Semantic Support for Security-Annotated Business Process Models

Ioana Ciuciu[1], Gang Zhao[2], Jutta Mülle[3], Silvia von Stackelberg[3], Cristian Vasquez[1], Thorsten Haberecht[3], Robert Meersman[1], and Klemens Böhm[3]

[1] STARLab, Vrije Universiteit Brussel, Brussels, Belgium
[2] Intelartes SPRL, Waterloo, Belgium
[3] Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
{iciuciu,cvasquez,meersman}@vub.ac.be,
{gang.zhao}@intelartes.com,
{jutta.muelle,silvia.stackelberg,
thorsten.haberecht,klemens.boehm}@kit.edu

Abstract. Service-Oriented Architectures (SOA) benefit from business processes (BP), which orchestrate web services (WS) and human actors in cross organizational environments. In this setting, handling the security and privacy issues while exchanging and processing personal data is essential. This lacks for secure business processes management. To achieve this, we represent security constraints descriptively by annotating process models, aiming to enforce these constraints by a secure business process management system (BPMS).To assist the process modeler in annotating process models, we introduce in this paper a tool which provides semantic interoperability during process design. By enforcing a shared conceptualization (ontology) of the security and privacy domains with an ontology base grounded in natural language this tool called knowledge annotator is able to make annotation recommendations according to knowledge stored in a knowledge base. The annotator is validated in an employability use case scenario.

Keywords: Business Process Model, Semantic Annotation, Semantic Interoperability, Ontology, Knowledge Management, Security Constraints, Employability.

1 Introduction

The Trusted Architecture for Securely Shared Services (TAS^3)[1] project provides a next generation trust and security architecture that is ready to (1) meet the requirements of complex and highly versatile business processes, (2) enable the dynamic, user-centric management of policies and (3) ensure secure end-to-end transmission of personal data and user-controlled attributes between heterogeneous, loosely coupled, context-dependent systems.

[1] http://tas3.eu/

T. Halpin et al. (Eds.): BPMDS 2011 and EMMSAD 2011, LNBIP 81, pp. 284–298, 2011.

One of the challenges, in this context, is to offer a secure business processes framework for sharing, accessing, and using personal data processing services in federated environments.

To make business processes secure, we proceed as follows. We annotate the business process model with security constraints in the first step. These annotations concern the handling of authentication, authorization, audit logging, and other security issues. The business process management system (BPMS) transforms security annotations into descriptive security policies or triggers process model extensions. Finally it executes secure business processes by dedicated system components. For example, these components allocate actors to activities, enforce data-specific authorizations, or trigger security-specific user involvements. This infrastructure guarantees that business processes will be performed according to the annotated security constraints. In order to ensure semantic interoperability between the different components of the system, we provide an ontology, embedded in the architecture, which explicitly documents the relationship between core security concepts. One goal is that all security-relevant business process specifications are annotated to a common, agreed upon semantic knowledge structure (ontology) in order to ensure alignment and interoperability between actors with respect to security concepts.

It is therefore of major importance to have a mechanism which ensures the correct specification of the security annotations. The solution we propose in this paper is a semantic security annotation tool for business processes. This semantic annotator tool aims to assist the process modeler in specifying security constraints for business process models. It uses a lower common ontology representing security constraints for business processes and a knowledge base storing a set of previously defined correct annotated rules. The system is able not only to support the process modeler with syntactically correct security concepts, but also to assist him with annotation suggestions. The suggestions are made according to information retrieved from the knowledge base which is matched against the process modeler input (knowledge).

The rest of the paper is organized as follows: Section II briefly describes related work. Section III provides background information on the technology being used. The approach is presented in Section IV. Section V shows the possible annotation use cases within an employability use case scenario developed by the University of Nottingham [1]. Section VI presents our conclusion and suggestions for future work.

2 Related Work

The idea of adding semantics to business processes has been adopted and its importance has been recognized by the business process community for several years now [2,3]. Ever since, Semantic Web technologies have been applied and new tools have been proposed in order to add semantics to business processes. The semantics are captured via semantic annotations specifying the process dynamics and behavior [4], or the meaning of process elements (as in e.g. the SUPER[2] project [5]).

Several semantic annotation models have been proposed, aiming at semantic interoperability of business process knowledge [6]. The focus of this research is on annotation tools aiming at assisting the process modeler with an ontology-based

[2] http://www.ip-super.org/

recommendation system. Betz [7] proposes an approach for the automatic user support based on an autocompletion mechanism during the modeling process (where business processes are represented as Petri Nets). Born [8] presents a tool for the user-friendly integration of domain ontology information in the process modeling, through match matching and filtering techniques. A similar approach, based on linguistic analysis of process element labels and of the concept names is presented in [9] in order to support process modelers with annotation suggestions.

