2009

Mitigating response distortion in IS ethics research

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

Patrick Delfmann
Universität Münster, delfmann@ercis.de

Sebastian Herwig
University of Muenster, herwig@ercis.uni-muenster.de

Lukasz Lis
University of Muenster, lis@ercis.uni-muenster.de

Armin Stein
University of Muenster, stein@ercis.uni-muenster.de

TOWARDS INCREASED COMPARABILITY OF CONCEPTUAL MODELS – ENFORCING NAMING CONVENTIONS THROUGH DOMAIN THESAURI AND LINGUISTIC GRAMMARS

Becker, Jörg, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany, becker@ercis.uni-muenster.de

Delfmann, Patrick, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany, delfmann@ercis.uni-muenster.de

Herwig, Sebastian, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany, herwig@ercis.uni-muenster.de

Lis, Łukasz, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany, lis@ercis.uni-muenster.de

Stein, Armin, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany, stein@ercis.uni-muenster.de

Abstract

Distributed construction of conceptual models may lead to a set of problems when these models are to be compared or integrated. Different kinds of comparison conflicts are known (e.g. naming conflicts or structural conflicts), the resolution of which is subject of different approaches. However, the ex-post resolution of naming conflicts raises subsequent problems that origin from semantic diversities of namings – even if they are syntactically the same. Therefore, we propose an approach that allows for avoiding naming conflicts in conceptual models already during modelling. This way, the ex-post resolution of naming conflicts becomes obsolete. In order to realise this approach we combine domain thesauri as lexical conventions for the use of terms, and linguistic grammars as conventions for valid phrase structures. The approach is generic in order to make it reusable for any conceptual modelling language.

Keywords: Conceptual Modelling, Model Comparison, Domain Thesaurus, Computational Linguistics

1 INTRODUCTION

Since IS modelling projects are often large-scaled, the required models are increasingly constructed in a distributed way in order to boost the efficiency of modelling (vom Brocke & Thomas 2006). I.e., different modellers participate in the modelling process developing different model parts – maybe at different places and at a different time. Furthermore, domain experts are increasingly integrated in the modelling process to comply with domain requirements as much as possible – an approach that is commonly known as *End-user Development* (Wulf & Jarke 2004; Fischer et al. 2004). Distributed modelling becomes more and more main-stream. This applies not only intra-corporately but also in the form of so-called modelling communities, e.g. the *Open Model Initiative* (Frank & Strecker 2007).

Empirical studies show that model sections developed in the course of distributed modelling can vary heavily concerning terms and abstraction level (Hadar & Soffer 2006). Therefore, the comparability of the regarded models cannot be guaranteed. Comparison conflicts may occur, which are commonly divided into *naming conflicts* and *structural conflicts* (Pfeiffer 2008). Consequently, the integration of the different model sections into one total model representing the original modelling issue may be extremely laborious. Usually, model parts are consolidated manually and with attendance of all modellers in order to reach a consensus. Sometimes, external consultants are involved additionally (Phalp & Shepperd 2000; Vergidis & Tiwari & Majeed 2008). Thus, the original sub-goal of distributed modelling – i.e. the increased modelling efficiency – may be missed.

Summarising, a considerable problem of distributed modelling is the insufficient comparability of conceptual models. In order to solve this problem, approaches are required that are able to assure comparability. In literature, there exist many contributions that propose approaches for resolving comparison conflicts subsequent to modelling (cf. Section 2).

The goal of this article is to introduce an approach that ensures the comparability of conceptual models by avoiding potential comparison conflicts already *during* modelling. This way, we prevent semantic problems that result from the ex-post resolution of conflicts. This article focuses on *naming conflicts*. We define naming conventions for elements of modelling languages and ensure their compliance by an automated, methodical guiding during modelling. The conventions are set up using the *domain terms* and the *phrase structures* that are defined as valid in the regarded modelling context. As a formal specification basis, we use thesauri that provide term conventions not only for nouns but also for verbs and adjectives. These thesauri also include descriptions of a term's meaning. In order to provide conventions for phrase structures, we use the *Head-driven Phrase Structure Grammar (HPSG)* (Pollard & Sag 1994), a formal grammar for the specification of natural language syntaxes. During modelling, model element names are validated simultaneously against both the term and phrase structure conventions. Our approach is generic so that it can be applied to any conceptual modelling language. Our approach is suitable for modelling situations, where it is possible to provide all involved stakeholders with the necessary information about the modelling conventions, i.e. modelling projects that are determined regarding organization and/or business domain. These modelling situations usually apply in companies, corporate groups or modelling communities.

The paper is structured as follows. First, we analyse related work in Section 2 and discuss the research gap that led to the development of the approach presented in this paper. Furthermore, we outline the research methodology used. In Section 3, we introduce a conceptual framework for the specification and enforcement of naming conventions. The feasibility of our approach is shown exemplarily with a demonstrator software in Section 4. Furthermore, an explorative analysis of real-world process modelling projects shows the potential of the approach reducing ambiguities in process models. We conclude the paper in Section 5 and motivate further research addressing e.g. the conflict classes not yet covered by this approach.

