An Ontological Approach for Organizing a Knowledge Base to Share and Reuse Business Workflow Templates

Thi Hoa Hue Nguyen, Tuan Phan-Hong, Nhan Le Thanh

An Ontological Approach for Organizing a Knowledge Base to Share and Reuse Business Workflow Templates

Thi-Hoa-Hue Nguyen
and Tuan Phan-Hong
Vietnam-Korea Friendship IT College
Da Nang, Vietnam
Email: huenth@viethanit.edu.vn and tuanph@viethanit.edu.vn

Nhan Le Thanh
The I3S laboratory
University of Nice Sophia Antipolis
Sophia Antipolis, France
Email: Nhan.LE-THANH@unice.fr

***Abstract*—Business process models have been used in a lot of enterprise applications. Along with their popularity, the problem of how to create them correctly in terms of semantics and syntax while effectively promoting the reuse of suitable parts of existing models is increasingly interest. This paper describes how to organize a knowledge base to facilitate the shareability and reusability of business workflow templates. We first introduce a repository consisting of business workflow templates which are well-checked at the syntactic and semantic level. An organizational mechanism for control flow-based business workflow templates is therefore provided to ensure an effective search of templates. We then propose a process for developing workflow templates. Thereby for each use case, users can select and modify suitable workflow templates from the knowledge base.**

***Index Terms*—Business process; Business workflow template; Knowledge base; Ontology; Reuse; SPARQL**

I. Introduction

The design of business workflow management systems (WfMSs) is generally independent from the concrete business area of employing enterprises. Consequently, this workflow technology follows the generic approach. Therefore, IT experts play an important role in implementing business processes of the enterprise and establishing the software infrastructure. It is important to note that the use of business workflows aims to automate and optimizes an organization's processes in an administrative context to reduce costs (e.g., human resources) and increase revenue. Up to now, there are more than a hundred business WfMSs, such as FileNet[1], SAP[2], JBPM[3] and Spiff Workflow[4]. Insurance, banking and health industries, for example, are domains using business workflows.

However, the specification of a real-world business process is generally manual and is thus vulnerable to human error. An incorrectly designed workflow may lead to failed workflow processes, execution errors or not meet the requirements of customers, etc. There is an inherent problem regarding the problem of modeling semantically rich business workflow templates[5], workflow templates sharing and subsequently their reuse need to be considered.

Let us consider the following scenario. A person plans to create an ordering process for his own purpose. He has either some experience in working on it or none at all. The question is how he can create his process model in the most effective way without developing it from scratch.

In fact, the different existing workflow templates extracted from a set of process models can support modelers to create new workflows or process models by providing the knowledge about potential and suitable workflow activities. Therefore, our objective is to organize of the knowledge base which guides the search for suitable workflow templates in order to reuse them. Users can adapt the resulting workflow templates for each specific use case. This is the knowledge on how to model a business process reusing control flow-based business workflow templates (CBWTs). Hence, the annotation and storage of workflow templates play a very important role in the success of reusable CBWTs, which guarantee an effective search for modeling a business process.

The rest of this paper is structured as follows: Section II presents a scenario which is considered as a typical example to understand the problem of modeling business processes and reusing them. In Section III, we give a short introduction to our CPN Ontology which is a representation of Coloured Petri Nets (CPNs) with OWL DL[6]. We also indicate that the SPARQL[7] query language is able to check the syntactic and semantic correctness of CBWTs. In Section IV, we propose an organization of the knowledge base of CBWTs. In Section V, a process for developing workflow templates is introduced. We give related work in Section VI. Finally, we conclude the paper with an outlook on the future research in Section VII.