Our approach is grounded in natural language. It is built on the ontology-based data matching principles [10]. The goal is to assist the process modeler with annotation suggestions retrieved from a security constraints ontology base and from a knowledge base storing previously defined annotations.

3 Background

We represent business process models as Business Process Model and Notation 2.0[3] (BPMN 2.0) diagrams. BPMN is the widely accepted de-facto standard for process models(OMG[4], 2011). It provides several elements to represent a process flow, such as pools and lanes, activities, events, data objects, flow objects, and artifacts. As BPMN artifacts enable the annotation of process diagrams, we embed security constraints as security-marked annotations into BPMN diagrams. Consequently, the security annotations are standard-conform to BPMN.

The knowledge annotator presented in this paper is based on Developing Ontology Grounded Methodology and Applications (DOGMA, [11]). DOGMA is a formal ontology engineering framework applying the principles of database design methodology (NIAM/ORM2, [12]) to ontology engineering. DOGMA ontology is grounded in natural language and based on the *double articulation principle* [13], which makes the ontology two layered:

1. The *lexon* base layer, containing a set of simple binary facts, called lexons;
2. The *commitment* layer that formally defines rules and constraints by which applications may make use of the lexons from the lexon base.

A lexon is defined as a quintuple $\langle \gamma, t_1, r_1, r_2, t_2 \rangle$ representing a fact type. γ is a context identifier that points to a context where two terms, t_1, t_2 are originally defined and disambiguated. r_1, r_2 are two roles that characterize the relationship between t_1 and t_2. For example, *‹SecBP, Security Annotation, defined for, annotated with, BPMN Element›* is a lexon which means “in the context of secure BP (SecBP), a security annotation is defined for a BPMN element and a BPMN element is annotated with a security annotation”. This example is depicted in Fig. 1.

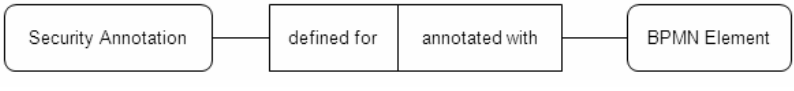

Fig. 1. A lexon example

[3] http://www.bpmn.org/
[4] http://www.omg.org/

A commitment contains a constraint on a (set of) lexon(s). For instance, we can apply the mandatory constraint on the above lexon, "there exists at least one BPMN element per annotation type". The commitment language needs to be specified in a language such as OWL[5] or SDRule language [14].

4 Knowledge Annotator for Secure Business Process Models

The knowledge annotator is designed as a user-friendly, intelligent system, intended to assist the process modeler during the specification of the security-specific constraints and to learn from the process modeler by using a dedicated knowledge base. This is realized by capturing the process modelers' modeling intentions via a user-friendly interface (UI) and by presenting him/her with recommendations. The recommendations are determined before by an ontology-based data matching operation between the user input, the security constraints ontology (see Section 4.2), and the collected security annotations retrieved from the knowledge base (see Section 4.4).

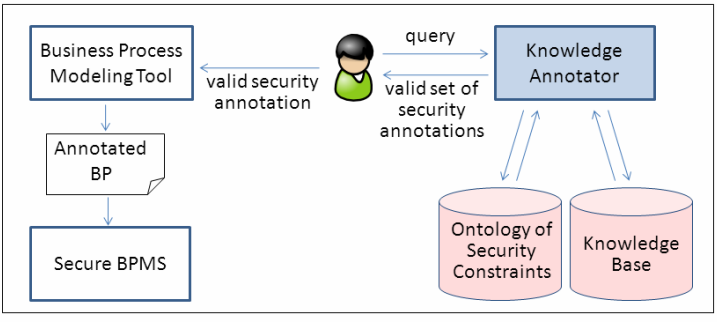

Fig. 2. User-system interactions for the annotation of security constraints

In our approach, the business process model is annotated with security constraints that make use of the concepts from the ontology of security constraints and of the knowledge stored in a knowledge base, as shown in Fig. 2.

Let us take the particular case when the process modeler needs to annotate an "activity" BPMN element using a "requestToUser" type of security annotation. For this he/she sends a query to the annotator, indicating "request to user" in the annotation search fields. The BPMN element of type "activity" is inferred from the business process (BP) modeling tool and passed to the query as well. The annotator assists the process modeler by performing several operations, as shown in Fig.3.

The embedding of the annotator tool into the design phase of security-annotated business processes eliminates the tedious task of manual search for the different options of correct annotations for a specific user query. It presents the modeler with a complete set of options according to the expressed modeling intentions (queries). This is extremely helpful in case of large sets of security annotations stored in the knowledge base and retrieved by the annotator components (see Section 4.5).

