2009

REDUCING THE VARIATIONS IN INTRA- AND INTERORGANIZATIONAL BUSINESS PROCESS MODELING – AN EMPIRICAL EVALUATION

Dominic Breuker
European Research Center for Information Systems

Daniel Pfeiffer
European Research Center for Information Systems

Jörg Becker
European Research Center for Information Systems

REDUCING THE VARIATIONS IN INTRA- AND INTERORGANIZATIONAL BUSINESS PROCESS MODELING – AN EMPIRICAL EVALUATION

Dominic Breuker, Daniel Pfeiffer, Jörg Becker[1]

Abstract

The objective of this paper is to evaluate the semantic building block-based approach as a means for intra- and interorganizational business process modeling. It is described whether and why the semantic building block-based approach reduces the variations in distributed modeling projects in comparison to traditional modeling approaches. Our argumentation is grounded on the assumption that the specification of a service-oriented architecture (SOA) requires a detailed understanding of the intra- and interorganizational business processes. In order to enable the collaboration of services the underlying process structure must be explicated. In a laboratory experiment the variations of distributed process modeling in the traditional and the building block-based approach have been compared. It could be shown that the semantic building block-based approach leads to considerably fewer variations and, thus, to a more consistent view on the intra- and interorganizational process landscape.

1. Introduction

The implementation of a service-oriented architecture (SOA) requires in-depth knowledge about the underlying business processes. The identification of appropriate services and the collaboration of these services within an organization and across organizational borders presuppose a detailed process documentation. In order to collect this process information it is not sufficient to acquire single, independent processes. Rather, it is necessary to create clarity about the process landscape of an organization and its environment [3]. Therefore, modeling projects in a SOA-context are distributed with respect to personnel, location as well as time and can involve multiple organizations.

Traditional business process modeling languages and methods disregard important aspects of intra- and interorganizational business process modeling. Languages such as Event Driven Process Chains (EPC), Business Process Modeling Notation (BPMN), or IDEF3 do not explicitly address the issues of distributed, cross-organizational process modeling. Variations such as a deviating terminology, a varying grade of abstraction, or a different understanding of the scope of a process are not considered in the languages. The semantic building block-based approach has been

[1] European Research Center for Information Systems, Leonardo-Campus 3; 48149 Münster, Germany.

developed to collect a large number of processes in different organizations [15]. It has been designed to minimize the variations that can occur when multiple modelers are involved. The objective of this paper is to provide empirical evidence that the semantic building block-based approach considerably reduces the variations of distributed business process modeling. The rationale behind this behavior is explained and empirically tested in a laboratory experiment.

The paper proceeds as follows: In the next section the different variations within and between business process models are described that can emerge in distributed modeling projects. In the following section of this paper the fundamental characteristics of the semantic building block-based approach are discussed. The specific structure of this approach is confronted with the properties of traditional modeling languages. Subsequently, the semantic building block-based approach is evaluated in a laboratory experiment. The distributed modeling conflicts that can emerge in the building block-based approach are compared to the traditional approach. The paper closes with a short summary of the main results and an outlook to future research.

2. Variations in Intra- and Interorganizational Business Process Modeling

The comparability of business process models (BPM) is an important quality criterion. In order to improve the operational efficiency of a company or public administration, BPMs are employed to suit the organizational structure to the process flow. The business processes of an organization or a value network must be analyzed in whole to get a coherent overview on the interactions between individual factors. For instance, an analysis might include questions such as: Does the process comply with the quality regulations of the organization? Are there any substantial weaknesses in the process? Is a service in two different organizations performed by the same process? How much money could be saved through the introduction of a Document Management System? As the answer to these questions usually leads to high costs, the effort needed to conduct it must be reduced. This can be achieved by creating comparable BPMs, i.e. models with a relatively small number of variations.

A real world phenomenon can be represented through BPMs in many different ways. BPMs are constructed by using two different languages. The first one is the modeling language. Its meaning is at least semi-formally specified, which makes this part of a process model unambiguous. The other component of a BPM consists of a domain language. It is used to make statements about real world phenomena. In order to create a BPM, both languages must be applied together. Domain languages are owned by a linguistic community that decides on the meaning of its statements by shared conventions, which have been established implicitly by using the language. Because of the ambiguity of such natural languages it is possible to express the same meaning by different combinations of constructs and domain statements.