[1]http://www-01.ibm.com/software/ecm/filenet/

[2]http://help.sap.com/saphelp_46c/helpdata/en/c5/e4a930453d11d189430000e829fbbd/content.htm

[3]http://www.jbpm.org/

[4]https://pypi.python.org/pypi/SpiffWorkflow

[5]In our work, we define a definition for business workflow template: **A business workflow template is a generic business workflow that can be customized according to the application.**

[6]https://www.w3.org/TR/owl-features/

[7]https://www.w3.org/TR/rdf-sparql-query/

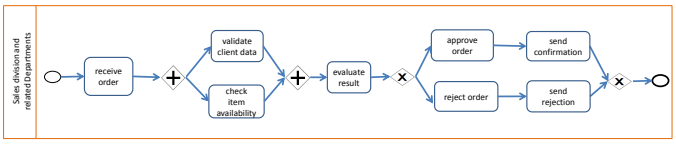

Fig. 1. *Order processing* template

II. Scenario

To better motivate our research, let us consider the following scenario, which can serve as a typical example for better understanding the problem of modeling business processes and reusing them. The scenario will illustrate the problem descriptions that will be used as examples to demonstrate our proposed solution in the next Sections.

In the scenario we will mention:

- A repository, called *CBWTRepository*, contains business workflow templates. The templates stored in *CBWTRepository* are generic and can be used to model specific process models according to the *CBWTRepository* customer's requirements;
- A customer company, named *CompanyA*, has imported workflow templates from *CBWTRepository* to build its own business application.

In the following we describe a set of workflow templates relating to the *fromOrdertoDelivery* (fOtD) process. We also present the requirements of *CompanyA* concerning its business policy. Customer companies can use the workflow templates to model their own fOtD process in compliance with their requirements. In Subsection II-A, the templates are mentioned and described in their generic form. In Subsection II-B, we introduce a *CompanyA* variant of the fOtD process and illustrate an adaptation of the templates used to model the fOtD process for *CompanyA*. There are a lot of workflow templates used to model the *fromOrdertoDelivery* process, such as templates for dunning, templates for returning purchased goods, templates for claims and templates for notification. However, to make this scenario easier to understand, we just highlight the four main templates as follows: *Order Processing*, *Invoicing*, *Payment* and *Shipment*.

A. fromOrdertoDelivery Process Model

1) Order Processing: The *Order Processing* template (see Fig. 1[8]) is used to model an order processing process. It is worth noting that a workflow-step can be a sub-workflow in itself. For example, the step *check item availability* contains some workflow-steps, e.g., *check internal item availability*, *check external item availability*, which are not illustrated in the figure for the sake of simplicity.

2) Invoicing: The *Invoicing* template (see Fig. 2) is used to model an invoicing process.

3) Payment: The *Payment* template (see Fig. 3) is used to execute a payment process in response to the received invoices.

[8]The templates are described in Section II based on BPMN [1]

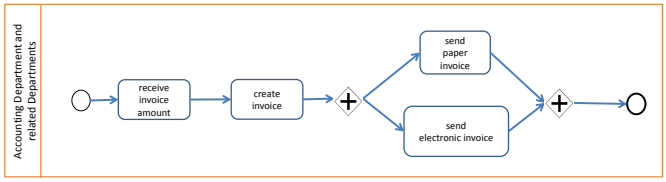

Fig. 2. *Invoicing* template

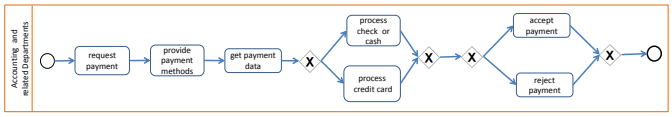

Fig. 3. *Payment* template

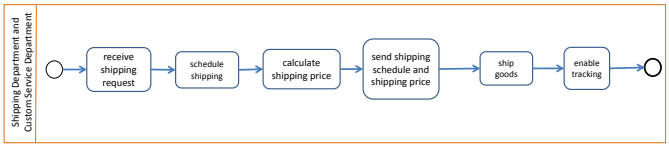

Fig. 4. *Shipment* template

4) Shipment: The *Shipment* template (see Fig. 4) is used to model a shipment process. In the upcoming Subsection, we present the business of a company, namely *companyA* and describe how to apply the above templates to its fOtD process.

B. Adapting templates stored in CBWTRepository to model the fromOrdertoDelivery Process for CompanyA

CompanyA, based in *France*, plans to create a fromOrdertoDelivery process. Instead of developing the process from scratch, this company has imported workflow templates from *CBWTRepository* to build its own business application.

