Autocompletion for Business Process Modelling

Karol Wieloch, Agata Filipowska, and Monika Kaczmarek

Department of Information Systems,
Faculty of Informatics and Electronic Economy,
Poznan University of Economics
{a.filipowska,k.wieloch,m.kaczmarek}@kie.ue.poznan.pl

Abstract. This paper presents an idea and prototype of the semantic-based autocompletion mechanism supporting development of business process models. Currently available process modelling tools support business analysts by suggesting elements that may be incorporated in the process, validating modelled processes, providing additional descriptions easing automation, etc. However, these solutions based mainly on syntactic data, disregard proper identification and usage of previously modelled process fragments. The mechanism described in this paper analyses context and annotations of process tasks (also on the semantic level) in order to deliver a list of suggestions for possible successor tasks: process fragments that may complete the model being developed.

We argue that the proposed autocompletion mechanism has an ability to improve the efficiency of the modelling process by among others reducing modelling errors and shortening the duration of the modelling process.

1 Introduction and Problem Description

Business process management (BPM) encompasses methods, techniques, and tools to design, enact, control, and analyse operational business processes involving humans, organizations, applications, documents, and other sources of information [1]. Typically, BPM follows a life cycle that consists of four phases, namely: design (modelling), implementation, enactment, and analysis [2].

Business process modelling being an introductory phase of the whole BPM lifecycle is a way of collecting, documenting and analysing processes. This is usually done by business analysts: experts being able to describe every detail of a business process taking place in a company or public administration.

There is a number of tools supporting the process modelling, some of which enable also process execution, monitoring and further analysis (supporting the whole process lifecycle). Among these tools the most popular ones are ARIS Platform, iGrafix and Proforma.

These tools, being very usable taking into account different aspects, however, provide limited support for users when it comes to intelligent process modelling. Typical process modelling resembles a scratchboard rather than a technical approach. The guidance offered concerns mainly process syntactic data and the

W. Abramowicz, L. Maciaszek, K. Węcel (Eds.): BIS 2011 Workshops, LNBIP 97, pp. 30–40, 2011.

process semantics is omitted. But in a company, some of the processes interact, some of them include similar process fragments. If an expert is unaware that a fragment was previously modelled, he spends time on re-modelling it. Often also a significant training is required to teach people how to model their processes using a given tool or notation.

Additional issues concern the fact, that business models are usually transformed manually into executable models and the new, executable process models are neither understandable nor available for changes to business experts. These process models stay out of reach of process experts and are not used while modelling new processes.

In this paper we address the problem of supporting business analysts while modelling processes by providing them with a set of process fragments that semantically match the process they are working on. This list is prepared based on the description of the process being modelled. The only requirement is that the process needs to be (at least partially) semantically annotated. We call this functionality an autocompletion mechanism similarly to what is available in the Integrated Development Environments (IDEs).

The work presented in the paper was a part of the approach developed in the FP6 EU SUPER[1] project regarding the Semantic Business Process Management. The SBPM is to close the Business-IT gap by using semantic technologies [3]. Similarly to how Semantic Web services achieve more automation in discovery and mediation as compared to conventional Web services, in SBPM more automation should be achieved in process modelling, implementation, execution and monitoring phases by using ontologies and Semantic Web services technologies.

The autocompletion mechanism and component that was developed is integrated with the BPMO editor (version of the WSMO Studio[2]).

The paper is structured as follows. Next section presents the overview of the related work concerning the issue of process autocompletion. Then a description of the scenario follows. Next, we present the completion strategies and the mechanism overview. The article concludes with an overview of the architecture of the solution and a brief summary.

2 Related Work

Our approach is related to the work of [4], which also concerns semantic business process modelling. In this paper the author describes methods and techniques to support modelling of semantically annotated business processes in Petri Nets focusing on measuring the similarity of process task labels to suggest matching process fragments. We developed our approach for Business Process Modeling Notation (BPMN) and its underlying metamodel BPMO, being less restrictive when it comes to process modelling. Here, we use the notion of the process fragment and their decomposition suggested by [5].

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

[6] present an approach for supporting the modelling of business processes using semi-automated Web service composition techniques. They take into account the functional part of service descriptions when making suggestions during modeling. Similarly to [6], [7] in their work focus on implementation aspects considering also the process context and non-functional properties in addition to the functional properties, thus increasing the level of precision of the suggestions.

