Management of Interorganizational Business Processes on the Basis of Composition Rules

Agnes Koschmider[1], Marco Mevius[2]

[1] Institute of Applied Informatics and Formal Description Methods
Universität Karlsruhe (TH), Germany
koschmider@aifb.uni-karlsruhe.de

[2] Research Center for Information Technologies
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
mevius@fzi.de

Abstract. Process modeling facilitates understanding and restructuring activities used to achieve business goals. The need for rapid deliveries of services to customers has fueled the integration of different business partners, into a single value creation chain. But, integration of interorganizational business processes in electronic markets is a difficult and time-consuming task. By using formal description languages such as Petri nets for modeling interorganizational business processes, purely syntactic integration problems of distributed business environments can be solved. In this paper we present syntactic and semantic rules, which have to be satisfied, in order to support holistic interconnectivity of interorganizational business processes.

Keywords. Business processes, Petri nets, interconnectivity, composition, OWL

1 Introduction

Process modeling facilitates understanding and restructuring activities used to achieve business goals. The need for rapid deliveries of services to customers has fueled the integration of different business partners, into a single value creation chain. But, the interconnectivity of interorganizational business processes in electronic markets is a difficult and time-consuming task. Business processes of different companies have to fit in another organizational environment and they have to complement each other. By using Petri nets [12] for modeling interorganizational business processes, purely syntactic interconnectivity problems of distributed business environments can be solved. Moreover, Petri nets obey an operational semantics that facilitates composition, simulation, and validation of distributed business processes. The missing semantic representation of Petri net components can hamper the interconnectivity of busi-

ness processes. Usually, different users do not use the same vocabulary in order to describe same process models. The detection of different terms for the same process object names requires a significant amount of experience in the field of process engineering, and may result in extra analysis efforts.
Another problem of interconnectivity of interorganizational business processes is to identify identical processes. Most users mix the abstraction levels of process activities, in that process activities on the same process level are sometimes modeled in-depth, and sometimes rather coarsely. Inconsistently modeled processes return distorted analysis results, and may lead to costly erroneous judgements or disregard relevant information.

In this paper, we describe how to interconnect interorganizational business processes, with respect to the syntax and fitness of processes. Petri nets can be used in order to model complex business process models and their process relevant objects. The formal foundation of Petri nets, and thus the available analysis methods for Petri nets, supports a correct syntactical interconnectivity of interorganizational business processes. Within available Petri net analysis methods process models can be validated and verified (e.g., [14]). It can be decided if the process model represents the reality or if the models are correct (deadlock freeness, liveness). But it cannot be decided if the interconnected business processes fit together semantically, with respect to the content.

Our paper is organized as follows: in Section 2 we describe how to model interorganizational business processes with Petri nets and the characteristics of such business processes. Section 3 outlines syntactical rules for interconnectivity of interorganizational business processes. In Section 4 similarity measures are sketched, which support to interconnect interorganizational business processes even when using a different vocabulary for same process object names. Furthermore, this section illustrates techniques to identify homogenous and heterogeneous abstraction levels of interorganizational business processes. Related works to our approach are presented in Section 5. The paper concludes with an outlook on future work.

2 Management of interorganizational business processes

In this section we will sketch characteristics of interorganizational business processes and describe how interorganizational business processes can be modeled.

2.1 Modeling

Petri nets constitute a formal graphical process description language, that combines the advantages of graphical representation with a formal semantics of behavior. Formally, a Petri net is a directed bipartite graph with two sets of nodes (places and transitions), and a set of arcs (flow relations). The Numerous Petri net variants that have been proposed, can be divided into two groups: elementary and high-level Petri nets. In elementary Petri nets (place/transition nets), the flow of tokens representing anonymous objects defines the process flow.

For illustration we present a place/transition net (p/t net) [13] that describes the following three-layered inter-organizational manufacturing process (Fig. 1): The assembler (third layer) instantly processes incoming “end-customer orders”. The “end-customer orders” are passed to at least three manufacturers. All manufacturers should respond within a pre-defined time interval. Because of business confidentiality, the order’s end user details are not sent to the manufacturers. Only the part of the order that is relevant to quoting the price is communicated. Each manufacturer calculates an offer and transmits it to the assembler. The assembler collects the lowest quoted price in time, sends a mandate to the selected manufacturer and informs end users about the assured delivery date. The manufacturer again authorizes a raw material distributor. For all raw material orders (both big ones and small ones), the delivery of the raw material is coordinated and the material is inspected. The raw material that is not rejected due to defects is conveyed to the second layer, and the components are manufactured. Thereafter, components are made available to the assembler (of the third layer) for assembling the complete products. The whole order is put on hold if at least one product fails the final testing. The rejected products are repaired and tested again. Figure 1 describes a situation with three end-customer orders, four accepted orders and one final order.