Let us take a brief look at the company's policy concerning the fromOrdertoDelivery process: *CompanyA* manages an online shopping website selling beauty products. About payment, with regard to online cosmetic orders, all orders must be prepaid. The company accepts credit cards, including *VISA*, *MasterCard*, and *American Express*. For the promotional codes, only one code (if applicable) may be used for one purchase.

An order can be shipped via an indicated shipping service. Back orders are not accepted. Customers are allowed to change their shipping method before completing their online order. Shipping charges are based on the order value and shipping address as follows:

- Within *France*, goods which cost in excess of *EUR 100* per order will be delivered free of charge, conversely, a flat rate delivery charge of *EUR 6.80* will be applied.
- Within the rest of the *European Union* (*EU*), goods which cost in excess of *EUR 150* per order will be delivered free of charge, conversely, a flat rate delivery charge of *EUR 7.50* will be made.
- Shipment to *NON-EU* countries will be free of charge for order values of *EUR 200* or over. If the order value

is less than *EUR 200*, a flat rate delivery charge of *EUR 10* will be made. Additional customs duties, taxes and charges may be incurred for delivering to the *NON-EU* countries.

Charges are for each shipment and will be added to the invoice.

An order can be cancelled by calling to the Customer Service Department but only if the shipment has not yet been confirmed.

Customers can return their purchased goods by sending them back to the indicated company's address. Returns must be accompanied by invoice and they can be accepted only within *30* days of purchase. All returned products must be unused, and in saleable condition.

Accepted returns will be re-credited to the corresponding customers. Requests for refunds must be made in writing and will be granted only if no account balance is due.

Therefore, when applying the fOtD process to company CompanyA, we can re-use two templates, i.e., *Shipment* and *Payment*. However, some steps of these templates have to being modified or deleted. For example, a set of steps, which is used to calculate shipping price, replaces the step *calculate the shipping price* in the *Shipment* template.

III. Foundations

A. Modeling Business Processes with Coloured Petri Net The CPN Ontology

In this Subsection, we shortly present the CPN Ontology [2] which is defined for business processes modeled with Coloured Petri Nets (CPNs)[9]. The main purpose is to make business process models easily to be shared and reused.

The CPN Ontology is developed by translating each element of CPNs into a corresponding OWL concept. Fig. 5 depicts the core concepts of the CPN Ontology. The ontology is described based on DL syntax and the axioms supported by OWL. In the next step, we describe the main constructs of the ontology modeled with OWL DL.

The CPN Ontology consists of the following concepts: The concept **CPNOnt** is defined for all possible business processes modeled with CPNs. To represent all places and transitions in a process model, we define the concept **Place** and the concept **Transition**, respectively. We define the concept **InputArc** for directed arcs from places to transitions and the concept **OutputArc** for directed arcs from transitions to places. The concept **Token** is defined to represent all tokens inside places. To express all transition expressions, we define the concept **GuardFunction**. The concept **CtrlNode** is defined for occurrence condition in control nodes and the concept **ActNode** is defined for occurrence activity in activity nodes. To express all expressions in input arcs and output arcs, two concepts, **Delete** and **Insert**, are defined. We define the concept **Attribute** to represent all attributes of individuals. Finally, the concept **Value** is defined for all subsets of $I_1 \times I_2 \times \ldots \times I_n$ where I_i is a set of individuals.

[9]According to [3], Coloured Petri Net (CPN) is a well-proven language that is suitable for modeling workflows or work processes. They have been developed into a full-fledged language for the design, specification, simulation, validation and implementation of large-scale software systems. Therefore, CPN is chosen as the workflow language in our work to transform a business process into a control flow-based business workflow template.