2 FOUNDATIONS

2.1 Related Work

Early approaches of the 1980s and 1990s discussing the resolution of naming conflicts address the integration of company *databases* and use the underlying *schemas* as a starting point (Batini & Lenzerini 1984; Batini & Lenzerini & Navathe 1986; Bhargava & Kimbrough & Krishnan 1991; Lawrence & Barker 2001). Hence, these approaches focus on data modelling languages, mostly dialects of the Entity-Relationship Model (ERM) (Chen 1976). Names of schema elements are compared and, this way, similarities are revealed. The authors state that such a semantic comparison can exclusively happen manually. Moreover, only single nouns are considered as names. In contrast, in common modelling languages and especially in process modelling languages, names are used that consist of sentence fragments containing terms of any word class. This aspect is not regarded by schema integration approaches. Thus, they are only suitable for the mentioned data modelling languages.

Other approaches make use of *ontologies* (Gruber 1993; Guarino 1998) in order to address the problem of semantic comparison of names. These approaches act under the assumption that there exists a "generally accepted" ontology describing a certain modelling domain. It is assumed that all considered models of this domain comply with its ontology, i.e. that modellers had a thorough knowledge of the ontology. E.g., Höfferer (2007) connects domain ontologies to the terms that are used as element names in conceptual models. This way, he establishes relationships between elements of different models that are to be compared and identifies similarities. In addition to ontologies, Ehrig, Koschmider, and Oberweis (2007) define combined similarity measures that consist of syntactic and semantic parts. These serve as a basis for the decision whether or not the model elements compared are equivalent. Consequently, it is argued that if identical terms – or those that are defined as synonymous within the ontology – are used in different models and by different modellers, they can be considered as semantically identical as well (Koschmider & Oberweis 2005; Sabetzadeh et al. 2007).

However, solely the circumstance that two modellers act in the same business domain does not guarantee at all that they share the same or an equivalent understanding of business terms. If a "generally accepted" ontology is available, it is suitable for model comparison if and only if it is explicated and can be accessed by all involved modellers already *during* the modelling process. Furthermore, in order to ensure comparability of the models, modellers have to comply *strictly* with the ontology. Most approaches make the implicit assumption that these preconditions are already given rather than addressing a methodical support. So far, there exist only few approaches that address such a support. Greco et al. (2004) propose adopting terms from existing ontologies for process models manually. However, due to manual adoption, correctness cannot be assured. Born, Dörr, and Weber (2007) propose semi-automated adoption of model element names. However, they restrict their approach to BPMN models (White & Miers 2008).

In order to avoid the problems of ontology based approaches, Pfeiffer (2008) suggests integrating domain semantics into modelling languages, i.e. the names of model elements are preset. Thus, he proposes *domain-oriented building block-based process modelling languages*. With the PICTURE modelling approach, he concretises them for the domain of public administration. Since PICTURE consists exclusively of 26 process building blocks with defined semantics, naming conflicts are a priori impossible. As the author states himself, this language class is restricted to mainly linear business processes.

Only few approaches, mainly from the German speaking area, suggest standardised phrases for model element names in order to increase the clarity of process models. The phrase standards are summed up as *modelling conventions*. E.g., Rosemann (1996) and Kugeler (2000) propose particular phrase structure guidelines for names of process activities (e.g. <*verb, imperative*> + <*noun*>; in particular e.g. "check invoice"). Moreover, the authors propose so-called *Technical Term Models* (Rosemann 2003) that have to be designed previously to process modelling and that specify the terms to be used within the phrases. Therefore, the scope of Technical Term Models is restricted to nouns. Similar approaches provided by Koschmider and Oberweis (2005) and Sabetzadeh et al. (2007) propose prepositioning generally accepted vocabulary. Further approaches recommend connecting names of model elements to online dictionaries in order to establish semantic relationships of terms (Rizopolous & McBrien 2005; Bögl & Kobler & Schrefl 2008). These online dictionaries consist of extensive collections of English nouns, verbs, and adjectives as well as their semantic relationships. Actually, the proposed approaches are promising regarding increased comparability of conceptual models since all of them aim at standardising names for model elements *prior to modelling*. However, up to now, a methodical realisation is missing.

To sum up, we identify the following need for development towards avoiding naming conflicts in comparing conceptual models: Up to now, methodical support for (1) the formal specification of naming conventions for all word classes and (2) the formal specification of phrase structure conventions is missing. Furthermore, there exists neither methodical support for (3) the integration of conventions in modelling languages nor (4) for guiding modellers in order to comply with the conventions.

In order to realise such a methodical support, we propose an approach that consists of (1) a formalism to specify thesauri covering nouns, verbs, and adjectives, (2) a grammar to specify phrase structures that can hold terms specified as valid within the term models, (3) a framework to integrate these naming conventions with modelling languages, and (4) a procedure to guide modellers automatically in complying with the conventions.

2.2 Research Methodology

The research methodology followed here complies with the Design Science approach (Hevner et al. 2004) that deals with the construction of scientific artefacts like methods, languages, models, or implementations. Following the Design Science approach, it is necessary to assure that the research addresses a *relevant problem*. This has to be proven. Furthermore, the artefacts to be constructed have to represent an *innovative contribution* to the existing knowledge base within the actual research discipline. I.e., similar or identical solutions must not be already available. Subsequent to the *construction* of the *artefacts*, these have to be *evaluated* in order to prove their fulfilment of the research goals.

In this contribution the *scientific artefact* is the modelling approach outlined in Section 1. This artefact aims at solving the *relevant problem* of the lacking comparability of conceptual models that are developed in a distributed way (cf. Section 1). Related work does not provide appropriate solutions up to now (cf. Section 2). Hence, the approach presented here (cf. Section 3) makes an *innovative contribution* to the existing knowledge base. In order to *evaluate* the approach, we have implemented a demonstrator software that shows the general applicability of the approach. Furthermore, we have conducted an explorative study that shows the potential of the approach (cf. Section 4). Additional evaluations concerning efficiency and acceptance issues will be performed in future studies (cf. Section 5).

