activity.** An activity is denoted by A(n, ctx, dom, pr, t), where n:Name, ctx: Context, dom: Domain, pr: Participant y t: Activity type, for instance 66:Begin, 99: End, 1:Activity.
- **Transition.** A transition is given by Tr(Ao, Ad, name). Ao: Activity Source and Ad: Activity Destination.
- **Domain:** A domain is composed of concepts and contextual tags. These tags determine the concept relevancy in a given context. Contextual tags are denoted by EtqCtx (concept).
- **Context:** It is denoted by Ctx (st, ent, c) where st are Situations, ent: Contextual Entities, c: Constraints. Note that it is necessary to determine what are the concepts that are affected by the context from the domain context labels.
- **Contextual Element:** A contextual element deals with aspects of each activity such as person, place, and applications involved in the implementation. Each contextual element also has contextual information, represented in its properties.
- **Situation:** A situation is given by S (Eve(A), eCtx, ICtx(eCtx), Ri), where Eve is the event that generated the variant by adding or removing activities, eCtx: Contextual Element, ICtx (eCtx): Contextual Information, and Ri: Restriction of associated information, either organizational or operational.
- **Structural Similarity:** This concept deals with the number of activities, participants (actors), gateways, transitions, and rules that make up each of the processes. In addition, a syntactic comparison is performed between process's name and its activity tags.
- **Semantic Similarity:** It seeks the degree of semantic similarity between the names of the processes and activities, which depends on the domain of execution and the context model.
- **Performance Similarity:** This concept refers to the execution time of activities and process variants. In addition, it determines the frequency with which activities are used.

4.2 Context Model Development

Five stages composing the context model development are presented as follows.

 •**Stage 1 - Conceptualization of variants.** In order to clarify the concept of variant and its relationship with contextual information associated with levels and situations of process changes, an ontology was developed using the methodology "*Ontology Development 101*" [21]. Figure 2 shows a class diagram fragment of the process variant ontology.
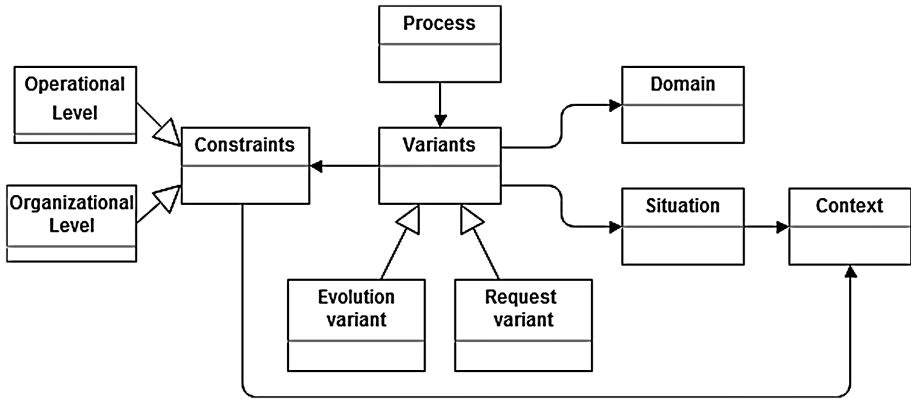
Fig. 2. Class diagram fragment of the business process variant ontology

It is important to highlight that a business process can generate several variants. Variants of a process have a relationship with execution domain information and contextual situations. In turn, depend on constraints levels, which can be operational and organizational type.

The definition of the levels and situations related to the context, can help to determine whether or not to incorporate a couple of variants to propagate. Then can be possible not propagate changes to process variants due to their constraint and contextual situations.

There are two kinds of constraint levels: (1) Operational levels of information, refer to the data that can be manipulated from the workflow process variants and its objects (activities, decision structures and actors) and (2) Organizational level, that takes into account information related to the organization and its units, which depend on internal rules and external.