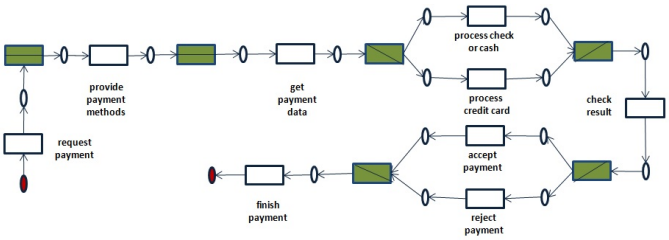

Fig. 5. *Payment template modeled with CPNs*

```
<rdf:RDF  xmlns:co="http://
  www.semanticweb.org/CPNWF#"
 ...>
<co:CPNOnt rdf:ID="Payment">
  ...
  <co:hasTrans>
     <co:Transition rdf:ID=
     "RequestPayment">
       <co:connectsPlace rdf:
       resource="#ReceivedRequest"/>
      ...
     </co:Transition>
 </co:hasTrans>
  ...
 </co:CPNOnt>
</rdf:RDF>
```

Fig. 6. *Payment template stored in RDF format (excerpt)*

Properties between the concepts in the CPN Ontology are also indicated. For example, **connectsPlace** and **hasGuardFunction** are two properties in a class **Transition**. Consequently, the concept **Transition** can be glossed as "The class **Transition** is defined as the intersection of: (i) any class having at least one property **connectsPlace** whose value is equal to the class **Place** and; (ii) any class having one property **hasGuardFunction** whose value is restricted to the class **GuardFunction**".

Fig. 5 shows the Payment template modeled with CPNs as an example. The template is used to execute a payment process in response to the received invoices. This template is stored in RDF format as depicted in Fig. 9. In the following, we introduce how to use SPARQL language to check the correctness of concrete business workflow templates.

B. Using SPARQL Queries to Verify Business Workflow Templates

Providing a high-level specification of business processes is the objective of process modeling. This makes process models independent of the target workflow management system. As mentioned previously, a workflow defined incorrectly may lead

to unintended consequences, for instance, a waste of time and effort, loss of trust in users. That is why a workflow definition should be analyzed and verified before it is put into use.

In [2][4], we introduce an approach to develop a workflow template relied on a set of semantic constraints and the structure of the CPN Ontology. The workflow template is formalized by an RDF graph in which the dependencies between its activities are also expressed. Our work focuses on checking the syntactic and semantic correctness of business workflow templates at the design phase. Regarding the syntactic verification issues, we introduce twelve syntactic constraints [5]. They are categorized into two groups, including constraints related to the definition of process model and constraints related to uses of control nodes. With regard to semantic verification, we concentrate on the following research question: Is the behavior of the individual activities satisfied and does it conform to the control flow? The answer to this question indicates the semantic verification issues that we have to deal with.

We initiate SPARQL queries to check the correctness of business workflow templates at the syntactic and semantic level. SPARQL is a query language, inspired by SQL for querying RDF data.

SPARQL verification queries are created based on the syntactic and semantic constraints. We use two query forms in our work, ASK and SELECT. The following query , for example, is used to check whether there exist syntactic errors related to the definition of process model or not. This query is used to find all transitions not having any input arcs, which means those transitions will never be enabled.

```
SELECT distinct ?t WHERE {
  ?cp rdf:type h:CPNOnt
  ?cp h:hasTrans ?t
FILTER NOT EXISTS{_:b h:connectsTrans ?t}}
```

We use Corese/KGRAM[10], a semantic engine, to match an RDF graph representing a business workflow template to graph patterns of those SPARQL verification queries. If there are no matches, i.e., no shortcomings, a workflow template is then stored in a knowledge base. We get an XML file resulting in nodes that contain required information (e.g., the name) and causes shortcomings as a result of the execution of each SPARQL verification query. For more details on verifying workflow templates, we refer the readers to [4][5].

A business workflow template is only stored in the repository if there are no syntactic and semantic errors. In order to share and reuse workflow templates, it is crucial to build a knowledge base for CBWTs management.

IV. Organization of the Knowledge Base of Control Flow-Based Workflow Templates

In literature, the main goals of workflow reuse are to improve workflow template quality and to increase its development productivity [6]. In other words, the more workflow templates are available, the more difficult they are to be suitable in a specific reuse case. It is worth noting that the reuse of workflow templates is only beneficial if the cost to find and adapt an existing workflow template is smaller than the cost needed to develop a new one from scratch.