Variations in BPMs arise from both, differing perceptions of reality and from the process of explicating this perception. A variation is a semantic or syntactic deviation between different BPMs which refer to the same or a similar real world phenomenon. They can be due to two different reasons [20].

- *Variations due to varying mental representations:* The mental representations of two model creators are most likely not exactly the same. This means the model creators perceive or structure real world phenomena differently. Likewise, they can, consciously or unconsciously, consider deviating aspects of the phenomenon as relevant. This can lead to BPMs at diverse levels of abstraction. Likewise, in these models the sequence of activities can vary or the model elements can be annotated with a different number of details.

- *Variations due to the explication:* Even when the model creators share “the same” mental representation variations can arise. These variations result from a different explication of the mental representations. Domain and modeling languages offer certain degrees of freedom to express a given fact. Model creators can utilize this freedom in diverse ways. For example, different domain statements can be chosen to express a specific aspect of the mental representation. Similarly, a model creator may have the choice between multiple constructs to describe a given fact. Thus even with an equivalent mental representation, different BPMs with corresponding conflicts can emerge.

Deviations between models have been investigated empirically especially in the context of structural models. UML Class Diagrams have been analyzed in multiple modeling experiments [e.g., 9]. Other empirical studies have focused mainly on the advantages of specific constructs in comparison to alternative forms of representation, such as entity types and attributes [18], properties of relations [6], optional properties [5], or whole-part relations [19]. There are only a very few empirical studies that refer to variations in process models. Mendling et al. [12], for example, have analyzed the SAP Reference Model to identify errors and inconsistencies. Gruhn and Laue [8] have investigated the role of OR-connectors in EPC models. Beneath these empirical studies, conflicts between models have theoretically been discussed in the database schema matching and integration literature [e.g., 1], in publications about metamodeling [e.g., 16], and ontology engineering [7]. In this paper we draw upon Pfeiffer [15] who has derived a comprehensive theoretical analysis of the variations in the context of business process modeling.

This encompasses type variations, occurring when two model elements of different types have the same meaning, synonym variations, occurring when the labels of two model elements with the same meaning differ, homonym variations, occurring when two model elements with different meaning have the same label, abstraction conflicts, occurring when model elements in two different model have a deviating level of abstraction, control flow variations, occurring when the number of control flows of two corresponding model elements differ, annotation variations, occurring when corresponding model elements in two different models have a different number of annotated model elements, order variations, occurring when the order of two model elements is permuted between two BPM, and separation variations, occurring when a model element has no corresponding model element in the second model with the same, a more specific or a more general meaning.

3. Traditional and Semantic Building Block-based Process Modeling

The application of traditional business process modeling languages leads to business process models that are hard to compare. Every model created with a traditional language can include many of the variations described in the previous section of this paper. For instance, an EPC basically consist of events and functions, whose semantics are essentially defined by the domain statement the modeler assigns to it [10]. Only by applying various rules and modeling conventions, comparability between the BPMs can be achieved. The creation as well as the implementation of such regulations within a specific modeling project involves significant efforts.

By using a business process modeling language which belongs to the semantic building block-based approach, the comparability of the resulting business process models can be significantly improved. These semantic building block-based languages (SBBL) achieve this advantage by avoiding the conflicts that occur when traditional modeling languages are used [14]. The semantic building block-based approach guides the modeler through the modeling process and restricts him in his decisions. By decreasing the choices a model creator can make during the model construction, the comparability of the BPMs can be increased [15].

The main modeling construct of the language class SBBL is the so called process building block (PBB). PBBs limit the degree of freedom within the process of model creation. Unlike traditional business process modeling languages the SBBLs employ PBB as their most important modeling constructs. Every PBB represents one or more reoccurring activities from a particular domain [11]. The difference between a PBB and a modeling construct from a traditional language is that the PBB already incorporates a domain statement. Modelers do not create and assign a domain statement to a construct, they can only choose from a given set of PBBs and, thereby, from a given pool of statements. Thus, the PBB are semantically specified and have a defined level of abstraction [17]. If additional information is needed, the PBB can be further described by a predefined set of attributes. Concerning their semantics, the PBB are unambiguously and mutually exclusively defined. To specify the constructs of a SBBL, a domain ontology is used. Every PBB stands for a set of elements taken from this ontology. Hence, the meaning of a PBB is explicitly defined. With the aid of the ontology, it is possible to ensure that no element of a SBBL contains semantics already covered by another element of this language. Given a real world phenomenon, there exists only a single possibility to represent it in a SBBL-based language. In ideal, every construct would be derived from the domain ontology, but from a practical perspective it is often necessary to include at least some constructs from other languages. For instance, this could be a construct to split up and join the control flow. Imagine that the ontology element 'encash/receive a payment' has been incorporated into a SBBL as a PBB. Also its corresponding attribute, 'Information System', is taken from the domain ontology. This encompasses not only the attributes themselves, but also their possible values. In the given example, the attribute 'Information System' may have only three allowed values: 'Open Office', 'MS Office' and 'MS Money'. The available labels for the PBB, which specify the domain task more detailed, are defined in the same manner. For the PBB 'encash/receive a payment', the labels 'encash/receive a cash payment', 'encash/receive a credit card payment', and 'encash/receive a money transfer' might be allowed.

