Patrick Delfmann, Sebastian Herwig, Łukasz Lis, Armin Stein

Supporting Distributed Conceptual Modelling through Naming Conventions

A Tool-based Linguistic Approach

Empirical studies attest that conceptual models created in distributed modelling environments often vary heavily in the way their respective model elements are labelled. Although the same issues are being modelled, different names are chosen by the involved persons. By this, the analysis and comparison of the models, which is required for their subsequent integration, is extremely challenging and time consuming. Literature analysis reveals several approaches addressing this problem by either manually or semi-automatically integrating existing models after their construction. However, this proves to be an exhaustive and error prone task. In this article we propose a domain and modelling language independent approach that prevents the emergence of naming conflicts already during the modelling process. This is done by formalising naming conventions consisting of context specific thesauri and customised phrase structures, which are both derived from natural language grammars and supplemented by domain-specific terms. These conventions serve as basis for a fully automated guidance of the modeller during the model creation process, resulting in semantically comparable conceptual models. For this, we present a research prototype that integrates our approach into a modelling tool.

1 Introduction

Empirical studies indicate that especially those conceptual models, which were constructed in a timely and regionally distributed way can vary heavily concerning terms and structure (Hadar and Soffer, 2006). Thus, naming conflicts and structural conflicts (Batini et al., 1986; Lawrence and Barker, 2001) may occur, even if the same issue was modelled. Furthermore, such variations occur in models created by only a single person as well. Consequently, the analysis of conceptual models (e.g., in reorganisation projects) may be extremely laborious. Information that is expressed differently has to be *normalised* in some way in order to make the models comparable. This usually requires discussions including all involved modellers in order to establish a consensus. Moreover, even additional external consultants are involved (Phalp and Shepperd, 2000; Vergidis et al., 2008).

To solve these deficiencies, approaches addressing the problem of model comparability are required. In the literature, many contributions propose the resolution of conflicts in conceptual models after modelling (cf. Subsection 2.1.1). Unlike these approaches, the goal of this article is to introduce an approach that ensures the comparability of conceptual models by avoiding potential conflicts already during modelling. This way, we avoid specific problems related to the ex post resolution of conflicts and make the normalisation process described above dispensable. For this, we provide means for the specification of naming conventions for elements of modelling languages. Through automated guiding integrated in a modelling tool, we are able to avoid violations directly at the very moment they can occur. The conventions are set up using domain terms and phrase structures that are defined as valid in a given modelling context. As a formal specification basis, we use a thesaurus defining nouns, verbs, and adjectives allowed in

the context. To provide conventions for phrase structures, we make use of a linguistic formalisation approach. During modelling, model element names are validated just in time against both the term and phrase structure conventions. Our approach is generic and can be applied to any conceptual modelling language. In our opinion, it is suitable for settings, in which naming conventions can be established and formalised. This process might take place on a peer to peer basis (e.g., in modelling communities) or have a top down form (e.g., in hierarchical, corporate settings).

The remainder of this article is structured as follows. In the next section, we present the foundations of our approach consisting of a literature analysis, an explorative empirical study, and the followed research methodology. In Section 3, we introduce a conceptual framework for the specification and enforcement of naming conventions as well as present our proposed procedure model. The subsequent section introduces the issue of linguistic parsing, which is essential for a feasible implementation of our approach. We provide a brief analysis of selected parsers and describe the modelling tool support in an application scenario. In Section 5, we conclude the article and discuss further research options.

2 Foundations

2.1 Related work

Resolving naming conflicts in conceptual models has been a research problem existing for the last few decades. A number of approaches propose means for the naming conflicts resolution in different areas of application (cf. Figure 1). They can be classified into two dimensions: On the one hand, there are approaches dealing with the problem either ex post or ex ante. Ex post approaches face the problem by analysing existing models, identifying naming conflicts and trying to solve them. Ex ante approaches aim at preventing the emergence of naming conflicts by restricting the modeller. On the other hand, approaches act on different linguistic objects, varying in their inherit complexity: Either they regard single terms or a respective assembly, called phrase.

2.1.1 Ex post approaches aiming at single terms

The integration of databases during the 1980s and 1990s lead to problems concerning the labelling of tables in database schemas. Even though, the databases serve the same or similar purpose, their tables and columns were named differently (e.g., address vs. contact details, see Batini and Lenzerini (1984), Batini et al. (1986), Bhargave et al. (1991), Lawrence and Barker (2001), or Rahm and Bernstein (2001)). Naturally, these approaches focus on data modelling languages in general and the Entity Relationship Model (ERM) notation and its dialects (Chen, 1976) in particular. Certain tools are presented, which are able to analyse given schemas and identify possibly matching fragments. Commonly, the intensity of correspondence is represented by a numerical measure. However, the necessity to involve domain experts to judge on the actual correspondence is explicitly stated. As the authors of the approaches solely present their solution for single nouns, they are not applicable for conceptual models, where labels of model elements often consist of sentence fragments containing terms of any word class (e.g., process models).

Scope / Time of application	Single terms	Phrases
Ex post	Batini and Lenzerini (1984); Batini et al. (1986); Bhargave et al. (1991); Lawrence and Barker (2001); Rahm and Bernstein (2001)	Ehrig et al. (2007); Höfferer (2007); Koschmider and Oberweis (2005); Sabetzadeh et al. (2007)
Ex ante	Born et al. (2007); Bögl et al. (2008); Greco et al. (2004); Rizopolous and McBrien (2005); White and Miers (2008)	Kugeler (2000); Kugeler and Rosemann (1998); Rosemann (1996, 2003)

Figure 1: Classification of approaches

2.1.2 Ex post approaches aiming at phrases

These approaches not only take single words as labels for model elements into consideration, but also concepts. These concepts consist of terms that are part of a domain ontology and thus interconnected. By this, nouns like *invoice* or *message* can be linked with verbs like *check* or *send*. Höfferer (2007) connects the labels of elements in existing models to a domain ontology, thus creating a model spanning understanding of the elements. Element labels referring to the same concepts in the ontology can be assumed being identical. Ehrig et al. (2007) propose similarity measures combined of a semantic and a syntactic part. These measures provide a basis for deciding whether model elements are equivalents or not (Koschmider and Oberweis, 2005; Sabetzadeh et al., 2007). All the approaches have in common that existing models have to be connected to the domain ontology or it has to be manually judged whether the rated elements are identical or not. It has to be questioned if the required ex post efforts are justified in comparison to a fully manual adaptation.