We also apply the ontology stack described in detail in [8]. In our work we focus on enhancing user experience while modelling processes by making the improving the effectiveness.

3 Scenario and User Interface

In this section we describe the principles of the autocompletion mechanism based on naming and functional annotations of process tasks and discuss its implementation within the semantic modelling tool developed within the SUPER project.

3.1 BPMO Editor and Available Tasks' Descriptions

The proposed autocompletion mechanism is envisioned to constitute an improvement within process modelling activities. Therefore, it needed to be implemented as a prototype within one of the available business process modelling tools. However, within our work we wanted to focus on the tool supporting the semantic annotation of business process models being the on-going trend in the area of BPM. Therefore, the decision has been reached to implement the proposed mechanism within the Business Process Modelling Ontology (BPMO) Editor for modelling semantic business processes developed within the EU 6FP SUPER project. The BPMO modelling environment [9] provides a basic functionality for adding semantic annotations to the process models. It operates on the Business Process Modelling Ontology [10] being an abstraction over the BPMN and EPC notations. Within the BPMO a task is defined as a Business Activity (Figure 1) and

```
Concept BusinessActivity subConceptOf upo#BusinessActivity
    hasName ofType (0 1) _string
    hasDescription ofType (0 1) _string
    hasNonFunctionalProperties ofType (0 1) BusinessActivityNonFunctionalProperties
    hasBusinessDomain ofType upo#BusinessDomain
    hasBusinessFunction ofType upo#BusinessFunction
    hasBusinessStrategy ofType upo#BusinessStrategy
    hasBusinessPolicy ofType upo#BusinessPolicy
    hasBusinessProcessMetrics ofType upo#BusinessProcessMetrics
    hasBusinessProcessGoal ofType upo#BusinessProcessGoal
    hasBusinessResource ofType upo#Resource
```

Fig. 1. An excerpt from the BPMO ontology

thus, can also refer to business attributes such as a business policy or a business process goal. Of course, tasks themselves have additional attributes to represent information about interactions with a partner process.

From our perspective the most important implications from using BPMO are as follows:

- All processes, process fragments and tasks are instances of BPMO concepts and as such are identified using IRIs (Internationalized Resource Identifiers);
- Some attribute types in Tasks are defined at a syntactic level, e.g. hasName ofType String.
- Some attribute types in Tasks are defined at a semantic level i.e. the specific values of task's attributes refer to the IRI of a concept within an ontology. The attribute of the special interest is the hasBusinessFunction pointing to the instances from the Business Functions ontology [8]. This allows for taking advantage of reasoning during the autocompletion procedure.

The scenario supported within the BPMO editor follows.

3.2 Supported Application Scenario

The following scenario is supported by the proposed solution. A business user is modelling a process and wants the system to suggest a task or a process fragment that would follow the one just added to the diagram (Figure 2).

As there might be many unfinished branches within the process flow, the user must explicitly select the task to be auto-completed. Before the auto-complete function is fully activated a user needs to select autocompletion strategy he wants to follow. Once the auto-completer returns a list of matching process fragments (Figure 3) the user selects the most appropriate fragment.

The chosen fragment replaces the selected task and possibly some more preceding ones depending on the strategy selected (Figure 4).