3 A FRAMEWORK FOR THE SPECIFICATION AND ENFORCEMENT OF NAMING CONVENTIONS

3.1 Procedure Model

In order to provide a framework for naming conventions, we propose the usage of a domain language that is used for naming model elements in a certain modelling context (i.e. a specific modelling domain, project, or company). This domain language is a subset of the respective natural language (here: English) used in the modelling context. The domain language consists of a set of valid domain terms that are allowed to be used for model element names exclusively. The set of domain terms is a subset of all terms available in the respective natural language. Furthermore, every natural language has a certain syntax that determines the set of grammatically correct phrases. In our framework, we restrict the syntax of the respective natural language as well. I.e., the possibilities to construct sentences for model element names are limited. In summary, we restrict the *grammar of a natural language* in order to provide a formal basis for naming model elements (cf. Figure 1).

Natural language grammars are usually defined by a formalism that consists of a *lexicon* and a *syntax specification* (Mitkov 2003). This grammar is complemented with *naming conventions*, which again consist of term and phrase structure conventions. Term conventions are specified by a *domain thesaurus* containing domain terms with a precise specification of their synonym, homonym, and word formation relationships as well as a textual description of their meaning. It is connected to the natural language's lexicon. Moreover, valid phrase structures are specified by *phrase structure conventions*. Hence, the natural language is customised for the needs of a specific modelling context. This allows for subsequent validation of the model element names and the enforcement of naming conventions. A conceptual overview of the naming conventions' specification is given in Section 3.2.

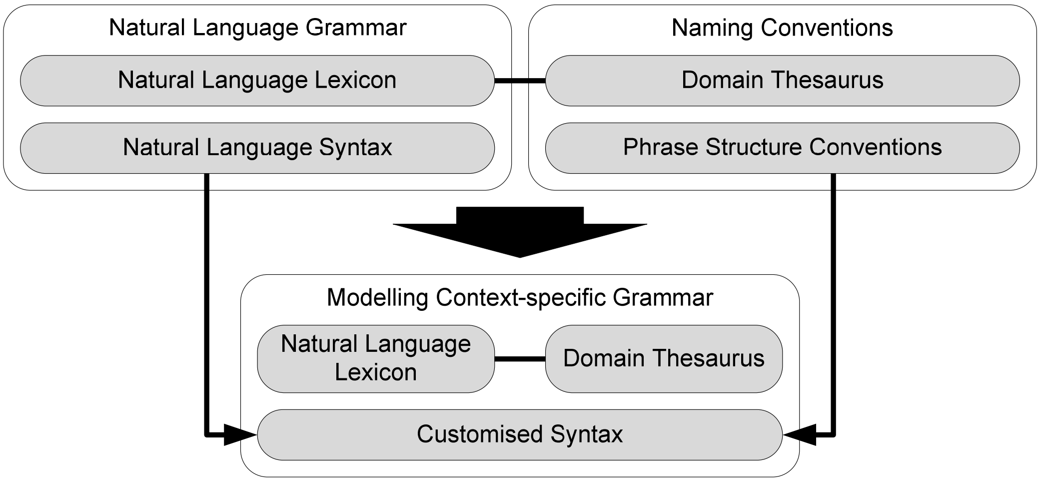

Figure 1. Customising the Natural Language Grammar with Naming Conventions

The domain thesaurus can be created from scratch, or by reusing possibly existing thesauri or glossaries. Contrary to most common domain ontologies, which consist of nouns (e.g. "invoice") or atomic domain concepts (e.g. "check invoice"), the thesaurus used here includes single nouns, verbs, and adjectives that are interrelated. Other word classes are generally domain independent. Thus, as they are already included in the lexicon, they do not need to be explicitly specified in the thesaurus. The terms in the thesaurus are linked to their synonyms, homonyms, and linguistic derivation in the lexicon. This additional term-related information can be obtained from linguistic services, which already exist for different natural languages. *WordNet* (http://wordnet.princeton.edu) is such a lexicon service for the English language providing an online interface. Therefore, in case of a later violation of the naming conventions by the modeller, synonymous valid terms can be automatically identified and recommended. The terms specified should be provided with short textual semantic descriptions, allowing modellers to look up the exact meaning of a term. The thesaurus should not be changed during a modelling project in order not to violate the consistency of application. When online services are used, the option of creating a local copy should be analysed.

Before starting a modelling project, the naming conventions have to be specified once for every modelling context. However, already existing conventions can be reused. Naming conventions are modelling language specific. For example, functions in Event-driven Process Chains (EPC) are labelled with *activities* (e.g. <*verb, imperative*> + <*noun*>; in particular e.g. "check invoice") and events are labelled with *states* (e.g. <*noun*> + <*verb, past participle*>; in particular e.g. "invoice checked") (Scheer 2000). For each model element type at least one phrase structure convention has to be defined. For the sake of applicability, the conventions should be specified in a manner, which is compatible with the formalism of the natural language grammar.

The conventions should be defined by a project team consisting of domain experts and modelling experts. I.e. the stakeholders responsible for the conventions should have thorough knowledge of the actual modelling context in order to reach a consensus. Most commonly, the thesaurus part of the conventions already exists in terms of corporate or domain-specific glossaries (e.g. http://www.automotivethesaurus.com; http://www.tradeport.org/library; http://www.logisticsworld.com/logistics/glossary.htm), which should be reused and adapted depending on the modelling situation (cf. Section 2.1).