2.1.3 Ex ante approaches aiming at single terms

Naming conventions provide means to limit the probability of using the wrong terminology during the modelling project in advance. Commonly, naming conventions are provided as written glossaries or as ontologies (Gruber, 1993; Guarino, 1998; Preece et al., 2001), containing terms that are suitable for the regarded domain. Assuming that a *generally accepted* ontology exists, describing a certain modelling domain, Greco et al. (2004) propose the manual adaptation of ontology terms for process models. Naturally, as every manual activity is again prone to error, this method has to be questioned in terms of reasonableness. Born et al. (2007) take a step forward by providing means for a semi automatic adoption of model element names from a domain ontology. There, only those terms are allowed, which are explicitly written down. However, their approach is limited to BPMN models (White and Miers, 2008). Furthermore, 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 labels on 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. Finally, it has to be questioned whether the fact that two modellers share the same business domain guarantees a common understanding of business terms. In our opinion, the modellers are only then able to make use of this *generally accepted* ontology, if they are assisted during building it in the first way. Otherwise, the *general acceptance* itself cannot be guaranteed. Automated approaches are provided by Rizopolous and McBrien (2005) and Bögl et al. (2008), who propose the use of online dictionaries like for example WordNet (Fellbaum, 1998). Model elements are then connected to the corresponding entries of those dictionaries. However, in this case it is still not clear how to differentiate between homonyms or how to choose the desired term out of two or more synonyms.

2.1.4 Ex ante approaches aiming at phrases

Mainly originating from the German speaking area during the 1990s, standardised phrase structures are provided as means for the generation of consistent model element names. Rosemann (1996) and Kugeler (2000) use these as guidelines for the labelling of process activities in Event Driven Process Chains (EPCs, Scheer, 2000). For example, the rule <verb, imperative> <noun, singular> restricts the label to a term like *check invoice*. Thus, only those phrases are allowed, that can be built in conjunction with the terms of a technical term

model (Kugeler and Rosemann, 1998; Rosemann, 2003), which has to be generated before the beginning of the modelling project. Although the term rules involve phrase structures, the technical term model is limited to nouns, thus not flexible enough to satisfy the requirements of complex modelling languages like the ones required for process modelling.

Actually, the proposed approaches are, by themselves, promising in regard to improving the comparability of conceptual models. However, methodical realisation as well as the respective IT support are still missing.

2.2 Naming practice in process models

The problem of naming conflicts in conceptual models becomes evident especially when looking at process models. In our opinion, this class of models is extra prone to naming conflicts, as process model elements are usually named with sentence fragments rather than with single terms. We have conducted an exploratory empirical analysis of an exhaustive English model base trying to verify this proposition. The analysed sample originates from two medium sized reorganisation projects, conducted at a Russian financial institution and a German retailer. It consists of overall 257 EPC models containing in turn overall 3,918 elements (1,827 activities and 2,091 events). In the corresponding modelling project, naming guidelines were available in terms of a corporate glossary and suggested phrase structures. However, these guidelines solely existed as textual recommendations.

In our analysis, we applied an automated part of speech (POS) tagging using TreeTagger (Schmid, 1994) to all activities' and events' names. In the second step, we revised the returned results manually and conducted punctual detailed exploratory analyses to identify common problems. The results show that, first, most elements are named with sentence fragments rather than with single terms (cf. Figure 2).

Second, element names containing a certain number of terms consisted of many different phrase structures. Examples for two word activity names are <verb, imperative> <noun, singular>, in particular *audit invoice* or <noun, singular> <verb, gerund>, in particular *invoice auditing* (cf. Table 1).

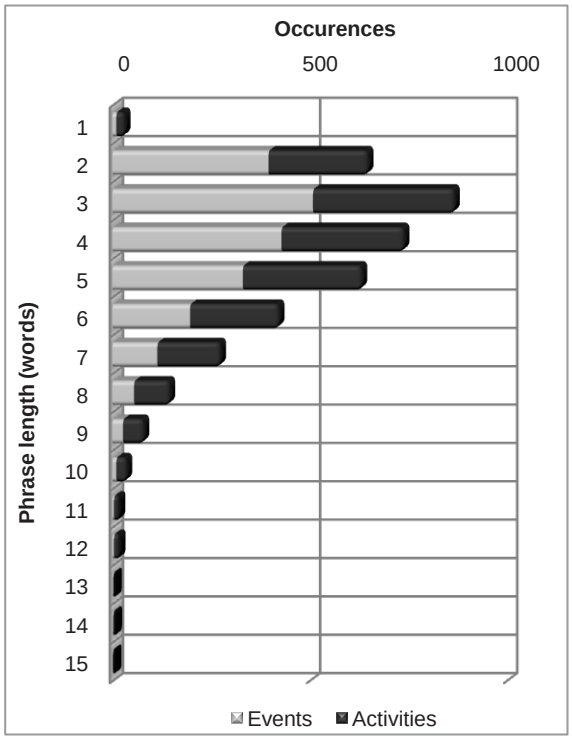

Figure 2: Occurrences of different phrase lengths used in process model element names (English model base)

At a glance, the great number of different phrase structures seems to promote inconsistencies in model element naming. A more detailed examination of the phrases supports this assumption. We could reveal the following common problem groups:

Synonyms Many phrases contained synonymous words and synonymous abbreviations. These synonyms cause ambiguities, which are hard to solve.

Meta information Several phrases contained information not belonging to the process seman-

tics. This means that, for example, structural information about the process itself was included in the model element name (e.g., *notification sent END OF PROCESS*).

Element type violations Some events were labelled as activities and vice versa (e.g., *order has to be dispatched* as a name for an activity).

Phrase structure inconsistencies Despite naming conventions, which were available in the construction process of the models, different phrase structures were used to express the same issue (e.g., *check if description necessary* vs. *verify necessity of description*).

Aggregated names Some elements showed combined names, such as *include remarks and send final version to proxy*.

Input errors These errors consisted of general spelling or typing errors (e.g., *checkinvoice*).

