Collaborative Specification of Semantically Annotated Business Processes

Marco Rospocher, Chiara Di Francescomarino, Chiara Ghidini,
Luciano Serafini, and Paolo Tonella

FBK–irst, Via Sommarive 18 Povo, I-38100, Trento, Italy
{rospocher,dfmchiara,ghidini,serafini,tonella}@fbk.eu

Abstract. Semantic annotations are a way to provide a precise meaning to business process elements, which supports reasoning on properties and constraints. The specification and annotation of business processes is a complex activity involving different analysts possibly working on the same business process.

In this paper we present a framework which aims at supporting business analysts in the collaborative specification and annotation of business processes. A shared workspace, theoretically grounded in a formal representation, allows to collaboratively manipulate processes, ontologies as well as constraints, while a dedicated tool enables to hide the complexity of the underlying formal representation to the users.

1 Introduction

Semantic annotation of business processes allows analysts to give a precise meaning to the process elements they are modelling and enables automated reasoning on the process and its properties. However, semantic annotation involves skills and competences that go beyond the typical background of a business analyst, such as ontology construction and extension, formulation of queries and constraints in descriptive logics. Moreover, the semantics of a business process is almost never unique. Different view points on the process elements and properties bring in different concepts and constraints. For example, for a security expert relevant concepts are *sensible data* or *authentication*, while for a warehouse expert important notions are *product supplier* or *order*.

Integrating and reconciling different views of the same process is not an easy task, and available tools for process construction (e.g., Hyperwave, InterPROM) provide functionalities for collaborative process definition. The problem becomes even harder when the process elements are given a precise semantics by means of an ontology. In fact, incremental ontology creation and extension is expected to be carried out in parallel with the incremental definition of the process. Available tools do not provide any explicit support to the complex activity of collaborative ontology creation/extension, neither they support the related activities of collaborative semantic annotation of process elements and constraint specification.

In this paper, we present a framework for the collaborative specification of semantically annotated business processes. The framework takes advantage of a shared workspace to store the main artefacts that are manipulated collaboratively, i.e., (1)

S. Rinderle-Ma et al. (Eds.): BPM 2009 Workshops, LNBIP 43, pp. 305–317, 2010.

process; (2) ontology; and, (3) constraints. Analysts work on these artefacts concurrently, without any notion of ownership (so they can modify artefacts initially created by others). Conflicts are managed through mutually exclusive lock, and disputes causing instabilities are resolved through discussion forums. To hide the complexity of the underlying formal ontology and descriptive logics formulas the framework includes a dedicated tool.

We have conducted a case study in which four analysts with different competences have collaboratively defined an on-line shopping process. The case study is described as a sequence of snapshots, which highlight the interactions among the work performed by different analysts on different parts of the process.

The paper is organized as follows: Section 2 describes the proposed framework and Section 3 presents the case study. Finally Related Works and Conclusions are presented.

2 Framework

We propose a framework for the collaborative specification of semantically annotated business processes based on the notion of shared workspace illustrated in Figure 1. This workspace makes the artefacts necessary for this activity visible to all actors who are contributing to the definition of the annotated process. These artefacts are then collaboratively developed by the actors, according to their role. Typical actors working concurrently on the business process specification are the analysts who are expert of different aspects of the business. For instance, an organization may ask a customer relationship expert, a logistic and warehouse expert, a payment expert, and a security expert to collaboratively define an on-line shopping process. These different analysts may specify different parts of the process. They can also modify the parts defined by others, so as to make them consistent with their own modifications. The usage of a collaborative workspace aims at supporting the integration of different perspectives.

The artefacts manipulated in the collaborative workspace are the process itself, a domain ontology used to annotate the process elements, and a set of constraints which makes use of both the business process and the domain ontology. The collaborative workspace includes also a “read only” part composed of a BPMN ontology and some BPMN axioms, which are used by the collaborative framework to give a precise semantics to the process elements.

