A Three-Tier Matching Strategy for Predesign Schema Elements

Peter Bellström
Department of Information Systems
Karlstad University
Karlstad, Sweden
e-mail: Peter.Bellstrom@kau.se

Jürgen Vöhringer
Institute for Applied Informatics
Alpen-Adria Universität Klagenfurt
Klagenfurt, Austria
e-mail: juergen.voehringer@ifit.uni-klu.ac.at

***Abstract*—Schema integration is a very complex task in which several schemata are merged into one global conceptual schema. Due to its complexity, computer-based applications and tools are needed that support and automate parts of the integration process. In our previous work, we have shown that schema integration on the predesign level allows for lower schema complexity, fewer conflicts and better end user feedback. In this paper, we focus on applied matching strategies, which are a central element of any semi-automatic integration process. We propose a set of matching methods that are suitable for the predesign level and discuss how they are intertwined and how their results regulate the integration process. As its main contribution, the paper offers an integration of previously presented methods and describes exemplary findings from the corresponding prototype.**

Keywords- predesign modeling; matching; integration

I. INTRODUCTION

When designing and developing information systems, we often have to deal with requirements, hereafter referred to as schemata, which are collected from different sources. These requirements are often divided into structural and behavioral schemata. In this paper, we focus on the structural aspect, meaning both what data should be stored in the database and what data the information system needs for processing its functionality. The application area is schema integration, a very complex task in which several conceptual schemata are merged into one global conceptual schema. In [3], the authors define schema integration as "the activity of integrating the schemas of existing or proposed databases into a global, unified schema" (p. 323). Due to complexity, computer-based applications and tools are needed in the integration process to help users not only to recognize but also to resolve similarities and differences between two source schemata.

In this paper, we mainly focus on the former of these: the recognition of similarities and differences. In doing so, we propose a three-tier matching strategy for predesign schema elements starting with an element level matching approach followed by structural level matching and ending with a taxonomy-based matching strategy. Our approach is rather unique since it is modeling-language independent, which means that the matched schemata can be formulated in any modeling language focusing on concepts and links between concepts. In our approach we focus on linguistic techniques to extract as much information as possible. Matching on the predesign level has various application areas, including

A. Integration of heterogeneous requirements during the early phases of information system development projects,
B. Consolidation of project schemata from a specific domain during ontology engineering.

Generally speaking, predesign matching is best used whenever natural language descriptions are available rather than more formal specifications (e.g., during requirements engineering), when semantic, project-overarching matching is needed (e.g., during ontology engineering) or when extensive user feedback by domain experts is expected or required [10][1]. In [26], we describe an experimental study that compared end user feedback for predesign models compared to feedback for standard conceptual models such as Unified Modeling Language class diagrams.

The paper is structured as follows: in section two, we address some related work and distinguish it from our own approach. In section three, we present our three-tier matching strategy consisting of element level matching, structural level matching and taxonomy-based matching. In section four, we show how these three tiers are interconnected and how the matching results are utilized for the purpose of schema integration. Finally, the paper closes with a summary and conclusions.

II. PREVIOUS AND RELATED WORK

Earlier work in the domain of schema integration might be roughly classified into three approaches [4]: manual, formal and semi-automatic approaches to schema integration. *Manual* means that all tasks are performed by hand, *formal* means in this context that a formal modeling language is applied and *semi-automatic* means that at least one computer-based application is used in the integration process. Looking at previous work, it can be concluded that much of that work has focused on the Entity-Relationship modeling language (ERML) [12] or some extension of it [25]. Lately, focus has shifted towards the Unified Modeling Language (UML) [20]. Even so, it should be noted that both the ERML and the UML are implementation-dependent modeling languages. In our approach, we instead work towards a method for modeling-language independent schema integration, meaning focus is on content instead of

 ISBN: 978-1-61208-121-2

implementation. In the rest of this section we give examples of semi-automatic approaches and distinguish them from our own approach.