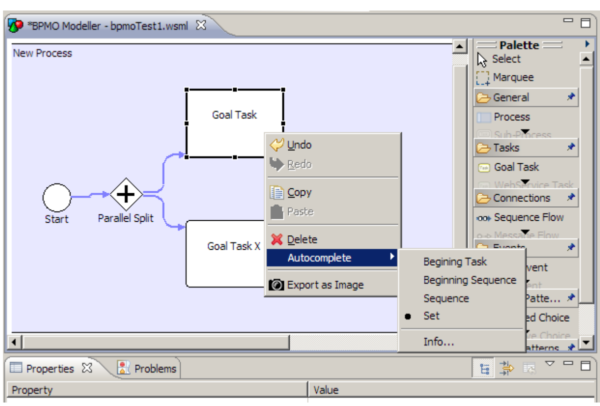

Fig. 2. An initial state of the modelling environment for the Auto-completer

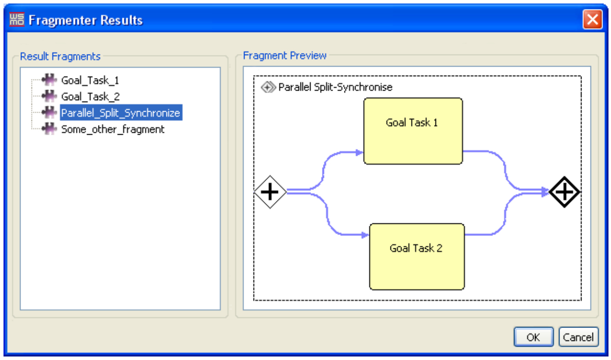

Fig. 3. A list of suggested fragments

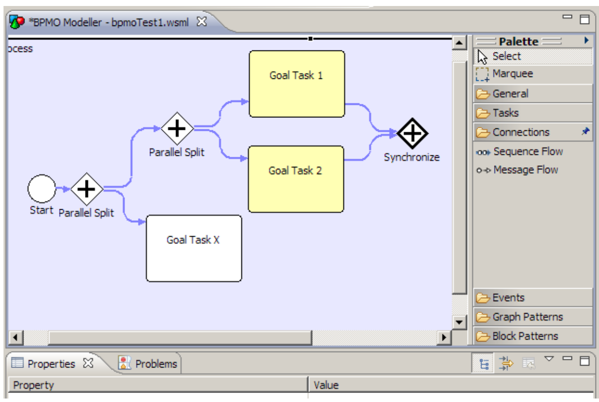

Fig. 4. The result of a single auto-complete action

The autocompletion strategy defines which part of an already modelled process is used as a query for finding potentially matching fragments and how the matching is performed. Five different autocompletion strategies have been defined and implemented within the tool. Their short overview follows.

3.3 Autocompletion Strategies

Beginning Task Match Strategy. The Beginning Task Match is the simplest autocompletion strategy. A user selects a task to be auto-completed. The auto-completer retrieves all tasks whose activities and names match the selected task (i.e. the similarity is above specific threshold) and ranks fragments they belong to according to their similarity to the user-provided task. After the user selects one of the suggested tasks, the task he initially selected is replaced with the chosen fragment. Effectively the selected task is auto-completed with a remaining part of the matched fragment.

An example follows:

a process modelled by a user: *StartEvent-Task0-Gateway0-Task1-Task2*
a problem to be solved by the auto-completer: *find all the fragments that begins with the task similar to Task2*
returned solution (suggested fragment): *Task2a-Task3*
a process model once a user accepts the suggestion: *StartEvent-Task0-Gateway0-Task1-Task2a-Task3*

Beginning Sequence Match. A user selects a task to be auto-completed. The auto-completer finds the longest sequence of tasks that ends with the task selected by the user (i.e. it "moves back" until a gateway or process beginning is reached). This pattern sequence is used to find matching fragments that start with the sequence of task matching the user-provided sequence of tasks. After the user decides to use one of the suggested fragments, the initially found pattern sequence of task will be replaced with the chosen fragment. Effectively the task selected by the user is auto-completed with a remaining part of the matched fragment.

An example follows:

a process modelled by a user: *StartEvent-Task0-Gateway0-Task1-Task2*
a problem to be solved by the auto-completer: *find all the fragments that begins with the sequence similar to Task1-Task2*
returned solution: *Task1a-Task2a-Task3-Task4*
a process model once a user accepts the suggestion: *StartEvent-Task0-Gateway0-Task1a-Task2a-Task3-Task4*

Sequence Match. The strategy is a more general version of the "Beginning Sequence Match". In this strategy the query is constructed in the same way as in the previous strategy. The difference is that the auto-completer discovers all fragments that contain sequence of tasks matching sequence of tasks provided by user at any location within them. As the result the task selected by the user is followed by a remaining part of the matched fragment and is preceded by the initial part of the selected fragment.

An example follows:

a process modelled by a user:
StartEvent-Task0-Gateway0-Task1-Task2

a problem to be solved by the auto-completer:
find all the fragments that contains the sequence similar to Task1-Task2

returned solution:
Task5-Task1a-Task2a-Task4-Task3

a process model once a user accepts the suggestion:
StartEvent-Task0-Gateway0- Task5-Task1a-Task2a-Task4-Task3

Set Match. This strategy generalises the "Sequence Match" strategy in a sense, that the auto-completer does not pose a sequence restriction on potentially matching fragments. It returns all fragments that contain tasks matching all user-provided tasks, regardless of their positions, order and existence of additional tasks between them.