During modelling, the model element names entered are verified simultaneously against the specified modelling context-specific grammar. On the one hand, the structure of an entered model element name is validated against the customised syntax specification. On the other hand, it is checked whether the used terms are allowed. Nouns, verbs, and adjectives are validated against the thesaurus. Other word classes are validated against the natural language lexicon.

In case of a positive validation, the entered model element name is declared as valid against the modelling context-specific grammar. In case of a violation of one or both criteria, alternative valid phrase structures and/or terms are suggested based on the user input. The modellers themselves have to decide, which of the recommendations fits their particular needs. By looking up the semantic descriptions of the terms, modellers can choose the appropriate one. Alternatively, they can choose a valid structure as a pattern and fill in the gaps with valid terms. However, the modeller should have the chance to propose a new term, which is then accepted temporarily. Hence, the modeller can continue without being distracted from his modelling session. It is then up to the modelling project expert team whether they accept the term or not. If the term is accepted, it is added to the thesaurus. Otherwise, the modeller is informed to revise the model element. Hereby, we ensure that equal model element names represent equal semantics, which is a precondition for comparability.

3.2 Conceptual Specification

In the following, we provide a conceptual framework for the specification and the enforcement of naming conventions using Entity-Relationship Models in (min,max)-notation (ISO 1982) (cf. Figure 2). *Phrase structure conventions (PSC)* are defined depending on distinct *element types* of conceptual modelling languages (e.g., *activities* in process models are named differently to *events*). Phrase structure conventions consist of *phrase types* or *word types.* A phrase type specifies the structure of a phrase, which can be used as a model element name. Therefore, a phrase type can be composed recursively of further phrase types or word types. Representing atomic elements of a phrase type, word types are acting as placeholders for particular words. An example of a word type is <*noun, singular*>, an example of a phrase type is <*noun, singular*> + <*verb, infinitive*>. The composition of phrase types is specified by the *phrase type structure*. At this, the allocation of sub-phrase types or word types to a phrase type and their *position* in the superordinated phrase type are defined. A word type consists of a distinct *word class* (*noun, verb, adjective, adverb, article, pronoun, preposition, conjunction,* or *numeral*) – and its *inflection*. Inflections are specialised as *case, number, tense, gender, mood, person,* and *comparative*, which are usually combined. E.g., a particular combined inflection is <*3rd person, plural*>. In respect to specific word classes, not every inflection is applicable. Based on the recursive composition of phrase types, the specification of arbitrary phrase structure conventions is possible.

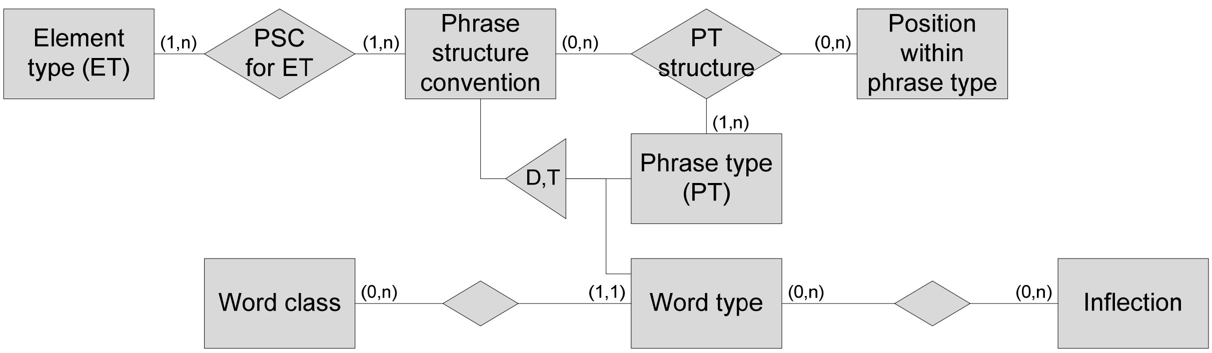

Figure 2. Specification of Phrase Structure Conventions on Type Layer

Phrase structure conventions restrict the English syntax and thus limit modellers in their freedom of naming of model elements. In order to facilitate the synchronisation between the syntax of the natural language and the applied phrase structure conventions, compatible formalisms for both syntax specifications are necessary. Hence, it should be possible to verify phrase structure conventions against the underlying natural language and to signalise potential conflicts directly during the specification process. For this purpose, we establish the connection to an appropriate linguistic method in Section 3.3.

Independent from their corresponding word class, particular uninflected words are called *lexemes* (e.g. the verb “check”). Inflected words are called *word forms* (e.g. past participle “checked” of the lexeme “check”). Word forms are assigned to the corresponding word classes and inflections, i.e. their word types. Thus, word forms represent lexemes of a particular word type (cf. Figure 3).

