2009

Configuration of actors and roles in establishing ICT

Jörg Becker
University of Muenster, joerg.becker@ercis.uni-muenster.de

Dominic Breuker
University of Muenster, dominic.breuker@ercis.uni-muenster.de

Daniel Pfeiffer
University of Muenster, daniel.pfeiffer@ercis.uni-muenster.de

Michael Rackers
University of Muenster, michael.raeckers@ercis.uni-munester.de

CONSTRUCTING COMPARABLE BUSINESS PROCESS MODELS WITH DOMAIN SPECIFIC LANGUAGES – AN EMPIRICAL EVALUATION

Becker, Jörg, University of Münster, Leonardo Campus 3, 4149 Münster, Germany, joerg.becker@ercis.uni-muenster.de

Breuker, Dominic, University of Münster, Leonardo Campus 3, 4149 Münster, Germany, dominic.breuker@ercis.uni-muenster.de

Pfeiffer, Daniel, University of Münster, Leonardo Campus 3, 4149 Münster, Germany, daniel.pfeiffer@ercis.uni-muenster.de

Räckers, Michael, University of Münster, Leonardo Campus 3, 4149 Münster, Germany, michael.raeckers@ercis.uni-muenster.de

Abstract

The objective of this paper is to evaluate the semantic building block-based approach as a means for improving comparability in business process modelling. It is described whether and why the semantic building block-based approach reduces the variations in comparison to traditional modelling approaches. Our argumentation is grounded on the assumption that business process modelling projects in large organisations have to be conducted in a distributed manner. However, the goal of these projects is to integrate single models into a consistent process landscape. This allows the organisation to mine the processes for potential improvements. A lack of comparability could deteriorate the quality of the process landscape and the analysis performed on its basis. In a laboratory experiment the variations of distributed process modelling in the traditional and the building block-based approach have been compared. Results indicate that the semantic building block-based approach leads to considerably fewer variations between business process models and, thus, improves the comparability of them.

Keywords: Model Comparison, Domain Specific Languages, Business Process Modelling, PICTURE.

1 INTRODUCTION

Comparability is an important quality criterion for business process diagrams (BPD). In large business process modelling projects, as they are often executed in huge companies, the documentation of relevant domain knowledge has to be accomplished in a distributed manner. This is even more important if the modelling project crosses organizational boundaries. The challenge of such projects is not to represent the domain knowledge in single process models that are later on analyzed separately, but rather to combine all the models to a process landscape providing an overview over the companies business. The involvement of multiple actors in the modelling process causes the resulting models to differ in terms of vocabulary, level of detail and level of abstraction (Hadar & Soffer 2006). Creating an integrated description based on such models is thus hampered. Hence, an overall view on the organization is blurred by deviating ways of describing it. The possibilities to set up a holistic business process management based on the complete set of process models is therefore limited (Becker & Algermissen & Falk & Pfeiffer & Fuchs 2006).

The problem of variations between business process models can be addressed in two different ways. On the one hand, variations can be seen as given, which makes it necessary to resolve conflicts. This can either be done in an automated fashion (van Dongen & Dijkman & Mendling 2008), which, however, delivers only approximate results, or manually (Pfeiffer & Gehlert 2005), which involves significant efforts. The second way of dealing with variations is to avoid them during the construction of a BPD. This can be accomplished by constraining the choices a modeller can make when he creates a BPD. Domain specific modelling languages lower the ambiguities during the construction of conceptual models (Pfeiffer 2007). Thus, we claim that the application of a domain specific modelling language reduces the variations between BPDs compared to a traditional process modelling language. The aim of this paper is to deliver empirical evidence to support this assumption.

