UNIFIED ENTERPRISE KNOWLEDGE REPRESENTATION WITH CONCEPTUAL MODELS – CAPTURING CORPORATE LANGUAGE IN NAMING CONVENTIONS

Completed Research Paper

Patrick Delfmann
University of Münster
European Research Center
for Information Systems (ERCIS)
Leonardo-Campus 3
48149 Münster, Germany
patrick.delfmann@ercis.uni-muenster.de

Sebastian Herwig
University of Münster
European Research Center
for Information Systems (ERCIS)
Leonardo-Campus 3
48149 Münster, Germany
sebastian.herwig@ercis.uni-muenster.de

Łukasz Lis
University of Münster
European Research Center
for Information Systems (ERCIS)
Leonardo-Campus 3
48149 Münster, Germany
lukasz.lis@ercis.uni-muenster.de

Abstract

Conceptual modeling is an established instrument in the knowledge engineering process. However, a precondition for the usability of conceptual models is not only their syntactic correctness but also their semantic comparability. Assuring comparability is quite challenging especially when models are developed by different persons. Empirical studies show that such models can vary heavily, especially in model element naming, even if they are meant to express the same issue. In contrast to most ontology-driven approaches proposing the resolution of these differences ex-post, we introduce an approach that avoids naming differences in conceptual models already during modeling. Therefore we formalize naming conventions combining domain thesauri and phrase structures based on a linguistic grammar. This allows for guiding modelers automatically during the modeling process using standardized labels for model elements, thus assuring unified enterprise knowledge representation. Our approach is generic, making it applicable for any modeling language.

Keywords: Knowledge Representation, Conceptual Modeling, Linguistics, Naming Conventions

Introduction

In early stages of the knowledge engineering process, conceptual models are commonly used to depict main concepts of the application domain along with their mutual relationships (Schreiber 2000). In communities where shared knowledge is to be managed, conceptual models provide a framework for sharing the common meaning of symbols exchanged during communication (Maedche et al. 2003; Motik et al. 2002). Especially when enterprise knowledge is concerned, meaning knowledge on the structure and business processes of an enterprise, its goals and their operationalization opportunities, conceptual models are a general means of (semi-)formal knowledge representation (Loucopoulos & Kavakli 1999; Kalpic & Bernus 2002).

However, representing corporate knowledge using conceptual modeling raises some problems. Due to the mostly extensive amount of modeling effort, modeling tasks are commonly distributed, meaning split and performed by different persons, at different times, and at different places. Empirical studies show that especially those conceptual models, which are developed in a timely, personally and regionally distributed way, can vary heavily concerning used terms (Hadar & Soffer 2006). Thus, so-called naming conflicts (Batini et al. 1986) may occur, even if the same issue is addressed. Moreover, even models of the same issue developed by the same persons at different times may show intense variations. The same applies even for situations where models are developed in modeling projects being strictly organized within a company or a corporate group. Enterprise knowledge that is modeled using a heterogeneous naming has to be transformed into the unified corporate language in order to make the models comparable. Usually, an according standardization process requires discussions including all involved modelers in order to reach a consensus. Sometimes, even external consultants are involved additionally (Phalp & Shepperd 2000; Vergidis et al. 2008). Thus, this task can be extremely laborious.

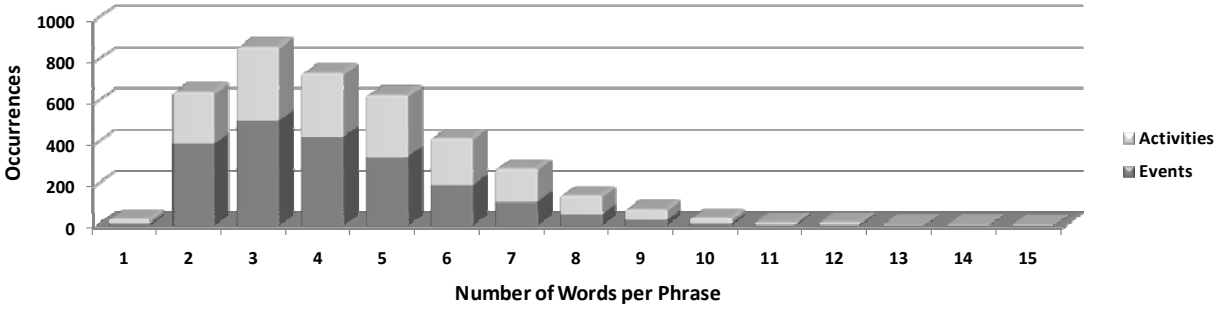

Figure 1: Average Number of Words Used in Process Model Element Names

The problem of naming conflicts in conceptual models becomes evident especially when looking at process models. Process models are extra prone to naming conflicts, since process model elements are usually named with sentence fragments rather than with single terms. We have conducted an exploratory empirical analysis of two modeling projects that supports this hypothesis. The analyzed model base consisted of overall 257 process models containing in turn overall 3918 elements (1827 activities and 2091 events). Within these modeling projects, naming guidelines were available in terms of a corporate glossary and suggested phrase structures. However, these guidelines solely existed as textual recommendations. We analyzed all model element names of the process models and found out that, first, most elements are named with sentence fragments rather than with single terms (cf. Figure 1).

Table 1: Phrase Structures in Process Model Element Names

# of terms	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of events	10	396	509	429	331	197	114	55	27	10	4	4	2	2	1
# of different phrase structures (event)	6	37	136	221	248	175	102	54	26	10	4	4	2	2	1
# of activities	21	252	358	310	301	225	160	90	52	26	12	13	2	3	2
# of different phrase structures (activity)	3	29	85	157	204	193	141	87	52	25	12	13	2	3	2

Second, element names containing a certain number of terms consisted of many different phrase structures (e.g., <verb, imperative> <noun, singular>, in particular "audit invoice" or <noun, singular> <verb, gerund>, in particular "invoice auditing"; cf. Table 1).

Approaches towards resolving or avoiding naming conflicts in process models therefore have to consider not only the terms but also the phrase structures used in model element names. Considering both aspects allows us to capture the corporate language to be used and support linguistic unification already during modeling in the knowledge externalization phase.

In literature, there exist many contributions that propose approaches for resolving naming conflicts in conceptual models subsequent to modeling (cf. "Related Work" section). Unlike these approaches, the goal of this article is to introduce an approach that ensures unified naming by avoiding potential conflicts already during modeling. This way, we prevent problems that result from the ex-post resolution of naming conflicts and make the costly language unification process described above dispensable. We define naming conventions for elements of modeling languages and ensure their compliance by an automated, methodical guiding during modeling. The conventions are set up using domain terms and phrase structures that are defined as valid in the regarded modeling context. As a formal specification basis, we use thesauri that provide term conventions not only for nouns but also for verbs and adjectives, including descriptions of their meanings. Furthermore, we specify phrase structure conventions formally. During modeling, model element names are validated simultaneously against both the term and phrase structure conventions using linguistic parsing. This way, only the given unified corporate language can be used during modeling. Certainly, a precondition for the applicability of such an approach is the possibility to provide all involved modelers with these naming conventions and the according parsing methods previously to modeling. Since process models are extra prone to naming conflicts as shown above, we illustrate our approach with examples based on process models. However, our approach is generic so that it can be applied to any conceptual modeling language maybe being less prone to naming conflicts.