# of words	Events		Activities	
	# of events	# of phrase structures	# of activities	# of phrase structures
1	10	6	21	3
2	396	37	252	29
3	509	136	358	85
4	429	221	310	157
5	331	248	301	204
6	197	175	225	193
7	114	102	160	141
8	55	54	90	87
9	27	26	52	52
10	10	10	26	25
11	4	4	12	12
12	4	4	13	13
13	2	2	2	2
14	2	2	3	3
15	1	1	2	2
Sum	2,091	1,028	1,827	1,008

Table 1: Phrase structures in process model element names (English model base)

All these problematic naming practices compromise the understandability, the comparability, and the consistency of the models. In the last resort, such naming practices might lead to uselessness of the models. In order to extend the analysis and to stabilize the findings, we conducted an analogous analysis of an even larger German model base consisting of 4,805 process models, which contained 13,381 events and 13,935 activities. The sample originates from a large reorganisation project conducted at a large German governmental institution. Despite coming from another natural language, the results are quite similar (cf. Figure 3):

The chart in Figure 3, showing the distribution of phrase structure lengths in the German model base, is similar to the one depicted in Figure 2 (the English model base). The distribution of different phrase structures is analogous as well (cf. Table 2). Moreover, we could reveal the same problem categories as in the first analysis.

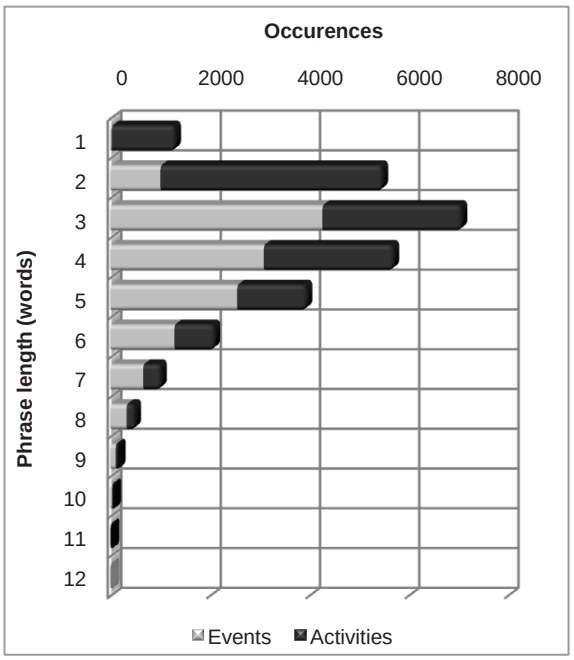

Figure 3: Occurrences of different phrase lengths used in process model element names (German model base)

Hence, we argue that approaches towards resolving or avoiding naming conflicts in process models have to consider not only the terms but also the phrase structures used in model element names.

In our opinion, considering both aspects allows to cover the complete linguistic scope of model element names and support linguistic unification through avoiding naming conflicts already during the modelling process.

# of words	Events		Activities	
	# of events	# of phrase structures	# of activities	# of phrase structures
1	26	6	1,276	4
2	1,009	22	4,466	40
3	4,271	144	2,799	119
4	3,091	330	2,605	286
5	2,555	549	1,394	398
6	1,291	566	802	399
7	662	445	355	273
8	327	255	168	142
9	107	97	48	44
10	34	33	18	18
11	5	5	4	4
12	3	3	0	0
Sum	13,381	2,455	13,935	1,727

Table 2: Phrase structures in process model element names (German model base)

2.3 Research methodology

The design science approach (Hevner et al., 2004) provides guidelines on how conduct research on relevant matters in a rigorous way. For this, the research should deal with the construction of scientific artefacts 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 artefacts to be constructed have to represent an innovative contribution to the existing knowledge base within the actual research discipline. Similar or identical solutions must not be already available. Subsequent to the construction of the artefacts, they have to be evaluated in order to prove their fulfilment of the research goals. The research methodology followed in this paper complies with the one described above.

Here, 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 (cf. Section 1 and Subsection 2.2). Related work does not provide satisfactory solutions up to now (cf. Subsection 2.1.1). Hence, the approach presented here (cf. Section 3) makes an innovative contribution to the existing knowledge base. In order to evaluate our artefact, we developed a research prototype that shows the general applicability of the approach (cf. Section 4). Further evaluations concerning acceptance as well as efficiency and increase of comparability will be the subject of empirical studies to be performed in the short term (cf. Section 5).

3 A Framework for the Specification and Enforcement of Naming Conventions

3.1 Procedure model

Our approach proposes the usage of a modelling project specific grammar, consisting of both, adjusted phrase syntax as well as a specialised set of allowed terms (cf. Figure 4).

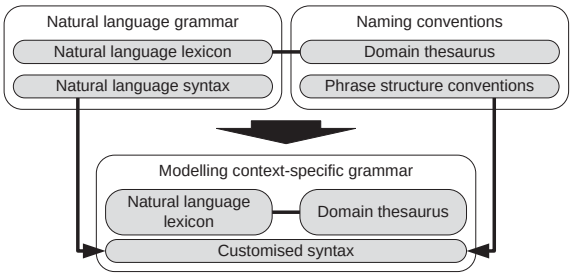

Figure 4: Specifying a context specific grammar

For this, besides the existing natural language grammar consisting of a natural language lexicon and natural language syntax (Kaplan, 2003), naming conventions have to be established. Similarly to the natural language grammar, the domain specific naming conventions consist of a domain thesaurus on the one hand and phrase structure conventions on the other hand. The domain thesaurus holds terms that are explicitly demanded by the modelling team. The terms can either ori-

gin from the natural language lexicon or, if not locatable there, explicitly be created. This might happen if brand names, artificial terms, or specific (e.g., not official) abbreviations are being used.

For the modelling project, both grammars have to be integrated, thus creating a modelling context-specific grammar. The term conventions are settled by a thesaurus containing domain terms along with a precise declaration of their synonym, homonym, and word formation relationships as well as textual description of their meaning. The thesaurus is then connected to the natural language lexicon while valid phrase structures are specified by phrase structure conventions. Hence, the natural language is trimmed for the needs of a specific modelling context. This context specific grammar allows for subsequent validation of the model element names and the enforcement of naming conventions.

At the beginning of a modelling project, it has to be decided whether the thesaurus has to be created from scratch or if it is possible to reuse existing thesauri or glossaries (e.g., Automotive Thesaurus, 2009; Tradeport, 2009; Virtual Library, 2009). It then has to include single nouns, verbs and adjectives that are interrelated. As other word classes like articles, prepositions, pronouns, or conjunctions are generally domain independent, they do not need to be explicitly specified in the thesaurus as they are already included in the natural language lexicon. The terms in the thesaurus have to be 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. WordNet is such a lexicon service for the English language providing an online interface. Therefore, in case of a violation of the naming conventions by the modeller, synonymous or derived valid terms can be automatically identified and recommended. The terms specified are provided with short textual semantic descriptions, allowing modellers for looking up the exact meaning of a term. Furthermore, in the case of a synonym relation between terms, the dominant term has to be specified. Once built up, the thesaurus should not be changed during a modelling project in order not to violate the consistency of application.