Languages from the class SBBL either avoid or at least decrease the previously described variations between BPMs. By using the semantic building block-based approach, some types of variations between models can be fully eliminated. Other variations can still occur, but their frequency can be significantly reduced. In the following the impact of the language class SBBL is discussed with regard to the five variation types considered:

- *Synonym variations:* Because of the fact that the constructs of languages from the class SBBL are derived from an ontology, they offer a controlled vocabulary to the modeler. Synonyms can be detected in the ontology, which makes it possible to eliminate them in advance of the model creation. Hence, as long as the modeler can only choose from the given vocabulary of a SBBL, no synonym variations can occur.

- *Type variations:* During the language construction, it is ensured that no semantically overlapping modeling constructs are included in the SBBL. If every PBB and every attribute of the language is semantically disjoint, it can be proven that no type variation can occur [14]. For every observable real world phenomenon only one single constructs exists which is able to represent it within the language. Therefore, every modeler who wants to describe the phenomenon is forced to use same construct.

- *Abstraction variations:* The type in combination with the label defines the semantics of a PBB. Because every PBB is semantically disjoint from the others, every modeler has to choose the same PBB to express a specific matter. Thus, the number of possible choices for the selection of domain statements and, thereby, also the number of abstraction variations is reduced. To completely avoid them, a specific level of the ontology has to be defined from which all the domain statements of a model have to originate.

- *Separation variations:* This type of variation cannot be entirely removed from models created with the language class SBBL. Nevertheless, it can be at least reduced because during model construction the modeler is guided by the ontology-based PBBs he can choose from. With the meaning of the PBBs in mind, he focuses on the semantics covered by them. Therefore, the models better fit to each other concerning the semantics they express.

- *Order variations:* Just like the separation variations, this type of variation cannot be completely avoided. In traditional modeling languages, it is hardly feasible to make any statements about the correct order of specific elements on the basis of their type. In contrast to that, the semantic building block-based approach allows to define heuristic order rules based upon the predefined semantics of the PBBs. For example, it is reasonable that the activity 'approve' always follows the activity 'perform a formal verification'.

The creation of languages from the class SBBL can only be accomplished successfully with a specific domain in mind. In order to be able to express every real world phenomenon by using a modeling language of this type, it is necessary to restrict the application to a specific domain. Otherwise, no appropriate ontology can be created due to the complexity of the real world. Hence, languages from the class SBBL are domain specific languages. A well documented example for such a language is the PICTURE-language, which is specifically designed for public administrations [2]. It consists of 24 PBB and over 50 attributes. The PBBs in PICTURE can only be connected in a sequential form. For an in-depth description of the language, we refer to [2]. A detailed analysis of the expressiveness can be found in [4].

4. Evaluation of Semantic Building Block-based Process Modeling

The hypothesis to evaluate is that modeling with a semantic building block-based language results in a smaller number of variation compared to traditional modeling languages. In order to do this, an empirical evaluation was conducted. EPC was chosen as an example of a traditional modeling language, PICTURE as an example for a domain specific one.