The approach is suitable for modeling situations, where it is possible to provide all involved modelers with the necessary information about the modeling conventions. These are modeling projects that are determined regarding organization and/or business domain. Nevertheless, the modeling tasks can take place in a distributed way, meaning at different places, by different persons and at different times. It is only important that the modeling conventions can be provided to each involved modeler, assuring that the conventions are applied to every model to be constructed.

This paper is structured as follows: First, we analyze related work on naming conflict resolution and discuss the research gap that led to the development of the approach presented in this paper. Furthermore, we outline our research methodology. As a next step, we introduce a conceptual framework for the specification and enforcement of naming conventions. The feasibility of our approach is shown exemplarily with a detailed application scenario. We finish the paper in a "Conclusions and Outlook" section and motivate further research.

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 et al. 1986; Bhargava et al. 1991; Lawrence & Barker 2001; Rahm & Bernstein 2001). Hence, these approaches focus on data modeling 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 conceptual modeling languages (especially process modeling languages), names are used that consist of sentence fragments containing terms of any word class. Thus, these approaches are only suitable for data modeling languages as a specific class of conceptual modeling languages.

Other approaches make use of ontologies (Gruber 1993; Guarino 1998; Preece et al. 2001) to address the problem of semantic comparison of names and avoiding ambiguities in knowledge representation. Those approaches can be distinguished into two different kinds. On the one hand, authors act under the assumption that there exists a "generally accepted" ontology describing a certain modeling domain. It is assumed that all considered models of this domain comply with its ontology. This means that modelers had a thorough knowledge of the ontology before the modeling took place. On the other hand, approaches suggest deriving an ontology from the models that have to be analyzed, which has to be performed after the modeling took place.

There are few examples for the former approach. For instance, Greco et al. (2004) propose adopting terms from existing ontologies for process models manually. A problem is that due to manual adoption, correctness cannot be assured. Born et al. (2007) propose semi-automated adoption of model element names based on concepts from a previously defined ontology. They restrict their approach to models of the Business Process Modeling Notation (BPMN) (White & Miers 2008) and describe a software implementation. However, their methodical support is limited to generating proposals for the naming of a given activity based on previous activities and the order of matching domain actions defined in the ontology. Users can, however, choose other naming of their own and thus abandon the convention provided by the ontology. Thus, again, naming correctness cannot be assured. Moreover, there is no support for the definition and use of more sophisticated naming conventions besides activities consisting of a verb and a noun, which have to be previously defined as domain actions in the ontology. The general problem with these approaches is that only because two modelers act in the same business domain does not guarantee 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 modelers already during the modeling process. Additionally, in order to ensure comparability of the models, modelers have to comply strictly with the ontology. The analyzed approaches make the implicit assumption that these preconditions are already given rather than addressing a methodical support.

For the latter approach, for example Höfferer (2007) connects domain ontologies to the terms that are used as names in conceptual models. This way, he establishes relationships between elements of different models that are to be analyzed. In addition to ontologies, Ehrig et al. (2007) define combined similarity measures that consist of syntactic and semantic parts. These serve as a basis for the decision whether the model elements compared are equivalent or not. 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 modelers, these can be considered as semantically identical as well (Koschmider & Oberweis 2005; Sabetzadeh et al. 2007). It has to be questioned whether the advantage of the subsequent connection of the models via the ontology warrants the efforts in comparison to a conventional manual analysis.

Only few approaches, mainly originating from the German speaking area, suggest standardized phrases for model element names in order to increase the clarity of process models. For example, Rosemann (1996) and Kugeler (2000) propose particular phrase structure guidelines for names of process activities (e.g., <verb, imperative> <noun, singular>; in particular "check invoice"). These are, however, guidelines only and, as no methodical support is provided, it remains up to modelers whether they apply the conventions or not. Moreover, the authors propose so-called Technical Term Models (Rosemann 2003) that have to be designed previously to process modeling and that specify the terms to be used within the phrases. However, the scope of Technical Term Models is restricted to nouns. Similar approaches provided by Koschmider and Oberweis (2005) and Sabetzadeh et al. (2007) propose the provision of a generally accepted vocabularies. Here again, the authors provide no methodical or technical support, which would force the modelers to use the preferred terms. Bögl et al. (2008) propose an approach for the automated ex post extraction of process pattern from EPC models. They use the online dictionary WordNet (2009) to provide a common sense vocabulary in addition to controlled domain vocabularies, which have to be maintained manually. As online lexical services provide extensive collections of nouns, verbs, and adjectives as well as their semantic relationships, we use them in our approach as well. Actually, the proposed approaches are promising regarding increased comparability of conceptual models since all of them aim at standardizing names for model elements prior to modeling. However, up to now, a methodical realization is missing.

To sum up, we identify the following need for development towards avoiding naming conflicts in 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 no methodical support for (3) guiding modelers in order to comply with the conventions. In order to realize 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 thesaurus, and (3) a procedure model to guide modelers automatically in complying with the conventions. Since process models are extra prone to naming conflicts (as indicated by the empirical study presented above), the case of process modeling seems to be an challenging application scenario for the methodical realization of naming conventions. Hence, to exemplify our proposed approach, we make use of process models. However, our approach is generic and not restricted to process models, making it applicable for further classes of conceptual models that are generally named with sentence fragments like UML Use Case or Interaction Overview Diagrams (OMG 2009).

Research Methodology

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

In this contribution the scientific artifact is the naming conventions approach outlined in the "Introduction" section. This artifact aims at solving the relevant problem of the lacking comparability of conceptual models (cf. "Introduction" section; cf. "Naming Practices in Process Models" section for further evidence). Related work does not provide satisfactory solutions up to now (cf. "Related Work" section). Hence, the approach presented here (cf. "Specification and Enforcement of Naming Conventions" section) makes an innovative contribution to the existing knowledge base. In order to validate the general feasibility of the approach, we developed a prototypical modeling software and applied it exemplarily to a detailed application scenario (cf. "Modeling Tool Support and Application Example" section). Further evaluations concerning applicability and acceptance as well as efficiency and increase of comparability will be subject of empirical studies to be performed in the short term (cf. "Conclusions and Outlook" section).

Specification and Enforcement of Naming Conventions

Procedure Model

In order to provide an integrated framework, we advocate the usage of a specific corporate language that is used for naming model elements in a certain modeling context (i.e., a specific modeling domain, project or company). We argue that such a unified corporate language can be captured in the form of naming conventions. This corporate language is a subset of the respective natural language (here: English) used in the modeling context. The language consists of a set of valid domain terms that are allowed to be used in model element names exclusively. That is, 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. This means that 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 2).