An example follows:

a process modelled by a user:
StartEvent-Task0-Gateway0-Task1-Task2

a problem to be solved by the auto-completer:
find all the fragments that contain tasks similar to Task1 and Task2

returned solution:
Task5-Task2a-Task3-Task1a-Task3

a process model once a user accepts the suggestion:
StartEvent-Task0-Gateway0- Task5-Task2a-Task3-Task1a-Task3

Auto-complete Name. This additional functionality provides a user with suggestions of the full name of a task whose name is only partially typed.

4 Matching Procedure

This section describes the matching procedure we have applied within the proposed solution. Matching consists of calculating distance value on four levels between following objects:

- task's attribute vs. task's attribute (attributes level),
- task vs. task (tasks level),
- set or sequence of tasks vs. other set or sequence of tasks (sequence/set level),
- one or more tasks (depends on strategy) vs. process fragment (fragment level).

Each level requires different procedure for calculating distance value as discussed within the following subsections.

4.1 Attributes Level

As already mentioned in the previous section, in the editor the tasks are described according to the BPMO ontology. At the level of task's characteristics we decided to consider two attributes: hasName and hasBusinessFunction.

Distance between task names is calculated using one of the standard string distance measures [11].

Distance between business functions assigned to a two given tasks is calculated using the following formula:

$$d(i,j) = \frac{CPL(i,j)}{\frac{H(i)+H(j)}{2}} \tag{1}$$

where $d(i,j)$ denotes the distance value, $CPL(i,j)$ is the length of the common path in the Business Function Ontology subsumption hierarchy (starting from the root) of the two concepts i and j; $H(i)$ is the length of the path in subsumption hierarchy from the root to the concept i.

4.2 Tasks Level

The distance between tasks is calculated as the average of the attribute level distances.

4.3 Sequence/Set Level

The similarity of two sequences is calculated as the arithmetic sum of similarity between respective pairs of tasks.

The similarity of two sets is calculated as the maximum of similarity of user-provided set and all possible subsets of the same size.

4.4 Fragment Level

The similarity of a user query to a specific process fragment is calculated as the maximum of similarities between set or sequence provided as user query and all possible sets or sequences belonging to specific fragment. For example in case of Beginning Task Match a single task is compared against first task of each fragment, while in case of Set Match set of n tasks is matched against each n-element subset of tasks of each fragment and the maximum of obtained similarity measures is used as query-to-fragment similarity.

The similarity measure is used while providing a user with a ranking of process fragments matching the process a user is working on. The more relevant fragments, the higher in the ranking they are presented. In case the similarity measure is 0, the process fragment is not included in the list presented to the user. When, the number of process fragments will increase, certain limits on the similarity measure (show only top 10% of matching processes) may be introduced.

5 Solution Architecture

This section provides some insights into the architecture of the proposed solution which is presented in the Figure 5.

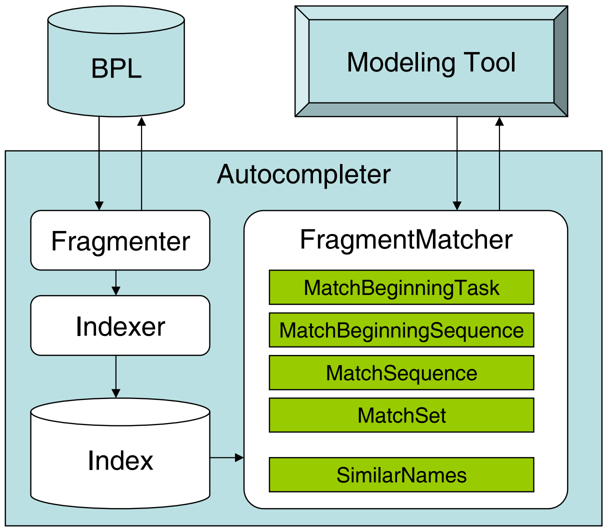

Fig. 5. Data flow within the Auto-completer and between other components

The Autocompleter is a separate component that communicates with the Business Process Library (BPL) and needs to have a graphical user interface (modelling editor). As such it can be used within any modelling tool.

The following components can be distinguished within the autocompleter itself:

– Fragmenter,
– Indexer along with the Index repository,
– Discoverer.