After finding suitable workflow templates, it is important for users to understand what the workflow templates actually do. Thus, there is a strong need that the knowledge base of workflow templates could provide enough information for modelers to be able to determine which template is suitable for the reuse case at hand. In this paper, we propose a method to semantically annotate workflow templates. Their retrieval through meta-workflow templates will model expert knowledge and guide the use of existing workflow templates. The idea of using content which characterizes workflow templates is to use explicit information to find suitable templates to build a business workflow. This is particularly important for workflow modelers to be able to deal with the great number of workflow templates.

Based on the analysis of the state-of-the-art concerning the organization and reuse of workflow templates, we annotate workflow templates by the following properties as follows:

- $templateName$: Description of the main task being enacted by the template.
- $description$: Description of the template.
- $keywords$: List of words that characterizes the template. It also includes the words that name the template.
- $listOfActivityLabels$: The labels are extracted from activity labels in the template.
- $creationDate$: The date when the template is created.
- $modificationDate$: The date the template is last modified.
- $relatedTemplates$: List of related templates (if any). The related templates can be predecessors and successors of the template.
- $bpOnt$: Indicating the business process ontology used to develop the template.

The properties $templateName$, $description$, $keywords$ and $relatedTemplates$ are determined by using expert knowledge. In contrast, the values of the properties $creationDate$ and $modificationData$ are automatically captured at the moment of storing the template. Depending on all the activity labels in the template, the value of the property $listOfActivityLabels$ is automatically retrieved. For example, to get all activity labels of the template $http://WFTemplate\#Payment\_Processing$, the following SPARQL query is first executed to get all IDs of its transitions:

```
SELECT distinct ?trans WHERE {
 k:Payment_Processing h:hasTrans ?trans
}
```

Then the labels of these transitions are cut from their IDs and added into the list of activity labels.

The property **bpOnt** captures the names (or URLs) of the business process ontology file. This property lead us to the representation of additional knowledge that facilitates

[10]http://wimmics.inria.fr/corese

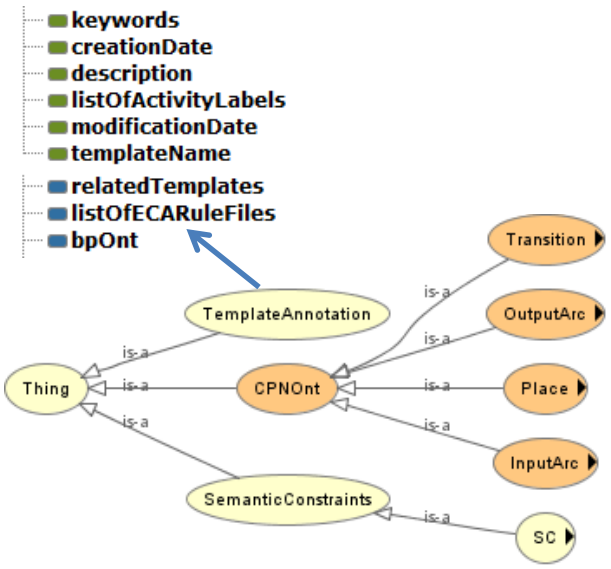

Fig. 7. Extract of the annotation ontology used to annotate workflow templates

modelers to search for suitable templates, which can be used to design a new one.