[5] http://www.w3.org/TR/owl-ref/

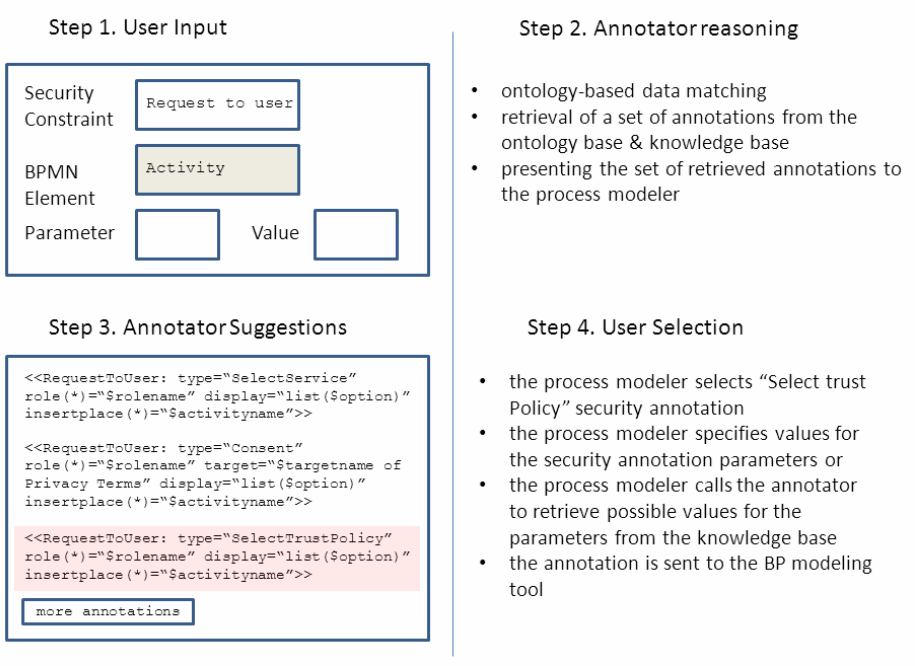

Fig. 3. Example of user-system interactions for a specific security annotation ("RequestToUser")

The ontology-based data matching operation ensures the exploration of different security ontologies developed in the past and the mapping of concepts between them and the ontology of security constraints (see Section 4.3). This allows a wide set of search options for the process modeler when he/she performs the search.

4.1 Security Annotation

A security annotation is a text annotation attached to a BPMN element. The syntax of a security annotation is specified by an annotation term, followed by a list of parameters (mandatory or optional) with their corresponding values:

```
<<AnnotationTerm: list(parameter="value")>>.
```

Our security language supports auditing, authentication, authorization, data and messageflow security, delegation, and user interactions [15].

Fig. 4 gives an example of several annotations of two BPMN lanes ("Placement Advisor" and "Student"). The <<Authn … >> annotations enforce authentications for all process participants executing tasks of these lanes according to the specified parameters, namely authentication "attributes" and the identity provider ("idp"). The role assignments of lane "Placement Advisor" and lane "Student" (<<Assignment type="Role">>) mean that only particular role holders, namely "Placement Advisors" or "Students" have the authorizations to execute tasks. The annotation <<BoD>> for

lane “Student” describes that all tasks must be performed by the same person (binding of duty), while the execution of tasks of lane “Placement Advisor” can be delegated due to spec=”weak” to other role holders.

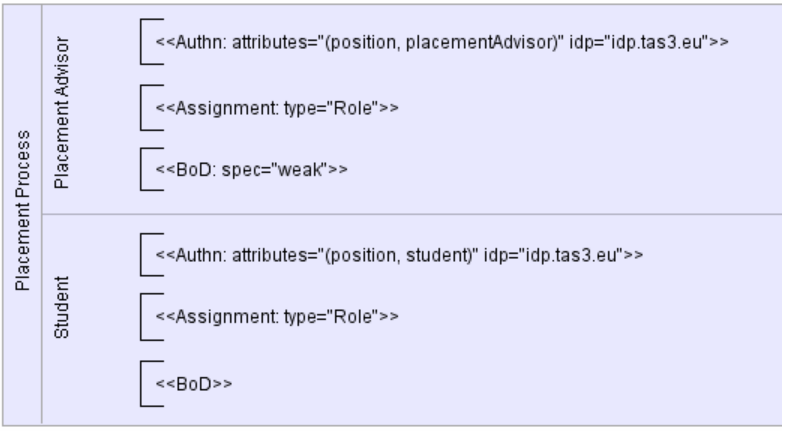

Fig. 4. Security annotations for BPMN lanes

4.2 Ontology of Security Constraints

A lower common ontology has been created to represent the security constraints applying to business processes. The security constraints ontology is used to assist the process modeler (see the approach described in [16]) for annotating the following BPMN 2.0 elements: activities, groups of activities, pools and lanes, data, message flows, and events.