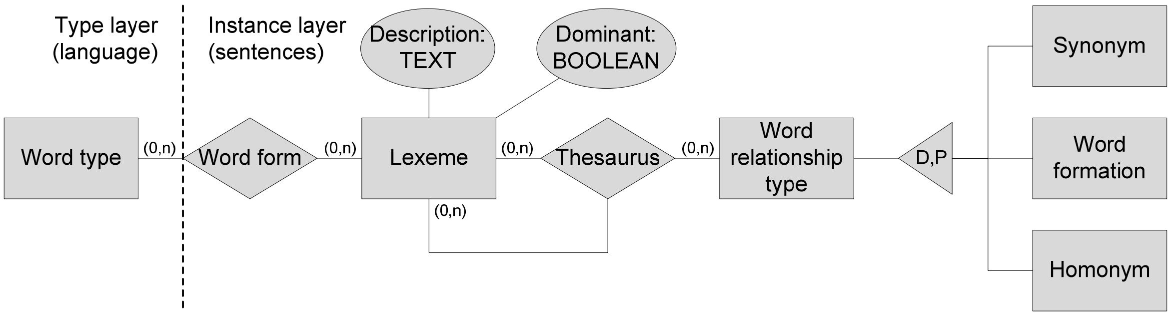

Figure 3. Specification of Term Conventions on Instance Layer

In order to specify the domain *thesaurus*, allowed words are stored in the form of lexemes that are related by different *word relationship types*. They are specialised as *homonym*, *synonym*, and *word formation* relations. *Word formation* means that a lexeme originates from (an)other one(s) (e.g. the noun “control” originates from the verb “to control”). In case of synonym relations, one of the involved lexemes is marked as *dominant* to state that it is the valid one for the particular modelling context. Homonym relations are necessary in order to distinguish lexemes that consist of the same string but have a different meaning and to prevent errors during modelling. Word formation relations are used to search for appropriate alternatives when a modeller has used invalid terms and phrase structures. E.g., if the phrase “order clearance” violates the conventions, the alternative phrase “clear order” can be found via the word formation relation of “to clear” and “clearance”. Based on the word relationship types, lexical services (cf. Section 3.1) are connected to the domain thesaurus. To specify the actual meaning of a lexeme, a textual semantic *description* is added at least to each dominant lexeme. This way, modellers are enabled to check if the lexeme they have used actually fits the modelling issue.

3.3 Specification of Linguistic Restrictions

We make use of formalisms for the syntax specification of natural languages both for assuring correctness of specified phrase structure conventions and for identifying the structure and words of model element names. The latter is necessary in order to detect convention violations during modelling and to suppose alternative valid phrase structures and terms.

Such formalisms are mainly established in the field of computational linguistics. A well-known and widely accepted class of such formalisms are *unification grammars* (Mitkov 2003). All formalisms of this class have in common that syntax rules are specified in relation to phrase structures and their components. In unification grammars, *feature structures* are used to formalise syntax rules of a given language. A feature structure is a set of features and their values, which are structured hierarchically. It describes syntactical characteristics of a phrase structure. These phrase structures declare for a given natural language, which word types or sub phrase types can occur at which position within a phrase structure. An exemplary phrase structure consists of a *phrase type*, which is composed of two *word types* (cf. Figure 4). The word types are characterised by the features *word class* and *inflection.* Depending on the word class, the feature *inflection* is composed of further features. Possible features are *number* and *case* for the word class *noun*, and *tense* for the word class *verb*. The position of word types and phrase types within the phrase structure is defined by the given order. In the class of unification grammars, the *Head-driven Phrase Structure Grammar (HPSG)* (Pollard & Sag 1994) is a widely established approach. Several formal specifications of natural languages based on HPSG are already available (cf. http://www.delph-in.net). A formal specification of the English syntax with HPSG was developed at the *CSLI LinGo Lab* of the University of Stanford (Copestake & Flickinger 2000).

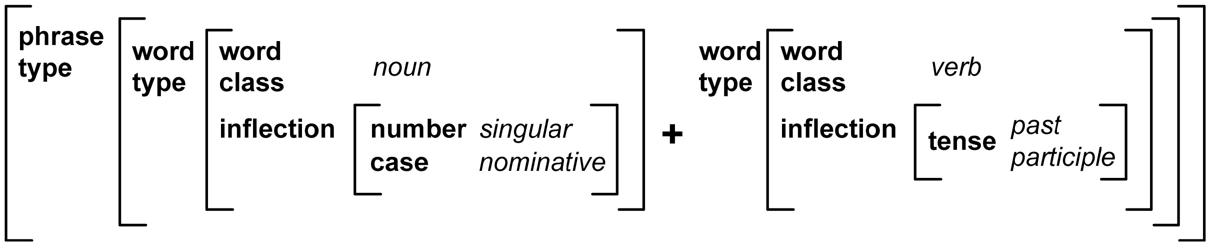

Figure 4. Exemplary Formalisation of a Phrase Structure

In our approach, we use this HPSG-based syntax of the English language. Hence, the set of possible phrase structure conventions for model elements is a subset of the syntax of the English language based on HPSG. In the modelling process, the model element names have to comply with the conventions. Moreover, the used words have to comply with the domain thesaurus. In order to assure this, the model element names are parsed against the used words, their inflection, and their phrase structure. Corresponding parsing methods are available as a part of the *LinGo English Resource Grammar* (Copestake & Flickinger 2000). The parser is able to detect the lexeme, its word class and its inflection for each word using a lexicon. In our approach, the lexicon consists of the domain thesaurus that is connected to common lexical services. If the terms used within model element names do not comply with the conventions, alternative valid lexemes are searched in the domain thesaurus via the defined word relationships and are proposed in the appropriate inflection form for proper use. If a phrase structure is violated in turn, alternative valid phrase structures are proposed that contain the valid terms.

4 EVALUATION

4.1 Modelling Tool Support

To validate the applicability of our approach, we developed a modelling prototype. The way of navigating through the software and its handling corresponds with the procedures motivated in Section 3. Please note that the screenshots were adapted to English, since the prototype solely exists in German.