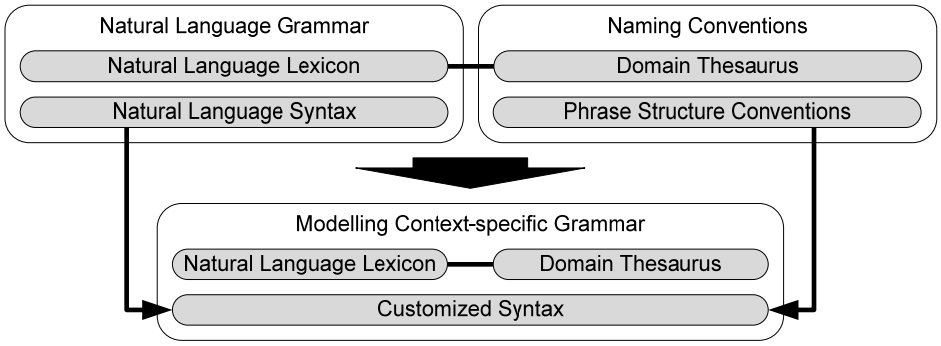

Figure 2: Customizing the Natural Language Grammar with Naming Conventions

Natural language grammars are usually defined by a formalism that consists of a lexicon and a syntax specification (Mitkov 2003). Such a grammar is complemented with naming conventions, which again consist of term and phrase structure conventions. Term conventions are specified by a 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. The thesaurus is then connected to the natural language's lexicon. Moreover, valid phrase structures are specified by phrase structure conventions. Hence, the natural language is customized for the needs of a specific

modeling 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 the next sub-section.

The thesaurus can be created from scratch, or by reusing possibly existing thesauri or glossaries. It includes single nouns, verbs and adjectives that are interrelated. Other word classes are generally domain independent. Thus, as they are already included in the general 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(s) in the general lexicon. This additional term related information can be obtained from linguistic services, which already exist for different natural languages (e.g., WordNet (2009), which is such a lexicon service for the English language). Therefore, in case of a later violation of the naming conventions by the modeler, synonymous or derived valid terms can be automatically identified and recommended. The terms specified are provided with short textual semantic descriptions, allowing modelers for looking up the exact meaning of a term. The thesaurus should not be changed during a modeling project in order not to violate the consistency of application.

The naming conventions have to be specified once for every modeling context, whereas already existing conventions can be reused (cf. Figure 3). For example, in the context of process modeling, activities like such in BPMN are labeled with actions (e.g., <verb, imperative> <noun, singular>; in particular "check invoice") and events are labeled with states (e.g., <noun, singular> <verb, past participle>; in particular "invoice checked"). Naming conventions are modeling language-specific. For each model element type of a certain modeling language (e.g., activities in BPMN) 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.

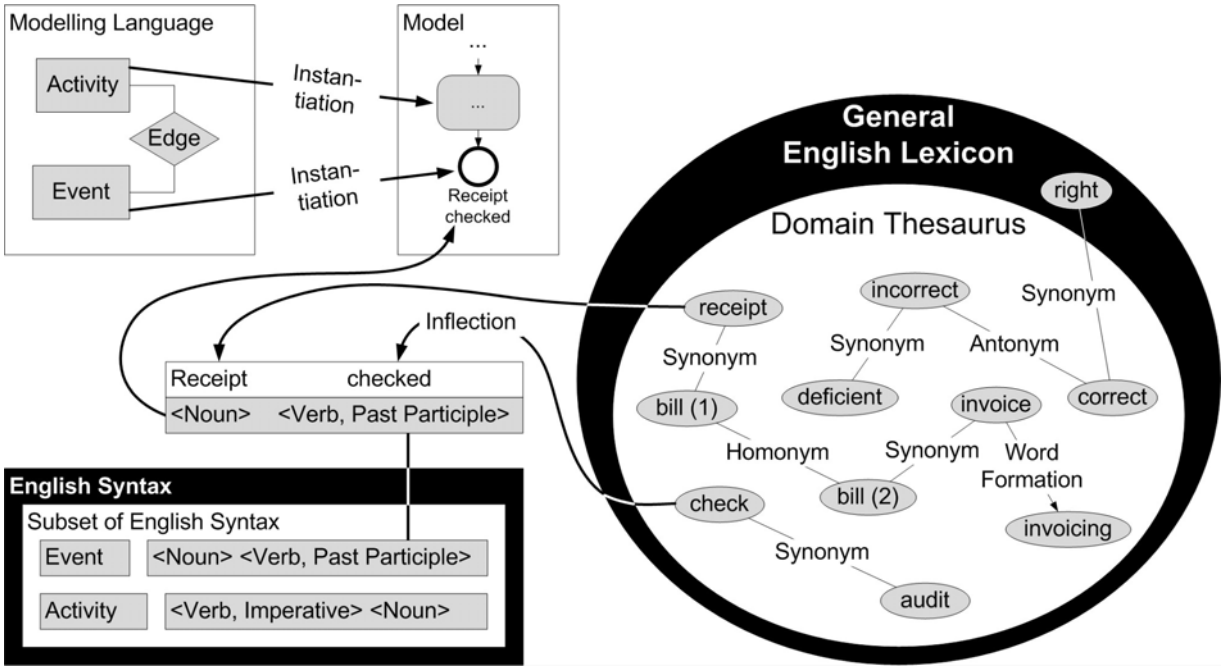

Figure 3: Using Formalized Naming Conventions

The conventions should be defined by a project team consisting of domain experts and modeling experts. This means that the stakeholders responsible for the conventions should have thorough knowledge of the actual modeling 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 (Automotive Thesaurus 2009; Tradeport 2009; WWW Virtual Library 2009), which should be reused and adapted depending on the modeling situation.

During modeling, the model element names entered by a modeler are verified simultaneously against the specified context-specific grammar. On the one hand, the structure of an entered model element name is validated against the customized syntax specification. On the other hand, it is checked whether the used terms are allowed. Nouns, verbs, and adjectives (i.e., word classes covered by the thesaurus) are validated against it. 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 modeling 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 modelers themselves have to decide, which of the recommendations fits their particular needs. By looking up the semantic descriptions of the terms, modelers can choose the appropriate one. Alternatively, they can choose a valid structure as a pattern and fill in the gaps with valid terms on their own. However, it should be possible for the modeler to propose a new term with a short textual semantic description. In order not to distract the modeler from his current modeling session, the proposed term is then accepted temporarily. In a next step, it is up to the modeling project expert team whether they accept the term or not. If the term is accepted, it is added to the thesaurus. Otherwise, the modeler is informed to revise the model element. Hereby, we ensure that equal model element names represent equal semantics, which is a precondition for comparability of conceptual models. A detailed description of the modeling process supported by our approach will be provided in the "Enforcing Naming Conventions during Modeling" subsection.

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 4). Phrase structure conventions (PSC) are defined depending on distinct element types of conceptual modeling languages (e.g., activities in process models are named differently to events).

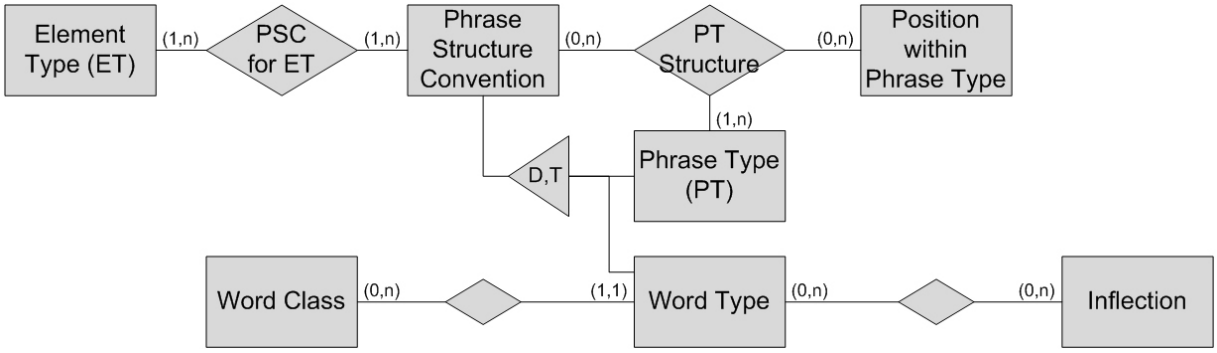

Figure 4: Specification of Phrase Structure Conventions on Type Level

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, we define the allocation of sub phrase types or word types to a phrase type and their *position* in the superordinate phrase type.