Within a laboratory experiment, twelve graduate students from the University of Muenster were asked to create an EPC and a PICTURE model independently from each other based on a given case description taken from the domain of public administrations. This case description was used to examine the variability between BPMs in both languages. This experimental setup simulates the process of distributed modeling and facilitates the validity of the analysis for two reasons. Firstly, all participants are modeling the same situation, which eliminates the case description as a source of variability. Secondly, every participant creates both an EPC and a PICTURE model. Thus, all variations resulting from a different understanding of the case description or from deviating opinions about the adequate degree of detail or abstraction influence the modeling process of both languages in the same way. The remaining variations can be fully explained by the process of explicating the mental representations of a participant in the form of a process model.
The analysis has been carried out in two steps:

- *Automated analysis:* In the first step, both EPC and PICTURE models are tested for similarity with an automated comparison algorithm [21]. This algorithm has been designed to quantify the similarity of the process flow as well as to detect and resolve problems resulting from the ambiguities of natural languages. The applicability of the algorithm has been demonstrated empirically by using the SAP Reference Model.

- *Manual analysis:* The second step is, in contrast to the first one, conducted manually to reconfirm the results from the automated comparison. In order to do this, the authors analyzed the BPMs from both groups to find and quantify variations from the types described above. If a high degree of similarity between the two models is found in the automatic analysis then a small number of variations can be expected in the manual analysis. The automated analysis of the models only provides a percentage value of similarity. Because the analysis is conducted manually in the second step, the nature of the variations can be explored in more detail.

4.1. Characteristics of the Automated Analysis

The comparison algorithm which has been used to determine the degree of similarity between the BPMs can be used for both PICTURE and EPC models in the same way. This is ensured by the fact that the models themselves are not used for the similarity calculation. Instead, the result is computed by using what is called a *causal footprint*. A casual footprint can be derived from the BPM. It is a directed graph whose vertices represent the various activities in the process. Vertices are connected by arcs whenever the corresponding activities of the vertices are always performed either before or after one another. In the first case, the arc is called a look-back-link, in the second case it is a look-ahead-link [22]. If, for example, there is an arc connecting the vertices A and B, this means that, depending on the type of the arc, activity A is either always performed before activity B or after it. In order to finally execute the comparison, the causal footprints of the models must be transformed into vectors. Their similarity is then determined by the deviation of their directions. For more details concerning the transformation, we refer to [21].

The comparison algorithm is able to identify ambiguities of natural languages within the labels of the model elements. To calculate the similarity of BPMs, common elements must be identified. Therefore, equivalent vertices need to be identified in order to compare two footprints. Natural languages allow expressing the same real-world concepts in different ways. This hampers the automatic identification of similar or equivalent activities. In order to deal with this problem, the comparison algorithm uses the lexical database WordNet, which allows to detect synonyms [13]. With the aid of this information, the semantic similarity of activities can be computed. Comparing the similarity score of an activity and of all elements connected to it, it is possible to map equivalent activities of different process models [21].

The comparison algorithm determines the similarity of process models regarding their content and their respective process flow. The causal footprint consists of both the vertices representing activities themselves and look-ahead as well as look-back-links, which stand for the procedural relations of the activities. Therefore, the comparison does not only consider the similarity regarding the content, but also takes the process flow into account.

4.2. Results of the Automated Analysis

12 BPMs from each group were compared pair-wise with each other. This resulted in a total of 66 comparisons for each group. Within the group of the EPC models, an average similarity of 0.54% has been measured. The maximum similarity was 4.02%, the minimum was 0%. This means that the comparison algorithm perceived the BPMs as being totally different. In contrast, the PICTURE

models achieved an average similarity of 43.75%. Some comparisons resulted in a value of 100%, which means that the models were identical. Other PICTURE models scored lower values as well. The minimum value was 13.99%.

Detailed results are described in Figure 1. In this diagram the average similarities of the individual BPMs compared to all other models are depicted. Figure 1-I presents the similarity values for the PICTURE and the EPC group on a single scale, Figure 1-II uses separate scales instead.