• **Stage 2 – Identification of Contextual Elements.** The Stage 2 of the model proposed regards the identification of contextual elements. For doing so, a taxonomy of business process context is given by Saidani and Nurcan in [17], context elements related to constraints and levels of consolidation schemes must be defined such as availability, associated skills, experience, mental condition, physical condition, business rules, process location, affinity actors and government, and regions laws.

• **Stage 3 – Definition of Constraints.** The definition of operational and organizational constraints, useful for developing Stage 3, is associated with the levels of process execution. In this way, operational constraints must be defined in terms of the data manipulated from the workflow process variants, and also in terms of the process objects like activities, decision structures, and actors. On the other side, organizational constraints consider information related to the organization and its units that depend on internal and external rules.

• **Stage 4 – Formalization of Situations.** In addition, the levels of integration constraints can be affected by contextual elements represented by contextual situations in the variants that must be initially formalized as follows: (a) process objects and decision structures: availability, duration, execution date, frequency of use. (b) Actors:

skills, experience, age, physical condition, mental state. (c) Organizational units: affinity actors, execution location, availability, business rules and governmental laws.

• **Stage 5 – Reasoning and Retrieval.** Finally, and after process variants — represented through an ontology—, contextual elements, operational and organizational constrains, and contextual situations had been identified from Stage 1 to 4, the construction of a reasoning mechanism is needed to be developed within Stage 5 in order to search and retrieve every process variants generated. It is important to highlight that the variant retrieval process uses the three following similarity techniques: structural similarity, semantic similarity, and performance similarity.

5   Model Validation

The validation of the model is composed of two parts. The first one describes the case study and the second follows all the stages of the context model development.

5.1   Case Study Description

The model validation is applied to the process called "Academic Self-Assessment" — generally used in university scopes—, considers as a case study, which consists of the following activities: Starting, Awareness Development, Software Process Opening, Password Assignment, Weighting Development, Information Source Selection, Survey Application, Software Process Ending, Improvement Plan Design, and Accreditation Process Authorization.

There are 12 activities identified by codes as follows: 1 of Starting type (66), 1 of Ending type (99), 1 of Exclusive-Gateway type (1), 2 of Inclusive-Gateway type (2), and 8 of Normal Type (0). On the other hand, there are 20 process models among them 1 reference model and 19 variants (12 evolution variants and 7 request variants).

Table 1 exhibits the structural information of the reference model and its associated variants where P: Participants, A: Activities, G: Gateways, T: Transitions, EV: Evolution Variants, RV: Request Variants, and R: Rules.

The domain is composed of concepts and contextual tags. Concepts are the following: Committee, faculty, academic institution, self-assessment, awareness, weighting, information sources, improvement plans, high quality accreditation, tracking. On the other hand, Contextual tags are the following: Place (institution), location (awareness), person (weighting), location (surveys).

Figure 3 deploys the Reference Model of "Academic Self-Assessment".

5.2   Context Model Development Based on Stages

The context model is initially defined in terms of situations and restrictions, then the process integration objective is presented and finally the variant search and retrieval which are based on similarity criteria.

• **Stage 1 - Conceptualization of Variants.** From the previously modeled contextual ontology emerges 12 variants by evolution and 7 variants by request. Variants

Table 1. Structural information of the reference model

Code	P	A	G	T	R	EV	RV
RM01	3	12	0	11	12	12	7
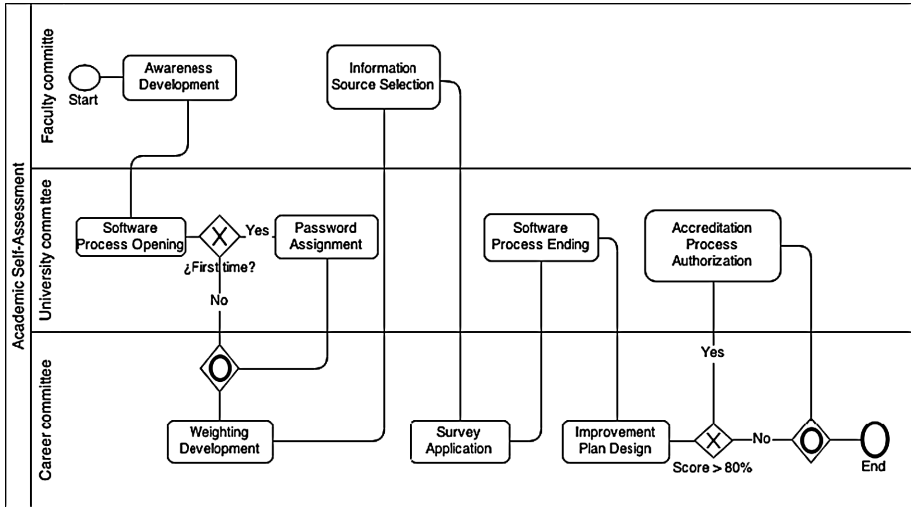
Fig. 3. Reference model of "Academic Self-Assessment"