A word type consists of a distinct *word class* (noun, verb, adjective, adverb, article, pronoun, preposition, conjunction or numeral) – and its *inflection*. Inflections modify a word according to its case, number, tense, gender, mood, person, or comparative. These are usually combined. For instance, a particular combined inflection is <3rd person, singular>. 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.

Independent from their corresponding word class, particular uninflected words are called *lexemes* (e.g. the verb "check"). Inflected words are called *word form* (e.g. past participle "checked"). Word forms are assigned to the corresponding word classes and inflections, that is their word types. Thus, word forms represent words belonging to a particular word type (cf. Figure 5).

In order to specify the domain thesaurus, allowed words are stored in the form of lexemes that are related by different *word relationship types*. These are *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 modeling 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 modeling. Word formation relations are used

to search for appropriate alternatives when a modeler has used invalid terms and phrase structures. For instance, 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. preceding sub-section) are connected to the domain thesaurus. To specify what is actually meant by a lexeme, a semantic *description* is added at least to each dominant lexeme. This way, modelers are enabled to check whether the lexeme they have used actually fits the modeling issue.

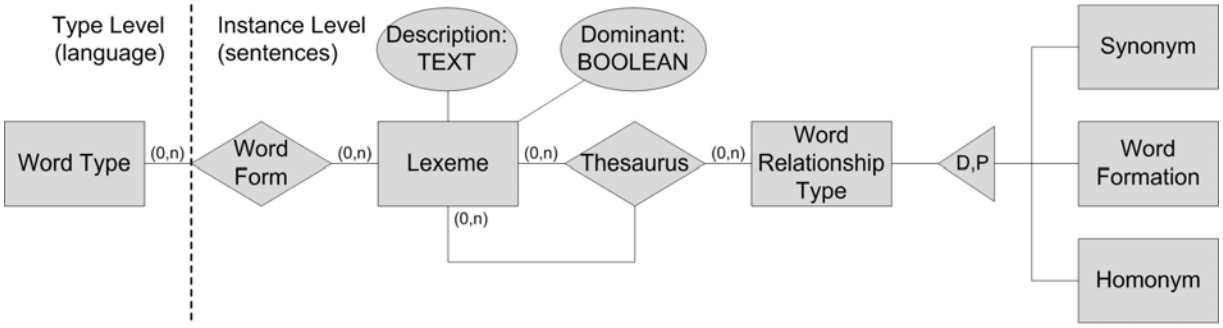

Figure 5: Specification of Term Conventions on Instance Level

Checking Compliance via Linguistic Parsing

The naming conventions specified via the framework introduced above have to be enforced during modeling to assure the compliance of the models with the conventions. Therefore, we make use of linguistic parsing being able to detect both lexemes and the phrase structure of a given phrase. Linguistic parsing methods return the parsing results in a formal grammar making it possible to reuse the results easily. In our approach, we make use of a parsing method based on the Head-driven Phrase Structure Grammar (HPSG) (Pollard & Sag 1994). HPSG is an established grammar of the class of so-called unification grammars (Mitkov 2003). These grammars are well-known in the field of computational linguistics and provide syntax specifications for natural languages. Several formal specifications of natural languages based on HPSG are already available (for an overview, cf. Delphin (2009)). A formal specification of the English syntax with HPSG was developed at the CSLI LinGo Lab of the University of Stanford (Copestake & Flickinger 2000). The parsing method used in this approach relies on the latter. An exemplary parsing result for the phrase "check invoice" is shown in Figure 6.

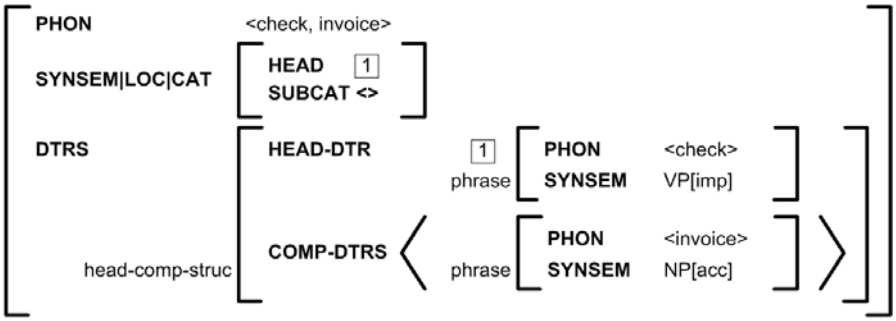

Figure 6: Exemplary HPSG Parsing Result

HPSG determines the phrase <check, invoice> and characterizes it as a verbal phrase. The verbal phrase consists of a so-called head, the constituting element of the phrase and one or more sub-components (cf. "SUBCAT <>"). The sentence "check invoice" is fractionized into its components, which are, in this case, the lexemes "check" and "invoice". The first component <check> is identified as a verbal sub-phrase in imperative form (cf. "VP[imp]"). The annotation "[1]" indicates that this sub-phrase is the head of the verbal super-phrase. Since the verbal sub-phrase consists of only one word, it is not further fractionized. The component <invoice> is identified as a nominal sub-phrase in accusative case (cf. "NP[acc]"). Here, the case is less relevant for the English language. However, for languages like for instance German, the distinction between different cases is important. The angle bracket indicates that there could be further sub-phrases, which is not the case here. Since the results of HPSG-based parsing are provided in a formal grammar, they can easily be reused to validate them against the naming conventions.

Enforcing Naming Conventions during Modeling

In the following, we outline the process of enforcing naming conventions during modeling, which is executed whenever the modeler creates a new model element and enters its name. The process is based on a heuristic validating the entered phrase against the naming conventions and suggesting alternatives when needed (cf. Figure 7).