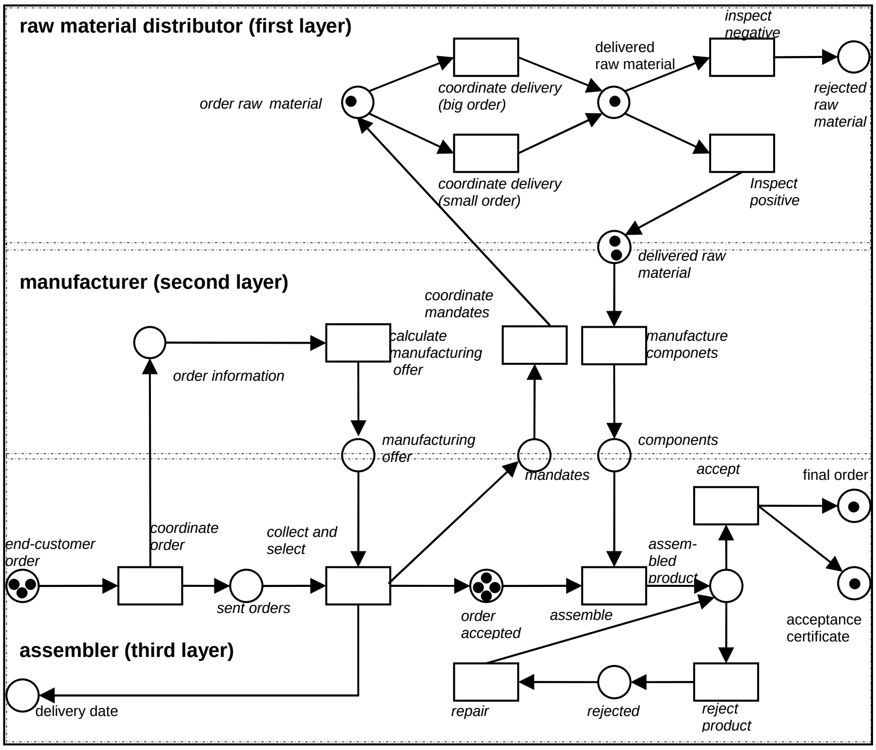

Fig. 1: Place/Transition net for an interorganizational manufacturing process

2.2 Characteristics

Usually, even if they share similar demands, business partners have their own specific vocabularies. A commonly adopted vocabulary is often too hard of a restriction for users in interorganizational cooperation. In Figure 1, the interorganizational business processes of the manufacturer and assembler are interconnected at the places "manufacturing offer", "mandates" and "components". Originally, the assembler had named the corresponding elements in his process "*fabricating offer*", "*mandates*" and "*elements*". The appropriate input/output elements have been computed by our similarity tool based on the similarity measures described in Section 4.

But for the similarity calculations, two modeling guidelines have to be satisfied. The business partners should comply with these guidelines, in order to facilitate a semantic interconnectivity of interorganizational business processes.

Generally, process element names are designated using a "verb and noun" expression, which should illustrate an unambiguous functionality of the process activity. Depending on the modeling scope, users name elements differently in order to describe in detail the flow semantics. Verbs such as *administrate*, *update*, *apply* or *change* usually indicate a function-oriented way of thinking. A review should clarify if the documentation is process-oriented. In order to guarantee appropriate process interconnectivity, we recommend precise element names, which simultaneously lay the foundation for calculating the similarities between different vocabularies that are used for same process element names.

