Semi-Automatic Schema Pre-Integration in the Integration of Modeling Language Independent Behavioral Schemata

Peter Bellström[1], Christian Kop[2], Jürgen Vöhringer[3]

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

[2]Institute for Applied Informatics, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria
e-mail: chris@ifit.uni-klu.ac.at

[3]econob GmbH, Klagenfurt, Austria
e-mail: juergen.voehringer@econob.com

***Abstract*—In this paper, we address schema pre-integration in the integration of modeling language independent behavioral schemata. In doing so, we propose and present a set of tasks that should be carried out not only to improve and clarify the meaning of a schema, but also to facilitate the resulting time consuming and error-prone phases in the integration process. Due to the complexity of schema pre-integration, domain experts and repositories (e.g. ontologies) are still important sources of knowledge and should therefore be involved during the whole integration process. As its main contribution, the paper offers new tasks to perform in the schema pre-integration process as well as an adjusted and enriched work of previously presented tasks for schema pre-integration.**

Keywords-Schema Integration; Pre-integration; Behavioral schemata; Dynamic Schemata

I. INTRODUCTION

When designing an information system, the designer and domain experts produce a set of conceptual schemata illustrating both the structural (static) and the behavioral (dynamic) aspects. During the development of larger proposed information systems or enterprise models, these models cannot be built at once into one schema. Often, initially different views are generated, which have to be merged afterwards into the proposed overall schema. Due to the fact that it is one information system, and not a set (cf. set of schemata) that is going to be developed, the conceptual schemata need to be integrated. In another integration scenario, separate information systems based on separate schemata already exist. However, due to various reasons (merging of enterprises, the need for consolidating federated databases for information retrieval etc.), these schemata have to be integrated in order to show the whole picture. Whereas the first scenario is often referred to as "view integration", the second one is called "schema integration". Since both integration scenarios are based on schemata (either final schemata or schema parts), we follow the definition given in [1] and will hereafter use the term "schema integration". Schema integration is described by [1] as "the activity of integrating the schemas of existing or proposed databases into a global, unified schema." (p. 323) and is by the same authors divided into four phases: *pre-integration*, *comparison of the schemata*, *conforming the schemata* and, *merging and restructuring*. We extend this definition only in the aspect that we do not only concentrate on databases. Databases are one application domain of the integration process.

The focus of this paper is the first phase: pre-integration. The reason for focusing on pre-integration is its important impact on the integration step. In the literature, it is already shown that this process can influence how efficiently different schemata can be merged.

This holds especially true if we do not focus on a specific modeling language but try to integrate schemata modeled in different languages for the same purpose. For instance, one schema could be modeled in the business process modeling notation (BPMN) [2] whereas another one could be modeled using ARIS event process chains [3]. In [4], it was shown that this can happen if the business process models of two enterprises that from now on will be in a consortium, must be merged.

Hence, our approach is based on previous approaches on schema integration and particularly on schema pre-integration. The approach is novel, since it tries to introduce modeling language independent schemata, which have the same aim and purpose, as input for the integration process. It focuses on the special tasks that must be considered in such a scenario (i.e. transforming the different modeling elements to a common abstract level).

This paper is therefore structured as follows: in section two, we describe the schema integration process as such. We mainly refer to the important work of [1]. In section three, we address related work on pre-integration and in section four the tasks to perform in schema pre-integration are described. Finally, the paper closes with conclusions and an outlook to future work.

II. THE SCHEMA INTEGRATION PROCESS

In [1], the authors divided the integration process into four activities: *pre-integration*, *comparison of the schemata*,

 ISBN: 978-1-61208-181-6

conforming the schemata and *merging and restructuring*. To grasp what schema integration is all about, each of these four activities will now be shortly addressed and described.

A. Pre-integration

In the pre-integration activity, general analyses of the schemata are applied in order to find strategies for how the schemata have to be integrated. This includes the:

- choice of schemata (views) that have to be integrated into a whole schema,
- collection of additional relevant information that are interesting during integration (i.e. assertions or constraints that must hold among the schemata), and
- strategy (policy) for the integration process (i.e. which schema comes first, which schema is integrated with which other schema).