Analysts working collaboratively on a given business process carry out four main activities: (1) incremental process construction; (2) ontology definition or extension; (3) constraint specification; and (4) addition of semantic annotations. These four activities are illustrated in detail in the final part of this section. What is important to note here is that there is no precedence relationship or prescribed workflow in the execution of these four activities. They can be executed concurrently, in any order, and the collaborative workspace must support multiple analysts working on different artefacts and carrying out multiple activities at the same time. To realize such a concurrent working environment we need to address two main problems: (i) concurrent modification of the same artefact; and (ii) instabilities deriving from incompatible modifications that are repeatedly done and undone. For the first problem, we adopt a solution, widely used to address concurrent database accesses, which is based on the acquisition of a *lock*

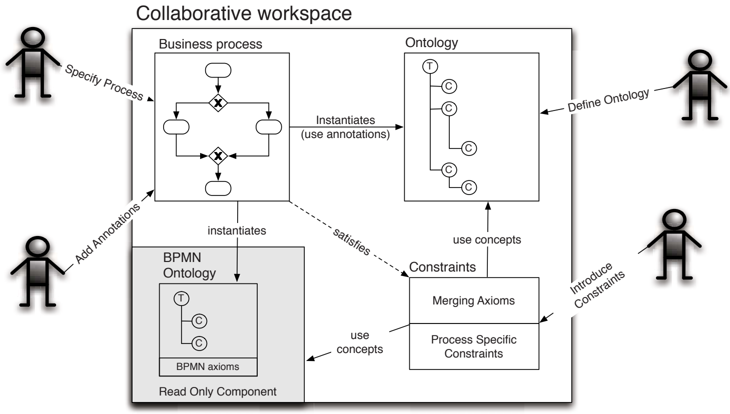


Fig. 1. The collaborative workspace

(i.e., a mutually exclusive access). When an analyst starts working on an artefact, she acquires the lock on it and the workspace provides such an artefact to the other analysts in "read only" mode. When the editing is finished, the analyst commits the changes. This produces an update of the workspace, which triggers an automated verification of the constraints on the new version of the workspace. It also results in the release of the lock on the changed artefact, which becomes available to the other analysts. Incompatible changes are instead managed by resorting to solutions widely used in collaborative content management systems (e.g., Wikipedia). Once a problem on some artefact modification is detected (with the help of automated change monitoring and analysis tools), the first attempt to solve the conflicts consists of initiating a discussion forum, which involves the contributors who made the conflicting changes, as well as experts about the object of the dispute. The project or team leader is in charge of starting such a forum. If no consensus is achieved by the discussants participating in the forum, the solutions used in collaborative content editing involve voting (with different voters having different weights) and/or authoritative decisions by an expert or by the project leader.

Another key characteristic of the collaborative workspace is that the four components illustrated in Figure 1 together with their inter-connections are theoretically grounded in a formal representation of semantically annotated business processes illustrated in [1]. In that work, we have defined and implemented these components as parts of a modular Business Processes Knowledge Base (BPKB), expressed using the semantic web language OWL, based on Description Logics [2]. The illustration of this formal representation is out of the scope of this paper. Nevertheless it is important to note here that an alignment between the informal representation provided in the workspace and their underlying formal representation is maintained by the tool implementing the workspace, as described in Section 2.5.

2.1 Process Construction

The main purpose of the collaborative workspace is to obtain annotated Business Process Diagrams (BPDs) specified using the Business Process Modeling Notation

(BPMN)[1]. The collaborative framework provides instruments for the graphical specification of BPDs. In addition to the graphical representation, each element of the process is also represented by means of a textual template. This is used to record additional properties of the element, such as its description, its annotations, or the logs of a discussion carried on to resolve a conflict on the element itself.

Formally, each BPD element is considered as an instantiation of an element specified in the "read only" BPMN ontology[2]. For instance a specific gateway in the process being drawn is considered by the system as an instantiation of the element Gateway in the BPMN ontology. This instantiation is automatic and transparent to the analysts' activity, but it is necessary to give a precise semantics to the process elements.

2.2 Ontology Construction

The domain ontology is necessary to give a precise semantics to the terms used to annotate business processes. This ontology is typically constructed together with the process and the collaborative workspace supports incremental process and ontology construction. In fact, even if existing domain ontologies can be used to annotate processes, they often need to be adapted to the specific needs of the process being designed. Ontology construction is also a collaborative activity. For instance the security expert may specify a portion of the process concerning her expertise and at the same time she may introduce the concepts needed for the security aspects of the business process. Other analysts may refine or extend the ontology later, with more concepts, as soon as the process grows. In addition, top level (upper) ontologies can be used as the starting point, to be refined later during process definition or to specify constraints as we describe in Section 2.3.

