Supporting Documentation and Evolution of Crosscutting Concerns in Business Processes

Chiara Di Francescomarino
supervised by Paolo Tonella

dfmchiara@fbk.eu - Fondazione Bruno Kessler, Trento, Italy

Abstract. Business processes can be very large and can contain many different concerns, scattered across the process and tangled with other concerns. Crosscutting concerns are difficult to find, locate and modify in a consistent way, thus making process maintenance and reuse hard, even for business experts.

In this work, we address the problem of supporting business designers when they need to document existing crosscutting concerns and when they work on their evolution. We propose to add semantic annotations from a domain ontology to BPMN process elements in order to be able, on their basis, to manually or semi-automatically mine concerns crosscutting the process. Once modularized separately from the principal process flow, we aim at supporting the consistent evolution of all the process parts they crosscut, by means of a BPMN-based rule language.

1 State of the Art

Business processes can be very large and involve many different perspectives. Business process (BP) description and modelling languages provide means for describing the main views which define the principal decomposition of the business process, such as the control flow, data flow and performers. Though allowing to represent these perspectives, the available languages are not able to capture all different types of relevant features in the process, above all if scattered across the process and tangled with one another. This lacking capacity in identification and explicit representation of crosscutting concerns (CCs) results not only in difficulties during process comprehension, but also in a challenging maintenance and evolution phase.

In order to improve process flexibility, to make maintenance and evolution easier and to improve reuse, the Business Process Community made several attempts, such as flexible and dynamic processes and generic processes [1, 2].

An aspect is a module composed by *pointcut designators* and *advices*. A pointcut provides a condition to specify a set of so called *join points*, i.e., precise points in the execution flow that the aspect intercepts, by means of some quantification mechanism. An advice, instead, specifies how to realize the concern and when in the join points of interest the concern execution has to be activated, thus realizing the AOP (Aspect Oriented Programming) *obliviousness* (the primary program ignores advices) [3].

Recently, AOP has investigated the migration of existing systems to aspects. Such transformation process involves at least two steps: (1) identification of scattered and tangled concerns (i.e. aspect mining); (2) migration of crosscutting concerns into actual aspects (i.e. aspect refactoring). A number of different techniques have been developed to (partially) automate aspect mining. Some of them rely on Formal Concept Analysis (FCA).

The AOP approach has already been applied to a specific process executable description language, BPEL (Business Process Execution Language). AO4BPEL [4] is a dynamic aspect oriented extension of BPEL, enriched with concepts for describing pointcut designators (XPath expressions to locate the places where the aspect is applied) and advices (fragments of BPEL code). A similar approach, mainly differing from AO4BPEL in the advice language has been proposed by Verheecke et al. [5]. Padus [6] is yet another aspect-oriented BPEL extension, but without dynamic weaving and improving portability.

All the cited works mainly focus on the developers' perspective, without considering that, in practice, business designers prefer a higher level modelling notation, such as BPMN (Business Process Modelling Notation). Moreover, explicit decomposition of the process into totally separate modules for the aspects, may hinder, instead of simplifying, process design and comprehension.

In order to address the SoC (Separation of Concerns) problem in general purpose code, several tools (e.g. JQuery [7]) support programmers in source code navigation and concern localization. We propose a similar approach for BPs. In order to help business experts to understand and maintain BPs, we introduce a visual language for querying BP models. However, since the raw syntactical information available in BPMN diagrams is not enough for characterizing the concerns being mined, we add semantic expressiveness to models by means of semantic annotations.

BP semantic annotation techniques have already been proposed by several authors (e.g. [8]), for different process description languages and addressing different issues. The main purpose of semantic annotations in this work is another one. By allowing to manually (via the query language) or semi-automatically (via FCA) mine for business domain CCs and to reason about domain ontology concepts, they aim to support concern documentation and domain constraint compliance.

This work supports business experts in BP crosscutting concern documentation and evolution, thus contributing to process understandability, maintainability and evolvability. In order to achieve this goal, we devise a four-fold contribution to the state of the art: (1) we propose to enrich BP models with semantic annotations and suggest to support the business designer both in the annotation phase, by suggesting correct annotations or verifying user-defined ones, and in the related domain ontology building and/or enrichment activity; (2) we define an easy-to-use, visual query language for querying BPs and we support the business user in formulating queries and visualizing the results; (3) we investigate a technique for the automatic mining of candidate CCs in BPs; (4) we propose a method to document CC evolution and, whenever any CC changes, to update all modules it crosscuts, in an automatic or semi-automatic way.

The paper is organized as follows: in Section 2 we give an overview of our research directions; in Section 3 we detail one of such directions (the visual query language for exploration and documentation of CCs in BPs); finally, conclusions and future works are presented in Section 4.

2 Crosscutting Concern Documentation and Evolution

Although BPMN allows the explicit representation of the main views of a process, it often leaves implicit the crosscutting features. Knowledge about the crosscutting functionalities pertains to the BP semantic (vs. syntactic) description, but ”BPM doesn’t provide a uniform representation of an organization’s process space as a whole on a semantic level which would be accessible to intelligent queries and inference” [8].