Most users mix the abstraction level of process activities, in that process activities on the same process level are sometimes modeled in-depth and sometimes abstractly. Petri nets support the representation of systems on different abstraction levels. When coupling interorganizational business processes, the elements to be interconnected should posses the same abstraction level, which represents the second guideline to comply with. We recommend maintaining an identical abstraction level of element names on the same process level. Different abstraction levels of business process models can be obtained in a top-down or bottom-up fashion. In top-down modeling, the top level process formulates an overview of process elements, without providing more detailed descriptions of process elements. Consequently, the top view of the process is structured in a more fine-grained way, by refining transitions to subprocesses. By using the bottom-up approach, modelers start modeling more specific process elements, which are subsequently linked together to coarse-grained processes. The linking or coarsening is done until a complete abstract view of the process model is determined. The main advantages of the bottom-up approach are a priori simulation and analysis capabilities of process fragments, before being integrated into a complete process model.

In the following sections we describe how to syntactically interconnect processes into a single value creation chain, as depicted in Figure 1.

3 Syntactic rules for interconnectivity of interorganizational business processes

A method for the syntactic interconnectivity of Petri nets is presented, which takes into consideration the formal requirements concerning process behavior. The method is the inverse of the one presented in [4].

3.1 Routing patterns

Four routing patterns exist for business processes, which represent the foundation when interconnecting interorganizational business processes. The routing patterns include sequential execution, choice, iteration and parallel execution. In sequential execution, tasks are executed one after the other. The choice of routing pattern allows to model alternative branching (a place or a transition has at least two postsets). Iteration allows the execution of a particular task, several times. By the use of the parallel routing pattern, tasks are executed independently. The two parallel branches are then integrated by a synchronization (place or transition has at least two presets).

3.2 Method

The business process fragments to be interconnected can be either disjoint (no overlapping input/output elements), or non disjoint (overlapping elements). Figure 2 shows non disjoint processes, in which the same process elements are highlighted in gray.

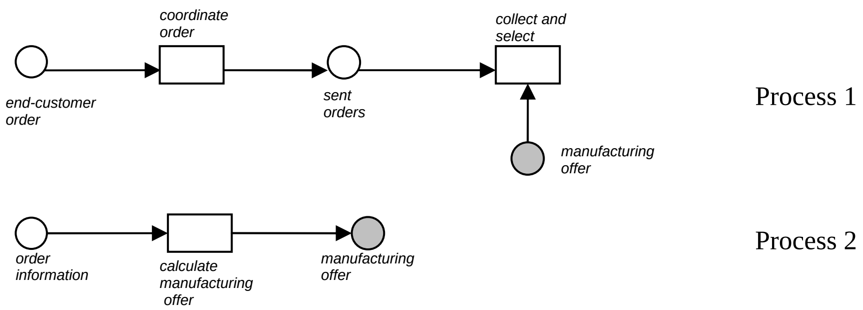

Fig. 2: Non disjoint processes

We distinguish between three types of interconnectivity: vertical, which is applicable for sequential processes; horizontal, which is applicable for processes with choice or parallel routing patterns, and diagonal, for processes with iteration. In a vertical composition, duplicate elements are replaced by one unique element (Fig. 3a). Processes are interconnected horizontally, if none of the processes depends on any other ones

(Fig. 3b). If processes depend on one another, then they can be interconnected diagonally (Fig. 3c).

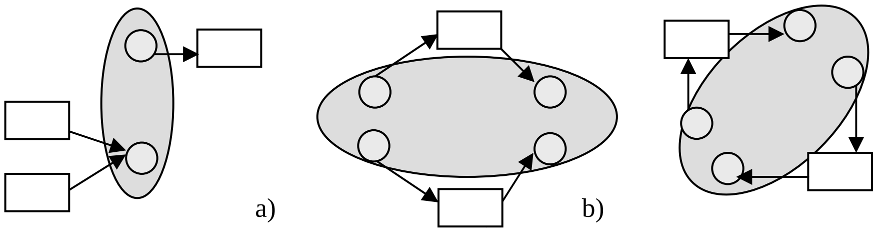

Fig. 3: vertical (a), horizontal (b) and diagonal (c) interconnectivity of Petri nets

As a commonly agreed vocabulary is often too hard of a restriction for the users, the next section describes how to calculate semantic similarities between business process element names.

4 Semantic rules for interconnectivity of interorganizational business processes

To support the interconnectivity of interorganizational business processes with respect to content, we describe Petri nets with the Ontology Language OWL [15], resulting in so-called semantic business process models. OWL was proposed to make it particularly easy to model data in a machine-interpretable form. OWL may enable automation of a variety of tasks, currently being performed "manually" by human agents.