The domain ontology is formally represented in OWL, but analysts interact with this artefact by using graphical and natural language templates which do not expose the formal ontology structure explicitly.

2.3 Constraints Definition

The collaborative workspace makes use of constraints to ensure that important semantic properties of process elements are satisfied. To support the analysts in this activity, it is possible to define a set of predefined templates in which (constrained) natural language is used to express the constraints. These templates are then formally translated to DL axioms. We distinguish among two different kinds of constraints: merging axioms and process specific constraints.

Merging axioms. These constraints are expressions used to state the correspondence between BPMN elements and elements of the domain ontology. Intuitively they define criteria for correct / incorrect semantic annotations. Examples of these criteria are:

---

[1] OMG - BPMN v1.1 - `http://www.omg.org/spec/BPMN/1.1/PDF`

[2] We assume availability of a BPMN ontology such as the one described in our previous work [1] and available online at: `http://dkm.fbk.eu/index.php/BPMN_Ontology`.

A BPMN data-object *can be annotated only* with objects of the domain ontology (1)

A BPMN activity *can be annotated only* with activities of the domain ontology (2)

A BPMN sub-process *cannot be annotated* with atomic activities of the domain ontology (3)

The action "to_manage" *can be used only to annotate* BPMN sub-processes (4)

Expressions (1)–(3) describe "domain independent" criteria as they relate elements of BPMN, such as data-objects, activities or sub-processes to elements of a top-level ontology, such as DOLCE [3]. These kinds of constraints can be thought of as "default" criteria for correct / incorrect semantic annotations. In this case DOLCE is provided as a "default" component of the ontology in the collaborative workspace. Note that these "default" criteria could still be modified by the analysts to reflect the actual annotation criteria for the specific domain at hand, although these changes should be agreed upon and justified carefully before the start of the annotation process. Other expressions, such as (4), are instead domain specific as they constrain the usage of a specific term, the action "to_manage", to annotate certain BPMN elements. Another characteristic of the expressions above is that they can describe positive constraints (see (1), (2) and (4)) or negative constraints (see (3)) for annotations. Finally these expressions can constrain a BPMN element to a domain specific element, as in (1)–(3), or vice-versa as in (4). To allow the business analysts to specify these kinds of positive and negative annotation criteria, in [1] we have introduced four different constructs:

- *annotatable only by.* The merging axiom $x \xrightarrow{\mathsf{AB}} y$ expresses that a BPMN element of type x can be annotated only with a domain specific concept equivalent or more specific than y;
- *not annotatable by.* The merging axiom $x \xrightarrow{\mathsf{nAB}} y$ expresses that a BPMN element of type x cannot be annotated with a domain specific concept equivalent or more specific than y;
- *annotates only.* The merging axiom $y \xrightarrow{\mathsf{A}} x$ expresses that any domain specific concept equivalent or more specific than y can be used to denote BPMN elements of type x;
- *cannot annotate.* The merging axiom $y \xrightarrow{\mathsf{nA}} x$ expresses that any domain specific concept equivalent or more specific than y cannot be used to denote BPMN elements of type x.

For instance, expression (1) can be represented with the merging axiom $\mathsf{data\_object} \xrightarrow{\mathsf{AB}} \mathsf{object}$ which in turn is formally represented with the DL statement $\mathsf{BPMNO{:}data\_object} \sqsubseteq \mathsf{BDO{:}object}$, where BPMNO and BDO are labels used to indicate the BPMN ontology and the domain ontology respectively.

The formal representation of the merging axioms allows reasoning to: (i) check whether the annotations satisfy or violate the annotation criteria expressed by the merging axioms, and (ii) identify the list of admissible annotations that can be suggested to the analysts in the collaborative framework. If a violation of a constraint occurs, explanation techniques similar to the ones described in [4] can be used to provide an indication of what went wrong and can be used by the analysts to repair the annotation or to trigger a revision of the merging axiom(s).