The paper proceeds as follows: In the next section the different variations within and between business process diagrams 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 NOTIONS OF VARIATION IN BUSINESS PROCESS MODELING

A real world phenomenon can be represented through BPDs in many different ways. BPDs are constructed by using two different languages. The first one is the modelling language. Its meaning can be formally specified, which makes this part of a process diagram unambiguous. The other component of a BPD consists of a domain language. It is used to make understandable statements about real world phenomena. In order to create a BPD, both languages must be applied together to create (human) readable process descriptions. 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 BPDs 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 BPDs which refer to the same or a similar real world phenomenon. They can be due to two different reasons (Soffer & Hadar 2007).

- *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 BPDs 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 BPDs 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 modelling experiments (Hadar & Soffer 2006, Lange & Chaudron 2006, Soffer & Hadar 2007). 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 (Shanks & Nuredini & Tobin & Moody & Weber 2003), properties of relations (Burton-Jones & Meso 2002, Burton-Jones & Weber 1999), optional properties (Bodart & Patel & Sim & Weber 2001), or whole-part relations (Shanks & Tansley & Nuredini & Tobin & Weber 2002). There are only a very few empirical studies that refer to variations in process diagrams. Mendling et al. (2006), for example, have analyzed the SAP Reference Model to identify errors and inconsistencies. Gruhn and Laue (2007) have investigated the role of OR-connectors in Event-driven Process Chains (EPC). Beneath these empirical studies, conflicts between models have theoretically been discussed in the database schema matching and integration literature (Kashyap & Sheth 1996, Parent & Spaccapietra 1998), in publications about metamodeling (e.g., Rosemann & zur Mühlen 1998), and ontology engineering (Davis & Green & Milton & Rosemann 2003). In this paper we draw upon Pfeiffer (Pfeiffer 2008) who has derived a comprehensive theoretical analysis of the variations in the context of business process modelling. The different variations are described in Table 1.

Variation name	Variation description
Type variation	Two model elements have the same meaning but a different construct (type) assigned.
Synonym variation	Two model elements have the same meaning but different labels.
Homonym variation	Two model elements have the same label but a different meaning.
Abstraction variation	Model elements in two different models have a deviating level of abstraction.
Control flow variation	The number of outgoing or incoming control flows of two corresponding model elements differs.
Annotation variation	A model element in the first model is annotated with a different number of model elements than a model element with a similar meaning in the second model.
Order variation	The order of the two model elements is permuted between two BPDs.
Separation variation	There is a model element that has no corresponding model element in the second model with the same, a more general, or a more specific meaning.

Table 1: Description of the variations between business process diagrams

In the following we will focus on type, synonym, abstraction, order and separation variations as part of the analysis. Homonym, control flow, and annotation variations are not taken into account. The reasons for this limitation are discussed in Section 4.

3 TRADITIONAL AND SEMANTIC BUILDING BLOCK-BASED PROCESS MODELLING

The application of traditional business process modelling languages leads to business process diagrams 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 modeller assigns to it (Kellger & Nüttgens & Scheer 1992). Only by applying various rules and modelling conventions, comparability between the BPDs can be achieved (Schütte & Rotthowe 1998). The creation as well as the implementation of such regulations within a specific modelling project involves significant efforts.

By using a business process modelling language which belongs to the semantic building block-based approach, the comparability of the resulting business process diagrams can be significantly improved. These semantic building block-based languages (SBBL) achieve this advantage by avoiding the conflicts that occur when traditional modelling languages are used (Becker & Pfeiffer & Räckers 2007d, Pfeiffer 2007). The semantic building block-based approach guides the modeller through the modelling process and restricts him in his decisions. By decreasing the choices a model creator can make during the model construction, the comparability of the BPDs can be increased (Pfeiffer 2008).

Analysing BPDs created with a SBBL on a semantic level can be more cost-effective compared to the analysis of BPDs created with traditional languages. While the effort of creating a SBBL is relatively high, various analysis methods can be predefined and executed automatically on models created with it (Pfeiffer 2008), which may compensate the creation effort. Using traditional languages to create BPDs requires business process experts to perform every analysis manually or semi-automatically. Such an approach does not scale.