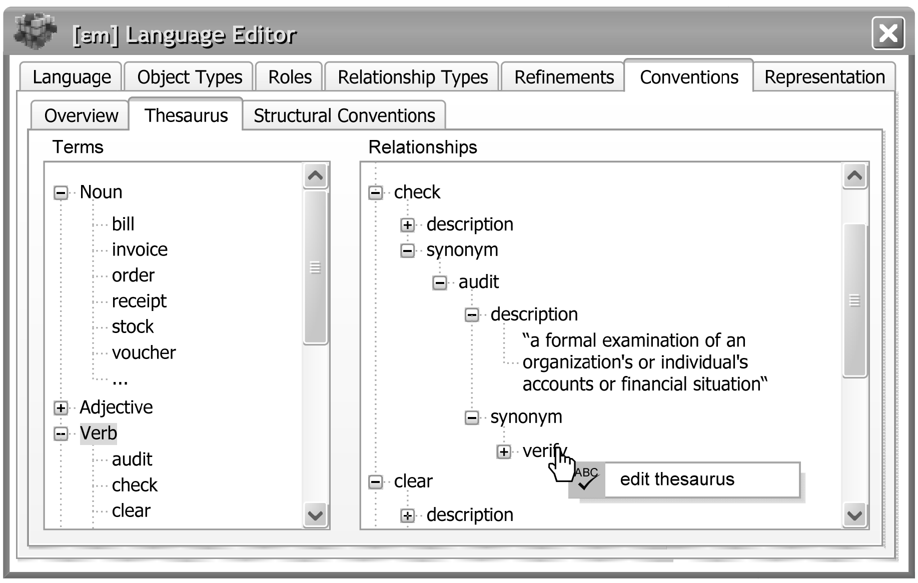

Figure 5. Specification of Term Conventions for Verbs

As a preliminary but unique step, the team responsible for constituting the domain grammar has to define the terms, which are allowed for the modelling context. Subsequently, the phrase structure conventions have to be specified. If the actual modelling context represents a domain which has been processed before, the existing set of terms and rules can be reused or adapted to the current requirements. Figure 5 shows an exemplary definition of domain verbs. It is sufficient to add the uninflected word, as the inflection can be looked up in the lexical services. Specifying nouns and adjectives is similar to specifying verbs. Word relationships like “synonym” or “word formation” can be defined between words. Besides the word relationships, the semantic descriptions are defined within the thesaurus editor as well.

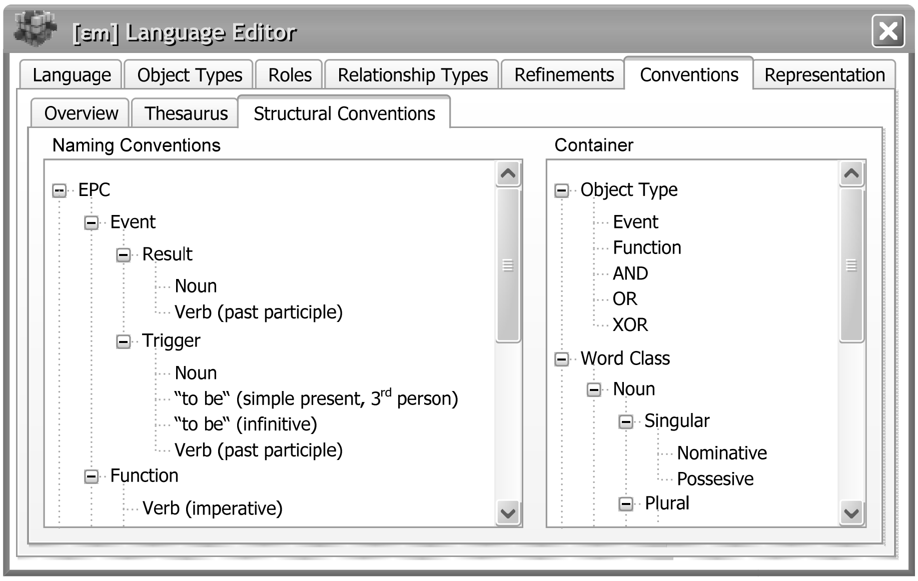

Figure 6. Specification of Phrase Structure Conventions

In the next step, phrase structure conventions are defined and connected to those language elements for which they are valid. Figure 6 illustrates this with the example of EPCs. Here, phrase structure conventions are created for the model element type *event.* Concerning their semantics (but not their type), events of the EPC can be distinguished in *trigger* events and *resulting* events. The former ones trigger action in activities, the latter ones conclude activities. Different phrase structures can be attached to each of them in regard to their different semantics. An example for a trigger event is “Invoice is to be checked”, hence an appropriate phrase structure convention called “Trigger” is <*noun*> + <*“to be”, simple present,* 3^{rd} *person*> + <*“to be”, infinitive*> + <*verb, past participle*>. With this phrase structure, a set of trigger events can be named. However, different aspects might require additional phrase structures to be defined. For result events, an adequate phrase structure is <*noun*> + <*verb, past participle*>, allowing phrases like “Invoice checked”.

Once generated, the phrase structure conventions in combination with the domain thesaurus are used during modelling. The modeller gets just-in-time hints as soon as he violates a convention (cf. Figure 7). First, he might have chosen invalid terms (e.g. *bill* instead of *invoice* or *audit* instead of *check*). As soon as a phrase is entered, it is parsed to determine its compliance with the naming conventions. Every term is transformed to its uninflected form and is compared to the domain thesaurus. If it is not found, synonymous valid terms are searched in the lexicon. If according alternatives are found, they are proposed to the modeller. Otherwise, the modeller has to rename the respective element – optionally by choosing a valid term from the domain thesaurus. Second, violations of phrase structure conventions are signalled, and alternative valid structures are proposed. Summarising, the name *audit bill* is suggested to be changed to *Invoice is to be checked.* Phrases corresponding with both the domain thesaurus and the phrase structure conventions are accepted without any feedback.