According to the results described in [1], the integration policy can be a binary and an n-ary policy. The binary integration strategy can be further divided into a ladder-procedure or a balanced procedure. The n-ary approaches can be divided into a one-shot integration and iterative strategy. Integration policy is called a ladder, if there are two schemata that are integrated at the beginning. The first intermediate integration result is then integrated with a third schema. The resulting second intermediate integrated schema is then integrated with a fourth schema and so on. In the balanced strategy, all the source schemata are integrated pairwise. The first intermediate integrated schemata are then integrated pairwise (i.e. the integration of schemata behaves like a binary tree). In the one shot approach, all the schemata are integrated into one global schema at once. If more than 2 schemata are integrated and it is not done in a one-shot strategy, then it is called an iterative strategy., The ordering of schemata and intermediate integrated schemata might be important, especially in the iterative and the ladder strategies. In these approaches, as well as in the balanced approach, it might be important which schema is integrated with which other schema. In [1], advantages and problems of the different strategies are discussed.

B. Comparision of the schemata

In this activity, it must be detected if concepts are the same or are different. Conflicts (i.e. concepts are identical or different) can be classified into naming conflicts. For instance are "employee" and "staff" the same concepts or not? Besides naming conflicts, structural conflicts can also appear in the schemata. For instance, the same concept can be modeled as an attribute in one schema and as an entity type in another schema.

C. Conforming the Schemata

In this activity, conflicts are resolved as best as possible and the schema is transformed (i.e. prepared) for the merging activity. The results of this activity are schemata with schema elements that conform (e.g. in the schemata to be integrated, a concept such as "customer" is an entity type in both schemata).

D. Merging and Restructuring

In this activity, the schemata will be integrated into one schema. Besides conflict resolution described in the previous section, it is also necessary to complete the integrated schema. For instance, if in one schema the concept "employee" exists and in another schema the concept "manager" exists, then it might be necessary to introduce a generalization relationship between manager and employee. Several other operations for this activity are introduced in [1].

III. Related Work

The work presented in [1] and [5] showed that only three works explicitly mentioned the pre-integration step. In the following, other integration approaches were published, which focused on several aspects of the integration problem.

A. Integration of Structure

In [6], the authors discovered that integration of structural schemata can be explained with attribute equivalence. They concluded that this is the basic concept throughout schema integration. The integration process starts with an existing (logical) database schema.

Another work on operators for deciding on the similarity or dissimilarity of schema construct was described in [7].

In [8], the author uses logical assertions to define which constructs of two conceptual models are equivalent. On the basis of these defined assertions, he proposed a method for the automatic integration of the two schemata.

The work presented in [9] concentrates on the automatic detection of naming conflicts.

Algorithms for structural schema integrations are introduced in [10].

Also in the approach presented in [11], the aim is to integrate existing database schemata. For this purpose, they enrich the schema semantically.

An object oriented framework for the integration of heterogeneous databases is presented in [12].

In [13] the authors discuss the impact of linguistic knowledge for the integration step. This approach is based on the well-established assumption that relationships are expressed by a verb. From a linguistic point of view, verbs always have a semantic structure in which nouns play a certain semantic role. For instance if a person buys something, then person has the semantic role of an actor. This knowledge can be used during the merging process (e.g. comparing only actors if two relationships are named with the same verb).

In [14] the authors introduce a black board architecture for schema integration of existing databases. A black board architecture system supports the sharing of knowledge from different knowledge sources. These knowledge sources are the designers and end users who feed the system with their knowledge.

The impact of similarity measures for schema matching and data integration is discussed in [15].

B. *Integration of Behavior and other Aspects*

Up to now, the described mentioned approaches were developed to integrate structural aspects (i.e. information needed for database design).

Integration of behavioral aspects in object oriented models is mentioned in the work of [16][17]. They describe the integration of state charts of an object type (e.g. life cycle of a book in a library [17]).

In [18], the authors present another interesting work of integration of dynamic object oriented models. Their work is based on the formalization of state chart constructs.

In [19], the authors provide a roadmap for behavior-based integration. They propose a meta class framework on which integration should be based.

An overview of business process integration is given in [20].