As mentioned above, the naming conventions have to be specified once for every modelling context whereas already existing conventions can be reused and customised (cf. in the following Figure 5).

Naturally, naming conventions are modelling language-specific. For example, activities in EPCs are labelled with actions (e.g., <verb, imperative> <noun, singular>; in particular e.g., *check invoice*) and events are labelled with states (e.g., <noun, singular><verb, past participle>; in particular e.g., *invoice checked*) (Scheer, 2000). These specifications are however not suitable for modelling languages that, for example, do not require process information, like organisational charts. Here, a single noun might be sufficient. Nevertheless, for each model element type at least one phrase structure convention has to be defined in order to prevent unlabelled elements. The definition of the conventions should be performed by a project team consisting of both, domain experts and modelling experts, hence the stakeholders responsible for the conventions should have thorough knowledge of the actual modelling context. If this is ignored, the modellers will be too much distracted from modelling by the time required for finding the correct term.

During the modelling process, the model element names entered 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 customised 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 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, al-

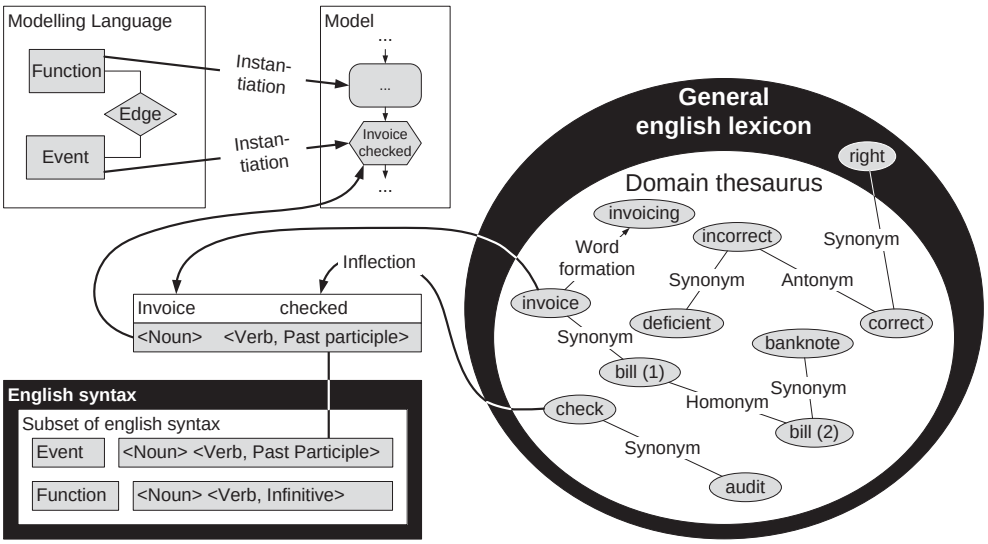

Figure 5: Application of formalised naming conventions

ternative valid phrase structures and/or terms are suggested based on the user's input. The modellers themselves have to decide, which of the recommendations suits 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 on their own. However, it should be possible for the modeller to propose a new term with a short textual semantic description. As one solution and in order not to distract the modeller from his current modelling session, the proposed term might be accepted temporarily. In a next step, it is 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. By this, we ensure that equal model element names represent equal semantics, which is a precondition for the comparability of conceptual models.

3.2 Conceptual specification

Making a conceptual specification of naming conventions easily reusable for a database implementation, we apply Entity Relationship Models in (min, max) notation (ISO, 1982). The starting point for the specification is a model element type, for which a naming convention is to be applied (e.g., activities in EPC). Therefore, we define phrase structure conventions depending on element types (cf. Figure 6). Any phrase structure convention consists either of a single word type or a phrase type. The former represents a set of words belonging to a specific word class (noun, verb, adjective, adverb, article, pronoun, preposition, conjunction or numeral) and being specifically inflected. Inflections turn a word from its basic form into a form complying with the syntax of a sentence or phrase, such as case, number, tense, gender, mood, person, or comparative. Inflections are usually combined, for instance <3rd person, singular>. In respect to different word classes, not every inflection is possible. For example, a phrase structure convention consisting of a single word type could be <noun, singular>. An according phrase structure convention consisting of more than one word type could be <noun, singular><verb, past participle>.

Phrase structure conventions that contain more than one word type are built recursively via the phrase type structure. The latter specifies which sub-phrase type or word type belongs to which super-phrase type, and what position the sub-