The main modelling 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 modelling languages the SBBLs employ PBB as their most important modelling constructs. Every PBB represents one or more reoccurring activities from a particular domain (Baacke & Rohner & Winter 2007, Becker et al. 2007d, Lang & Glunde & Bodendorf 1997). The difference between a PBB and a modelling construct from a traditional language is that the PBB already incorporates a domain statement. Modellers 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 (Rupprecht & Funffinger & Knublauch & Rose 2000). 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. In Figure 1, for example, the ontology element 'encash/receive a payment' has been incorporated into a SBBL as a PBB. Also the corresponding attributes of the PBB are 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' has 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’ are allowed.

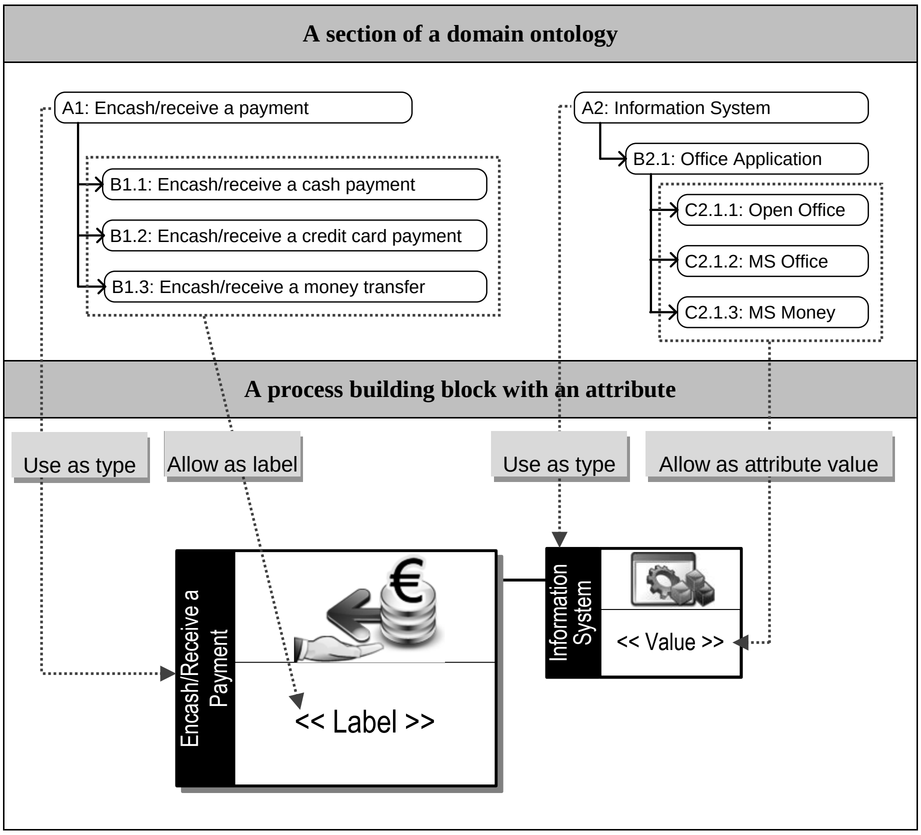

Figure 1: A Process Building Block and a Section from a Domain Ontology

Languages from the class SBBL either avoid or at least decrease the previously described variations between BPDs. By using the semantic building block-based approach, some types of variations between models can be fully eliminated. As the PBB do not offer multiple ways to express a specific fact, variations due to explication cannot occur. However, variations due to varying mental representations are still possible. But, because of the ontology incorporated into the language that is guiding the modeller though the modelling process, 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 (Pfeiffer 2007). 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 modelling 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 (Becker & Algermissen & Falk 2007a, Becker & Algermissen & Pfeiffer & Räckers 2007b, Becker et al. 2007d, Falk 2007). 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 (Becker et al. 2007a).