Process specific constraints. These constraints are expressions used to state specific properties that apply to the process under construction. Differently from merging axioms these expressions can have many different forms to match a variety of different properties of the process. In this paper we focus on two types of process specific constraints: (i) precedence relationship constraints, and (ii) BPMN constraints.

Precedence relationship constraints express restrictions over the sequence of activities contained in a process. For instance, in an on-line shopping process, the security expert may wish to introduce properties related to privacy, so that before providing any kind of sensible data, the customer has to read the company's privacy policy related with personal data management. This can be done by imposing that all activities annotated with the concept to_provide_sensible_data are always preceded by an activity annotated with the concept to_read_policy as follows:

$$\text{to\_provide\_sensible\_data } \textit{always\_preceded\_by} \text{ to\_read\_policy} \quad (5)$$

A similar constraint, holding for messages, is: *preceded_by_message_from*. Constraint (5) can be formalized in DL by means of two statements

$$\text{BDO:to\_provide\_sensible\_data} \sqsubseteq \forall \text{BPMNO:connect}^{-}.\text{BDO:to\_read\_p}^{*}$$
$$\text{BDO:to\_read\_p}^{*} \equiv \neg \text{BPMNO:se} \sqcap (\text{BDO:to\_read\_policy} \sqcup \forall \text{BPMNO:connect}^{-}\text{BDO:to\_read\_p}^{*})$$

where BPMNO:connect is the transitive closure of the connections provided by the connecting elements contained in the BPMN ontology (see [1]) and BPMNO:se denotes the start event element.

BPMN constraints are expressions which are used to impose additional restrictions over the semantics of BPMN. In other words, these are constraints used to impose limitations on the usage of certain BPMN elements. Examples are:

Inclusive_gateway cannot be used in the business domain (6)

Each gateway must have at most 2 outgoing gates (7)

Differently from the precedence relationship constraints, these expressions do not depend upon the domain ontology or the annotations of the different BPMN elements. Nevertheless they also depend upon the specific business domain to be modelled and not from the specification of BPMN. For instance the constraint (7) could be specified by the customer relationship expert to keep the structure of an on-line shopping process simple and to limit alternative choices available to the customer. Constraints (6) and (7) can be formally expressed by means of the DL statements $\text{BPMNO:inclusive\_gateway} \sqsubseteq \bot$ and $\text{BPMNO:gateway} \sqsubseteq (\leq 2)\text{BPMNO:has\_gateway\_gate}$.

2.4 Process Semantic Annotation

Analysts are required to annotate process elements with concepts taken from the domain ontology. The collaborative framework allows the analysts to associate elements of the domain ontology with the BPMN elements that are supposed to refer to that particular concept. This is done in the natural language based template which describes the element to be annotated. To simplify the task, the collaborative framework provides also suggestions, and presents the analysts with a list of admissible annotations, based on the constraints specified upon it.

Formally, an annotated process element is also an instance of the domain ontology concept it is annotated with. Thus if a certain activity is annotated with the concept to_provide_sensible_data, then that activity is an instance of Activity concept in the BPMN ontology but also of to_provide_sensible_data concept in the domain ontology.

2.5 Tool

To support the collaborative specification of semantically annotated business processes we are currently developing BP-MoKi, a collaborative tool based on Semantic MediaWiki (SMW)[3]. Inspired by the work presented in [5], BP-MoKi extends SMW offering specific support to edit and semantically annotate business processes, and it allows to build an OWL implementation of the BPKB.

The main idea behind BP-MoKi is that the ontologies and the process are represented as a collection of interrelated wiki pages connected by typed links. A wiki page is associated to each concept of the BPMN and Business Domain ontologies, and to each element of the Business Process. A typical page contains: (1) an informal description of the element in natural language (images or drawings can be used as well), whose purpose is to document and clarify the models to users not familiar with their formal representation; (2) a structured part, where the element is described by means of triplets of the form *(subject, relation, object)*, to represent the intra/inter-connection between the elements of the models (e.g., the subclass relation between elements of ontologies). This part is used to automatically create the formal representation in the BPKB.