The taxonomy of BPMN elements that can be annotated is illustrated in Fig. 5. Activities are considered to be either tasks or sub-processes. Data subsumes, according to the BPMN 2.0 standard, data objects, data stores, data inputs, and data outputs.

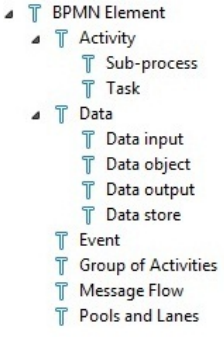

Fig. 5. Taxonomy of BPMN elements

The BPMN elements are annotated with security annotations, as illustrated in Fig. 6. Each security annotation applies to at least one BPMN element.

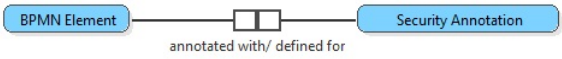

Fig. 6. BPMN element and security annotation concepts

The concept of security annotation is illustrated in Fig. 7.

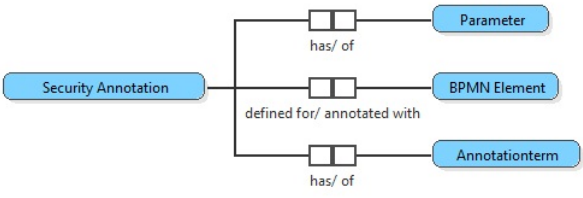

Fig. 7. Representation of the security annotation concept

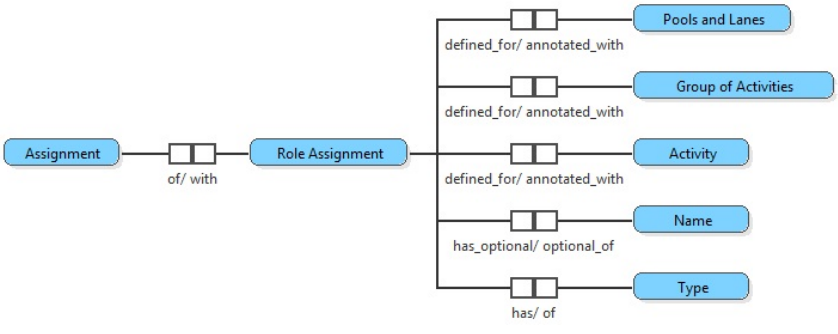

Fig. 8. Representation of the "role assignment" security annotation

Fig. 8 shows an example for the security annotation "role assignment" (annotation term, parameters, and its domain of BPMN elements are defined).In the authorization context, a role assignment specifies which role holders have to perform the annotated object (activities, group of activities, or all activities of annotated pools and lanes). The compulsory parameter of this type of annotation is "type"; "name" is an optional parameter. "Assignment" represents the annotation term of the "role assignment" security annotation.

The security constraints ontology is represented by the DOGMA ontology.

4.3 Ontology-Based Data Matching

Multiple ontologies of security concepts exist in TAS[3]. In this approach, we allow the process modeler to specify his/her queries by using generalized concepts from these

ontologies (seen as hierarchies, due to the domination relation specific to the security domain). This implies an ontology-based data matching between a concept from a general security ontology or from a user dictionary and the ontology of security constraints, used for specifying the security annotations.

Prior to the annotation, the system performs an ontology-based data matching step. This implies:1) mapping data into semantic networks (Tree, Directed Acyclic Graph/lattice or any directed graphs); 2) performing semantic computation, such as path recognition (shortest path, connectivity), path strength in scores (e.g., semantic vicinity), composite semantic similarity of semantic networks; 3) performing literal computation, such as fuzzy similarity of literals (strings); and 4) performing lexical computation, such as synonymous similarity (based on WordNet[6]) and similarity based on a user dictionary.

The searching task is performed via two modules: the interpreter and the comparator (as shown in Fig. 9).

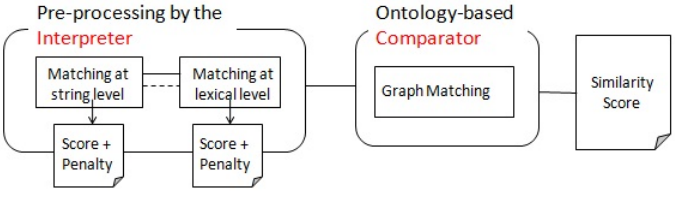

Fig. 9. Ontology-based data matching model

The interpreter makes use of a lexical dictionary and of the domain ontology to interpret the input term(s). Given a term that denotes a concept in the domain (security) ontology, the interpreter returns the correct concept defined in the ontology. The comparator then uses any combination of the different available graphs algorithms for the path recognition between two concepts originating from two different ontologies (or from a dataset and the ontology of security constraints). Currently the annotator searches for an exact match of a pattern, allowing similarity "1" (i.e. equality) only. Once the target concept is found, the annotation process is ready to start.