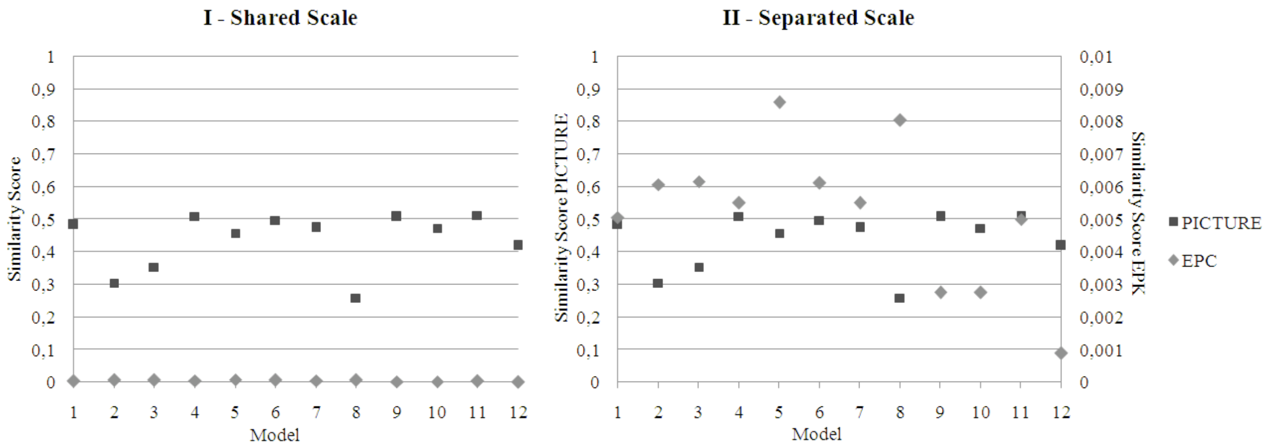

Figure 1: Average similarity degrees for PICTURE and EPC models

4.3. Characteristics of the Manual Analysis

Detailed statements about the nature and the degree of variability between BPMs can only be given manually. A framework, which classifies possible variations between process models into different categories, was introduced in Section 2. To identify these variations in process models, a semantic analysis of BPMs is necessary. Thus, a specific meaning needs to be assigned to every model element according to the modeler's intention. By this means, an ontology which describes the whole semantic of the case description has been developed. Thereafter, it was possible to assign statements of this ontology to every model element. The intended meaning had to be carefully explored by the authors. With the resulting assignments, the basis for the identification of variations was established.

When variations are identified they need to be counted in compliance with strict rules to assure a reasonable quantification of the variability. With the previously given definitions, variations can easily be identified. But the definition alone was not sufficient to generate a meaningful result. A set of rules for quantifying the identified variations had to be developed. They allowed for a consistent and uniform measurement. For example, rules were designed to prevent counting some variations multiple times. Different types of variations were not weighted, because there was no information about the extent to which an individual type of variation influences the comparability of BPMs.

With the given experimental setup, a reasonable measurement of homonym, control flow, and annotation variations was not possible. All models were created on the basis of the same case description. This makes the measurement of homonym variations difficult, because they occur when different concepts are expressed by the same terms. This usually happens in complex systems of different BPMs, however, not within a single case. Annotation conflicts were not measureable because no attributes were used within the EPC and only a fixed set of attributes within the PICTURE models. The PICTURE as well as the EPC language has strict rules concerning the incoming and outgoing control flows. In fact, only the AND, OR, and XOR operators from the EPC language allow for deviating numbers of control flows. Hence, no control flow variations were detectable during the analysis.

4.4. Results of the manual analysis

Within the variation analysis an average of 31.93 variations between EPC models were identified. An average of 12.59 of these variations were synonym variations, 5.95 were abstraction variations, 10.70 were separation variations, 2.15 were type variations, and 0.53 were order variations. The group of the PICTURE models scored an average value of 4.59 variations. It consists of 0.63 synonym variations, 0.83 abstraction variations, 1.77 separation variations, and 1.32 type variations. Order variations were not fount between PICTURE models. A comparison of the results can be found in Figure 2.