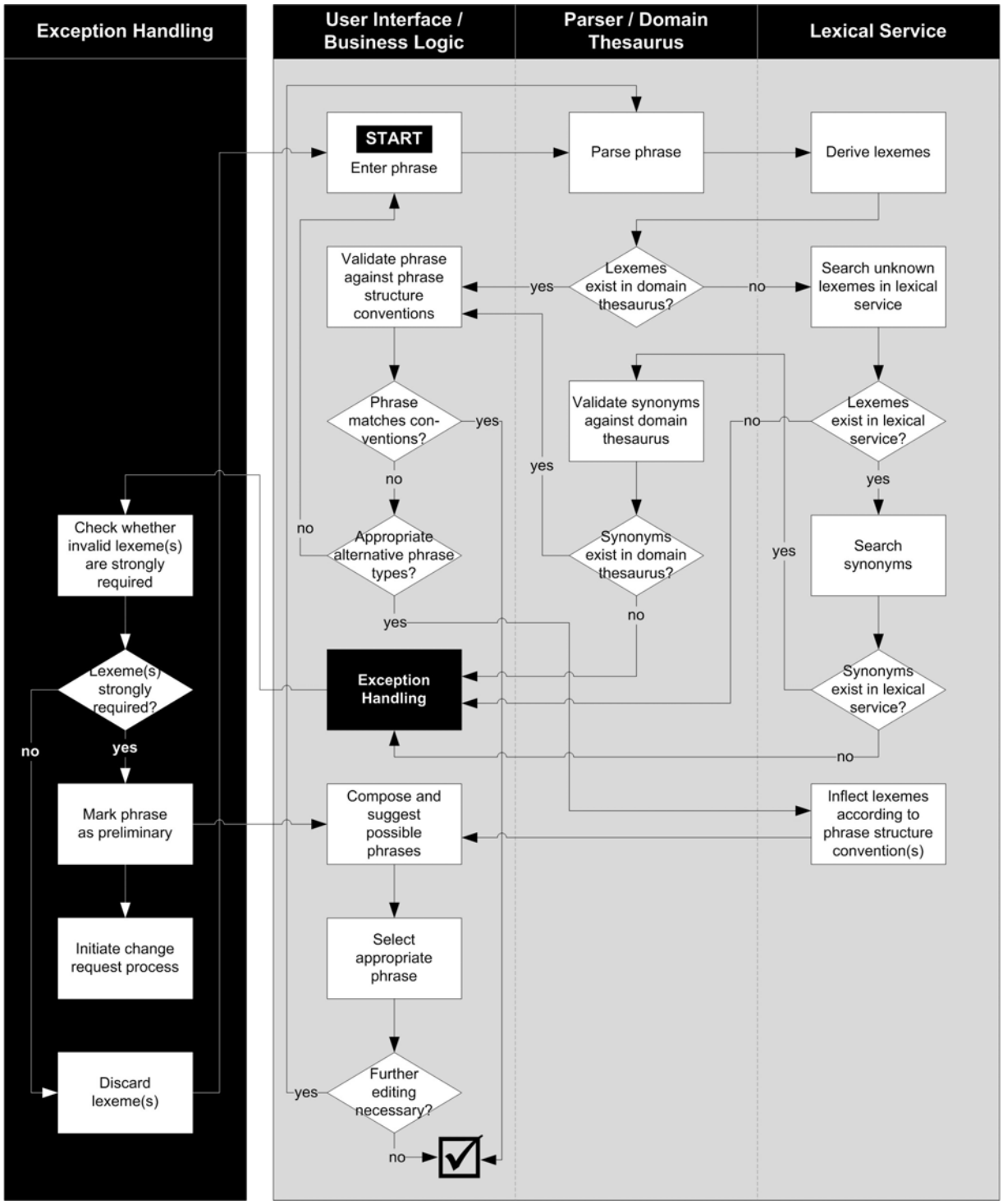

Figure 7: Validation, Suggestion, and Exception Handling Heuristic

As soon as the modeler has entered the name of a model element, a linguistic parser analyzes it concerning the phrase structure and the used words. Starting from the phrase structure and the used words, the heuristic checks the compliance of the model element name with the naming conventions. First, the words detected by the parser are re-inflected to their lexeme form. All of the lexemes belonging to the word classes *noun*, *adjective/adverb* or *verb* are validated against the domain thesaurus. If they are found in the domain thesaurus, they are valid, meaning they comply with the naming conventions. If one or more lexemes are not found in the domain thesaurus, the heuristic searches for synonyms of each invalid lexeme in a general lexicon. The synonyms found are matched against the domain thesaurus. If one of the synonyms matches the thesaurus entry, the original lexeme is replaced with the valid synonym (e.g., the invalid lexeme "bill" is replaced with the valid lexeme "invoice").

Second, the phrase structure used as model element name is validated against the phrase structure conventions. If the phrase structure and all of the original lexemes are valid, the original phrase as a whole is marked as valid, and the heuristic terminates. If the phrase structure is valid, and the original lexemes had to be replaced by synonyms, the synonyms are inflected automatically according to the phrase type, and the new phrase is proposed to the modeler as an alternative to his original suggestion (e.g., "The name 'audit bill' you entered violates the naming conventions. Did you mean 'check invoice'?").

If the phrase structure entered by the user is invalid, the heuristic calculates all possible phrases complying with both the phrase structure conventions and the domain thesaurus. Of course, this can cause multiple phrase alternatives, which are not necessarily complete each. As a consequence, the modeler has to complete an incomplete phrase, if s/he chooses one. In this case, the phrase is parsed once again. In some cases, the heuristic must provide an exception handling mechanism. This applies for the following situations:

- a synonym search is not possible due to a totally unknown original lexeme
- no synonyms are found for an invalid lexeme
- synonyms are found, but none of them matches the domain thesaurus

In these cases, the user is prompted whether the word s/he has used is strongly required to express the semantics of the model element name, or whether it can be discarded or replaced by a word specified in the domain thesaurus. In the latter case, the user can search the domain thesaurus for an appropriate word. If the original term is strongly required and cannot be replaced by an alternative one, the user can propose the term as a new entry for the domain thesaurus. Then a modeling expert committee has to decide whether the term is added to the domain thesaurus or not. In this case, the model element name is marked as preliminary until the decision of the committee. Once a decision has been made, the domain thesaurus is either updated accordingly, and the model element marked as preliminary is finally accepted, or the model element name marked as preliminary is changed by the modeling expert committee according to the naming conventions. In both cases, the modeler is informed. Accepting a new term requires a preceding synonym analysis of the thesaurus to prevent ambiguities.

Modeling Tool Support and Application Example

To validate the general applicability of our approach, we developed a modeling prototype. The way of navigating through the software and its handling is tightly connected to the procedure model motivated in the preceding section.

As described above, the connection of our approach with modeling languages requires the adoption of the respective meta model. For the tool support, this indicates the necessity of meta modeling abilities, which is supported by our research prototype. Hence, virtually any modeling language that can be created or exists inside the prototype can be extended with naming conventions. In the following, we show the application of the tool using a particular application scenario.

Table 2: Excerpt of a Domain Thesaurus

Nouns	Verbs	Adjectives
invoice	check	valid
sum	receive	invalid
goods receipt	compare	new

As a preliminary step, the person responsible for specifying the modeling conventions has to define the terms, which are allowed for the modeling context. An exemplary excerpt of a domain thesaurus is shown in Table 2. The section contains terms that are necessary to express invoice auditing issues.

As a second step, the phrase structure conventions are specified. Table 3 contains exemplary phrase structure conventions feasible for BPMN. On the one hand, *activities* are to be named with phrases, which express that *something is done*. On the other hand, phrase structure conventions for *events* express that *something has been done*. The example in Table 3 represents only a section of possible phrase structure conventions. Thus, further phrase types for both events and tasks are conceivable.