In [23], the authors present a survey of approaches to automatic schema matching. They distinguish schema-based and instance-based matching. Our work is classified as a schema-based approach, since it is applied early in the information system development process in which schemata are focused. In [23], the authors further state that schema-based matching can be performed on the element level (concept) and on the structural level (neighborhood) and it can be either linguistic or constraint-based matching. Our approach is a composite schema-based matching approach since we apply element level matching, structural level matching and taxonomy-based matching. The work in [23] was also adapted and refined in [24].

In [17], the authors present a method for structural conflict resolution while applying the ERML. The authors pinpoint that in their method, structural conflicts are automatically resolved resulting in less manual effort. Finally, in [17], the authors state that "the key structural conflict is that between an entity type and an attribute" (p. 227). In our approach, we do not distinguish between entities (classes) and attributes because we work on a higher level of abstraction compared with the ERML and the UML.

In [14], the authors once again adopt the ERML while addressing schematic discrepancy. The authors present algorithms that resolve the problem. In the algorithms, meta-data are transformed into entities and the authors pinpoint that the information and constraints given in the source schemata are kept. Similar to the work presented in [17], the work presented in [14] is classified as work on an implementation-dependent level.

Several algorithms for calculating concept similarity have also been proposed such as the Wu and Palmer metric [30], the Hirst and St Onge metric [15] and the Lesk metric [2]. All three algorithms will be presented in more detail in section 3C, taxonomy-based matching.

Finally, we have found that some techniques of our matching strategy are similar to the ones used in the DIKE approach [21] and the GeRoMeSuite [16]. However, the DIKE approach is quite different compared to ours since in our approach we do not focus on any specific modeling language but instead only on concepts and links between concepts. In the GeRoMESuite [16] the authors present a system for holistic generic model management but their approach focuses on implementation dependent models such as SQL, XML and OWL, while our focus is on implementation independent conceptual schemata, meaning the approaches are complementary rather than exclusively.

III. MATCHING STRATEGIES

An important aspect of our semi-automatic three-tier matching approach is its independence from any specific modeling language [9], meaning it can be used for the integration of schemata that are available in different source languages. Of course, this also means that our strategy cannot depend on language specific modeling concepts but has to utilize other, e.g., linguistic, information to analyze the models.

In our approach we first perform comparisons on the *element level* for gathering preliminary matching proposals. Then *structural level matching* is applied to identify potential contradictions to the original assumptions that might hint at homonym or synonym conflicts. Finally, we use an optional *taxonomy-based approach* to identify previously undetected concept relationships. The latter step is especially relevant when concepts are matched in the context of ontology engineering, since it has the potential of detecting hidden, easily overlooked information.

All of these strategies are applied on concept pairs with both members of the pair coming from one of the matched schemata. Thus, in preparation for the matching process, all relevant concept pair permutations are generated – since the pairs are symmetric, the order of the concepts in the pair is irrelevant. In a further preprocessing step, linguistic tools like stemmers and lemmatizers are used to reduce the words from the concept designations in the target schemata to their base forms [8].

The following sections will describe the different levels of the matching approach in more detail.

A. Element level matching

On the element level, concepts are directly compared to each other without considering the context. The main matching criteria on this level are the names of concepts; element level matching therefore presupposes that the concept names are available in their linguistic base form. Other matching criteria on the element level are potentially available metadata such as definitions, indications of quantity or data types, though the latter is implementation-dependent and thus typically not available on the predesign level.

The eventual goal of element level matching is to decide whether a concept pair matches. The process has the following possible outcomes:

- Equivalence/Synonymy,
- Relatedness,
- Independence.

At first glance, equal words/definitions suggests *equivalence*, although the concepts might be later identified as homonymous. If the compared concept names are not equivalent, but domain ontology is available and both compared words describe known ontology concepts, then information about the *relatedness* of the compared concepts is queried. If they are not synonymous they may be directly related in another way, indirectly related via several intermediate concepts or completely unrelated. If potential relatedness is detected in the ontology, this information is incorporated in the integration proposal.