Discoverer is the main subcomponent of the auto-completer. It is responsible for implementing various autocompletion strategies. Each of matching strategies is implemented as a plug-in using the same index structure and implementing the same interface. Thus, a number of additional matching strategies may be implemented.

Fragmenter is responsible for identifying SESE fragments (Single Entry Single Exit fragments) taking advantage of the algorithm used e.g. in [5], being able to identify all SESE fragments in linear time, within processes stored in the

Business Process Library. Identified fragments are stored in the BPL again and send to the Indexer.

The role of the Indexer is to retrieve all relevant information regarding process fragments that is required during the discovery phase. The retrieved information is stored within an internal data structure (index).

Two structures are used to index process fragments and utilised in answering user queries. First and the simplest structure consists of a list of all tasks present (with IRIs of corresponding process fragments), identified by an IRI. The second structure contains all n-grams (including 1-grams) of tasks, specifying also if a specific sequence corresponds to the beginning of specific fragment.

The index is used basically to select candidate sequences or tasks that may be similar to a user query. In case of Beginning Task Match and Beginning Sequence Match strategies n-grams (n being the number of tasks in user query; n=1 in case of Beginning Task Match) that correspond to a start of any fragment are selected. In case of Sequence Match all n-grams are selected (whatever their position in the task is). Finally, in case of the Set Match, first the search space is limited to all fragments that contain at least n elements (i.e. that contain a sequence of at least n elements); next all subsets of n elements of all these fragments are considered for similarity measurement.

The index is implemented as an SQL database (HSQLDB Java library is used). Each of match strategies is responsible for construction of an SQL query performing selection of IRIs of process fragments. The similarity measure for two tasks in accessible directly in a query (as a stored procedure, implemented in Java). The results of such query are handled by the Discoverer module.

6 Conclusions

This paper presents a novel approach to improve existing tools that support business process modelling. Our solution supports the business analysts actively during their work helping them to find process fragments completing the process they are working on. This may lead to increasing effectiveness of the process modelling by avoiding the duplication of work (working on process models that were already developed).

The presented approach consists of two distinct phases: process fragments identification and their description and application of previously identified process fragments in processes being modelled. Although our algorithm operates on the semantic annotations of process models, we may also operate also on the syntactic level what increases the application possibilities of our approach.

The approach was tested by business users, that underlined its usefulness while process modelling. Some of the major benefits offered by our approach are i) identification of process elements that may enrich the process being modelled, ii) suggestions based not only on syntactic, but also semantic information, iii) improved model quality and faster modelling process.

References

1. Van Der Aalst, W.M.P., Hofstede, A.H.M.T., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Muehlen, M.Z., Rosemann, M.: Multi-paradigm process management. Development (Bpmds), 169–175 (2004)
3. Hepp, M., Leymann, F., Bussler, C., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: a vision towards using semantic web services for business process management. In: IEEE International Conference on eBusiness Engineering, ICEBE 2005, vol. 5(1), pp. 535–540 (2005)
4. Koschmider, A.: Ähnlichkeitsbasierte Modellierungsunterstützung für Geschäftsprozesse. Universitätsverl., Karlsruhe (2007)
5. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 43–55. Springer, Heidelberg (2007)
6. Schaffner, J., Meyer, H., Weske, M.: A Formal Model for Mixed Initiative Service Composition. In: IEEE International Conference on Services Computing, pp. 443–450. IEEE Computer Society (2007)
7. Born, M., Brelage, C., Markovic, I., Pfeiffer, D., Weber, I.: Auto-completion for Executable Business Process Models. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2009. LNBIP, vol. 17, pp. 510–515. Springer, Heidelberg (2009)
8. Born, M., Markovic, I., Filipowska, A., Starzecka, M., Kaczmarek, M., Walczak, A.: Business Functions Ontology and its Application in Semantic Business Process Modelling, pp. 136–145 (2008)
9. 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. CEUR Workshop Proceedings, vol. 251, CEUR-WS.org (2007)
10. Cabral, L., Norton, B., Domingue, J.: The Business Process Modelling Ontology. In: Proceedings of the 4th International Workshop on Semantic Business Process Management, SBPM 2009, pp. 9–16. ACM, New York (2009)
11. Markovic, I., Pereira, A.C.: Towards a formal framework for reuse in business process modeling. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 484–495. Springer, Heidelberg (2008)