OWL-S ontologies are proposed as a support in a method for business process integration [21]. The business process models are firstly transformed into OWL-S models, which are then integrated.

In [22], the authors describe the integration of use cases. The authors exploit information of modular petri nets, which describe these use cases.

Finally requirement statements on behavior are integrated using the behavior tree approach [23]

C. *Pre-Integration*

Among these above-listed approaches, pre-integration was either explicitly mentioned or, according to the needed input for the integration step, it could be concluded which kind of tasks are necessary in the pre-integration step.

Because the research work of both [6] and [14] used sources that were relational schemata, these sources are firstly brought into a canonical form (i.e. a specific conceptual model). This can be seen as a pre-integration step. In [14], the authors also explicitly mention the pre-integration step. In their work, schema translation and pre-integration are separate steps. Their pre-integration is about making a decision about the policy used to integrate the schemata (i.e. binary, ladder or n-ary strategy).

The need for defining assertions and constraints as mentioned in [6] and in [8] might also be treated as part of a pre-integration step, since the schemata are prepared in order to make integration easier. The step denoted as "assertion specification" or "schema integration assertion" in [6] and in [8] respectively, is subsumed as the schematic inter-schema relation integration in the work of [14]. A similar policy of pre-integration can be found in [11]. Their first step of integration is called "semantic enrichment phase". This step contains a knowledge acquisition step and a schema conversion step. In the knowledge acquisition, the schemata are analyzed to discover semantics and implicit restrictions (e.g. keys, attribute dependencies). Afterwards, everything is also converted to a canonical form. In their work, this canonical form is a BLOOM schema.

The methodology presented in [13] implies that there are linguistic knowledge bases (i.e. lexicons) that can be used. Furthermore, elements of an entity relationship model must be extended with this extra information. Hence, the use of the lexicons and the generation of extra information are part of a pre-integration step. The outcome of this step is then the input for the subsequent steps.

The work of [16] and [17] has some sort of pre-integration work as well. The first phase is called "integration in the large", whereas the second phase is called "integration in the small". Integration in the large can be seen as a kind of pre-integration, because relationships between state charts are built manually. These relationships express the "dependencies" between state charts which are considered for integration. These relationships support the search for an integration plan in order to reduce expensive integration operations. Integration according to this plan is done in the second phase of their approach (integration in the small).

In the work presented in [19], a pre-integration step is briefly discussed. The pre-integration starts with the definition of the behavior and structure of the meta classes for a domain, which can then be specialized for a certain application.

In the work presented in [21], on an ontology based method for business integration, OWL-S can be seen as the canonical form within a pre-integration step.

The modular petri nets used in [22] can be interpreted as the canonical form to describe the behavior within use cases, which support the integration process.

The behavior tree model is the canonical form for integration of requirements in [23].

As shown in section II and section III, pre-integration is only partially in focus. In this paper, we therefore focus and highlight the pre-integration phase and present new tasks as well as adjusted and enriched tasks that should be used in the pre-integration phase. In the long run, these tasks should not only reduce the time needed for integration but also reduce the risk of errors occuring.

IV. PRE-INTEGRATION IN SCHEMA INTEGRATION

Pre-integration is the first step that should be carried out when conducting schema integration. However, studying related work within the research field shows that this phase has often been overlooked [24], and in some of the early methods, it was not even mentioned [1].

In the semi-automatic method for the integration of modeling language independent behavioral schemata that we are currently researching, we propose conducting the following tasks in the pre-integration step: a) *translate the schemata into one modeling language*, b) *schema constituent name adaption*, c) *schema constituent disambiguation*, d) *standardization of the abstraction level*, e) *recognition and resolution of intra-schema conflicts*, f) *introduction of missing constituents*, g) *selecting the integration strategy*, and h) *selecting the order of integration*.

It should be noted that similar phases have been mentioned for the integration of structural schemata (see for instance [25]). However, in [25], the authors researched the integration of structural schemata, whereas we research the integration of behavioral schemata. It should also be noted that not all tasks were addressed in [25] and if the task was addressed it needs to be adjusted to fit integration of behavioral schemata, as will be shown and exemplified in this section. Most of the described tasks can be atomized or at least partly atomized contributing to a semi-automatic approach to schema pre-integration.