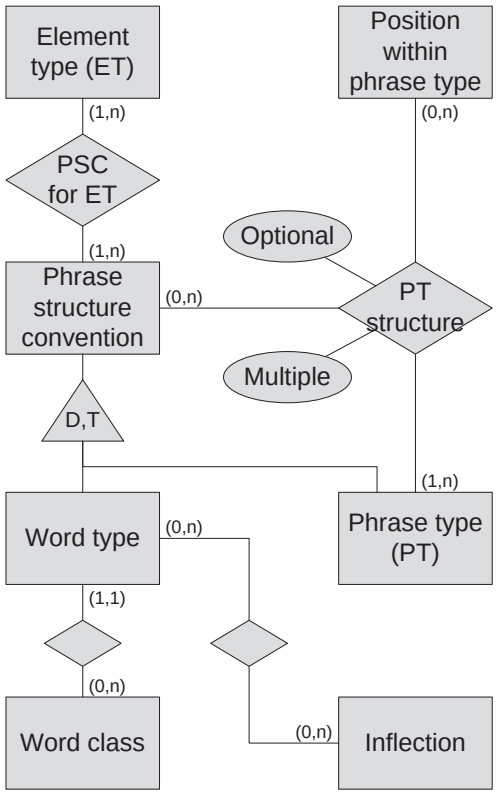

Figure 6: Specification of phrase structure conventions

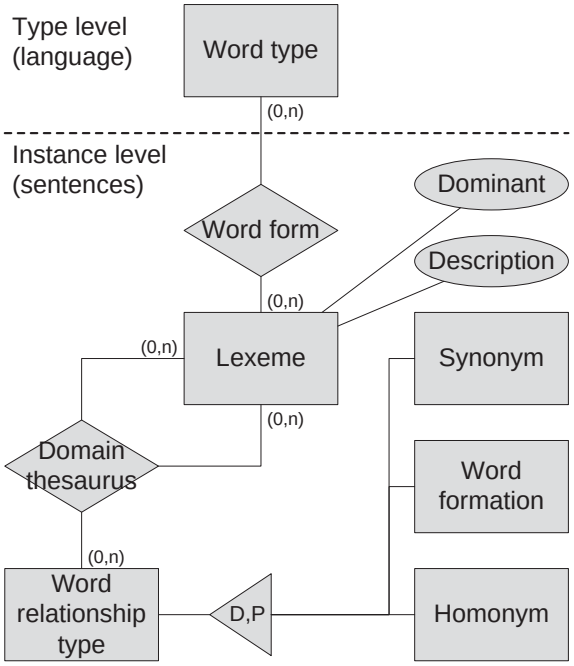

Figure 7: Specification of the domain language

phrase type or word type has in the super-phrase type. Some phrases require word chains, for instance noun chains (e.g., *phrase structure convention*). To allow such chains, we introduce the Boolean attribute multiple. If it is TRUE, the according word type or sub-phrase type is allowed to be repeated any number of times. Other word types or sub-phrase types are optional in a phrase structure convention. This is indicated by the Boolean attribute optional. Using this specification environment, any phrase structure can be defined. However, it should be noted that the conventions specified comply with the syntax of the underlying natural language.

The instantiation of phrase structures leads to particular phrases or sentence fragments containing particular inflected words. An inflected word is called word form and consists of its basic form – the lexeme – and its word type (cf. Figure 7). The words to be used in a modelling domain or project are stored in their lexeme form in the *domain thesaurus*.

Besides the valid lexemes, the domain thesaurus provides word relationships. These are homonym, synonym, and word formation relationships. Word formation means that a lexeme originates from (an)other one(s). 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 to prevent naming elements with the same string but with different meanings. Word formation relationships allow for searching valid alternatives, whenever a modeller uses invalid terms, which may have originated from valid ones. For example, if the phrase *order clearance* violates the naming conventions, alternative terms can be found via word formation relationships that may match the conventions (e.g., *clear order*).

Finally, a semantic description is added at least to each dominant lexeme to specify what is actually meant by a lexeme. This way, modellers are en-

abled to check whether the lexeme they have used actually fits the modelling issue.

4 Modelling Tool Support

4.1 Linguistic Parsing

The naming conventions specified according to the framework introduced above have to be actively enforced already during modelling to assure the compliance of the models with the conventions. Therefore, we need a measure to analyse the phrases entered by the modeller regarding their structure and their used terms. Analysing natural language fragments is the subject of the research discipline of computational linguistics. Several formalisms have been proposed which are able to represent phrases of natural languages in a formal way. An established and well known class of such formalisms are so called Unification Grammars (Kaplan, 2003). These consist of feature structures describing both the syntactic role of words or phrases and the base form of the used words, meaning their lexemes. Exemplary unification grammars are the Head-Driven Phrase Structure Grammar (HPSG, Pollard and Sag, 1994), the Generalized Phrase Structure Grammar (GPSG, Gazdar et al., 1985), the Lexical Functional Grammar (LFG, Bresnan, 2001) and the Unification Categorical Grammar (UCG, Calder et al., 1988).

Software artefacts called linguistic syntax parsers exist, which take natural language phrases as input and provide an output, which complies with such a formal grammar. An exemplary parsing result in HPSG notation for the phrase *check invoice* returned by a linguistic parser is shown in Figure 8.

The parser determines the phrase <check, invoice> and characterises it as a verbal phrase. The verbal phrase consists of a so called head, the constituting element of the phrase and one or more subcomponents (*SUBCAT* <>). The sentence *check invoice* is fractionised into its components, which are, in this case, the lexemes *check* and *invoice*. The first com-

Parser	Characteristics
Stanford Parser (Klein and Manning, 2003)	• Several languages supported (e.g., English, German) • Fair performance • Integration opportunities through a .NET interface (Proxem, 2009) • Public licence model available • Standalone version available
Xerox Incremental Parser (XIP) (Xerox, 2009)	• Several languages available, however not German • Slow response time • Standalone version available • Online demo available • Result set provided in XML • No public licence, however research licence available
Connexor-Machinese Syntax (Connexor, 2009)	• Several languages supported (e.g., English, German) • Fair performance • Online demo available • Only online availability • Only commercial licence model available
TAGH (Geyken and Hanneforth, 2005)	• Only available for German • Online demo available • Web service interface available • Compound decomposition support
IPS (Wehrli et al., 1992)	• Several languages supported (e.g., English, German) • Online demo available • No public licence available • No identification of unknown words
Attribute-Logic Engine (ALE) (Penn and Carpenter, 1999)	• Limited to English • Limited integration opportunities • Public licence model • Only limited grammar • No further development
Babel (Müller, 1996)	• Limited to German • Online demo available • Limited integration opportunities • Public licence model • Only limited grammar (esp. lexicon) • Parsing of complete sentences only
Linguistic Knowledge Builder (LKB) (Copestake, 2001; Copestake and Flickinger, 2000)	• Limited performance • Limited integration opportunities • Public licence model • Parsing not in the focus • Integrated grammar development environment
PET (Callmeier, 2000)	• Several languages supported (e.g., English, German) • Fair performance • Limited integration opportunities • Public licence model
Enju (Hara et al., 2005; Miyao and Tsujii, 2005)	• Limited to English • Fair performance • No public licence, however research licence available
Stefs (Selj, 2000)	• Public licence model • Only a grammar prototype provided