To support creation and editing of a business process in BPMN, BP-MoKi integrates Oryx[4], a state of the art collaborative web-based tool for the graphical modelling of business processes. To each element of the process corresponds a page in BP-MoKi where a semantic annotation of the element can be added, and additional information/documentation about that element can be inserted.

To support creation and editing of the Business Domain ontology, BP-MoKi provides ontology modelling functionalities similar to the ones described in [5]: among them, we have *import functionalities*, which allow to set up BP-MoKi with available business domain knowledge, *editing functionalities*, which provide the basic support for creating, editing and deleting ontology elements, and *visualization functionalities*, which allow to produce different types of graphical overviews of the ontology.

Currently, BP-MoKi allows to specify the merging axioms (via a form included in the pages associated to the concepts of the BPMN and the Business Domain ontology), and we are working to extend the support to the other constraints described in the paper.

3 Use Case

In this section, we describe the definition and the semantic annotation of an on-line shopping process. Four domain experts are involved in its collaborative creation: (1)

[3] See http://semantic-mediawiki.org

[4] See http://bpt.hpi.uni-potsdam.de/Oryx

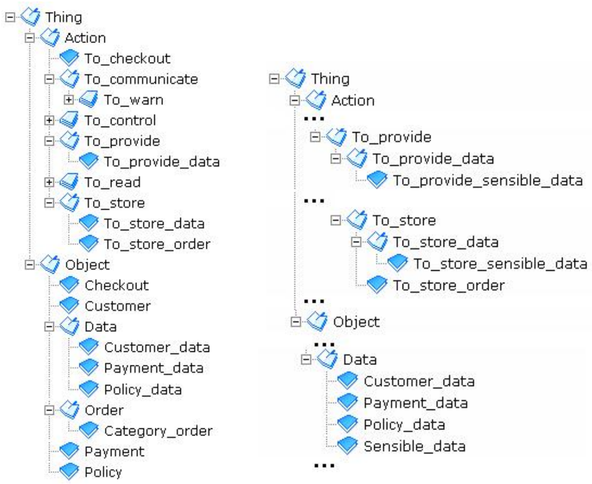


Fig. 2. Ontology evolution

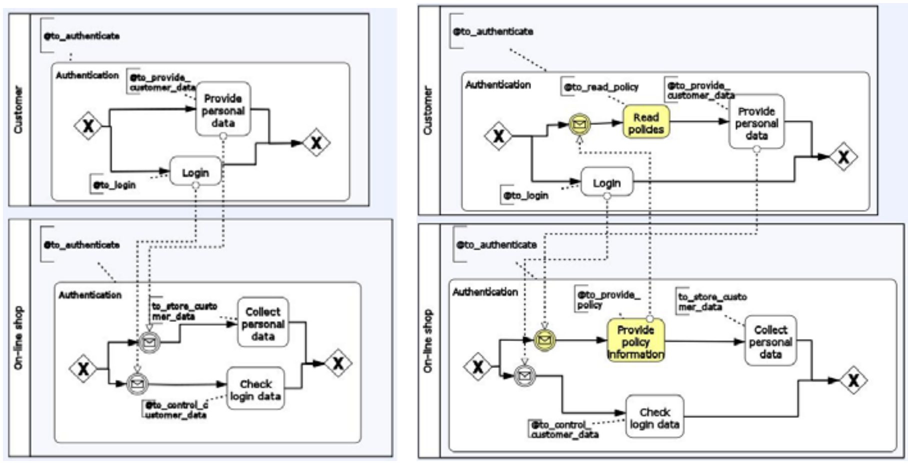
a *customer* relationship *expert*, responsible for the product presentation and selection; (2) a logistic and *warehouse expert*, responsible for the warehouse management (e.g. availability checks and order preparation); (3) a *payment expert*, responsible for the checkout process; and (4) a *security expert*, responsible for privacy and security aspects.

In the following we comment snapshots, captured at different times, of the artefacts in the workspace, providing an example of how collaborative process construction may work in practice.