Business Process Semantic Annotation. In order to supply BPs with semantic information from the business domain, we propose to semantically annotate them with concepts taken from a business domain ontology. For the semantic annotation of BPMN elements, we use a standard BPMN textual annotation, augmented with an ”@” before the semantic concept. Such annotations open to the possibility of reasoning about semantic properties, thus facilitating the compliance with possible predefined constraints, the process of manual or semi-automatic crosscutting concern mining and the consistent update of all the modules crosscut by a modified documented feature.

An example of a semantically annotated BPMN diagram is shown in Figure 2. It represents a process for realizing an assembled product starting from three raw products. All the flow objects in the process are annotated with concepts taken from a “buying and selling” domain ontology.

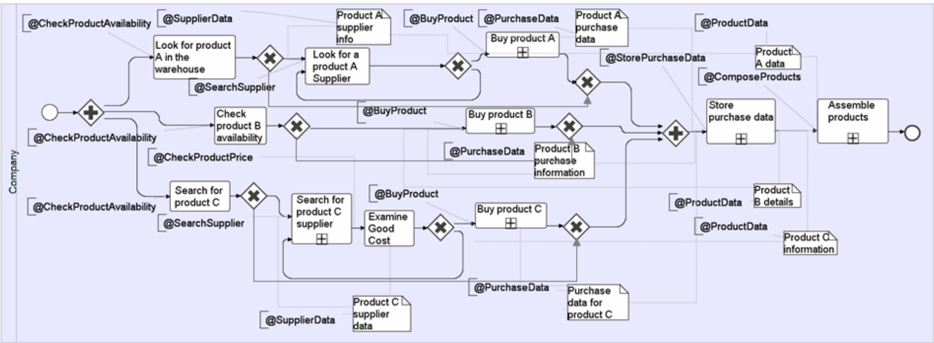


Fig. 1. An example of a semantically annotated BPMN process.

Clearly business designers need to be supported in the semantic annotation phase. On one hand they can be supplied with annotation suggestions compliant with possible predefined constraints and, on the other, mechanisms for the automatic compliance verification of user defined annotations with constraints can be provided. Moreover, they will also be supported in ontology building or enrichment by means of dedicated tools and techniques. We plan to use an ad-hoc linguistic analysis for ontology extraction from process entity labels, since available ontology learning techniques are not applicable to small textual corpora such as those represented by process labels.

BPMN Visual Query Language. In order to support crosscutting concern documentation, understanding and evolution, we propose the use of a query language able to quantify on the BP elements, thus localizing the concerns of interest. By carefully considering usability of the query language (it is supposed to be used by business designers), we chose a language as close as possible to what business experts already know: BPMN itself. We slightly extended BPMN to BPMN VQL (Visual Query Language), so as to make it usable for query purposes as well. BPMN VQL is described in more detail in Section 3.

Concern Mining. The support provided by query languages and concern browsers is of fundamental importance, but it still involves substantial effort from the users, thus breaking the ease-of-use requirement of business designers. It can be complemented by an automated concern mining tool, which automatically suggests candidate CCs to designers, both structural (e.g. workflow patterns) and business domain oriented (e.g. purchasing activities). In order to achieve this goal, we investigated novel aspect mining techniques for BPs, based on FCA (more details in [9]).

Crosscutting Concern Aspectization. Once CCs are known and their existence is documented (through a collection of queries), their evolution too has to be supported and documented. Changing a crosscutting concern, due to its scattered and tangled nature, is not a trivial job. The update has to be applied in a consistent way to all the concern occurrences and the tangled subparts. Manually executing the change is a tedious and error-prone task. In order to support business designers, we propose to use a BPMN-based rule-language, that extends BPMN VQL with an “update” part in order to support BP manipulation at selected points. The rule could be either weaved and stored, in order to only document the concern changes (and potentially re-weaved whenever the designer wants) or it can be kept alive during workflow editing, in order to detect any possible violation of the rule, giving rise to a warning and a (semi-)automated fix (more details in [9]).

3 BPMN Visual Query Language

The exploration of large BPs and the retrieval of interesting CCs can be a difficult and time consuming task for business experts. In order to support them, we propose a query language able to quantify over the BP elements, localize interesting concerns and, once identified, present them to users by visually highlighting their occurrences in the process diagram. Since a critical issue of the query language is usability, the proposed language is a *visual* language based on BPMN, the standard language business designers use for representing BPs.

The BPMN Visual Query Language (BPMN VQL) allows to formulate queries by using standard BPMN graphical notation and new specific operators. Moreover, the language provides a different notation to distinguish between the "matching" part of the query, that determines the criterion to match, and the "selection" part (characterized by darker background, thicker lines and bold font style), that allows to visualize only the selected subpart of the matching result.