4.4 The Knowledge Base

The basic data element used by the knowledge annotator is represented by the Security Annotation Term, Element, Parameter, Value (STEPV) object. The STEPV object encapsulates the four entities needed to completely define a security annotation: the security constraint, the BPMN element being annotated, the parameters and their corresponding values.

For every security annotation that the process modeler intends to define, he must indicate as many fields (STEPV elements) as possible (according to his/her knowledge) corresponding to what he/she has in mind (see example from Fig.3). The process

[6] http://wordnet.princeton.edu/

modeler input is captured by annotator and analyzed in order to make recommendations to the process modeler. After performing the analysis, the system returns to the process modeler the STEPV elements that are related to his/her initial (usually incomplete) specification. The STEPV elements returned are considered valid annotations according to the constraints defined in the commitment layer (see Section 3).

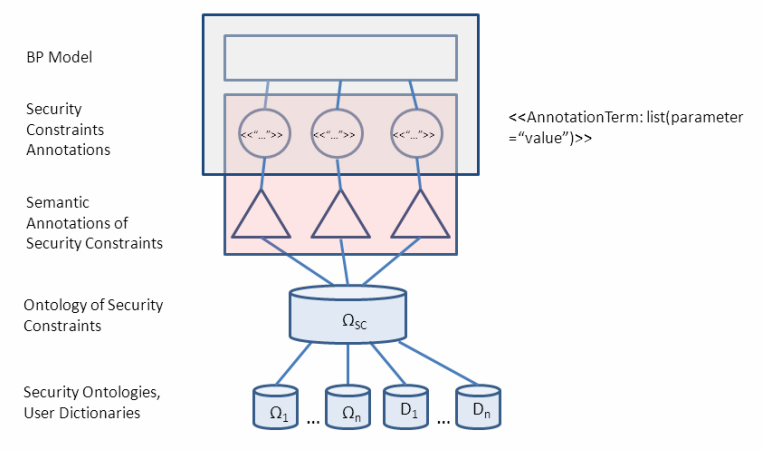

Fig. 10. Semantic annotation of security constraints

Once a STEPV object is complete and correct, our system stores it in the knowledge base. It will be used for later knowledge retrieval when the process modeler queries the system for recommendations. Note that the first three entities in every possible STEPV object (i.e., annotation term, BPMN element and parameters) are already stored in the ontology base, together with the relations they share with respect to one another (see Section 4.2).

When values are associated to parameters for a specific security annotation defined by the process modeler, we say that the security constraints ontology for that particular security annotation is instantiated. It is at this point that the knowledge base is capturing and storing the process modeler's knowledge (the way the process modeler chooses to instantiate the security annotation).

In order to ensure semantic interoperability between different organisations and actors regarding security concepts, we add an extra layer to the knowledge base, that is, the semantic annotations (as shown in Fig.10). Semantic annotations are added to the STEPV elements from different security-related ontologies or user dictionaries. The approach is explained in Section 4.3.

4.5 Knowledge Annotator Architecture

The knowledge annotator system was designed to assist the process modeler in designing security-annotated business processes with a user-friendly interface. Several functions are encapsulated in a web service, supported by six architectural components: A) capturer; B) annotator; C) indexer; D) retriever; E) comparator; and F)

presenter. The architecture is illustrated in Fig. 11. The annotator is implemented as a multi-tiered application, using J2EE[7] and Jena2[8] for reasoning.

The input to the web service call is collected by the UI. It is a query combining different terms expressing the process modeler's modeling intentions (such as "Request to user" AND "Activity" in Fig. 3). The result of a query sent to the knowledge annotator will be a set of correct security annotations proposed to the user via the UI. The added value of our approach is that the system allows query terms originating from multiple sources (e.g., user dictionary, security ontologies) which are mapped to concepts of the ontology of security constraints. This is realized prior to the actual annotation process, via an ontology-based data matching operation explained in Section 4.3.