Table 3: Characteristics of selected linguistic parsers

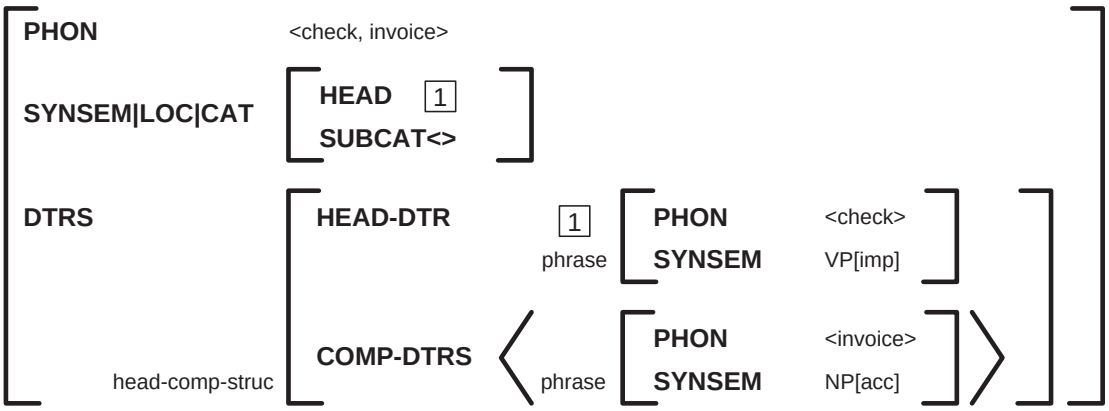

Figure 8: Exemplary parsing result

ponent <check> is identified as a verbal sub-phrase in imperative form (*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 fractionised. The component <invoice> is identified as a nominal sub-phrase in accusative case (*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.

Due to their well structured representation, such parsing results can easily be reused by formalisms, for instance by an algorithm that checks the phrase structure and the lexemes against naming conventions. However, not every linguistic parser is suitable for our approach to the same extent. We formulated five desired characteristics the parser should feature to be specifically applicable in the implementation of our method:

- support for at least English and German,
- good performance (short response time),
- good integration opportunities,
- comprehensive output,
- public (free) licence model.

We studied a number of different parsers to identify those best suiting our needs. The brief results of our analysis are shown in Table 3.

For the implementation of our approach, we decided to use the Stanford Parser (Klein and Manning, 2003). Similar to PET (Callmeier, 2000) it offers broad language support, good performance, and a public licence model. The critical advantages here were excellent integration potentials with both the modelling tool and WordNet, which are provided by Proxem (2009).

As shown in Figure 9, linguistic parsing allows us to decompose a given model element name into single terms and derive their uninflected forms (1). In a next step, we validate the lexemes against the domain thesaurus (2). Lexemes contained in the domain thesaurus are denoted as valid. For those lexemes that do not exist in the domain thesaurus, we search synonyms in the general lexicon and match them against the domain thesaurus (3). If no such synonyms are available or a lexeme is not contained in the general lexicon, we exclude them from further validation steps. Based on the defined structure conventions, we suggest possible model element names to the modellers that contain the valid lexemes in the appropriate inflection form (4). If a phrase structure is violated in turn, alternative but valid phrase structures are proposed that contain the valid terms.

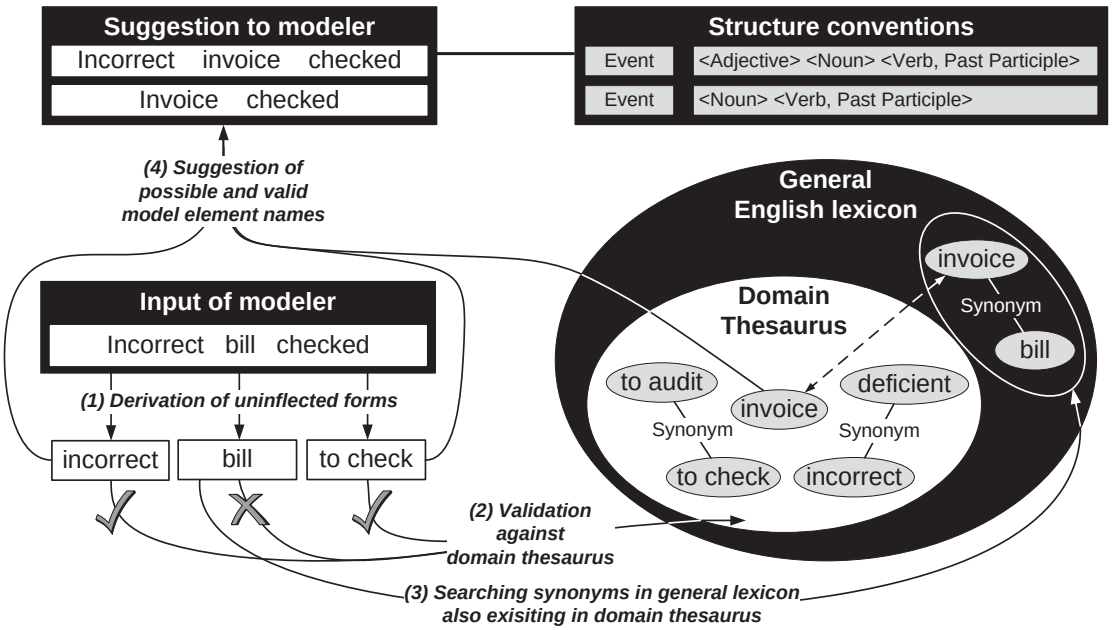

Figure 9: User input validation and generation of suggestions

4.2 Modelling Environment

In order to actively enforce naming conventions already during modelling, it is necessary to integrate our approach into a modelling tool. As described above, the connection of our approach with modelling languages requires the adoption of the respective meta model. For the tool support, this indicates the necessity of meta modelling abilities, which are featured by our research prototype. Hence, virtually any modelling language that can be created or is already part of the prototype can be extended with naming conventions. The software follows a fat client/thin server three tier architecture, thus enables distributed modelling. We chose widespread Microsoft Visio as the underlying drawing engine. We communicate with this engine using a generic interface realised through the Microsoft .NET Framework.

As a preliminary step in the application scenario, the person responsible for specifying the modelling conventions 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 addressed before, the existing set of terms and rules can be adapted to the current requirements. It is generally sufficient to add uninflected words, as the inflection can be automatically looked up in the lexical services.

Phrase structure conventions have to be assigned to those language elements for which they are valid. For example, it is necessary to create different phrase structure conventions for EPC events (i.e., separate conventions for trigger events and result events). The former represent states that trigger some activities and the latter represent states resulting from activities. Different phrase structures can be attached to each of them in regard to their different semantics.