The OWL-based description of Petri nets [8] makes it possible to (semi-) automatically manipulate business process models. Each Petri net element has a corresponding concept in the OWL-based description. The set of places corresponds to the concept “Place”, the set of transitions to the concept “Transition”, the set of arcs connecting places with transitions to “FromPlace”, and arcs connecting transitions with places to “ToPlace”. Figure 4 shows a simple example of an OWL-based description. The concept `Place` is a subconcept of the concept `PetriNet` and where `Place` has an ObjectProperty `transRef` referring to the subsequent element `Transition` (expressed via the domain `Place` and the range `Transition` of the ObjectProperty `transRef`):

```
<owl:Class rdf:ID="Place">
 <rdfs:subClassOf rdf:resource="#PetriNet"/>
  <owl:ObjectProperty rdf:ID="transRef">
   <rdfs:domain rdf:resource="#Place"/>
   <rdfs:range rdf:resource="#Transition"/>
  </owl:ObjectProperty>
 ...
</owl:Class>
```

Fig. 4: Example for OWL syntax

The OWL serialization forms the input for the similarity calculation of distributed business processes. In the first step, our similarity tool automatically extracts names of the same concept (places versus places and transitions versus transitions). Then, in the second step, it calculates the similarity as described in the following subsections. We sketch five similarity measures for calculating the semantic similarities. The aim of similarity measurements is twofold: to indicate the degree of similarity, and based on the degree of similarity, to decide if processes should be interconnected or not.

4.1 Classification

In order to classify similarity measures, we rely on the lexical taxonomic structure of WordNet [3]. WordNet is an English online lexical reference system, which provides synonym-, and hyperonym-, hyponym sets, consisting of nouns, verbs, adjectives, and adverbs, as listed below:

- synonym: two terms have an identical meaning
- hyperonym/hyponym: two terms have an is-a relationship; subclass/class relationship

We extend the existing structure of WordNet with homonyms (two terms have same pronunciation, but different meaning), and use this extended structure of terms as our classification scheme for similarity measurements.

4.2 Interconnectivity regarding the similarity

In order to automatically uncover synonyms, homonyms, or terms with different abstraction levels in elements to be interconnected, we have defined several semantic similarity measures. The similarity measures ensure that all appropriate process elements can be proposed for interconnectivity, even when using different vocabularies. The similarity measures that were considered are syntactic, linguistic, structural and abstraction level-based, as reported in [2].

Intuitively, the degree of similarity between process models correlates positively with the number of used synonyms, and negatively with the number of used homonyms. Synonyms can be detected by a linguistic similarity measure that exploits all senses of a term, as proposed by WorldNet’s synonym relationship. Typos in process element names can hamper the correct calculation of linguistic similarity degrees. For instance, the terms (receive letter vs. recive leter) have a linguistic similarity of 0.0 due to spelling mistakes. Therefore we have extended an existing syntactic similarity measure for ontologies, to support character string comparison. Some results of the linguistic (sim_{ling}) and syntactical (sim_{syn}) similarity measurement are given in Table 1.

Name	sim_{ling}	sim_{syn}
manufacturing offer vs. fabricating offer	0.5	0.5294
component vs. element	0.3334	0.1429

Tab. 1: Result of linguistic and syntactic similarity calculations

Neither of the similarity measures can support the detection of homonyms. The syntactic and linguistic similarity for the terms (order vs. order) equals 1.0, because of identical character strings; but the terms may have different meanings (on the one hand an order may be a commercial document used to request someone to supply something, and on the other hand it may be a formal association of people with similar interests). A so-called context of process elements supports homonym detection, by defining the set of all elements which influence the name's similarity. Furthermore, we assign individual weights to each context element, in order to consider the different influences of each context element.

To detect hyperonyms/hyponyms, we compute abstraction level-based similarities, which take into account the depth of terms in lexical reference systems such as WordNet. The abstraction level similarity is only calculated in the case of a linguistic or structural similarity of 0.0. Finally, we aggregate the four similarity measures with particular weights, into a combined similarity measure (sim_{com}).

An excerpt of similarity results is shown in Table 2. The second value is the result of the syntactic calculation, the third one from the linguistic measurement, and the fourth result returns from structural measurement. The last corresponds to abstraction level-based measurements. The aggregation of all similarity measures is the first value.