If the compared concepts are classified as *independent* after the first matching steps (i.e., no potential relatedness

 ISBN: 978-1-61208-121-2

was found), but one or both of their names consist of compound words, then these names are deconstructed. For endocentric compounds – the most common ones in the English language [11] – the right-most element of the compound word is its head. Thus, the following percolative rules are applicable for identifying automatic relationships between the words:

A. If the compared concept names are available in the form of A and AB (i.e., A corresponds to the compound AB minus the head B), then the relationship "AB belongs/related to A" can be assumed.

B. If the compared concept names are available in the form B and AB, where A is the head of the compound AB, then the relationship "AB is a B" can be assumed.

To exemplify the first rule, the concept "car color" is identified as a potential attribute of "car" ("car color" belongs to "car"), and the concept "student name" is identified as an attribute candidate for "student" ("student name" belongs to "student"). Regarding the second rule, the exemplary concept "dialysis patient" would be interpreted as a "patient" ("dialysis patient is a patient"), and "blood pressure measurement" is a (specific form of) "measurement".

On a related note, if no definition or ontology data is available about a schema concept, semi-automatic disambiguation can be attempted, using generic lexicons such as WordNet [19] that contain word sense definitions. The word in question is looked up in the lexicon, which results in all possible word senses and their definitions being returned. The following four outcomes are possible:

- exactly one definition is returned;
- more than one definition is returned;
- no fitting definition are returned;
- the returned definition is on the wrong detail level.

If more than one meaning is returned, the senses are ranked according to their likelihood of occurrence in the English language or the domain in question. If no meaning is returned, other searches are automatically attempted with linguistic decompositions or variants of the word. If the returned definition is on the right track, but on the wrong detail level, the search is repeated for the candidate concept's hypernyms or hyponyms respectively. The entire process is described in detail in [28].

B. Structure level matching

On the structural level, comparisons of the concepts' neighborhoods are conducted, meaning that those concepts that are directly connected to concepts, which have been identified as equivalent or similar to concepts in another source schema, are compared. In doing so, several similarities and differences might be recognized that otherwise could pass unnoticed. Besides that, structure level matching is also used to verify or decline the results of element level matching. In structure level matching, we propose to use a set of "IF THEN" rules. Moreover, certain influence factors such as *polysemy count*, *valency* and *domain weight* might be used to complement the rules. The influence factors could even be used to decide whether neighborhood comparison is necessary [8] at all.

Polysemy count gives the number of meanings a word has in a given language, valency gives the number of attribute slots a word has in a given language and domain weights can be manually assigned to concepts by domain experts [8].

We propose to use two types of "IF THEN" rules: *rules for equivalent concept names* and *rules for similar concept names* (see also [6][7]). As the rule names indicate, *equivalent* means that two concept names are recognized as equivalent in element level matching, e.g., "Name" in schema 1 and "Name" in schema 2. *Similar* means that the concept names are not equivalent but recognized as similar in element level matching, e.g., "Order" in schema 1 and "OrderItem" in schema 2. In total, at least six rules should be used for equivalent concept names and three rules for similar concept names. *The rules for equivalent concept names* can be stated as:

IF comparison of concept names yields equivalent and comparison of concept neighborhoods yields:

- Equivalent THEN *equivalent* concepts are most likely recognized.
- Different THEN *homonyms* are most likely recognized.
- Similar AND one concept in each schema is named different, THEN *synonyms* are most likely recognized.
- Similar AND one concept name is a composite of another concept name with a following addition, AND cardinality is indicating 1:1, THEN an *association* between the two concepts is most likely recognized.
- Similar AND one concept name is a composite of another concept name with a prior addition, THEN a *hypernym-hyponym* pair is most likely recognized.
- Similar AND one concept name is a composite of another concept name with a following addition AND cardinality is indicating 1:M with or without uniqueness, THEN a *holonym-meronym* pair is most likely recognized.