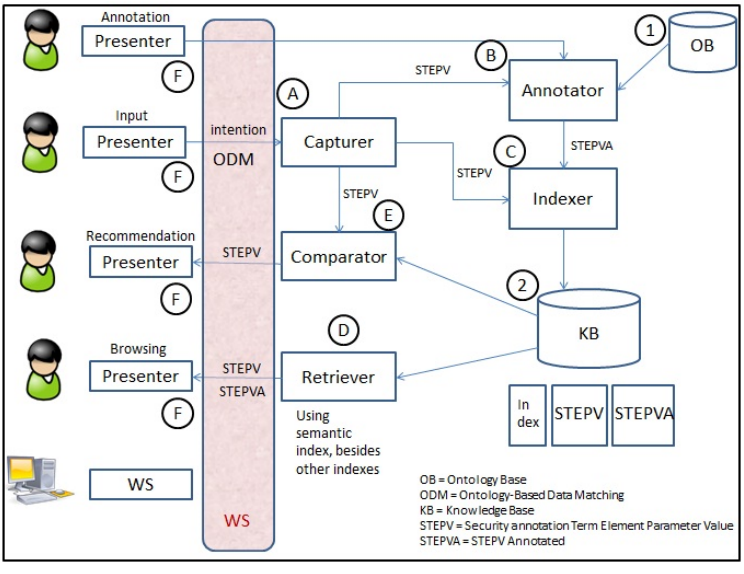

Fig. 11. Knowledge annotator architecture

The *Capturer* component captures the process modeler's intent and transforms it into an STEPV object. The STEPV objects are passed from one component to the other along the process (e.g. from the capturer to the indexer, from the indexer to the knowledge base, from the knowledge base to the retriever and from the retriever to the presenter). The process modeler specifies its input via a user-friendly interface. The UI presents a template with several fields for the process modeler to textually specify his security constraints. At this stage, the process modeler can be assisted by an ontology browser.

The STEPV object is annotated in a semi-automatic manner with concepts from the ontology base (OB), using the *Annotator* component.

[7] http://java.sun.com/j2ee/overview.html
[8] http://jena.sourceforge.net/inference/

The *Indexer* is used for the indexing of the STEPV elements stored in the knowledge base. This will facilitate the retrieval operation.

The *Retriever* component retrieves similar fragments (STEPV objects) from the knowledge base (e.g., all existing security annotations which share at least one common STEPV element with the input STEPV object). The similarity measure can be defined according to the user needs. For example, the user could only be interested in STEPV objects with a particular value for the "name" parameter. The knowledge base contains semantic annotation instances (STEPVA objects) of the security constraints.

The *Comparator* component performs a matching in order to compare the process modeler's demand (STEPV input object) with the resulted STEPV elements retrieved from the knowledge base in the past step.

Finally, the *Presenter* displays the user recommendations based on the design fragments (STEPV objects) retrieved from the KB. It is also the place where the user defines his/her queries. This component interacts with the process modeler via the UI. The WSs interaction is intended to provide interoperability in case of collaborative annotations done by members from different organizations. It implies human-system interactions at each organization end.

The following section shows how the knowledge annotator was applied in order to model secure business processes in an employability use case scenario.

5 Use Case Scenario

Within the TAS^3 project, we have developed an employability demonstrator focusing on the management of internships and work placements for university students [1]. Timely and accurate presentation and secure exchange of verified skills data and personal data is key factor to the success of this scenario. For example, recruiters and prospective employers want to access verified data in a standardized format (e.g., HR-XML[9]) to facilitate matching of students with job profiles. Similarly, candidates want to retain control over how their personal data is accessed, processed and stored by third parties.

Fig. 12 illustrates one of the employability scenarios in TAS^3. In this scenario, Betty is a student at a UK university and seeks a work placement. Betty contacts a placement service approved by the university to discuss the details of her application. Her placement advisor, called Paul, informs her that he first needs to verify that she is a registered student at the university. Once Paul has received the confirmation, he contacts Betty to get permission to access relevant personal data to match her to available placements. Betty agrees to share her data provided that the data is not being shared to third parties without her approval. Based on this information, Paul identifies a number of placement providers that he believes to have suitable placements for students like Betty. Betty wishes to be put forward for two placements and agrees that the placement advisor can act on her behalf and she consents to have relevant personal data to be disclosed to them. Paul forwards Betty's personal data to the placement providers for consideration.

[9] http://ns.hr-xml.org/schemas/org_hr-xml/3_0/

There are several security constraints embedded in this process. Our goal is to technically realize this scenario as a business process and to use system support particularly for enforcing such security constraints. E.g, to check Betty's admission to the placement application, an identity provider component has to identify and authenticate Betty at the beginning of the process. In the same way, authorized access to Betty's personal data requires to specify access rules and to enforce these constraints by security components. To this end we model the placement process and descriptively annotate security constraints. A secure business process management system will enforce these constraints during process execution, i.e., in our example scenario, when Betty performs her placement application.