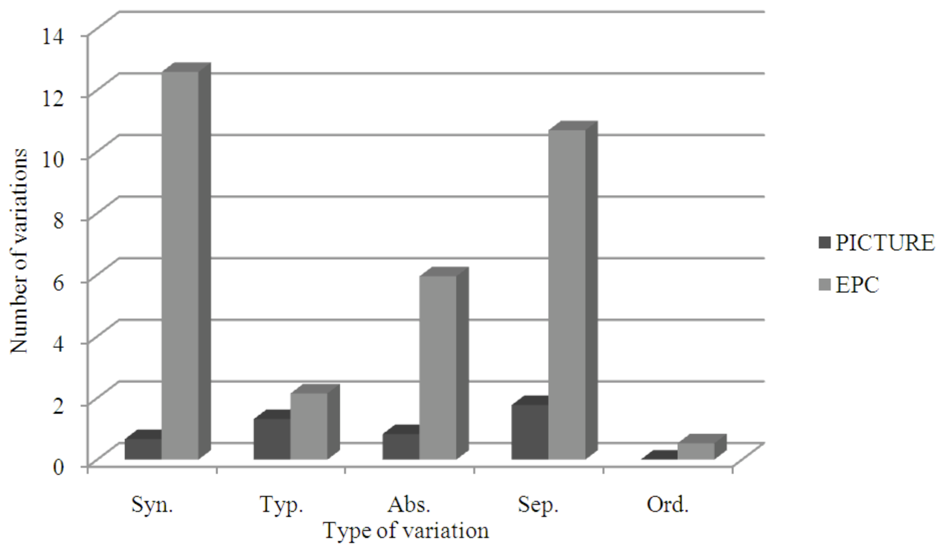

Figure 2: Numbers of variations for PICTURE and EPC according to the different variation types

4.5. Discussion of the results

The results of the automated similarity calculation are confirmed and further detailed by the manual analysis. While the automatic analysis can hardly find any commonalities between EPC models, it provides very good results for PICTURE models. In compliance with these results, the manual analysis shows a significantly higher number of variations of any kind for EPC models compared to PICTURE models. These results support assumption that the automated analysis is correct and further specify the results by categorizing the variations.

The semantics of BPMs that contain natural language elements cannot be captured automatically. The use of ontology-based labels for the PBBs in PICTURE actually results in a massive reduction of synonym variations compared to EPC. Although the algorithm used is build to detect synonyms, the low similarity degrees for EPC models imply that it fails to do so in most of the cases. The avoidance of many synonym variations by PICTURE in parallel with the high similarity degrees indicates that synonym variations cannot be resolved automatically.

The degree of detail and abstraction are fixed when using a SBBL-based modeling language. The limitation of the number of choices a modeler can make within the modeling project when he is using a SBBL in fact increased the comparability of the created models. A significant decrease of abstraction and separation variations in the manual analysis supports this conclusion.

It remains to be demonstrated that the expressiveness of a SBBL is sufficient. The increased comparability of models created with a SBBL leads to a decreasing expressiveness because of the predefined semantics of the PBBs. It is possible that the modeler is that limited in his decisions that he is not able to represent all relevant real world facts by using the PBBs. Hence, the creation of a SBBL is very time consuming and error prone. This analysis only shows that the language class SBBL produces models with a higher degree of comparability, but it does not take the expressiveness of the models into account.

5. Conclusion

The starting point of this paper was the insight that service-oriented thinking presupposes detailed knowledge about the business processes of an organization and its environment. We identified business process modeling as a way to explicate the relevant process knowledge. However, traditional business process modeling languages only provide little support for distributed modeling scenarios. The objective of this paper was to evaluate the semantic building block-based approach for the purpose of interorganizational business process modeling, in particular with respect to semantic variations within and between BPMs. In a laboratory experiment that simulated a distributed modeling project the potential advantages of the language class SBBL have been analyzed. Both an automated and a manual approach were chosen to compare the performance of the two languages EPC and PICTURE. The results of the analysis demonstrate that the type of the language has a strong influence on the number of variations in the resulting BPMs. PICTURE as an example of the language class SBBL considerably decreased the number of variations and, thereby, improved the quality of the corresponding BPMs.

However, the number of variations is only one component of the evaluation of the semantic building block-based approach. Furthermore, it is necessary to assess the efficiency and the effectiveness of the resulting languages. Efficiency means that a SBBL-based modeling language is able to acquire a specified number of processes at minimal cost. Effectiveness requires that a language of the class SBBL is expressiveness enough to describe the relevant phenomena of the domain at hand. In other words, effectiveness makes sure that the modeling language can indeed be successfully applied in a given domain. An empirical analysis of these two aspects is open to further research.

Acknowledgements

The work published in this paper is partly funded by the European Commission through the STREP PICTURE. It does not represent the view of European Commission or the PICTURE consortium and the authors are solely responsible for the paper's content.

References

[1] BATINI, C., LENZERINI, M., NAVATHE, S. B., A comparative analysis of methodologies for database schema integration. ACM Computing Surveys Vol. 18 (1986), pp. 323-364.
[2] BECKER, J., ALGERMISSEN, L., FALK, T., Prozessorientierte Verwaltungsmodernisierung. Springer, Berlin 2007.
[3] BECKER, J., ALGERMISSEN, L., FALK, T., PFEIFFER, D., FUCHS, P., Model based identification and measurement of reorganization potential in public administrations: the PICTURE-approach. In: Proceedings of the 10th Pacific Asia Conference on Information Systems (PACIS 2006), 2006, pp. 860-875.