An ontology is thus developed to annotate workflow templates. The ontology describes the main classes and properties for RDF annotations of workflow templates as shown in Fig. 7. In fact, the semantic annotations of workflow templates have been inspired by this idea: the knowledge added into these annotations will be helpful for the (re-)use of workflow templates. Those meta-workflow templates allow retrieving a list of workflow templates that correspond to different criteria. For example, to acquire all existing workflow templates relating to payment by credit card, two criteria are used: (i) one keyword of such a template is *credit card*; (ii) description of such template contains *payment procecss*. This can be performed by the SPARQL[11] query as follows:

```
SELECT * WHERE
{
?workflow anno:keywords ?keyword
FILTER (?keyword ~ "credit card"
    ^^xsd:string)
?workflow anno:description ?descr
FILTER (?descr ~ "payment process"
    ^^xsd:string)
?workflow anno:templateName ?name;
 anno:listOfActivityLabels ?actLabel;
 anno:relatedTemplates ?relatedTemp;
 anno:listOfECARuleFiles ?ecaRule;
 anno:creationDate ?crtDate;
 anno:modificationDate ?modDate
}
```

It is important to emphasize that those meta-workflow templates allow retrieving workflow templates, which are annotated with additional expert knowledge formalized with the help of the CPN ontology, the BP ontology (for more details, please see [4]). In the following we introduce an excerpt of the RDF annotation related to the workflow template $http://WFTemplate\#Payment$ depicted in Figure 8.

[11] $PREFIX\ anno :< http://ontWFTemplateAnnoURI.owl\# >$

```
<rdf:RDF
xmlns ="http://ontWFTemplateAnnoURI.owl#"
xmlns:wf="http://WFTemplate#"
xmlns:rule="http://ECARule#"
... >
<TemplateAnnotation rdf:ID="wf0012">
 <templateName rdf:resource=
   "http://WFTemplate#Payment"/>
 <keywords>Cash;Credit card;Payment;
   Payment processing </keywords>
 <listOfActivityLables>Request payment;
 Provide payment methods; Get payment
 data; Process check or cash; Process
 credit card; Check result; Accept
 payment; Reject payment; Finish payment
 </listOfActivityLables>
 <description>Template payment processing
 is used to handle the payment process...
 </description>
 <relatedTemplates rdf:resource=
 "http://WFTemplate#Invoicing"/>
 <relatedTemplates rdf:resource=
 "http://WFTemplate#OrderProcessing"/>
  ...
  <bpOnt rdf:resource=
  "http://BPOntology#Payment"/>
  ...
</TemplateAnnotation>
</rdf:RDF>
```

Fig. 9. An excerpt of the RDF annotation related to http://WFTemplate#Payment

V. Process for Developing Workflow Templates

In this section, we introduce a process for developing workflow templates, which is regarded as part of the process for developing an encompassing workflow application. The process consists of the main following phases (see Figure 10):

1) **Search for reusable workflow templates**: An analysis of the process(es) is performed before implementing it. This results in a set of requirement descriptions as well as a business process model. The information is then used to start the process for developing workflow templates which may involve the search for reusable workflow templates.
2) **Understand and select potential, suitable templates**: In this phase, modelers have to carefully consider the found workflow templates. They try to understand them to decide which ones are (partly or fully) reused for their application.