4 EVALUATION OF THE SEMANTIC BUILDING BLOCK-BASED APPROACH

4.1 Setting of laboratory experiment

The hypothesis to evaluate is that modelling with a semantic building block-based language results in a smaller number of variation compared to traditional modelling languages. In order to do this, an empirical evaluation was conducted. EPC was chosen as an example of a traditional modelling 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. This case description was used to examine the variability between BPDs in both languages. This experimental setup simulates the process of distributed modelling and facilitates the validity of the analysis for two reasons. Firstly, all participants are modelling 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 modelling 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 diagram. Because the order in which the models had to be created was not specified, the participants were able to correct mistakes they made during the creation of their first model if they notice a misunderstanding during the creation of their second one. Thus, in case the understanding of the case description changed over time due to a learning process, the validity of the results is not threatened.

As PICTURE is a domain specific language designed to be used in public administrations, the case description was taken from this domain. It is about handling an application for a resident parking permit. First, the application has to be checked according to various criteria. Depending on the results, it is either accepted or rejected. In the latter case a rejection letter is created and send to the applicant. If the application is accepted, the application data must be entered into a software system, the parking permit has to be created, an annual fee must be encashed and the parking permit has to be delivered to the applicant. Various methods of payment as well as different ways to deliver the parking permit are possible. The analysis has been carried out in two steps:

- *Automated analysis:* In the first step, both EPC and PICTURE models were tested for similarity with an automated comparison algorithm (van Dongen et al. 2008). 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 was, 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 BPDs 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 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. As the analysis is conducted manually in the second step, the nature of the variations can be explored in more detail.

4.2 Characteristics of the Automated Analysis

The comparison algorithm which has been used to determine the degree of similarity between the BPDs 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*. This is a construct that can be derived from the BPD. 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 (van Dongen & Mendling & van der Aalst 2006). 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 (van Dongen et al. 2008).

The comparison algorithm is able to identify ambiguities of natural languages within the labels of the model elements. To calculate the similarity of BPDs, 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 (Miller 1995). 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 diagrams (van Dongen et al. 2008).

The comparison algorithm determines the similarity of process diagrams 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.

4.3 Results of the Automated Analysis

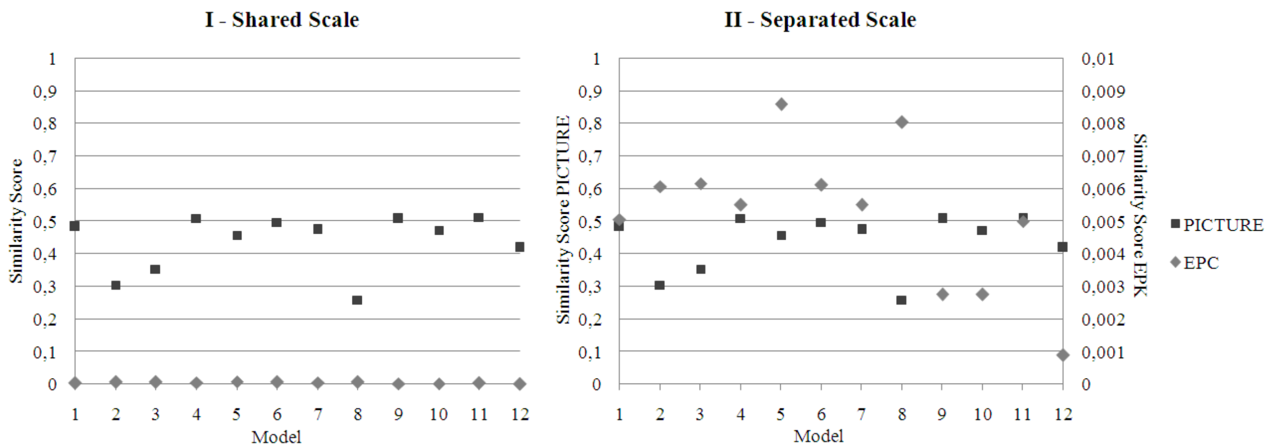

Figure 2: Average similarity degrees for PICTURE and EPC models

12 BPDs 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 BPDs 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 2. In this diagram the average similarities of the individual BPDs compared to all other models are depicted. Figure 2-I presents the similarity values for the PICTURE and the EPC group on a single scale, Figure 2-II uses separate scales instead.