[4] BECKER, J., ALGERMISSEN, L., PFEIFFER, D., RÄCKERS, M., Local, participative process modelling: the PICTURE-approach. In: Proceedings of the 1st International Workshop on Management of Business Processes in Government (BPMGOV 2007) at the 5th International Conference on Business Process Management (BPM 2007), 2007, pp. 33-48.
[5] BODART, F., PATEL, A., SIM, M., WEBER, R., Should optional properties be used in conceptual modelling: a theory and three empirical tests. Information Systems Research Vol. 12 (2001), pp. 384-405.
[6] BURTON-JONES, A., MESO, P., How good are these UML diagrams: an empirical test of the Wand and Weber good decomposition model. In: Proceedings of the 23rd International Conference on Information Systems (ICIS 2002), 2002, pp. 101-114.
[7] DAVIS, I., GREEN, P., MILTON, S., ROSEMANN, M., Using meta models for the comparison of ontologies. In: Proceedings of the 8th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2003) at the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003), 2003, pp. 1-10.
[8] GRUHN, V., LAUE, R., What business process modelers can learn from programmers. Science of Computer Programming Vol. 65 (2007), pp. 4-13.
[9] HADAR, I., SOFFER, P., Variations in conceptual modeling: classification and ontological analysis. Journal of the Association for Information Systems Vol. 7 (2006), pp. 568-592.
[10] KELLGER, G., NÜTTGENS, M., SCHEER, A.-W., Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Veröffentlichungen des Instituts für Wirtschaftsinformatik Vol. 89 (1992)
[11] LANG, K., GLUNDE, J., BODENDORF, F., A framework for reusable reference process building blocks. ACM SIGGROUP Bulletin Vol. 18 (1997), pp. 68-70.
[12] MENDLING, J., MOSER, M., NEUMANN, G., VERBEEK, H. M. W., VAN DONGEN, B. F., VAN DER AALST, W. M. P., Faulty EPCs in the SAP reference model. In: Proceedings of the 4th International Conference Business Process Management (BPM 2006), 2006, pp. 451-457.
[13] MILLER, G. A., WordNet: a lexical database for English. Communications of the ACM Vol. 38 (1995), pp. 39-41.
[14] PFEIFFER, D., Constructing comparable conceptual models with domain specific languages. In: Proceedings of the 15th European Conference on Information Systems (ECIS 2007), 2007, pp. 876-888.
[15] PFEIFFER, D., Semantic business process analysis - building block-based construction of automatically analyzable business process models Westfälische Wilhelms-Universität Münster 2008
[16] ROSEMANN, M., ZUR MÜHLEN, M., Evaluation of workflow management systems: a meta model approach. The Australian Journal of Information Systems Vol. 6 (1998), pp. 103-116.
[17] RUPPRECHT, C., FUNFFINGER, M., KNUBLAUCH, H., ROSE, T., Capture and dissemination of experience about the construction of engineering processes. In: Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000), 2000, pp. 294-308.
[18] SHANKS, G., NUREDINI, J., TOBIN, D., MOODY, D. L., WEBER, R., Representing things and properties in conceptual modelling: an empirical evaluation. In: Proceedings of the 11th European Conference on Information Systems (ECIS 2003), 2003, pp. 1-17.
[19] SHANKS, G., TANSLEY, E., NUREDINI, J., TOBIN, D., WEBER, R., Representing part-whole relationships in conceptual modeling: an empirical evaluation. In: Proceedings of the 23rd International Conference on Information Systems (ICIS 2002), 2002, pp. 89-100.
[20] SOFFER, P., HADAR, I., Applying ontology-based rules to conceptual modeling: a reflection on modeling decision making. European Journal of Information Systems Vol. 16 (2007), pp. 599-611.
[21] VAN DONGEN, B. F., DIJKMAN, R., MENDLING, J., Measuring similarity between business process models. In: Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008), 2008
[22] VAN DONGEN, B. F., MENDLING, J., VAN DER AALST, W. M. P., Structural Patterns for Soundness of Business Process Models. In: Proceedings of the Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06), 2006, pp. 116-128.