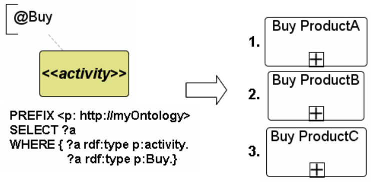
In the following subsection we present some examples of uses of the BPMN VQL referring to the product assembly process shown in Figure 2. In order to formalize the queries and give them a precise semantics, we provide their translation into SPARQL, an RDF-based query language More details about the language can be found in our Technical Report ([9]).

3.1 BPMN VQL Use Cases.

The BPMN VQL can be used to query for any concern of interest that is part of the business process design. A simple example is provided in Figure 2.

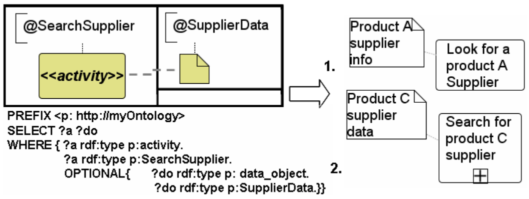
For particularly relevant concerns, it makes sense to store the query as part of the design artefacts and to keep it as a precise documentation of the design choices related to the specific concern. For example, the business designer, in order to be compliant with specific privacy laws or business strategies, could be interested in storing the concern related to the product supplier search management as a relevant design feature. The query is shown in Figure 3.

Moreover, availability of design artefacts documenting the relevant concerns of a BP is particularly useful when the process design evolves, due to new customer requests or to environmental changes.



```
PREFIX <p: http://myOntology>
SELECT ?a
WHERE { ?a rdf:type p:activity.
             ?a rdf:type p:Buy.}
```

Fig. 2. Query asking for all the activities (both tasks and subprocesses) that buy something (left) and query result (right).



```
PREFIX <p: http://myOntology>
SELECT ?a ?do
WHERE { ?a rdf:type p:activity.
             ?a rdf:type p:SearchSupplier.
             OPTIONAL{      ?do rdf:type p: data_object.
                              ?do rdf:type p:SupplierData.}}
```

Fig. 3. Query asking for all the activities that search suppliers and, if any, the "SupplierData" data objects they produce (left). On the right the query results.

4 Conclusion and future works

We aim at realizing a framework able to support business users in BP CC documentation and evolution. Up to now we have: (1) studied and partially implemented techniques both for semi-automatic semantic annotation of BPs and for domain ontology building and enrichment; (2) defined a BPMN VQL for querying BPs; (3) investigated, adapted and developed approaches for semi-automatic mining of business domain CCs; (4) studied rule languages for CC evolution.

In the future we will complete the implementation of tools for the user support in BP semantic annotation and domain ontology building; in the visual querying of business processes; in the semi-automatic mining of CCs; and in their evolution. Moreover, we will continue our research in the following directions: (1) linguistic analysis and normalization techniques, aimed at ontology construction; (2) ontology reasoning and constraint definition and compliance; (3) FCA analysis criteria, to provide more automation during the identification of concern candidates; (4) new mining techniques, based on structural properties of the workflow; (5) weaving and re-weaving of aspectized concerns; (6) validation of the benefits gained by applying the suggested methodologies to case studies involving business users; (7) extension to the process execution phase.

References

1. van der Aalst, W.M.P.: Generic workflow models: How to handle dynamic change and capture management information? In: Conference on Cooperative Information Systems. (1999) 115–126
2. A. Schnieders, F.P.: Variability mechanisms in e-business process families. In: 9th International Conference on Business Information Systems (BIS 2006), W. Abramowicz, H. Mayr (2006) 583–601
3. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley (October 6, 2004)
4. Charfi, A., Mezini, M.: Aspect-oriented web service composition with AO4BPEL. In: Proceedings of the 2nd European Conf. on Web Services (ECOWS). Volume 3250 of LNCS., Springer (2004) 168–182
5. Verheecke, B., Cibràn, M.A., Jonckers, V.: Aspect-oriented programming for dynamic web service monitoring and selection. In Zhang, L.J., ed.: ECOWS. Volume 3250 of LNCS., Springer (2004) 15–29
6. Braem, M., Verlaenen, K., Joncheere, N., Vanderperren, W., Straeten, R.V.D., Truyen, E., Joosen, W., Jonckers, V.: Isolating process-level concerns using padus. In Dustdar, S., Fiadeiro, J.L., Sheth, A.P., eds.: Business Process Management. Volume 4102 of LNCS., Springer (2006) 113–128
7. Janzen, D., De Volder, K.: Navigating and querying code without getting lost. In: AOSD '03: Proceedings of the 2nd int conf on Aspect-oriented sw development, New York, ACM Press (2003) 178–187
8. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: a vision towards using semantic web services for business process management. (2005) 535–540
9. Tonella, P., Di Francescomarino, C.: Business process concern documentation and evolution. Technical report, Fondazione Bruno Kessler (FBK-IRST), Trento (2008) http://se.fbk.eu.