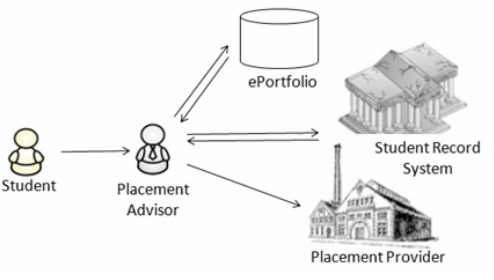

Fig. 12. The employability scenario

We now show exemplarily how security constraints of the employability scenario can be annotated efficiently using our tool. Fig. 13contains a business process activity "call matching service" of this scenario. It represents the matching of the personal data with placements available. The activity is annotated with three security annotations. In our example we use trust policies for the selection of adequate web services (i.e. placement providers) to be called. To this end, a modeler may start with an annotation "RequestToUser: SelectTrustPolicies". The annotator will find an annotation in the knowledge base, in particular a "RequestToUser: ServiceSelection" annotation which denotes that the student should be allowed to interactively select a web service. This is a typical involvement of users in an environment where trust policies for web services determine their use. The execution of a service discovery results in a list of web services adequate not only with respect to the required functionality but also to the trust level demanded by the caller. Additionally, the annotation "RequestToUser: SetDataPolicy" introduces a user interaction to set the data access policy for the user's personal data. The secure BPMS employs this policy when calling the matching service.

Analyzing this scenario, we identified the following use cases:

Use case 1. The first situation represents the basic use case for the knowledge annotator, when the process modeler is interrogating the ontology base (components A, B, D, F and 1 in Fig. 11) to retrieve and browse the correct concepts he/she needs in order to specify security constraints. The process modeler can ask the system to make faceted search on the BPMN elements (e.g., which annotations are defined for activities?), on the particular security annotation types, and on the parameters. If the

process modeler decides to annotate pools and lanes with the "Roleassignment" security annotation to specify that both, lane "Student" and lane "Placement Advisor" represent roles (see Fig. 4), the result looks like:

Table 1. Results returned for the faceted search: Roleassignment

SecurityAnnotation	BPMN Element	Parameters
Roleassignment	Pools&Lanes, Activities, Group of Activities	Type Name

Following these options, the process modeler either decides to make a choice or launches new queries in case the results do not correspond to his modeling intentions.

Use case 2. The second case represents the situation when the process modeler needs to instantiate the security annotation, i.e. to give values to the parameters of the annotation. In this situation, the system is performing matching operations between the process modeler input and the knowledge base content in order to retrieve instantiated security annotations of the same type. All components in Fig. 11 are involved.

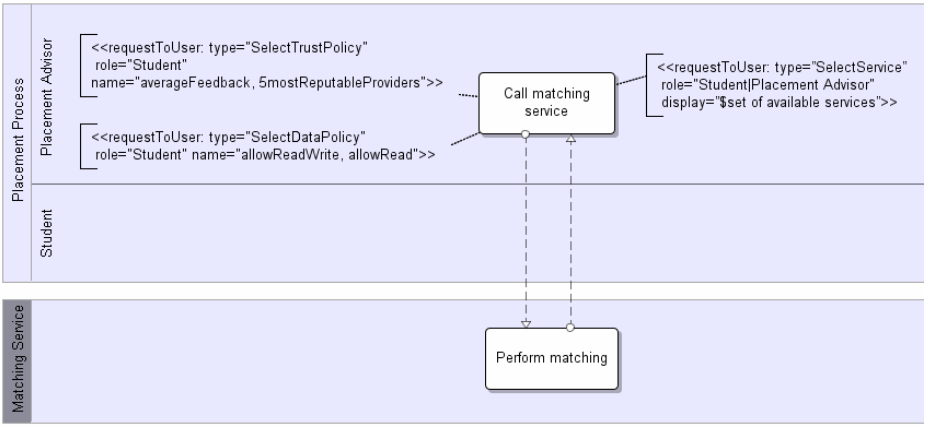

Fig. 13. Example of an annotated BPMN activity

Use case 3. The third use case is represented by the situation when the process modeler has created the security annotation by memory and needs to check for its syntactical correctness. In this case, the system performs similarity measures between the input and the ontology base and presents to the process modeler recommendations. All components in Fig. 11 are involved, except component 8 (i.e. the knowledge base).

Use case 4. The fourth situation is when the process modeler asks the system to check an instantiated security annotation for correctness. In this case the matching is done at the knowledge base level (matching performed not only at the ontology (type) level, but also at value level). The difference between use case 3 and use case 4 is that in the first case the correctness is validated only with the ontology base (syntax only), while

for the second case it is validated against the knowledge base. All components in Fig. 11 are involved.

Use case 5. The fifth situation represents the case when the actors are two web services which are interoperating via an interface. This situation represents the case when members of multiple organizations make collaborative annotations. The process modeler is involved in the annotation process at the organization end. All components in Fig. 11 are involved.

Currently, use cases 1 and 3 are completely supported. The implementation of other use cases is work in progress. We are actually working on enriching the knowledge base by domain experts via an online form and on the integration of a knowledge annotator user interface embedded in a BPMN editor.

6 Conclusion and Future Work

For specifying security constraints of business processes we have developed a security language for annotating process models. This paper presents a knowledge annotator which assists the process modeler during the specification of these constraints by providing semantic interoperability.