The rules for equivalent concept names verify or decline the result from the first part of element level matching, while the rules for similar concept names verify or decline the result from the second part in which the percolative rules are applied. The *rules for similar concept names* can be stated as:

IF comparison of concept names yields:

- Similar, one concept name is a composite of another concept name with a following addition, AND comparison of concept neighborhoods yields similar

 ISBN: 978-1-61208-121-2

or equivalent with an indication to a 1:1 cardinality THEN an *association* between the two concepts is most likely recognized.

- Similar, one concept name is a composite of another concept name with a following addition, AND comparison of concept neighborhoods yields similar or equivalent with or without an indication to a unique 1:M cardinality THEN a *holonym-meronym* (aggregation) pair is most likely recognized.
- Similar, one concept name is a composite of another concept name with a prior addition, AND comparison of concept neighborhoods yields similar or equivalent THEN a *hypernym-hyponym* pair is most likely recognized.

In [6] and [7] it was described how the rules could be applied while applying the Karlstad Enterprise Modeling approach [13]. However, in this paper we do not focus on any specific modeling language and therefore we have also refined and adapted the rules to be useful for any modeling language; in other words the rules are modeling-language independent.

C. Taxonomy-based matching

The previous matching strategies for concept pairs were all based on their names and context or the use of domain ontologies. Domain ontologies, however, are not always available, and concepts may have a sparse neighborhood, which can make analysis of their context unreliable. Using general-purpose taxonomies that are not restricted to one domain, general assumptions about the relationship between two words can be made: isolated words are compared based on their position in the taxonomy instead of on their structure or context.

A particularly extensive domain-independent taxonomy for the English language is provided by the lexical database WordNet [19], which is freely available and thus widely used in scientific research projects. It is important to note that using WordNet for calculating concept similarity is completely separate from using WordNet for disambiguation purposes as discussed on the element level. In [18], the authors compared a number of different approaches for calculating semantic similarity metrics based on WordNet. Perl-based implementations for deriving concept similarity measures from WordNet were also presented in [22]. Among them, Wu and Palmer, Hirst and St Onge and Lesk will be shortly discussed here because they are three very different forms of WordNet-based similarity measures.

The *Wu and Palmer metric* was first suggested in [30]. The similarity value is calculated using formula 1:

$$wup = \frac{2 * depth\ (LCS\ (concept\ 1, concept\ 2))}{depth\ (concept\ 1) + depth\ (concept\ 2)} \quad (1)$$

In a first step, the least common subsumer (LCS) is determined, i.e., the first common parent of the compared concepts in the taxonomy. The similarity score is derived from dividing the double of the taxonomy depth of the LCS (since two concepts are compared) by the sum of the taxonomy depths of the compared concepts. Further separation of the concepts from their first common father concept means a lower similarity score.

The *Hirst and St Onge metric* [15] allows measuring the similarity between two concepts by determining the length of the taxonomy path between them. Three different kinds of paths for connecting concepts can be distinguished based on their strength: *extra-strong*, *strong* and *medium* paths. Extra-strong paths exist between two equivalent concepts. Strong path are identified by a direct connection between two concepts. Medium-strong paths finally mean that two concepts are indirectly connected. In the latter case, the number of path direction changes is relevant for determining the concept similarity. Direction changes occur every time a medium-strong connection switches between upwards-paths (generalizations), downward-paths (specializations) and horizontal paths (other relationships between concepts). Frequent direction changes lower the similarity score, as shown by formula 2:

$$hso = C - pathLength - k * numberDirectionChanges \quad (2)$$

The calculation returns zero if no path at all exists between the concepts. In that case, the concepts are interpreted as unrelated. C and k are constants used for scaling the metric.

Finally, the *Lesk metric* [2] is a context-based similarity score that does not require taxonomic structures. Instead it presupposes a lexicon, in which different word senses are distinguished and detailed definitions for each meaning are available. Because the WordNet taxonomy contains definitions and examples for each concept, it is a popular choice for this task. For determining Lesk similarity, the definitions of both involved concepts must be provided; then a numerical estimation of their degree of separation is calculated by counting the word overlap.