A. *Translate the schemata into one modeling language*

The first task to perform in schema pre-integration is to translate all schemata into one modeling language. In [24], this task was called canonization. Choosing the right canonical model for the current project is also emphasized in [26] in which the author focuses on schema translation in federated information systems stating that "[…] the canonical data model must have an expressiveness which is equal to or greater than that of any of the native models in the federation. […] a canonical data model should contain as few basic constructs as possible." (p. 15). In our method, this means translating all schema elements to conditions (pre- and post) and process types, which are the minimal modeling constituents for describing and modeling the behavior of an information system [27]. In [4], this was demonstrated using a small library system prepared for integration. In [4], it was also emphasized that during this transformation, it is important that all labels of the original schema elements are represented in the process type e.g., *Reserve Educational Book* becomes *Customer Reserves Educational Book*, adding the actor into the process type label. This task can often be partly atomized, since tools exist that can aid in the process of translating a schema from one modeling language to another. However, the schema produced by these tools should be viewed as an intermediate schema, since these most likely need to be manually adjusted to meet all rules of the chosen modeling language. In the end, it is still the domain expert that has the domain knowledge and therefore can decide how the processes and states should be combined and described.

B. *Schema constituent name adaption*

The second task to perform in schema pre-integration is to adapt schema constituent names to specific naming rules. The names of constituents are very important and if they are put together, they should reflect the meaning of either a condition or a process type. However, to facilitate semi-automatic schema integration, they should not only be readable for humans, e.g. domain experts and designers, but also be readable to a computerized application. This means that a formal language is not useful but neither is natural language since it is ambiguous. We therefore use naming guidelines to adjust the language used in the schemata and thereby end up with a controlled subset of natural language. In [27] and [28], this was first mentioned as an important guideline, called *standardization of concept notions*.

Three examples on how to use this task and when it is applicable are as follows:

- name static concepts in ***singular*** (e.g. *Books* → *Book*)
- name process types with the ***verb*** + ***noun*** rule (e.g. order → order Book)
- name conditions with the ***noun*** + ***verbal principal*** rule (e.g. *reserved* → *Book reserved).*

Since we assume that integration of structural schemata is conducted before integration of behavioral schemata, the integrated structural schema can also be used as an information repository; it can even be used as a template on how to adjust constituent names.

Finally, similar and complementary approaches are given in [28] and [29], where the authors described a controlled langauge approach for OWL verbalization [28] and schema constituent adjustment [29].

C. *Schema constitutent disambiguation*

The third task to perform in schema pre-integration is schema constituent disambiguation. In this task, we add descriptions and definitions of process types and conditions. This task could either be done manually by domain experts or automatic suggestions could be generated using domain ontologies or general lexicons such as Wordnet [30]. However, to get a good and reliable result from this step, it is important that prior to this task schema constituent name adaption has been conducted. Since we also assume that integration of structural schemata is done prior to integration of behavioral schemata, the results from that task could also be used within schema constituent disambiguation. For instance, having conducted schema element disambiguation for static schemata [25], we have already collected definitions and explanations of important structural concepts that could be reused. The integrated structural schema also indicates which structural concepts need to be processed and given conditions (see also schema constituent name adaption). This task is rather complex and often we need to split sentences or sequences of words into single words and from that move on with the disambiguation task [28].

D. *Standardization of the abstraction level*

The fourth task to perform in pre-integration is standardization of the abstraction level. In [16] and [17], the authors address integration of state charts, mentioning the problem of *state overlapping*, meaning one state in a specific state chart corresponds partially to a specific state in an another state chart. In [27], it was mentioned that similar problems were identified in the integration of behavioral schemata, where they were called *process type overlapping* and *condition overlapping*.

In our method, we address the overlap problem by trying to standardize the abstraction level in each behavioral schema. In doing so, we agree with [28] who also address the problem of different abstraction levels, stating that the schemata should be detailed without addressing implementation issues and each modeling element should be atomic. If for instance a process type is recognized as not being atomic, the process type needs to be analyzed and modified to fulfill this criterion.