4.4 Characteristics of the Manual Analysis

Detailed statements about the nature and the degree of variability between BPDs can only be given manually. A framework, which classifies possible variations between process diagrams into different categories, was introduced in Section 2. To identify these variations in process diagrams, a semantic analysis of BPDs is necessary. Thus, a specific meaning needs to be assigned to every model element according to the modeller'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 BPDs. To measure synonym variations, all words in the EPC models were stemmed and frequently occurring words like "a", "an" or "and" were not considered in the comparison. Taking variations resulting from such differences into account would distort the results, because transforming strings into the described form can be easily accomplished automatically (Porter 1980).

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 BPDs, 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.

4.5 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 combined results can be found in Figure 3.

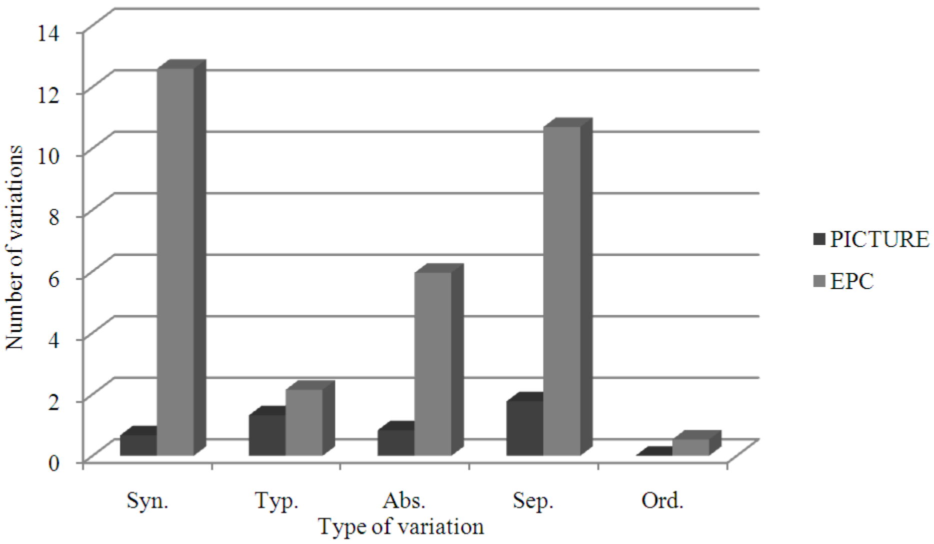

Figure 3: Numbers of variations for PICTURE and EPC according to the different variation types

4.6 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 BPDs 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 modelling language. The limitation of the number of choices a modeller can make within the modelling 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 modeller 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. Although there is a study focusing on this issue by evaluating PICTURE against workflow patterns (Becker & Algermissen & Pfeiffer & Räckers 2007c) and despite of the fact that PICTURE has been successfully used to create a total of more than 1,000 process diagrams in 12 german administrations until now (Bergener & Pfeiffer & Räckers 2009), a comprehensive empirical study is up to future research.

5 CONCLUSION

In the beginning of this paper we argued that BPD created in a distributed process modelling project exhibit a significant degree of variability which is a major obstacle to the creation of a sound overview over the organizations process landscape. In order to overcome this problem in BPD, we proposed to use languages from the SBBL class. Such languages reduce the variations between BPD by limiting the freedom of choice during model creation. To support our argumentation, an empirical evaluation of the PICTURE language, a well known example for a language coming from the class SBBL, was conducted. In a laboratory experiment that simulated a distributed modelling 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 BPDs. PICTURE considerably decreased the number of variations and, thereby, improved the quality of the corresponding BPDs.