Traditionally, the Lesk algorithm is used for disambiguating words in full natural language texts: a context window containing an equal number of words on both sides of the observed word is defined. Then all available definitions for the observed concept and the other content words in the context window are examined and compared, ignoring non-content words such as pronouns or articles. The word sense that has the greatest overlap with the definitions from the surrounding text is assumed to be the correct one.

In our use of the Lesk algorithm, already disambiguated concept-pairs are presupposed and the Lesk metric is used to calculate similarity scores for them. The scores describe the concept pair's relative similarity compared to other concept pairs and – if an according threshold value has been defined – the conflict potential of the word pair. For example, using

our optimized Lesk algorithm, the concept pair "car"-"bicycle" has a similarity score of 198, "car"-"motorcycle" has a score of 321 and "car"-"bus" is assigned the score 688, which indicates their relative similarity.

The algorithm and potential optimizations of the Lesk algorithm for our purposes was described in detail in [28]. Lesk is the most relevant WordNet similarity measure for the matching purpose since it is rather robust against inadequacies in the taxonomic structure and its results can be improved by relatively simple, light-weight enhancements of the taxonomy, such as filling gaps in concept definitions.

The results of taxonomy-based concept matching are a starting point for future ontology extensions. If, for instance, a high matching score is identified between two previously unrelated concepts, then a relationship between them can be assumed, which is a candidate to be incorporated in the domain ontology.

IV. FURTHER USE OF THE MATCHING RESULTS

Any integration attempt needs to follow a predetermined workflow that combines the various techniques that have previously been described. In [10], this process was called Concept Determination Method (CDM), because at the heart of it is the disambiguation of concepts, which is a precondition for conflict recognition and resolution, which in turn enables model consolidation. The following parties are involved in the process: system designers, domain experts and ontology supervisors. The process input typically consists of two schemata, which are to be integrated. A single schema is also a permitted input; in this case only the schema-preprocessing phase is traversed, which involves the optimization of its modeling element designations and the resolution of any potential inner-schema conflicts. In all cases, the output of the CDM consists of one single (integrated) schema. The integration process is supported by a number of external repositories, namely the domain ontology and an optional domain-overarching taxonomy/lexicon. For the purpose of testing the CDM, an integration prototype is currently under development at Alpen-Adria-Universität Klagenfurt.

The CDM starts by choosing the source schemata that should be consolidated, one of which is typically the current integrated schema. If more than two schemata need to be integrated, they are processed pair-wise one after the other, with the current integrated schema always being one member of the pair. In cases where only one schema is chosen as input, it is preprocessed and returned in optimized form. The integration process itself follows the typical phases (1) schema preprocessing, (2) schema matching and (3) schema consolidation.

In preprocessing, schemata are examined for internal conflicts and prepared for the following phases. Afterwards, the matching phase begins, which consists itself of several sub phases that were described earlier in the article. How the several matching techniques are utilized in a common workflow was first discussed in detail in [8].

In the first step of schema matching, all permutations of concept pairs from the two source schemata are prepared for comparison. The eventual matching goal is to decide whether the compared concepts are the same or different. The proposed workflow is as follows: every concept pair is first matched on the element level using the direct comparison of the base form and the application of linguistic rules. This step results in a preliminary matching decision. If the result is "independent" and a domain ontology is available, then information about potential connections between the concepts are looked up in the ontology. Technically speaking, this is still a part of element level matching, because the concepts' context is irrelevant for this step.

Concept pairs that have been classified as "independent" or "equivalent" during element level matching then undergo structural matching, which aims to identify potential homonym and synonym conflicts based on the neighbors of the compared concepts. Additionally, structural matching should also recognize hypernym-hyponym pairs and holoynm-meronym pairs. If such conflicts are identified as likely, a respective warning is added to the preliminary matching decisions.