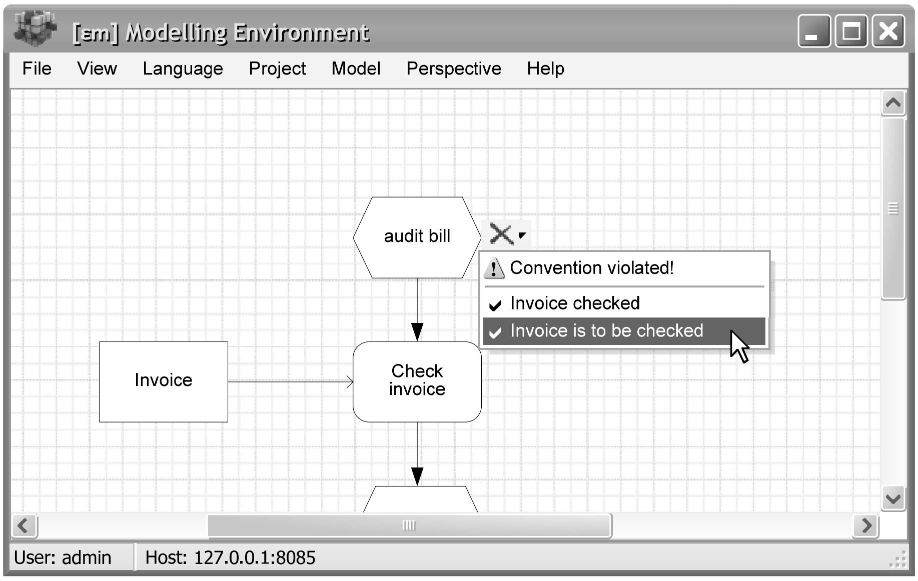

Figure 7: Automatic Guidance in Order to Comply with Naming Conventions

4.2 Explorative Analysis

Naming practices in process models provide evidence concerning the danger of naming conflicts as well as show the feasibility of approaches that aim at resolving or even avoiding them. In order to empirically evaluate our phrase structure conventions approach we conducted an exploratory empirical analysis. We analysed two process modelling projects consisting of overall 257 EPC models containing in turn overall 3918 elements (1827 functions and 2091 events). Within these modelling projects, modelling conventions were available in terms of glossaries and phrase structures. However, these conventions solely existed as textual recommendations rather than methodical support. All model element names were analysed and revised. We then manually analysed 280 phrase structure types, which occurred at least twice. These structures covered 1770 model element names in total. We were able to match the found structures to 47 identified phrase structure conventions, which had the same power of expression. E.g. we were able to express the same semantics concurrently achieving less naming variability. This analysis shows that having applied our approach would have substantially reduced naming differences and thus supported model comparability.

5 CONCLUSION AND OUTLOOK

Integrating naming conventions into conceptual modelling languages is promising for increasing the comparability of conceptual models. Two characteristics are significant to avoid common problems:

- Defining and providing naming conventions *previously to modelling* is the basis for avoiding naming conflicts rather than resolving them. Therefore, time-consuming alignment of namings in the course of model comparison becomes dispensable.
- *Guiding the modeller automatically* during modelling is of substantial importance, since only this way the compliance with the modelling conventions can be assured.

Certainly, specifying naming conventions in the proposed way is time-consuming. Our approach is therefore mainly suited for large-scaled modelling projects. Nevertheless, for every project, business domain or company, the conventions have to be specified only once and are reusable. Moreover, term models, thesauri, or glossaries that may already exist in companies or business domains can be reused.

Future research will focus on the further evaluation of the proposed approach. In the short-term, the approach will be instantiated for different modelling languages, different natural languages and different application scenarios. In particular, the capability of our approach to increase the efficiency of distributed conceptual modelling and its acceptance will be evaluated in empirical studies. In order to assure the applicability of the approach, the demonstrator software will be enhanced in order to make it usable in practice. In the course of evaluation, it will also be investigated if ambiguities play a role in model element names. E.g. the sentence “They hit the man with a cane” is ambiguous, even if the meanings of all of the used words are considered definite. Thus, we will analyse further conceptual models and determine if phrase structures are common in conceptual modelling that promote ambiguities. A result of this analysis could be a recommendation to restrict phrase structure conventions to phrases that do not lead to ambiguities.

Middle-term research will address further approaches in order to facilitate the comparison of models that are developed in a distributed way. Here, the comparison of conceptual models based on the model structure is a promising research area. Moreover, we will question if modelling conventions on the basis of structure patterns that are provided at the beginning of modelling are able to increase the comparability of conceptual models as well.

References

Batini, C. and Lenzerini, M. (1984). A Methodology for Data Schema Integration in the Entity Relationship Model, in: IEEE Transactions on Software Engineering 10 (6), pp. 650-663.

Batini, C., Lenzerini, M. and Navathe, S. B. (1986). A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18 (4), pp. 323-364.

Bhargava, H. K., Kimbrough, S. O. and Krishnan, R. (1991). Unique Name Violations, a Problem for Model Integration or You Say Tomato, I Say Tomahto. ORSA Journal on Computing 3 (2), pp. 107-120.