Snapshot 1: security constraint specification. Initially, the security expert introduces some constraints related to privacy. Before providing any kind of sensible data, the customer has to read the company's privacy policy related with personal data management. Symmetrically, before storing any sensible data, the on-line shop application must show the company's privacy policy to the customer. We assume that an initial domain ontology (left in Figure 2) was previously imported into BP-MoKi. However, the initial ontology does not contain the concept sensible_data so that the security expert needs first of all to modify the ontology, by introducing the new concepts: sensible_data, to_store_sensible_data and to_provide_sensible_data (right in Figure 2). Once the lacking concepts are available, the following three constraints ($C1$, $C2$ and $C3$) can be added to the process specific constraint set:

($C1$) to_provide_sensible_data *always_preceded_by* to_read_policy;
($C2$) to_store_sensible_data *always_preceded_by* to_provide_policy;
($C3$) to_read_policy *preceded_by_message_from* to_provide_policy.

Snapshot 2: modelling presentation and selection. The customer expert starts drawing the process in BP-MoKi, by modelling the optional authentication activity as well as the information exchange between the customer and the on-line shop (i.e. product presentation, browsing and selection). Using the ontology artefact edited by the security expert, the customer expert can semantically annotate his process elements. He



(a) (b)

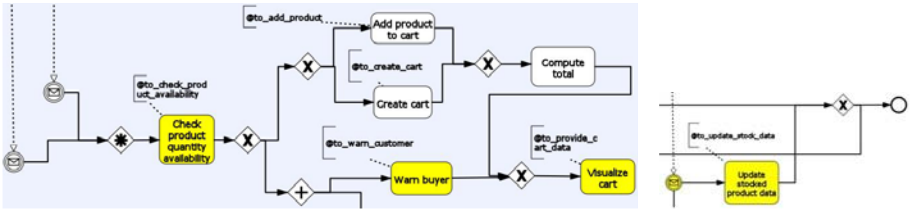
Fig. 3. Authentication sub-processes before and after inconsistency fixing

extends the ontology with specific concepts as, for example, to_provide_customer_data and to_store_customer_data. The customer expert's authentication sub-process is shown in Figure 3(a). Semantic annotations are visualized as standard BPMN textual annotations preceded by the symbol "@".

Snapshot 3: managing sensible data. At this point, in our scenario, the security expert realizes that customer_data and payment_data are actually sensible_data, and therefore she modifies the ontology in BP-MoKi by introducing an is_a relationship between them. As soon as she commits the change, the automatic constraint check will detect an inconsistency in the workspace, due to the violation of the constraints $C1$ and $C2$. The customer expert will fix the problem, by introducing the activities annotated with to_read_policy and to_provide_policy_data in customer and on-line shop pools, respectively (as shown in Figure 3(b)).

Snapshot 4: modelling warehouse management. When the warehouse expert is called to design her part of the process, besides modelling the sub-process related to ordering missing or scarse products, she may notice that no check about the actual product availability in the warehouse is performed. Therefore she modifies the customer expert's process, by introducing in the control flow new activities realizing this kind of control. The new activities and decision points are depicted with a dark background in the process fragment in Figure 4(a).

Snapshot 5: discussion forum to rename a concept. The warehouse expert needs also to extend the ontology with specific concepts (e.g. to_store_category_order). However, when annotating the activity for checking the product availability in the warehouse, she notices that the to_control concept is already in the ontology. In order to avoid to introduce too many synonym concepts in the ontology and with the assumption that the verb "to_check" would be more suitable for the annotation of her activities, she starts a discussion forum about renaming to_control into to_check. Since all other analysts agree



(a) (b)

Fig. 4. Warehouse expert's modifications

on such a change, the warehouse expert can rename the concept and its subconcepts. The modification is automatically propagated to other experts' annotations.

Snapshot 6: modelling payment and delivery. The payment expert is able, at this point, to complete the process by designing the checkout sub-process (Figure 5(a)). He has to model the exchange of information between customer and on-line shop related to product delivery (i.e. customer's address and delivery preference) and payment (i.e. credit card data). The checkout process strictly requires authentication, already introduced as an optional activity by the customer expert. However, since the needed authentication sub-processes (customer and on-line shop sides) are exactly the same as those modeled by the customer expert, the payment expert decides to reuse them.