Finally, taxonomy-based matching (e.g., the Lesk metric) can be optionally performed for concept pairs, which are still presumed "independent" after structural matching. The goal is to detect potential hidden relatedness between the concepts. This is especially recommended if at least one of the compared concepts is yet unknown in the domain ontology. The final matching proposals, including any warnings, are presented to domain experts, who then have the chance to accept the proposals or override them. For instance they can decide if and how potential homonymy/synonymy conflicts should be resolved. If no domain expert is available, the default proposals are pursued.

Based on the matching results, specific integration proposals are generated in the schema consolidation phase. In summary, the following strategies are applied [29]: For matching concepts, the integration proposal is to merge them and make sure that both concept names are stored in a repository otherwise this could result in semantic loss [5]. Unrelated concepts are transferred to the integrated schema independently. For (directly) related concepts, both concepts are transferred to the integrated schema and a relationship between them is introduced. Concepts are indirectly related when they have no direct connecting relationship in the domain but are connected via several other concepts. For example two concepts might have a common concept with which they are connected via hypernym-hyponym and holonym-meronym relationships. It is principally possible to also transfer such more complex relationships – including all intermediate concepts – to the integrated schema, as a proper connection for the indirectly related concepts.

A central requirement regarding the integration workflow states that the process should be automatized as much as possible. This means that domain experts should be supported by preferably accurate proposals and the tool should generate a default integrated schema even when no user input is made at all. The integration prototype provides the option to adjust the preferred degree of automatization.

 ISBN: 978-1-61208-121-2

Currently, the prototype focuses on certain matching techniques and was mainly tested for exemplary cases. However the preliminary results give reason to hope that the suggested workflow is a suitable default process for most projects.

V. SUMMARY AND CONCLUSIONS

In this paper, we have presented a three-tier matching strategy for predesign schema elements. Our strategy is modeling-language independent and should be applied early in the information system development process. Modeling-language independent means that detailed implementation and design information is not dealt with at this stage and that we only use the most essential modeling elements: concepts and links between concepts. Our approach should be viewed as one step towards a semi-automatic method for modeling-language independent schema integration. The presented and proposed multi-level matching strategy is composed of *element level matching*, followed by *structural level matching* and ending with *taxonomy-based matching*. When applied in schema integration, the three matching strategies should facilitate the recognition of similarities and differences between two source schemata.

REFERENCES

[1] A., Bachmann, W. Hesse, A. Russ, C. Kop, H.C., Mayr, and J. Vöhringer J., "OBSE – An Approach to Ontology-Based Software Engineering in the practice," in EMISA, 2007, pp. 129-142.

[2] S. Banerjee and T. Pederson, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," in Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics, 2002, pp. 136–145.

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

[4] P. Bellström, View Integration in Conceptual Database Design – Problems, Approaches and Solutions, Licentiate Thesis, Karlstad University Studies 2006:5, 2006.

[5] P. Bellström, "On the Problem of Semantic Loss in View Integration," in Information Systems Developent Challenges in Practice, Theory, and Education, Vol. 2, C. Barry et al., Eds. Heidelberg: Springer, 2009, pp. 963-974.

[6] P. Bellström, Schema Integration – How to Integrate Static and Dynamic Database Schemata, Dissertation, Karlstad University Studies 2010:13, 2010.

[7] P. Bellström, "A Rule-Based Approach for the Recognition of Similarities and Differences in the Integration of Structural Karlstad Enterprise Modeling Schemata," in Proceedings of the 3rd IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling, P. van Bommel et al., Eds. Heidelberg: Springer, 2010, pp. 177-189.

[8] Bellström P. and J. Vöhringer, "Towards the Automation of Modeling Language Independent Schema Integration," in Proceedings of the 1st International Conference on Information, Process, and Knowledge Management, A. Kusiak and S.G. Lee, Eds. IEEE Computer Socity, 2009, pp. 110-115.

[9] P. Bellström, J. Vöhringer, and C. Kop, "Towards Modeling Language Independent Integration of Dynamic Schemata," in Information Systems Development Toward a Service Provision Socity, G.A. Papadopoulos et al., Eds. Heidelberg: Springer, 2009, pp. 21-29.