by evolution only have events dealing with addition of activities. On the other hand, variants by request have both events corresponding to addition and elimination of activities.

Events refer to the action carried out on the process model structure, which did consider the generation of a variant. Adding events include activities and participants of the process. On the other hand, elimination events consider activities and transitions.

Table 2 gathers all the information of the variants.

• **Stages 2 and 3 - Identification of Contextual Elements and Definition of Constraints.** The following contextual elements are identified in the case study:

– eCtx1(place, institution), where the "place" is the contextual entity and "institution" is the contextual information.
– eCtx2(person, size), where the "person" is the contextual entity and "size" is the contextual information.

Constraints. From contextual properties following constraints are identified:

– R1: If eCtx1 (institution) == departmental => update (key). This constraint classifies as organizational type. It refers to whether the institution allows to "departmental" order the input key into the management system is automatically assigned each semester.
– R2: If eCtx2 (size) < 5 => activity (participant, teacher). Although the origin of the information is organizational, this constraint has a structural nature, since it directly affects the structure while adding a new participant to the process.

Table 2. Variants defined

Evolution	Request	Adding events	Elimination events
12	7	3	12

Table 3. Search and retrieval of variants

RM	V	NCF	CF
RM02 – "Academic Self-Assessment"	19	15	12

• **Stage 4 - Formalization of Situations.** The context model must be defined in terms of situations considering their contextual elements. In the case study considered situations are only identified for the MR2 – "Academic self-assessment", since starting from this model we will define the integration objective. The situations are the following:

– S1 (Elimination (first time), place, institution (place), R1: departmental institution => upgrade key). This situation arises from the elimination event of activity type Gateway "first time". This activity is focused on determining whether it was the first time performing the "self-assessment" process. If true, the input key is assigned to the management software. Otherwise, it follows using the preceding key. This situation directly depends on the organizational constraint "R1".
– S2 (addition (make tracking), person, size (person), R: Committee members < 5 - => do not track). This situation arises from the adding event of the activity 'Track' in charge of the Programme Committee and thus depends on the operational constraint "R2".

• **Stage 5 – Reasoning and Retrieval.** At first, the integration goal is defined, and then, the variant search and retrieval are performed by applying similarity criteria.

Integration Objetive. It is described in terms of the reference model structure to be affected and therefore it considers associated variants. For the case study an activity called "generate follow-up report" is attempted to be added in charge of the Programme Committee. However, it is expected that contextual model notifies that such an activity can be added, but only taking into consideration the contextual situation, related to the number of members of the Programme Committee.

Search and Retrieval of Variants. From similarity techniques and inference mechanisms those variants concerning the integration objective —that will not be affected by the change propagation— are retrieved. The retrieval process applies three types of similarity known as structural, semantic, and performance thus obtaining following results gathered in Table 3, where RM: Reference Model, V:Variants, NCF: No contextual filter, and CF: Contextual Filter.

From 19 variants of the reference model, 15 variants were retrieved accomplishing the minimum similarity degree. However, once the contextual filter was applied, 12 variants were finally obtained.

6   Conclusions and Future Work

The instance representation of reference models using both evaluation variants and variants-by-request allows determining whether generated changes are specific to the execution domain or even if they are related to the context.