An exemplary phrase structure convention for trigger events is <noun, singular><verb, passive infinitive> allowing for names like *invoice is to be checked*. For resulting events, an adequate phrase structure can be <noun, singular><verb, past participle>, allowing for phrases like *invoice checked*. Once generated, the phrase structure conventions in combination with the domain thesaurus are

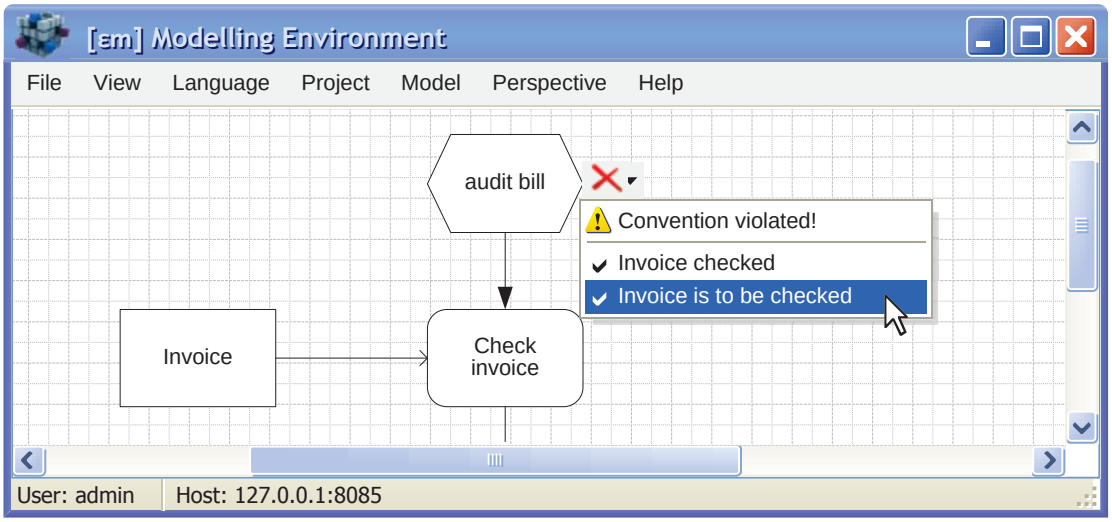

Figure 10: Automated guidance implemented in the modelling prototype

used during modelling. Modellers get suggestion as soon as they violate a convention (cf. Figure 10).

On the one hand, the modeller might have chosen terms, which are not allowed in the domain thesaurus (e.g., bill instead of invoice). Every entered phrase is parsed to determine its compliance with the conventions. Each term is brought to its uninflected form and compared with the domain thesaurus. If the term is not found, synonymous valid terms are searched in the natural language lexicon. If corresponding alternatives are found, they are included in the suggestions. In our example, for the phrase *audit bill*, the allowed terms would be *check* and *invoice*.

On the other hand, violations of phrase structure conventions are detected and alternative valid structures are proposed. In the example, only a term substitution would result in the phrase *check invoice*. However, the underlying phrase structure of <verb, imperative><noun, singular> is not allowed for events. Thus, similar but allowed structures are identified and included in the suggestions. Finally, the suggestions are presented to the modeller, who has to choose a valid option or provide a new name, which complies with the conventions.

To summarise the example, the modeller is automatically guided to change the name *audit bill* to *invoice is to be checked*. Names already complying with both the domain thesaurus and the phrase structure conventions are accepted without any feedback. This way, the modeller is actively assisted during the modelling process, thus naming conventions are enforced even before violations can occur.

5 Conclusion and Outlook

Integrating naming conventions into conceptual modelling languages the way presented here is promising for increasing the comparability of conceptual models and fills the gap presented in Subsection 2.1.1. We identified two characteristics that are significant to avoid common problems:

- Defining and providing naming conventions previously to modelling is the basis for avoiding naming conflicts in advance rather than having to resolve them afterwards. Therefore, time consuming alignment of labels becomes dispensable.
- Automatically guiding the modeller during the modelling process is of substantial importance,

since only this way the compliance with the modelling conventions can be assured. However, it has to be guaranteed that the guidance does not distract modellers from their core tasks.

Certainly, initially specifying naming conventions in the proposed way is an exhaustive and time consuming task. Thus, our approach is especially suited for large scaled regionally distributed modelling projects. Nevertheless, for every project, business domain, or company, the conventions need to be specified only once, as term models, thesauri and glossaries that may already exist in companies or business domains can be reused. Hence, one future research goal targets the creation of domain specific thesauri on the one hand and modelling language specific naming conventions on the other hand. Once created, they can be provided as starting point for domain and modelling specific customisation, what we expect to greatly reduce the required initial efforts.

Concerning the evaluation, the research prototype presented in Subsection 4.2 proves the technical feasibility of the approach. Although several software components were integrated into the modelling software, the achieved performance seems promising for a successful application in real life settings. Still, future research will continue focussing on further evaluating the proposed approach in certain dimensions. In the short term, we will instantiate the approach for different modelling languages, different natural languages, and different application scenarios. In particular, we are going to evaluate the capability of our approach to increase the efficiency of distributed conceptual modelling and its acceptance. For this, the efficiency of the approach will be observed in a laboratory experiment by comparing three different modelling groups. Given the same business case, each group has to model it in a distributed way, whereas one group has to model it without any guidance, the second one with paper based glossaries, and the third one with the respective software support. The insights from this experiment should help improving the prototype in a manner that its applicability in general and its responsiveness in particular may prove themselves in a real life scenario.

In a wider sense, we will also investigate whether ambiguities play a role in model element names. For example, the sentence *He sees the man with the binoculars* is ambiguous, even if the meanings of all 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 modelling. A result of this analysis could be a recommendation to restrict phrase structure conventions to phrases that do not lead to ambiguities.

References

Automotive Thesaurus (2009) Automotive Thesaurus. Website, URL http://automotivethesaurus.com, available online at http://automotivethesaurus.com; Visited on August 6th

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

Batini C, Lenzerini M, Navathe SB (1986) A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18(4):323–364

Bögl A, Kobler M, 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] (MKWI)

Bhargave HK, Kombrough SO, Krishnan R (1991) Unique Name Violations, a Problem for Model Integration or You Say Tomato, I Say Tomahto. ORSA Journal on Computing 3(2):107–120

Born M, Dörr F, 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) at the 8th International Conference on Web Information Systems Engineering (WISE), pp 260–271

Bresnan J (2001) Lexical Functional Syntax. Blackwell Publishers

Calder J, Klein E, Zeevat H (1988) Unification Categorial Grammar: a concise, extendable grammar for natural language processing. In: Vargha D (ed) Proceedings of the 12th Conference on Computational Linguistics, Morristown, vol 1, pp 22–27

Callmeier U (2000) PET — A Platform for Experimentation with Efficient HPSG Processing Techniques. Natural Language Engineering 6(1):99–108