When committing his modifications, the payment expert has to deal with new inconsistencies detected by BP-MoKi. In fact, (1) the constraints $C1$ and $C2$ have been violated, due to the is_a relationship between payment_data and sensible_data; (2) the "default" merging axiom restricting the semantic annotations of activities to subconcepts of Action has been violated too, due to the sub-process annotation checkout, a subconcept of Object. The payment expert fixes these problems (as shown in Figure 5(b)), by replacing the concept checkout with the concept to_checkout (subconcept of Action) and introducing the tasks in charge of providing (shop side) and reading (customer side) the policies. While annotating the activity labelled with "Read policies", the payment expert decides to modify the ontology by introducing a new is_a relationship between the concepts to_communicate and to_read. Since this ontology modification is just a refinement that does not impact the other analysts' work, there is no need to start a discussion to reach an agreement.

Snapshot 7: modelling the warehouse update. Finally, after the purchase is confirmed, stock quantities in the warehouse need to be updated. The warehouse expert will therefore complete the process by inserting the missing activities in the payment expert's flow (Figure 4(b)).

4 Related Works

The problem of adding formal semantics to business processes has been extensively investigated in the literature [6,7,8,9,10,11].

Thomas and Fellmann [10] consider the problem of augmenting EPC process models with semantic annotations. They propose a framework which joins process model and

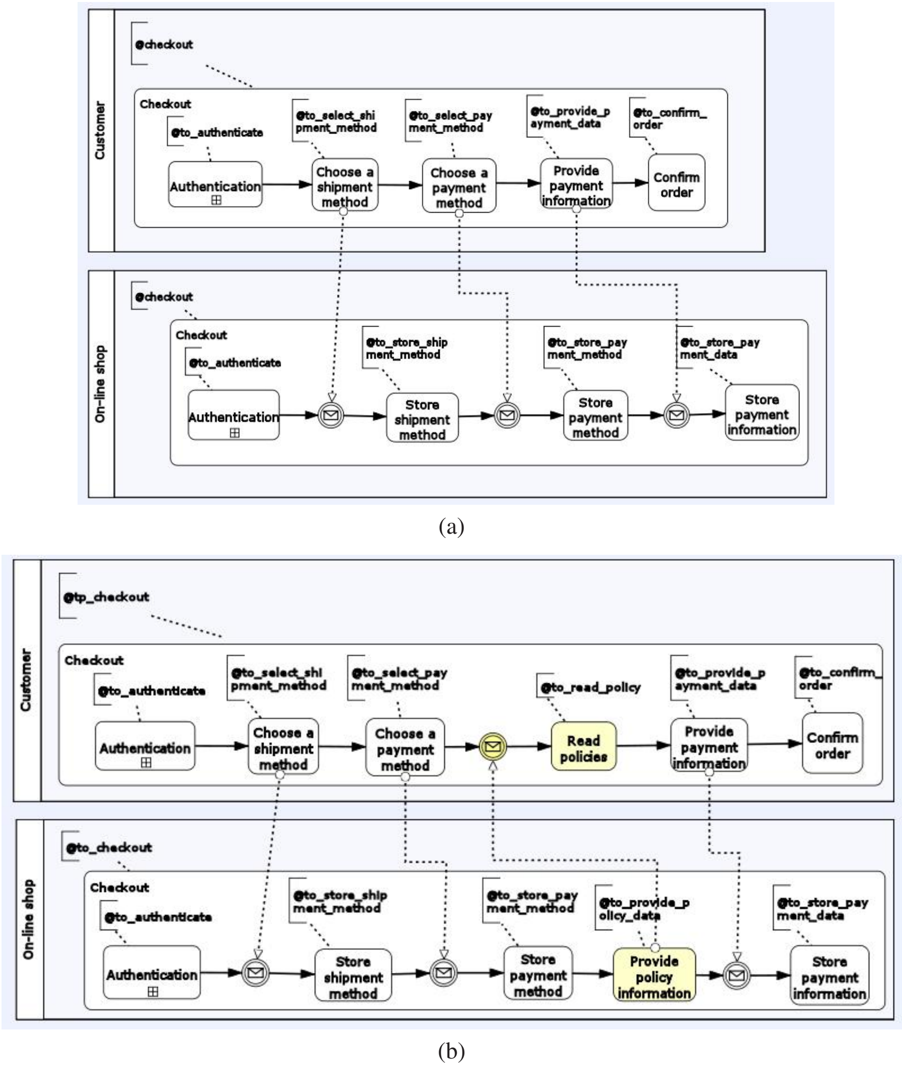


Fig. 5. Checkout sub-processes before and after inconsistency fixing

ontology by means of properties (such as the "semantic type" of a process element). Using is_a relations between the classes of the two ontologies being integrated (BPMN and domain ontology), instead of associating annotation properties to the process instances [10], allows us to simplify consistency verification.