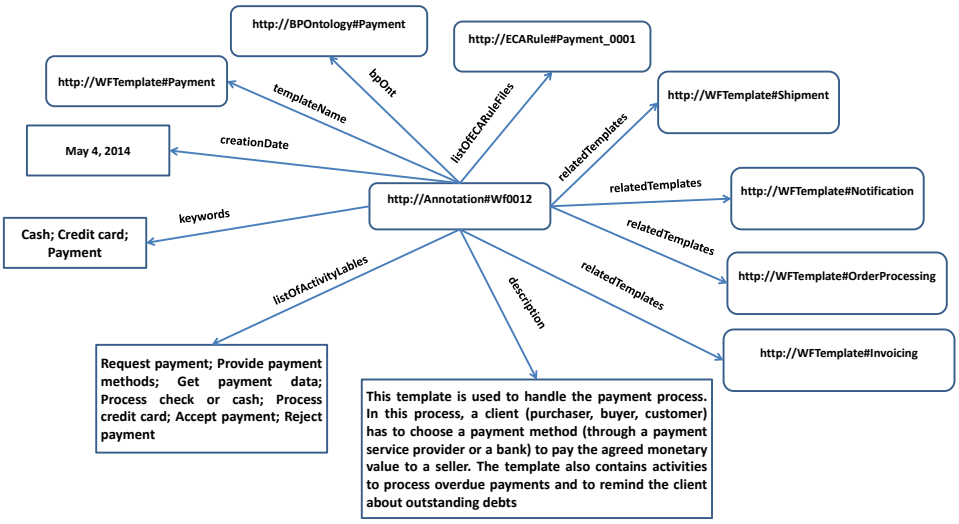

Fig. 8. Example of the semantic annotation of a workflow template

3) **Modify selected templates**: If the selected templates do not comply with all the requirements, they have to be modified accordingly. For example, some new activities can be added into a selected template.
4) **Create new sub-workflow templates**: Besides reusing part or all of the existing templates, modelers might have to create new sub-workflow templates to meet all the requirements. However, the creation of a new sub-workflow template is only necessary if no existing templates can be reused instead for the same purpose.
5) **Complete workflow templates**: The last phase is to complete a new workflow template. The existing unmodified, modified and new sub-workflow templates are integrated into a new workflow template for a specific use case. Each of these workflow templates is considered as a sub-workflow of the new workflow template. It is then verified at the syntactic and semantic level. In case of errors, the errors have to be solved. The new workflow template is stored in the CBWT repository if and only if: there exist no syntactic errors nor semantic errors.

To find suitable workflow templates, users can define their criteria by keyword, by description or by activity label. If the search process returns only one template, users can easily make their decision that the template is selected or not selected. Otherwise, the value of the property $RelatedTemplates$ can be used to provide more information for users to make their decision.

To sum up, the semantic annotations of workflow templates integrating expert domain knowledge formalized by an RDF graph are used to organize and retrieve workflow templates and their business process ontologies. The resulting templates can be used in a process for implementing software components or in a process for developing workflow templates.

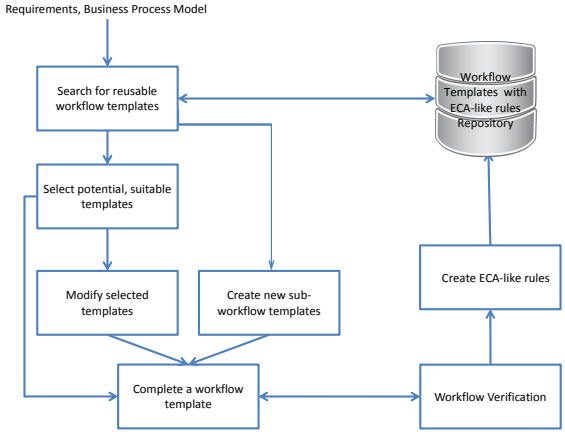

Fig. 10. Development of reuse-based workflow template

VI. Related Work

Up to now, the problem of reusing process models or workflows is mentioned in some existing approaches. In general, workflows can be reused manually or semi-automatically [7], [8], [9]. Moreover, modelers can partly or fully reuse a workflow [10], [11], [12], [9].

The authors in [10] specify a method for business process design by view integration which takes two process views as input. At first, semantic relationships between elements of different process models are formalized. On this basis, the integrated process model applying the merge operator is calculated. [11] also presents a formal approach for constructing customized process views on structured process models to improve effective cross-organizational collaborations. Each customized process is constructed by hiding and/or omitting activities not requested by the process consumer. However, neither of them considers content-based reuse. In order to

overcome this issue, the authors in [9] introduce a set of Domain Process Patterns (DPPs) that capture process model parts. A DPP represents a specific business function of a process model part in a modeling domain. DPPs facilitate reuse from a content perspective by focusing on domain-centered reuse of process model content. Nevertheless, DPPs do not provide any syntactic needs for modeling business processes. However, by capturing process model parts with a particular structure, DPPs do not support syntactic checks which are supported in our approach. In our approach, a workflow template is stored in the repository if and only if it is checked at the syntactic and semantic level, and no errors exist.

In [13], the authors propose a framework to enable ontology-driven process modeling. By utilizing the framework, users can define, analyze and re-engineer their process models in complex and dynamic contexts with semantically enriched processes. But they are mainly interested in structuring and exploiting design knowledge. They do not focus on the meaning of all the concepts and relationships in the knowledge base, this is in contrast with our approach.