The knowledge annotator is designed to be (1) intuitive, acting as an intelligent system which is able to capture the process modeler modeling intentions and to provide him/her with design recommendations; (2) based on an ontology of security constraints grounded in natural language; (3) interoperable, being designed as a web service which operates in an open, distributed and dynamic environment; and (4) secure, enabling query-only requests via SSL/TLS links.

An emerging work is the consolidation of the knowledge base with security annotations designed for the two TAS^3 pilots (employability and e-Health).

Future work will involve linking the ontology with a more general ontology of security concepts (upper common ontology), which already exists in the TAS^3 project. The purpose is to assist the process modeler with more abstract security concepts when performing the search, providing him/her with a hierarchy of concepts for exploration.

Another future work is to integrate the annotator with an ontology-based interoperation service in order to be able to accept organization-specific vocabularies which map to the security constraints ontology.

Future directions also include user studies, aiming to analyze the user context and behavior in order to provide him/her with improved design suggestions.

Acknowledgments. This paper is supported by the EC FP7 TAS^3 (Trusted Architecture for Securely Shared Services) project. The authors would like to thank all TAS^3 project partners for their contribution to the research.

References

[1] Claerhout, B., Carlton, D., Kunst, C., Polman, L., Pruis, D., Schilders, L., Winfield, S.: Pilots Specifications and Use Case Scenarios, TAS3, Deliverable D9.1, Trusted Architecture for Securely Shared Services (2009), `http://tas3.eu/`

[2] Jenz, D.E.: Ontology-Based Business Process Management: The Vision Statement. Jens & Partner GmbH, 1st edn. (2003), http://www.bptrends.com/publicationfiles/12-03%20WP%20BP%20Ontology%20Vision%20-%20Jenz.pdf
[3] 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. In: Proceedings of the IEEE IECBE 2005, Beijing, China, pp. 535–540 (2005)
[4] Wetzstein, B., Ma, Z., Filipowska, A.: Semantic Business Process Management: A Lifecycle Based Requirements Analysis. In: Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N. (eds.) SBPM 2007, Innsbruck, Austria (2007)
[5] Dimitrov, M., Simov, A., Stein, S., Konstantinov, M.: A BPMO Based Semantic Business Process Modelling Environment. In: Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N. (eds.) SBPM 2007, Innsbruck, Austria (2007)
[6] Lin, Y.: Semantic Annotation for Process Models; Facilitating Process Knowledge Management via Semantic Interoperability. PhD Thesis, Norwegian University of Technology (2008) ISBN 978-82-471-5160-0 (printed version)
[7] Betz, S., Klink, S., Koschmider, A., Oberweis, A.: Automatic User Support for Business Process Modeling. In: Proceedings of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, Budya, Montenegro, pp. 1–12 (2006)
[8] Born, M., Dorr, F., Weber, I.: User-friendly Semantic Annotation in Business Process Modeling. In: Proceedings of the Workshop on Human-friendly Service Description, Discovery and Matchmaking (2007)
[9] Di Francescomarino, C., Tonella, P.: Supporting ontology-based semantic annotation of business processes with automated suggestions. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. Lecture Notes in Business Information Processing, vol. 29, pp. 211–223. Springer, Heidelberg (2009)
[10] De Baer, P., Tang, Y., Meersman, R.: An Ontology-Based Data Matching Framework: Use Case Competency-Based HRM. In: Proceedings of the 4th Int. OntoContent 2009 Workshop, On the Move to Meaningful Internet Systems. LNCS, pp. 514–523. Springer, Heidelberg (2009)
[11] Spyns, P., Tang, Y., Meersman, R.: An Ontology Engineering Methodology for DOGMA. J. of App. Ontology 3(1-2), 13–39 (2008)
[12] Halpin, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufmann, San Francisco (2001)
[13] Spyns, P., Meersman, R., Jarrar, M.: Data Modeling Versus Ontology Engineering. SIGMOD Record: Special Issue on Semantic Web and Data Management 31(4) (2002)
[14] Tang, Y., Meersman, R.: SDRule Markup Language: Towards Modeling and Interchanging Ontological Commitments for Semantic Decision Making. In: Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches. IGI Publishing, USA (2009) ISBN: 1-60566-402-2
[15] Mülle, J., von Stackelberg, S., Böhm, K.: A Security Language for BPMN Process Models. Technical Report, Karlsruhe Institute of Technology (KIT), no. 2011-09, Karlsruhe (2011)
[16] Mülle, J., Müller, J., Haberecht, T., von Stackelberg, S., Ciuciu, I., Reul, Q., Hoppenbrowers, J., Blandin, A., Boisvert, A.: Design of a Semantically underpinned, Secure & Adaptable Process-Management Platform, TAS3, Deliverable D3.1 (3rd iteration), Trusted Architecture for Securely Shared Services (2010), http://tas3.eu/