Bögl, A., Kobler, M. and Schrefl, M. (2008). Knowledge Acquisition from EPC Models for Extraction of Process Patterns in Engineering Domains. In: Proceedings of the Multi-Conference on Information Systems [in German: Multikonferenz Wirtschaftsinformatik] 2008 (MKWI 2008). Munich.

Born, M., Dörr, F. and Weber, I. (2007). User-friendly semantic annotation in business process modeling. In: Proceedings of the International Workshop on Human-Friendly Service Description, Discovery and Matchmaking (Hf-SDDM 2007) at the 8th International Conference on Web Information Systems Engineering (WISE 2007). Editors: Weske, M., Hacid, M.-S. and Godart, C., Nancy, pp. 260-271.

Chen, P.P.-S. (1976). The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems 1 (1), pp. 9-36.

Copestake, A., Flickinger, D. (2000). An open-source grammar development environment and broad-coverage English grammar using HPSG. In: Proceedings of the 2nd Conference on Language Resources and Evaluation (LREC 2000). Athens.

Ehrig, M., Koschmider, A. and Oberweis, A. (2007). Measuring Similarity between Semantic Business Process Models. In: Proceedings of the 4th Asia-Pacific Conference on Conceptual Modelling (APCCM) 2007. Ballarat.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G. and Mehandjiev, N. (2004). Meta-design: a manifesto for end-user development. Communications of the ACM 47 (9), pp. 33-37.

Frank, U., Strecker, S. (2007). Open Reference Models – Community-driven Collaboration to Promote Development and Dissemination of Reference Models. Enterprise Modelling and Information Systems Architectures 2 (2), pp. 32-41.

Greco, G., Guzzo, A., Pontieri, L. and Saccà, D. (2004). An ontology-driven process modeling framework. In: Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA 2004). Editors: F. Galindo, F., Takizawa, M. and Traunmüller, R., Zaragoza, pp. 13-23.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5 (2), pp. 199-220.

Guarino, N. (1998). Formal Ontology and Information Systems. In: Proceedings of the 1st International Conference on Formal Ontologies in Information Systems. Editor: N. Guarino. Trento, pp. 3-15.

Hadar, I. and Soffer, P. (2006). Variations in conceptual modeling: classification and ontological analysis. Journal of the AIS 7 (8), pp. 568-592.
Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly 28 (1), pp. 75-105.
Höfferer, P. (2007). Achieving business process model interoperability using metamodels and ontologies. In: Proceedings of the 15th European Conference on Information Systems (ECIS 2007). Editors: H. Österle, J. Schelp and R. Winter. St. Gallen, pp. 1620-1631.
ISO (1982). ISO/TC97/SC5/WG3: Concepts and Terminology for the Conceptual Schema and the Information Base.
Koschmider, A. and Oberweis, A. (2005). Ontology Based Business Process Description. In: Enterprise Modelling and Ontologies for Interoperability, Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Co-located with CAiSE'05 Conference. Porto.
Kugeler, M. (2000). Organisational Design with Conceptual Models. Modelling Conventions and Reference Process Model for Business Process Reengineering [In German: Informationsmodellbasierte Organisationsgestaltung. Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation]. Berlin.
Lawrence, R. and Barker, K. (2001). Integrating Relational Database Schemas using a Standardized Dictionary. In: Proceedings of the 2001 ACM symposium on Applied computing (SAC). Las Vegas.
Mitkov, R. (2003). The Oxford handbook of computational linguistics. Oxford et al.
Pfeiffer, D. (2008). Semantic Business Process Analysis – Building Block-based Construction of Automatically Analyzable Business Process Models. Münster.
Phalp, K. and Shepperd, M. (2000). Quantitative analysis of static models of processes. Journal of Systems and Software 52 (2-3), pp. 105-112.
Pollard, C. J. and Sag, I. A. (1994). Head Driven Phrase Structure Grammar. In: University of Chicago Press, Studies in Contemporary Linguistics. London, Chicago.
Rizopolous, N. and McBrien, P. (2005). A General Approach to the Generation of Conceptual Model Transformations. In: Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05). Porto.
Rosemann, M. (1996). Complexity Management in Process Models. Language-specific Modelling Guidelines [In German: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung]. Wiesbaden.
Rosemann, M. (2003). Preparation of Process Modeling. In: Process Management – A Guide for the Design of Business Processes. Editors: Becker, J., Kugeler, M. and Rosemann, M., pp. 41-78.
Sabetzadeh, M., Nejati, S., Easterbrook, S. and Chechik, M. (2007). A Relationship-Driven Framework for Model Merging. In: Proceedings of the Workshop on Modeling in Software Engineering (MiSE'07) at the 29th International Conference on Software Engineering. Minneapolis.
Scheer, A.-W. (2000). ARIS – Business Process Modelling. 3rd Edition. Berlin.
Vergidis, K., Tiwari, A. and Majeed, B. (2008). Business process analysis and optimization: beyond reengineering. IEEE Transactions on Systems, Man, and Cybernetics 38 (1), pp. 69-82.
vom Brocke, J. and Thomas, O. (2006). Designing Infrastructures for Reusing Conceptional Models. A General Framework and its Application for Collaborative Reference Modeling. In: Proceedings of the 9th International Conference on Business Information Systems (BIS2006). Editors: Abramowicz, W. and Mayr, H. C. Klagenfurt, pp. 501-514.
White, S. A. and Miers, D. (2008). BPMN Modeling and Reference Guide. Understanding and Using BPMN. Lighthouse Point.
Wulf, V. and Jarke, M. (2004). The economics of end-user development. Communications of the ACM 47 (9), pp. 41-42.