The main issue of defining a context model for business process integration is to allow linking external and independent variables of the domain affecting the process performance once a change propagation has been performed starting from a reference model to its variants.

From the definition of a situation-based context model by using operational and organizational constraints the process of variant recovery is improved since contextual filtering prevents spreading unnecessary changes. However, the model presents some weaknesses that will be addressed in following issues as future work.

- Design a functional prototype that allows the interactive and automatic representation of the integration objective and process variants.
- Perform a comparative study of similarity metrics in order to select those appropriate for handling the context.
- Formalize and integrate evaluation criteria of similarity metrics to the functional prototype in order to determine minimum and maximum ranges that facilitate the evaluation among processes.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Berlin (2013)
2. Zhiqiang, Y., Remco, M., Dijkman, M.: Business process model repositories - framework and survey. Inf. Softw. Technol. **54**(4), 380–395 (2012)
3. Li, C., Reichert, M., Wombacher, A.: Discovering reference models by mining process variants using a heuristic approach. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 344–362. Springer, Heidelberg (2009)
4. Döhring, M., Reijers, H., Smirnov, S.: Configuration vs. adaptation for business process variant maintenance: an empirical study. Inform. Syst. **39**, 108–133 (2014)
5. Lohrmann, M., Reichert, M.: Modeling business objectives for business process management. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 106–126. Springer, Heidelberg (2012)
6. Kumar, A., Yao, W.: Design and management of flexible process variants using templates and rules. Comput. Ind. **63**(2), 112–130 (2012)
7. Ploesser, K., Peleg, M., Soffer, P., Rosemann, M., Recker, J.: Learning from Context to Improve Business Processes. BPTrends (2009)

8. Li, C., Reichert, M., Wombacher, A.: Mining business process variants: challenges, scenarios, algorithms. Data Knowl. Eng. **70**(5), 409–434 (2011). ISSN: 0169-023X

9. Gerth, C., Küster, J.M., Engels, G.: Language-independent change management of process models. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 152–166. Springer, Heidelberg (2009)

10. Jason, J.: Semantic business process integration based on ontology alignment. Expert Syst. Appl. **36**(8), 11013–11020 (2009)

11. Weidlich, M., Mendling, J., Weske, M.: Propagating changes between aligned process models. J. Syst. Softw. **85**(8), 1885–1898 (2012). ISSN: 0164-1212

12. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features – enhancing flexibility in process-aware information systems. Data Knowl. Eng. **66**(3), 438–466 (2008)

13. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Comput. Ind. **63**(2), 148–167 (2012). ISSN: 0166-3615

14. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Business process model merging: an approach to business process consolidation. ACM Trans. Softw. Eng. Meth. - TOSEM (2012)

15. Ruopeng, L., Shazia, S., Governatori, G.: On managing business processes variants. Data Knowl. Eng. **68**(7), 642–664 (2009). ISSN: 0169-023X

16. Ploesser, K., Peleg, M., Soffer, P., Rosemann, M., Recker, J.: Learning from Context to Improve Business Processes. BPTrends, January 2009

17. Saidani, O., Nurcan, S.: Context-awareness for adequate business process modelling. In: Proceedings of the Third IEEE International Conference on Research Challenges in Information Science, RCIS 2009, pp. 177–186 (2009). ISBN: 978-1-4244-2864-9

18. Bernal, J., Falcarin, P., Morisio, M.: Dynamic context-aware business process: a rule-based approach supported by pattern identification. In: Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, pp. 470–474, March 2010

19. Tavares, V., Lima, C., Santoro, F.: Dynamic process adaptation: a context-aware approach. In: Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design – CSCWD 2011 (2011)

20. Mattos, T., Santoro, F., Revoredo, K., Tavares, V.: Formalizing the situation of a business process activity. In Proceedings of CSCWD 2012, pp. 128–134 (2012)

21. Noy, N., McGuiness, D.: Ontology Development 101: A guide to creating tour first ontology (2001). http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html