Table 3: Exemplary Phrase Structure Conventions for BPMN

Activity	Event
<*verb, imperative*> <*noun, singular, object case*>	<*noun, singular, subject case*> <*verb, past participle*>
<*verb, imperative*> <*noun, singular, object case*> <*preposition*> <*noun, singular, object case*>	<*noun, singular, subject case*> <*verb, past participle*> <*preposition*> <*noun, singular, object case*>
<*verb, imperative*> <*conjunction*> <*noun, singular, subject case*> <*auxiliary verb, 3rd person singular, simple present*> <*adjective*>	<*noun, singular, subject case*> <*auxiliary verb, 3rd person singular, simple present*> <*adjective*>
<*verb, imperative*> <*conjunction*> <*noun, singular, subject case*> <*auxiliary verb, 3rd person singular, simple present*> <*noun, singular, object case*>	<*noun, singular, subject case*> <*auxiliary verb, 3rd person singular, simple present*> <*noun, singular, object case*>

During the modeling process, the heuristic presented in the preceding section assures the compliance with the conventions. As soon as a modeler enters a model element name, it is validated against the modeling conventions. For example, the modeler enters the activity name “bill is controlled”. At a glance, this name matches neither the domain vocabulary, nor the phrase structure conventions. The heuristic analyzes the phrase as follows: First, the name phrase is parsed via an HPSG-based linguistic syntax parser (cf. Figure 8).

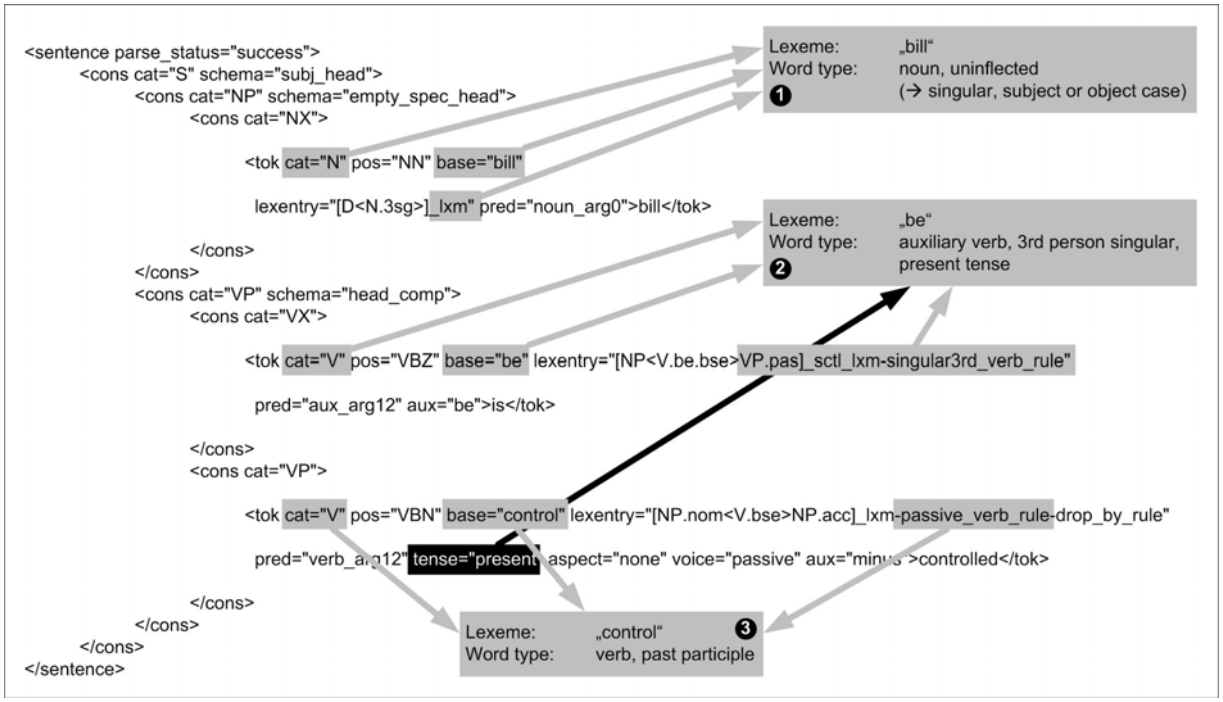

Figure 8: Exemplary Parsing Result

Besides the sentence structure, the parser reveals the lexeme, the word class, and the inflection of each of the used words. Here, the following words are detected:

1. the noun “bill” being uninflected, meaning singular and either subject or object case
2. the auxiliary verb “be”, 3rd person singular, present tense
3. the verb “control” in passive mood, meaning inflected as past participle (please note that the used parser returns the tense of the auxiliary verb through the attributes of the verb in passive mood)

In a next step, the heuristic checks whether the detected lexemes comply with the domain thesaurus. Neither the noun “bill” nor the verb “control” is contained in the thesaurus. As a consequence, the heuristic searches for appropriate synonyms in a general lexicon (cf. Figure 9). In the example, we have used the general English thesaurus provided by Merriam Webster, which is available through an online interface and is therefore suited well for the illustration of this example (Merriam Webster 2009). Note that, in our modeling prototype, we use another lexicon provided by WordNet, which can be accessed efficiently through a native offline interface but is less suited for illustrating this example. The synonyms found are compared to the lexemes contained in the domain thesaurus. The matches found are “invoice” and “check”. The auxiliary verb “be” does not have to be validated since auxiliary verbs do not contain any domain semantics but are only necessary to comply with the syntax of a natural language.

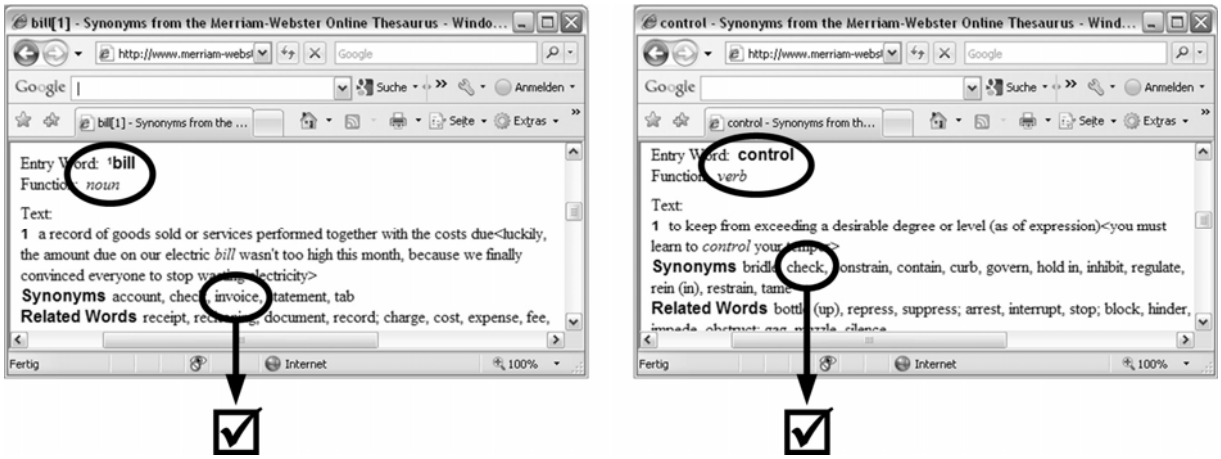

Figure 9: Retrieving Valid Synonyms

Third, the phrase structure used in the model element name is validated against the phrase structure conventions. Since it is not compatible, the phrase types specified within the conventions are now “filled” with the validated lexemes successively. Possible results are shown in Figure 10. The lexemes “invoice” and “check” do not have to be inflected due to the phrase structures. The auxiliary verb “be” is inflected to its 3rd person singular.