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 modelling language is able to acquire a specified number of processes at minimal cost. The creation of a domain ontology, whose existence is a necessary precondition for a language from the SBBL class, requires a considerable amount of effort, which could outweigh quality improvements. Effectiveness requires that a language of the class SBBL is expressive enough to describe the relevant phenomena of the domain at hand. In other words, effectiveness makes sure that the modelling 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

Baacke, L., Rohner, P. and Winter, R. (2007). Aggregation of reference process building blocks to improve modeling in public administrations. In Electronic government: 6th international EGOV conference, proceedings of ongoing research, project contributions and workshops (Grönlund, A., Scholl, H. J. and Wimmer, M. A. Eds.), pp. 149-156, Trauner Druck, Linz.

Becker, J., Algermissen, L. and Falk, T. (2007a). Prozessorientierte Verwaltungsmodernisierung. Springer, Berlin.

Becker, J., Algermissen, L., Falk, T., Pfeiffer, D. and Fuchs, P. (2006). 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), pp. 860-875, Kuala Lumpur, Malaysia.

Becker, J., Algermissen, L., Pfeiffer, D. and Räckers, M. (2007b). Bausteinbasierte Modellierung von Prozesslandschaften mit der PICTURE-Methode am Beispiel der Universitätsverwaltung Münster. Wirtschaftsinformatik, 49 (4), pp. 267-279.

Becker, J., Algermissen, L., Pfeiffer, D. and Räckers, M. (2007c). 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) (Gray, J. Ed.), pp. 33-48, Brisbane, Australia.

Becker, J., Pfeiffer, D. and Räckers, M. (2007d). Domain specific process modelling in public administrations: the PICTURE-approach. In Proceedings of the 6th International Conference on Electronic Government (EGOV 2007) at the 18th International Conference on Database and Expert Systems Applications (DEXA 2007) (Wimmer, M. A., Scholl, H. J. and Grönlund, A. Eds.), pp. 68-79, Regensburg, Germany.

Bergener, P., Pfeiffer, D. and Räckers, M. (2009). How to inform the point of single contact? – A business process based approach. In Proceedings of the Proceedings of the 9. Internationale Tagung Wirtschaftsinformatik: Business Services: Konzepte, Technologien, Anwendungen, pp. 635-644, Vienna.

Bodart, F., Patel, A., Sim, M. and Weber, R. (2001). Should optional properties be used in conceptual modelling: a theory and three empirical tests. Information Systems Research, 12 (4), pp. 384-405.

Burton-Jones, A. and Meso, P. (2002). 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) (Applegate, L., Galliers, R. and DeGross, J. Eds.), pp. 101-114, Barcelona, Spain.

Burton-Jones, A. and Weber, R. (1999). Understanding relationships with attributes in entity-relationship diagrams. In Proceedings of the 20th International Conference on Information Systems (ICIS 1999) (De, P. and De Gross, J. I. Eds.), pp. 214-228, Charlotte, NC.

Davis, I., Green, P., Milton, S. and Rosemann, M. (2003). 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) (Siau, K., Krogstie, J. and Halpin, T. Eds.), pp. 1-10, Velden, Austria.

Falk, T. (2007). Prozessmodellierung in der öffentlichen Kommunalverwaltung: Konstruktion einer bausteinbasierten Prozessmodellierungstechnik. Dissertation. University of Westfälische Wilhelms-Universität Münster, Münster, Germany.

Gruhn, V. and Laue, R. (2007). What business process modelers can learn from programmers. Science of Computer Programming, 65 (1), pp. 4-13.

Hadar, I. and Soffer, P. (2006). Variations in conceptual modeling: classification and ontological analysis. Journal of the Association for Information Systems, 7 (8), pp. 568-592.

Kashyap, V. and Sheth, A. (1996). Semantic and schematic similarities between database objects: a context-based approach. The VLDB Journal, 5 (4), pp. 276-304.