Name	sim_{com}	sim_{syn}	sim_{ling}	sim_{str}	sim_{abs}
manufacturing offer vs. fabricating offer	0.8179	0.5294	0.55	1.0	0.0
component vs. element	0.6142	0.1429	0.4	0.8	0.0

Tab. 2: Excerpt of result of similarity calculations

The calculated similarity degrees allows making assertions, whether or not specific elements should be interconnected. It is up to the user to define a meaningful threshold, which indicates "good" and "bad" elements. Our user study has shown that a similarity degree of 0.4 is a satisfying threshold. In Table 2, both element pairs satisfy this threshold, and therefore the corresponding elements will be interconnected.

4.3 Interconnectivity regarding the abstraction level

Next, when realizing methods for interconnectivity of interorganizational business processes, we have to consider the abstraction level of the interconnected business processes (under the assumption that syntactic properties are fulfilled). Each decomposition level describes process elements from a different abstraction level. Top level process models formulate an overview of process activities, while lower level models provide more detailed descriptions. However, in order to improve model consistency, users have to maintain particular modeling requirements, such as the homogenous abstraction of process element names on the same decomposition level. The abstraction level can be identified when measuring the linguistic specificities of element names. Our measurement is based upon the hypothesis that in fine-grained processes, element names are more specific than in coarse-grained ones. The specificity of names indicates their process decomposition level, and thus their abstraction level.

In order to reach our objective, we identified three different so-called coarsening patterns: “Identical Nouns”, “Heterogeneous Nouns” and “Specialization of Nouns” [7]. We called them coarsening patterns, because our detection system isolates element name sequences of one process regarding abstraction homogeneity and heterogeneity, and proposes coarsening in case of nonuniformly specified names. Interrelated names with heterogenic abstraction as their pre– and postsets will be suggested as coarsening candidates as well.

Users may name elements in subprocesses with a noun that is identical to the one of the transition to be refined. Figure 5 shows the first coarsening pattern, where the term “product” occurs in all subprocess element names, in connection with additional verbs.

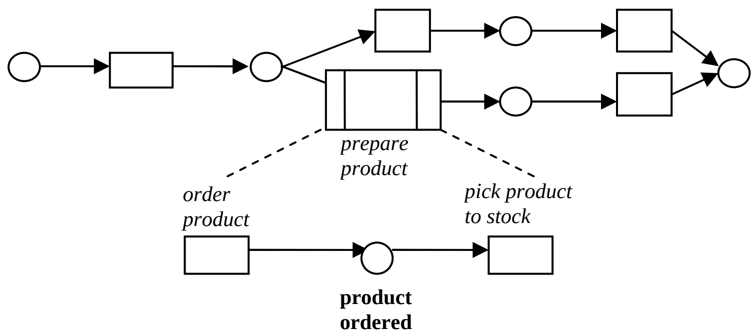

Fig. 5: Coarsening Pattern 1 “Identical Nouns”

As a second pattern, we identified that the name (respectively noun) of the transition to be coarsened is heterogeneous from the subprocess elements, as shown in Figure 6.

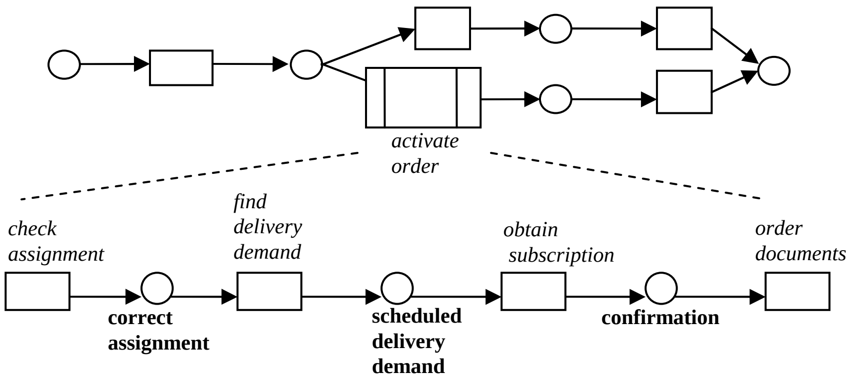

Fig. 6: Coarsening Pattern 2 “Heterogeneous Nouns”

These two explained patterns can be combined where subprocess elements are named with heterogeneous and identical nouns. In the third coarsening pattern, the subprocess nouns can be regarded as a specialization of the specific transition to be refined. In Figure 7, “contract application” and “contract certificate” represent a specialization of the noun “contract”.