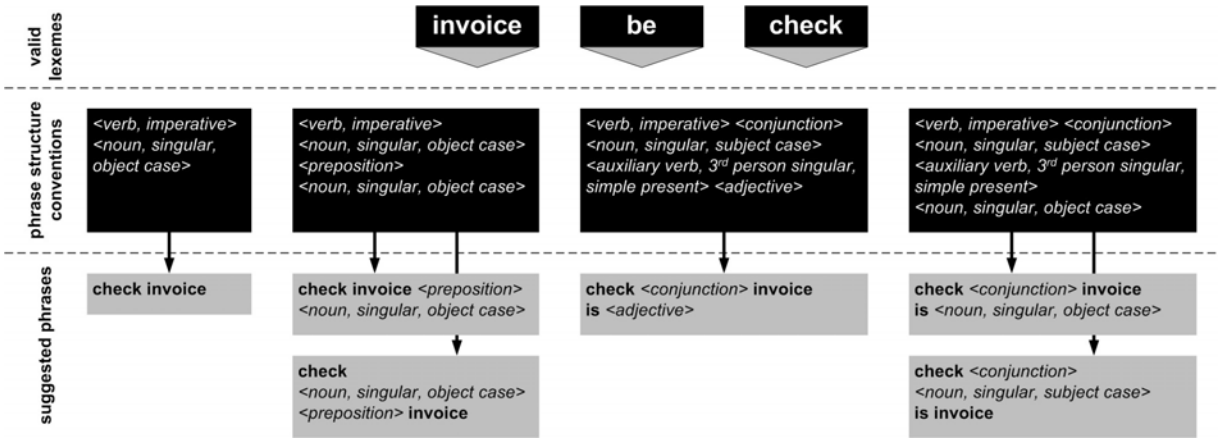

Figure 10: Suggestion of Valid Phrases

The modeler can choose from the suggested phrases and has to complete them, if s/he chooses an incomplete one. In the latter case, the parsing process starts again. If the modeler chooses a complete phrase, the model element name is valid. As a result, the invalid phrase "bill is checked" is for example replaced by the valid phrase "check invoice". If the modeler chooses an incomplete phrase, s/he has to complete it and the heuristic starts from the beginning (e.g., the second phrase suggested in Figure 10 is completed to "check invoice against price"). Of course, this can cause multiple review cycles.

Should the modeler not agree with the suggestions of the heuristic, s/he has to claim adding a term to the domain thesaurus. Then, s/he has to specify the term and provide a description in order to make the new term understandable for the modeling expert team, who decide whether or not the new term is accepted. In this particular example, the modeler may claim to add the new term "to control", which may – in her or his opinion – express a very rigorous check being different from a common one.

The graphical user interface of our research prototype provides the user with according hints and a drop-down list containing the suggested phrases (cf. Figure 11).

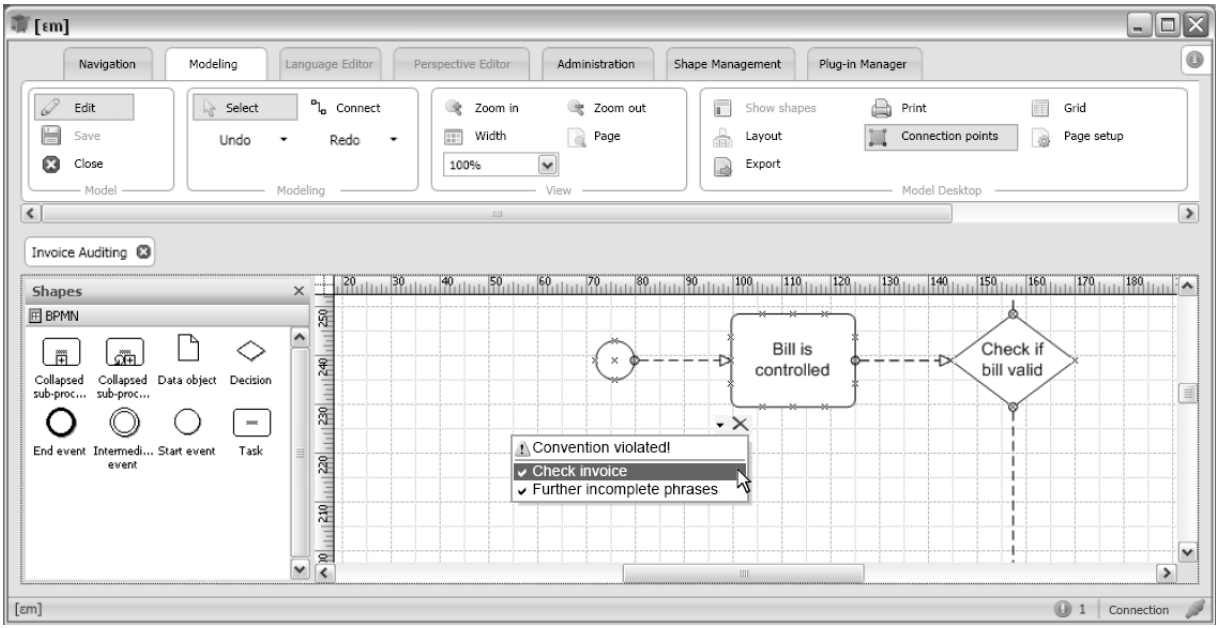

Figure 11: Automatic Guidance in Order to Comply with Naming Conventions

Conclusion and Outlook

Integrating naming conventions into conceptual modeling languages is promising for increasing the comparability of conceptual models and avoiding ambiguities in knowledge representation. Two characteristics are significant to avoid common problems:

- Defining and providing naming conventions previously to modeling is the basis for avoiding naming conflicts rather than resolving them. Therefore, time-consuming ex-post linguistic unification becomes dispensable.
- Guiding the modeler automatically during modeling is of substantial importance, since we can assure the compliance with the modeling conventions and, thus, the corporate language, only this way.

Thus, the main contribution to practitioners is that modeling tools can be extended in order to streamline distributed modeling projects concerning time and costs. However, analyzing cost-benefit of the approach, several issues have to be taken into account, which seem to decrease the benefit of the approach at a first glance. First, the naming conventions have to be specified, and this can be time-consuming. Second, the modeling process itself could be slowed down through the interventions of the heuristic. We found that defining the conventions is indeed quite time-consuming. Taking into account that in most modeling projects modeling conventions are defined anyway (although in absence of methodical support), and corporate glossaries can be reused in parts in the most cases, this cost issue is

moderated. Moreover, once the modeling conventions are defined, they are reusable for further projects in the regarded organization or domain. Looking at the modeling process itself, the suspected slow-down turned out not to be significant. The execution of the heuristic including parsing, looking up synonyms and inflection is fast enough not to be recognized by a modeler. After some first application experiments we have conducted, modelers expressed that they were not slowed down by the heuristic, but even sped up, because they did not need to think about modeling conventions or look them up in modeling standards any more. The problem of incomplete phrase suggestions already outlined in the preceding section turned out to occur only rarely. This means that the suggested complete phrases were selected in most cases, or the incomplete phrases could be completed easily. Furthermore, we could not observe any change requests. However, the most promising aspect of the approach is that the linguistic alignment of models subsequent to the modeling process can be completely abandoned, since the modeler has no chance to use a different language than the standardized corporate language. In our experience, this is the most time-consuming part of modeling projects not being supported by formalized modeling conventions.