VII. Conclusion

In this paper, we have presented a process for developing workflow templates, which specially emphasizes the different phases of workflow template reuse comprising the tasks of searching, understanding and modifying workflow templates. Each phase provides useful support to facilitate the reuse of workflow templates.

Moreover, in order to better support the search for suitable workflow templates, the annotation ontology has been developed to annotate workflow templates. The ontology provides adequate information about the workflow templates for workflow modelers to determine whether a workflow template is able to be reused.

At the moment, only build-time is supported and we know that verifying workflow templates at build-time is not enough to guarantee workflows can be executed correctly. The correctness of workflow execution must also be checked. Therefore, in future work, we plan to develop a run-time environment for validating concrete workflows.

References

[1] "Business process model and notation, v2.0," http://www.bpmn.org/, 2011.

[2] T.-H.-H. Nguyen and N. Le-Thanh, "An ontology-enabled approach for modelling business processes," in *Beyond Databases, Architectures, and Structures*, ser. Communications in Computer and Information Science. Springer International Publishing, 2014, vol. 424, pp. 139–147.

[3] J. B. Jørgensen, K. B. Lassen, and W. M. P. van der Aalst, "From task descriptions via colored petri nets towards an implementation of a new electronic patient record workflow system," *STTT*, vol. 10, no. 1, pp. 15–28, 2008.

[4] T.-H.-H. Nguyen and N. Le-Thanh, "Ensuring the Semantic Correctness of Workflow Processes: An Ontological Approach," in *Proceedings of 10th Workshop on Knowledge Engineering and Software Engineering (KESE10) co-located with 21st European Conference on Artificial Intelligence (ECAI 2014)*, Grzegorz J. Nalepa and Joachim Baumeister, Ed. Prague, Czech Republic: CEUR Workshop Proceedings, Aug. 2014, vol. 1289. [Online]. Available: https://hal.inria.fr/hal-01081339

[5] T. Nguyen and N. L. Thanh, "Ensuring the correctness of business workflows at the syntactic level: An ontological approach," in *Intelligent Information and Database Systems - 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14-16, 2016, Proceedings, Part II*, 2016, pp. 533–543.

[6] M. Kradolfer, "A workflow metamodel supporting dynamic, reuse-based model evolution," Ph.D. dissertation, 2000.

[7] I. Markovic and A. C. Pereira, "Towards a formal framework for reuse in business process modeling," in *Proceedings of the 2007 International Conference on Business Process Management*, ser. BPM'07. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 484–495. [Online]. Available: http://dl.acm.org/citation.cfm?id=1793714.1793769

[8] R. Lu, S. Sadiq, and G. Governatori, "On managing business processes variants," *Data Knowl. Eng.*, vol. 68, no. 7, pp. 642–664, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2009.02.009

[9] A. Koschmider and H. A. Reijers, "Improving the process of process modelling by the use of domain process patterns," *Enterprise IS*, vol. 9, no. 1, pp. 29–57, 2015. [Online]. Available: http://dx.doi.org/10.1080/17517575.2013.857792

[10] J. Mendling and C. Simon, "Business process design by view integration," in *Business Process Management Workshops, BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings*, 2006, pp. 55–64. [Online]. Available: http://dx.doi.org/10.1007/11837862_7

[11] R. Eshuis and P. W. P. J. Grefen, "Constructing customized process views," *Data Knowl. Eng.*, vol. 64, no. 2, pp. 419–438, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2007.07.003

[12] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data Knowl. Eng.*, vol. 70, no. 6, pp. 483–503, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2011.02.002

[13] G. Greco, A. Guzzo, L. Pontieri, and D. Saccà, "An ontology-driven process modeling framework," in *Database and Expert Systems Applications*, ser. Lecture Notes in Computer Science, F. Galindo, M. Takizawa, and R. Traunmüller, Eds. Springer Berlin Heidelberg, 2004, vol. 3180, pp. 13–23. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30075-5_2