Kellger, G., Nüttgens, M. and Scheer, A.-W. (1992). Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Veröffentlichungen des Instituts für Wirtschaftsinformatik, 89.
Lang, K., Glunde, J. and Bodendorf, F. (1997). A framework for reusable reference process building blocks. ACM SIGGROUP Bulletin, 18 (1), pp. 68-70.
Lange, C. F. J. and Chaudron, M. R. V. (2006). Effects of defects in UML models: an experimental investigation. In Proceedings of the 28th International Conference on Software Engineering (ICSE 2006) (Osterweil, L. J., Rombach, D. and Soffa, M. L. Eds.), pp. 401-411 Shanghai, China.
Mendling, J., Moser, M., Neumann, G., Verbeek, H. M. W., van Dongen, B. F. and van der Aalst, W. M. P. (2006). Faulty EPCs in the SAP reference model. In Proceedings of the 4th International Conference Business Process Management (BPM 2006) (Dustdar, S., Fiadeiro, J. L. and Sheth, A. Eds.), pp. 451-457, Vienna, Austria.
Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38 (11), pp. 39-41.
Parent, C. and Spaccapietra, S. (1998). Issues and approaches of database integration. Communications of the ACM, 41 (5), pp. 166-178.
Pfeiffer, D. (2007). Constructing comparable conceptual models with domain specific languages. In Proceedings of the 15th European Conference on Information Systems (ECIS 2007) (Österle, H., Schelp, J. and Winter, R. Eds.), pp. 876-888, St. Gallen, Switzerland.
Pfeiffer, D. (2008). Semantic business process analysis - building block-based construction of automatically analyzable business process models PhD thesis. University of Westfälische Wilhelms-Universität Münster, Münster.
Pfeiffer, D. and Gehlert, A. (2005). A framework for comparing conceptual models. In Proceedings of the 1st International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2005) at the 24th International Conference on Conceptual Modeling (ER 2005) (Desel, J. and Frank, U. Eds.), pp. 108-122, Klagenfurt, Austria.
Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14 (3), pp. 130-137.
Rosemann, M. and zur Mühlen, M. (1998). Evaluation of workflow management systems: a meta model approach. The Australian Journal of Information Systems, 6 (1), pp. 103-116.
Rupprecht, C., Funffinger, M., Knublauch, H. and Rose, T. (2000). 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) (Wangler, B. and Bergman, L. Eds.), pp. 294-308, Stockholm, Sweden.
Schütte, R. and Rotthowe, T. (1998). The guidelines of modeling: an approach to enhance the quality in information models. In Proceedings of the 17th International Conference on Conceptual Modeling (ER 1998) (Ling, T. W., Ram, S. and Lee, M. L. Eds.), pp. 240-254 Singapore.
Shanks, G., Nuredini, J., Tobin, D., Moody, D. L. and Weber, R. (2003). Representing things and properties in conceptual modelling: an empirical evaluation. In Proceedings of the 11th European Conference on Information Systems (ECIS 2003) (Ciborra, C. U., Mercurio, R., de Marco, M., Martinez, M. and Carignani, A. Eds.), pp. 1-17, Naples, Italy.
Shanks, G., Tansley, E., Nuredini, J., Tobin, D. and Weber, R. (2002). Representing part-whole relationships in conceptual modeling: an empirical evaluation. In Proceedings of the 23rd International Conference on Information Systems (ICIS 2002) (Applegate, L., Galliers, R. and DeGross, J. Eds.), pp. 89-100, Barcelona, Spain.
Soffer, P. and Hadar, I. (2007). Applying ontology-based rules to conceptual modeling: a reflection on modeling decision making. European Journal of Information Systems, 16 (5), pp. 599-611.
van Dongen, B. F., Dijkman, R. and Mendling, J. (2008). Measuring similarity between business process models. In Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008), Montpellier, France.
van Dongen, B. F., Mendling, J. and van der Aalst, W. M. P. (2006). 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) (IEEE Ed.), pp. 116-128.