[10] P. Bellström, J. Vöhringer, and A. Salbrechter, "Recognition and Resolution of Linguistic Conflicts: The Core to a Successful View and Schema Integration," in Advances in Information Systems Development New Methods and Practice for the Networked Socity, Vol. 2, G. Magyar et al., Eds. Heidelberg: Springer, 2007, pp. 77-87.

[11] L. Bloomfield, Language, Chicago - London: The University of Chicago Press, 1933.

[12] P. Chen, "The Entity-Relationship Model – Toward a Unified View of Data," ACM Transactions on Database Systems, vol. 1(1), 1976, pp. 9-36.

[13] R. Gustas and & P. Gustiené, "Towards the Enterprise Engineering Approach for Information System Modelling Across Organisational and Technical Boundaries," in Enterprise Information Systems V, O. Camp et al., Eds. Dordrecht: Kluwer, 2004, pp. 204-215.

[14] Q. He and T.W. Ling, "Resolvning Schematic Descrepancy in the Integration of Entity-Relationship Schemas," in: Proceedings of ER 2004, P. Atzeni et al., Eds. Heidelberg: Springer, pp. 245-258.

[15] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," in WordNet: An Electronic Lexical Database (Language, Speech, and Communication), 1998.

[16] D. Kensche, C. Quix, X. Li, and Y. Li, "GeRoMeSuite: A System for Holistic Generic Model Mangement," in Proceedings of the 33rd international conference on Very large data bases, C. Kock et al., Eds. 2007, pp. 1322-1325.

[17] M.L. Lee and T.W. Ling, "A Methodology for Structural Conflict Resolution in the Integration of Entity-Relationship Schemas," Knowledge and Information Systems, vol. 5(2), 2003, pp. 225-247.

[18] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," in AAAI'06, 2006, pp. 775-780.

[19] G. A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, vol. 38, 1995, pp. 39-41.

[20] Object Management Group, OMG Unified Modeling Language (OMG UML), Superstructure, [Electronic], Available: http://www.omg.org/spec/UML/2.2/Superstructure/PDF/ [20101201]

[21] L. Palopoli, G. Terracina, and D. Ursino, "DIKE; A System Supporting the Semi-Automatic Construction of Cooperative Information Systems From Heterogeneous Databases," Software–Practice and Experiences, vol. 33, 2003, pp. 847-884.

[22] T. Pederson, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity - measuring the relatedness of concepts," in Proceedings of the 19th National Conference on Artificial Intelligence, 2004, pp. 1024-1025.

[23] E. Rahm and P.A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," The VLDB Journal, vol. 10, 2001, pp. 334–350.

[24] P. Shvaiko and J. Euzenat, "A Survey of Schema-Based Matching Approaches," Journal of Data Semantics, vol. 4, 2005, pp. 146-171.

[25] W. Song, Schema Integration – Principles, Methods, and Applications, Dissertation, Stockholm University, 1995.

[26] J. Vöhringer, P. Bellström, D. Gälle, and C. Kop, "Designing a study for evaluating user feedback on predesign models," in Proceedings of ISD2009 Conference, 2010, pp. 411-425.

[27] J. Vöhringer, D. Gälle, G. Fliedl, C. Kop, and M. Bazhenov, "Using Linguistic Knowledge for Fine-tuning Ontologies in the Context of Requirements Engineering," International Journal of Computational Linguistics and Applications, Vol. 1(1-2), 2010, pp. 249-267.

[28] J. Vöhringer and G. Fliedl, "Adapting the Lesk Algorithm for Calculating Term Similarity in the Context of Ontology Engineering," in Proceedings of ISD2010 Conference, 2011, In Print.

[29] J. Vöhringer and H.C., Mayr, "Integration of Schemas on the Pre-Design Level Using the KCPM-Approach," in Advances in Information Systems Development Bridging the Gap between Academia and Industry, A.G. Nilsson et al., Eds. Heidelberg: Springer, 2006, pp. 623-634.

[30] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 1994, pp. 133-138.

 ISBN: 978-1-61208-121-2