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

Connexor (2009) Connexor: Machinese Syntax. Website, URL http://www.connexor.eu/technology/machinese/machinesesyntax/, available online at http://www.connexor.eu/technology/machinese/machinesesyntax/; Visited on August 6th, 2009

Copestake A (2001) Implementing Typed Feature Structure Grammars. Center for the Study of Language and Information

Copestake A, Flickinger D (2000) An Open-Source Grammar Development Environment and Broad-Coverage English Grammar using HPSG. In: Proceeding of the Conference on Language Resources and Evaluation, pp 591–600

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

Fellbaum C (1998) WordNet: An Electronic Lexical Database. The MIT Press

Gazdar G, Klein E, Pullum G, Sag IA (1985) Generalized Phrase Structure Grammar. Harvard University Press, Oxford

Geyken A, Hanneforth T (2005) TAGH: A Complete Morphology for German based on Weighted Finite State Automata. In: Finite State Methods and Natural Language Processing. 5th International Workshop, FSMNLP 2005, Helsinki, Finland, pp 55–66

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

Gruber TR (1993) A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2):199–220

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

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

Hara T, Miyao Y, Tsujii J (2005) Adapting a probalistic disambiguation model of an hpsg parser to a new domain. Natural Language Processing 3651:199–210

Hevner AR, March ST, Park J, Ram S (2004) Design Science in Information Systems Research. Management Information Systems Quarterly 28(1):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), pp 1620–1631

ISO (1982) Concepts and Terminology for the Conceptual Schema and the Information Base. Tech. Rep. ISO/TC97/SC5/WG3, International Organization for Standardization, New York

Kaplan RM (2003) The Oxford handbook of computational linguistics., Oxford University Press, chap Syntax, pp 70–90

Klein D, Manning CD (2003) Accurate unlexicalized parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, Sapporo, Japan, vol 1, pp 423–430

Koschmider A, Oberweis A (2005) Ontology Based Business Process Description. In: Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability, Co-located with CAiSE'05 Conference

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]. PhD thesis, University of Münster, Germany

Kugeler M, Rosemann M (1998) Modelling of Terms for Business Information Systems to Support Business Communication. [In German: Fachbegriffsmodellierung für betriebliche Informationssysteme und zur Unterstützung der Unternehmenskommunikation]. Informationssystem Architekturen Fachausschuss 52 der Gesellschaft für Informatik eV (GI) 5(2):8–15

Lawrence R, Barker K (2001) Integrating Relational Database Schemas using a Standardized Dictionary. In: Proceedings of the 2001 ACM symposium on Applied computing (SAC)

Miyao Y, Tsujii J (2005) Probalistic Disambiguation Models for Wide-Coverage HPSG Parsing. In: Proceedings of the 43rd Annual Meeting of the ACL, pp 83–90

Müller S (1996) The Babel-System: An HPSG Prolog Implementation. In: Proceedings of the 4th International Conference on the Practical Application of Prolog, pp 263–277

Penn G, Carpenter B (1999) ALE for Speech: A Translation Prototype. In: Proceedings of the 6th Conference on Speech Communication and Technology (EUROSPEECH)

Phalp K, Shepperd M (2000) Quantitative analysis of static models of processes. Journal of Systems and Software 52(2-3):105–112

Pollard CJ, Sag IA (1994) Head Driven Phrase Structure Grammar. Studies in Contemporary Linguistics, University of Chicago Press

Preece A, Flett A, Sleeman D, Curry D, Meany N, Perry P (2001) Better Knowledge Management through Knowledge Engineering: A Case Study in Drilling Optimisation. IEEE Intelligent Systems 16(1):36–42

Proxem (2009) Proxem. Website, URL http://www.proxem.com/Antelope/tabid/55/Default.aspx, available online at http://www.proxem.com/Antelope/tabid/55/Default.aspx; Visited on September 12th

Rahm E, Bernstein PA (2001) A Survey of Approaches to Automatic Schema Matching. The International Journal on Very Large Data Bases 10(4):334–350

Rizopolous N, McBrien P (2005) A General Approach to the Generation of Conceptual Model Transformations. In: Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSEŠ05)

Rosemann M (1996) Complexity Management in Process Models: Language-specific Modelling Guidelines [In German: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung]. PhD thesis, University of Münster

Rosemann M (2003) Process Management: A Guide for the Design of Business Processes, Springer Verlag, chap Preparation of Process Modeling, pp 41–78

Sabetzadeh M, Nejati S, Easterbrook S, Chechik M (2007) A Relationship-Driven Framework for Model Merging. In: Proceedings of the Workshop on Modeling in Software Engineering at the 29th International Conference on Software Engineering

Scheer AW (2000) ARIS — Business Process Modelling. Springer Verlag, Berlin

Schmid H (1994) Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Proceedings of the 1st International Conference on New Methods in Natural Language Processing, pp 44–49

Selj VK (2000) Stefy: Java Parser for HPSGs, Version 0.1. Tech. Rep. CS-99-26, Department of Computer Science, University of Waterloo

Tradeport (2009) Tradeport Ű Reference Library for Global Trade. Website, URL http://tradeport.org/library, available online at http://tradeport.org/library; Visited on August 6th

Vergidis K, Tiwari A, Majeed B (2008) Business process analysis and optimization: beyond reengineering. IEEE Transactions on Systems, Man, and Cybernetics 38(1):69–82

Virtual Library (2009) WWW Virtual Library: Logistics. Website, URL http://logisticsworld.com/logistics/glossary.htm, available online at http://logisticsworld.com/logistics/glossary.htm; Visited on August 6th

Wehrli E, Clar R, Merlo P, Ramluckun M (1992) The IPS system. In: Boitet C (ed) Actes du quinzième colloque international en linguistique informatique, pp 870–874

White SA, Miers D (2008) BPMN Modeling and Reference Guide: Understanding and Using BPMN. Future Strategies Inc., Lighthouse Point

Xerox (2009) Xerox Incremental Parser (XIP). Website, URL http://www.xeroxtechnology.com/ip1.nsf/sedan1?readform&unid=9C7EE64CFD78931585256FCD005C454D&nav=nav_cat_7, 2009-08-06

Patrick Delfmann, Sebastian Herwig, Łukasz Lis, Armin Stein

European Research Center for Information Systems (ERCIS)
University of Münster
Leonardo-Campus 3
48149 Münster
Germany
{delfmann | herwig | lis | stein}@ercis.uni-muenster.de