Summarizing, first application experiments showed that the approach is promising, as long as it is possible to provide every modeler with the naming conventions. The modeling experiments we have conducted are able to get a first understanding of the benefit of the approach, but they have to be extended to a significant population of modelers to be able to score it precisely. Such extended experiments will reveal as well, whether or not exception handling plays a significant role concerning time and costs in real-world scenarios. In particular, extended experiments will consist of with-without tests combined with an analysis concerning the effort that has to be spent until a comparable and consistent model base is created.

As already noted, the approach works only if the naming conventions as well as the methodical support to comply with them can be provided to all involved modelers. It is then suitable for any type of modeling project, no matter whether or not distributed in terms of time, place, or person. If this precondition is not fulfilled, the approach is not suitable. For instance, this is the case in situations where models, which already exist, are to be aligned linguistically. In such a case, the approach fails, because the meaning of the model element names originally intended by the modelers of the already existing model cannot be determined exactly. Here, the only way to align these models is to discuss their contents involving all of their modelers.

Our main contribution to research is the combination of the two so far separated research areas of conceptual modeling and computational linguistics. We have introduced an approach to enhance the comparability of conceptual models that differs considerably from well-known and popular ontology-driven approaches and that is based on differing premises. Therefore, it enhances previous research results by closing the methodical gaps outlined in the "Related Work" section. Furthermore, standardization of model element naming in conceptual models opens new research opportunities, for instance in the area of automated model analysis, model comparison or model integration. The automated translation of models from one natural language into another one, which is nearly impossible without standardization of their naming, is a further very promising area of research. Finally, applying concepts from computational linguistics in conceptual modeling permits a new field of application and the evaluation of these concepts.

Our future research will focus on further evaluating the proposed approach. In the short-term, the approach will be instantiated for different modeling languages, different natural languages and different application scenarios. In particular, the capability of our approach to increase the efficiency of distributed conceptual modeling and its acceptance will be evaluated. In order to assure the applicability of the approach, the demonstrator software will be enhanced in order to make it usable in real-world projects. It has to be evaluated on a larger scale to what extent naming conventions can improve the knowledge representation process. In the course of evaluation, it will also be investigated if semantic ambiguities play a role in model element names. For example, the sentence "They hit the man with a cane" is semantically ambiguous, even if the meanings of all of the used words are considered definite. Thus, we will perform further studies on existing conceptual models and determine if phrase structures promoting ambiguities are common in conceptual models. 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 approaches in order to facilitate the comparison of models itself, for instance in order to find semantically equivalent structures in different models. Here, exclusive matching of model element names is not sufficient, since equal elements can be part of different structures. The comparison of models based on the model structure in combination with the involved model element names is therefore a promising research area.

References

Automotive Thesaurus 2009. *Automotive Thesaurus*, http://automotivethesaurus.com [2009-05-04].

Batini, C., and Lenzerini, M. 1984. "A Methodology for Data Schema Integration in the Entity Relationship Model," *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, Germany.

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., Godart, C., Nancy, France, 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., and 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, Greece.

Delphin. 2009. "Deep Linguistic Processing with HPSG," http://www.delph-in.net/index.php. [2009-05-04].

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, Australia.

Frank, U., and 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: Galindo, F., Takizawa, M. and Traunmüller, R., Zaragoza, Spain, 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: Guarino, N., Trento, Italy, 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: Österle, H., Schelp, J., and Winter, R., St. Gallen, Switzerland, pp. 1620-1631.

ISO 1982. ISO/TC97/SC5/WG3: *Concepts and Terminology for the Conceptual Schema and the Information Base*.

Kalpic, B., and Bernus, P. "Business process modelling in industry – The powerful tool in enterprise management," *International Journal of Computers in Industry* (47:3), pp. 299-318.

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, Portugal.

Kugeler, M. 2000. *Organizational Design with Conceptual Models. Modeling Conventions and Reference Process Model for Business Process Reengineering [in German: Informationsmodellbasierte Organisationsgestaltung. Modellierungskonventionen und Referenzvorgehensmodell zur prozessorientierten Reorganisation]*, Logos, Berlin, Germany.

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, NV, USA.

Loucopoulos, P. and Kavakli, E. 1999. "Enterprise Knowledge Management and Conceptual Modelling," in *Selected Papers From the Symposium on Conceptual Modeling, Current Issues and Future Directions*, Editors: Chen, P. P., Akoka, J., Kangassalo, H., and Thalheim, B., Lecture Notes In Computer Science, vol. 1565. Springer, London, UK, pp. 123-143.

Maedche, A., Motik, B., Stojanovic, L., Studer, R., and Volz, R. 2003. "Ontologies for Enterprise Knowledge Management," *IEEE Intelligent Systems* (18:2), pp. 26-33.

Merriam Webster. 2009. *Merriam Webster Online Search*, http://www.merriam-webster.com. [2009-08-26]

Mitkov, R. 2003. *The Oxford handbook of computational linguistics*, Oxford University Press, Oxford, UK.

Motik, B., Maedche, A., and Volz, R. 2002. "A Conceptual Modeling Approach for Building Semantics-Driven Enterprise Applications," in *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics (ODBASE-2002)*, Springer, Berlin, Germany, pp. 1082-1099.

OMG 2009. *Unified Modeling Language (OMG UML), Infrastructure, V2.1.2*, http://www.omg.org /docs/formal/07-11-04.pdf, [2009-09-09].

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 *Studies in Contemporary Linguistics*. University of Chicago Press, Chicago, IL, USA.

Preece, A., Flett, A., Sleeman, D., Curry, D., Meany, N., and Perry, P. 2001. "Better Knowledge Management through Knowledge Engineering," *IEEE Intelligent Systems* (16:1), pp. 36-42.

Rahm, E., and Bernstein, P. A. 2001. "A Survey of Approaches to Automatic Schema Matching," *The International Journal on Very Large Data Bases* (10:4), pp. 334-350.

Rosemann, M. 1996. *Complexity Management in Process Models. Language-specific Modeling Guidelines [in German: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung]*, Gabler, Wiesbaden, Germany.

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., Springer, Berlin, Germany, 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, MN, USA.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., and Wielinga, B. 2000. *Knowledge engineering and management: the CommonKADS methodology*, MIT Press, Cambridge, MA, USA.

Tradeport. 2009. *Tradeport. Reference Library for Global Trade*, http://tradeport.org/library. [2009-05-04].

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.

White, S. A., and Miers, D. 2008. *BPMN Modeling and Reference Guide. Understanding and Using BPMN*, Future Strategies Inc., Lighthouse Point, FL, USA.

WordNet. 2009. *WordNet. A lexical database for the English language*, http://wordnet.princeton.edu. [2009-05-04].

WWW Virtual Library. 2009. *Logistics*, http://logisticsworld.com/logistics/glossary.htm. [2009-05-04].