De Nicola *et al.* [11] propose an abstract language (BPAL) that bridges the gap between high-level process description (e.g. in BPMN) and executable specification (e.g. in BPEL). The formal semantics offered by BPAL refers to notions such as activity, decision, etc., while the problem of integrating process model and *domain* ontology (e.g., by semantically annotating the process) is not their focus.

In the SUPER project [9], the SUPER ontology is used for the creation of semantic annotations of both BPMN and EPC process models in order to support automated composition, mediation and execution. Due to these goals, the focus of SUPER annotations is more on process data than on process elements. On the contrary, we focus on process element semantic annotations, since our purpose is to provide a precise and shared semantics to process elements in collaborative scenarios.

Collaborative business process definition and process integration is supported by commercial tools (e.g., Hyperwave) and by research prototypes (e.g., InterPROM). It has also been the subject of some research works (e.g., [12,13]). However, no work attempted so far to define a framework for the collaborative definition of *semantically* annotated business processes, even though the need for creating a symbiosis between project management and collaborative tools has been recently recognized [14].

5 Conclusions and Future Work

We have presented a framework, supported by a tool, for the collaborative specification of semantically annotated business processes. In our case study, we have shown how the functionalities provided to business analysts support them in the: (1) incremental definition of the process and the ontology; (2) compliance with enforced constraints; (3) evolution of ontology concepts; (4) reuse of the work carried out by other analysts.

In our future work, we will extend the functionalities available in BP-MoKi, supporting additional templates for common patterns of constraints. We will conduct further case studies and we will investigate domain independent top level ontologies that may be used as a starting point for ontology construction.

References

1. Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P.: Reasoning on semantically annotated processes. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 132–146. Springer, Heidelberg (2008)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
3. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002)
4. Horridge, M., Parsia, B., Sattler, U.: Explanation of owl entailments in protege 4. In: Bizer, C., Joshi, A. (eds.) Proc. of ISWC 2008. CEUR-WS, vol. 401 (2008)
5. Ghidini, C., Kump, B., Lindstaedt, S., Mahbub, N., Pammer, V., Rospocher, M., Serafini, L.: MoKi: The Enterprise Modelling Wiki. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 831–835. Springer, Heidelberg (2009)
6. Dijkman, R.M., Dumas, M., Ouyang, C.: Formal semantics and automated analysis of bpmn process models (2007), `http://eprints.qut.edu.au/archive/00005969/`
7. Wong, P., Gibbons, J.: A Relative Timed Semantics for BPMN. (2008) (submitted), `http://web.comlab.ox.ac.uk/oucl/work/peter.wong/pub/bpmntime.pdf`
8. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes. In: VLDB 2006, pp. 343–354 (2006)

9. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M.: A bpmo based semantic business process modelling environment. In: Proc. of ESWC 2007. CEUR-WS, vol. 251 (2007)
10. Thomas, O., Fellmann, M.: Semantic epc: Enhancing process modeling using ontology languages. In: Proc. of SBPM 2007, June 2007, pp. 64–75 (2007)
11. Nicola, A.D., Lezoche, M., Missikoff, M.: An ontological approach to business process modeling. In: Proc. of IICAI 2007, December 2007, pp. 1794–1813 (2007)
12. Leser, F., Alt, R., Osterle, H.: Implementing collaborative process management: the case of net-tech. International Journal of Cases on Electronic Commerce 1(4), 1–18 (2005)
13. Chen, Q., Hsu, M.: Inter-enterprise collaborative business process management. In: Proceedings of 17th International Conference on Data Engineering, pp. 253–260 (2001)
14. Donker, H., Blumberg, M.: Collaborative process management and virtual teams. In: Proc. of CHASE 2008, pp. 41–43. ACM, New York (2008)