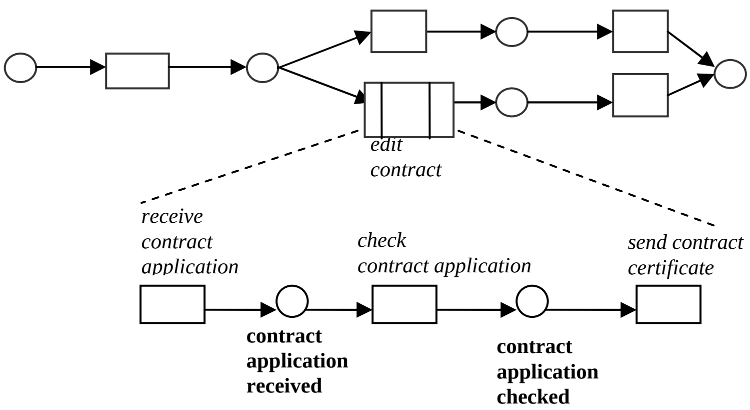

Fig. 7: Coarsening Pattern 3 “Specialization of Nouns”

The three defined patterns, and the abstraction level-based similarity degree for each element name, promise to detect non-uniformly specified process elements to be interconnected. But, such (semi-)automatic detection requires that the system autonomously subsumes elements to one of the patterns. The subsumption of elements can be particularly performed by first integrating all processes/subprocesses into one process and then analyzing the linguistic specificity of each name. The integration approach is adopted from [9]. For the integration procedure, we assume that processes are modeled in a top-down manner. If a transition to be refined has a cycle, then the cycle is broken into two subprocesses. Besides linguistical aspects of names, we have to consider routing patterns (as explained in Section 3.1) in order to isolate appropriate sequences.

In [7], two algorithms are presented which compare the abstraction level of interconnected business processes. In the following, we explain the algorithm to detect pattern 1. After the integration of all processes from the top-down to one process, the algorithm traverses the process regarding choice or concurrency branches. The algorithm regards process branching as a breakpoint, and tries to find one of the three coarsening patterns in a branch. In our consideration, only a branch B ($\in B_1,...,B_n$), having more than two transitions or at least four elements $e_1,\ldots,e_n$ from E_B, is a reasonable analysis sequence. First of all, the algorithm traverses as long as a branching occurs. This (branching) sequence B will be analyzed considering coarsening patterns. Subsequently, the algorithm traverses to the next sequence and so on. If more than half of the elements of a branch are interrelated elements, with lower similarity as the remaining elements, then these elements might be subsumed to one of the three coarsening patterns.

In Figure 1, the interorganizational process of the manufacturer and the assembler has a parallel branching, which is again integrated by transition "collect and select". The first breakpoint for the algorithm is the branching at transition "coordinate order". As the sequence has only two elements, the algorithm traverses to the next branching ("collect and select"). The algorithm regards elements from "end-customer order" till "collect and select" (the upper branch included) as one sequence, and "end-customer order" till "collect and select" (the bottom branch included) as a second one. In the upper branch, the algorithm finds no coarsening possibility. A consistent hierarchical decomposition would return no coarsening possibilities in the bottom branch either. As no coarsening possibilities can be found, the processes can be interconnected with respect to their abstraction level homogeneity.

5 Related work

Business processes can be regarded as a composition of loosely-coupled web services. A lot of research has been done in the field of web services. We refer to [10] for a list of current solutions for web service composition.

With respect to similarity calculations, [11] presents an extensible business modeling tool, supporting the semantics of business vocabulary and business rules standard, and allowing business modelers to capture and formalize business knowledge in a fact-oriented and natural language approach. In a preceding section, we described our similarity measurement approach, which makes it obsolete to use a controlled vocabulary.

The composition of subsystems through hierarchic classification of process models has been proposed for several modeling languages. A series of tools for process modeling make it possible to insert process hierarchies via coarsening concepts. To the best of our knowledge there exists no work which supports the feature of (semi-) automatic detection of non-uniformly specified process elements, on the same abstraction level.

The concepts presented in this paper lay the foundation for realizing a modeling support for business processes. Users can be assisted in modeling business processes, by recommending process fragments during the modeling process, from a process repository. The recommendations can be used to complete the process model being edited. In this scenario, even a commonly agreed-on vocabulary for process element names stored in the process repository is a too hard of a restriction. Furthermore, the recommended process fragments have to maintain an identical abstraction level as the edited business process. Processes with consistent hierarchical specifications guarantee correct process analysis and workflow views.

In [1], the basic idea of such a recommendation mechanism is presented. In [6], we described for the first time, how autocompletion for business processes can be performed when considering business rules.

6 Conclusion and outlook

The rapid growth of data and communication technologies demand that companies focus on the content of data. In this paper, we presented methods that allow us to interconnect interorganizational business processes, on the syntactic and semantic level.

In order to decide about similarity degrees between process elements to be interconnected, we have sketched syntactic, linguistic, structural, and abstraction level-based similarities. Three so-called coarsening patterns were illustrated, in order to help subsume elements to one of the patterns, and then detect the modeling abstraction level homogeneity and heterogeneity of the processes to be interconnected.

The benefits of our approach are flexibility and automation of involved systems, in order to facilitate the interconnectivity of business processes, and to shorten communication among process-implementing software components. Our approach facilitates rapid adoptions to a changing environment, due to a reduced process modeling effort.

References

1. Betz, S., Klink, S., Koschmider, A., Oberweis, A.: Automatic user support for business process modeling. In Proceeding of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, pp. 1–12, Budva, Montenegro, 2006.
2. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM 2007), volume 67, pp. 71-80. Australian Computer Science Communications, Ballarat, Australia, 2007
3. Fellbaum, C.: WordNet: An electronic lexical database, MIT Press, 1998
4. Guth, V., Lenz, K., Oberweis, A.: Distributed Workflow Execution Based on Fragmentation of Petri Nets, in: R. Traunmüller, E. Csuháj-Varjù (eds.): Proceedings of 15th IFIP World Computer Congress 'Telecooperation - The Global Office, Teleworking and Communication Tools', Vienna, Sept. 1998, pp. 114-125

5. Hamadi, R., Benatallah, B.: A Petri Net-based Model for Web Service Composition. In: Schewe, K.-D.; Zhou, X. (eds.): Database Technologies, Proc. 14th Australasian Database Conference, (2003), pp. 191-200
6. Hornung, T., Koschmider, A., Oberweis, A.: Rule-based Autocompletion Of Business Process Models. In CAiSE Forum 2007. Proceedings at the 19th Conference on Advanced Information Systems Engineering (CAiSE). Trondheim, Norway, June 2007.
7. Koschmider, A., Blanchard, E.: User Assistance for Business Process Model Decomposition. In First IEEE International Conference on Research Challenges in Information Science, pp. 445-454. April, Ouarzazate, Maroc, April 2007
8. Koschmider, A., Oberweis, A.: Modeling semantic business process models. In Rittgen, P. (eds.): Handbook of Ontologies for Business Interaction. Idea group publishing, 2007
9. Lausen, G.: Modeling and Analysis of the Behavior of Information Systems," IEEE Transactions on Software Engineering, vol. 14, no. 11, pp. 1610–1620, 1988
10. Milanovic, N., Malek, M.: Current solutions for Web service composition, Internet Computing, IEEE, vol.8, no.6, pp. 51-59, 2004
11. Tommasi, M.D., Corallo, A.: SBEAVER: A Tool for Modeling Business Vocabularies and Business Rules. In: Knowledge-Based Intelligent Information and Engineering Systems. Volume 4253 of Lecture Notes in Computer Science, Springer, 2006, 1083–1091
12. Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets I: Basic Models. Lecture Notes in Computer Science, Vol. 1491, Springer-Verlag, Berlin, 1998
13. Reisig, W.: Place/Transition Systems. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.): Advances in Petri Nets. Part I. Lecture Notes in Computer Science, Vol. 254, Springer-Verlag, Berlin Heidelberg New York, 1987, pp. 117-141
14. van der Aalst, W.: The Application of Petri Net to Workflow Management. The Journal of Circuits, Systems and Computers, Department of Mathematics and Computing Science, Eindhoven University of Technology, http://is.tm.tue.nl/staff/wvdaalst/publications/ p53.pdf
15. W3C. OWL Web Ontology Language Overview, February 2004, Recommendation, http://www.w3.org/TR/owl-features/