E. *Recognition and resolution of intra-schema conflicts*

The fifth task to perform in schema pre-integration is recognition and resolution of intra-schema conflicts. This means analyzing one single source schema aiming to recognize conflicts (similarities and differences) within the schema. This is an important task since oftentimes two process types or conditions are named the same but the actual meaning is very different, or that one process type or condition is given different names but the actual meaning is the same. In other words, in this task we look for potential homonym and synonym conflicts. To do so, we not only analyze and compare the name of the constituents, but also the neighborhoods (surrounding). Comparing the neighborhood of the constituent has also been addressed in relation to integration of structural schemata. In [31] and [32] for example, the authors use neighborhood comparison as a matching strategy during semi-automatic integration of modeling language independent structural schemata. Similar techniques are also used in the DIKE approach [33] and the GeRoMeSuite [34]. However, our approach is placed much closer to the work presented in [31] and [32], due to the focus on modeling language independent integration.

F. Introduction of missing constituents

The sixth task to perform in schema pre-integration is the introduction of missing constituents. More precisely, this means introducing missing process types, conditions or connections between constituents. During several of the described pre-integration tasks, e.g. translate the schemata into one modeling language and standardization of the abstraction level, the domain expert and designer might identify holes: some constituents are missing in the schema that is currently being prepared for integration. In this task, the process types, conditions and connections between them that are identified by the domain experts and designers are added manually. If a domain ontology and/or taxonomy are available, these can be used to recognize a missing connection between two constituents. The possibility of using a behavioral taxonomy to enrich behavioral schemata was also addressed by [19]. However, their approach focused on the object oriented modeling paradigm, while we instead research modeling language independent integration of behavioral schemata.

G. Selecting the integration strategy

The seventh task to perform in schema pre-integration is selecting the integration strategy. In this task, the order of integration is decided. The integration order can be divided into binary and n-ary integration. Binary integration can further be divided into binary ladder and binary balanced and n-ary integration into n-ary one-shot and n-ary iterative [1].

In our approach, we have decided to use binary ladder, meaning two schemata are always integrated (see section II A). Using binary ladder is preferred since the complexity is reduced due to only processing two schemata within each iteration and we can also in a semi-automatic way decide upon a first suggestion of the order of integration.

During the 1980's, this task was also the main task researched for pre-integration in the integration of structural schemata (e.g. [5][35]).

H. Selecting the order of integration

The last task to perform in schema pre-integration is selecting the order of integration. Having decided to use binary ladder (see the former task), this task should result in a decision regarding the specific integration order, meaning which schemata should first be integrated and so on. To decide upon the specific integration order. we not only analyze and compare the pre-condition(s) and the post-condition(s) of each schema, but we also analyze the process type descriptions. By analyzing and comparing the conditions, we mean the first and last conditions of each schema. For instance, the first condition of schema one is the pre-condition *Book Not Reserved* and the last post-condition of the same schema is *Book Reserved*. This means that conditions within the schema are in this step viewed as a black box. In other words, in this task we are looking for conditions that might be the same. For instance, two disjoint schemata have the same post-conditions, a post-condition in one schema is a pre-condition in another (consecutive), two parallel schemata have the same pre- and/or post-conditions, two alternative schemata have the same pre- and/or post-conditions and finally two schemata are viewed as equivalent having the same pre-and post conditions. For a more detailed discussion and description of schema relationship types, please see [16][17], who research the integration of state charts and [28], who research integration of Klagenfurt Conceptual Pre-Design Models [36].

To complement the analysis and comparison of conditions, we also count the occurrences of the most important terms used within the schemata. This should be done since the resulting figures could aid in the process of deciding the order or integration. For instance, if schema one describes the process of storing an order and schema two describes the process of updating an already stored order, counting the number of "order" should most likely indicate that these two schemata should be integrated in one iteration.

Selecting the order of integration is also facilitated if the task of standardization of the abstraction level has already been carried out. Additionally, the integrated structural schemata can be used as a knowledge source, since in it the static data are defined and described.

V. CONCLUSION AND FUTURE WORK

In this paper, we have addressed schema pre-integration in the integration of language independent behavioral schemata. In doing so, we have presented and described a set of tasks that are important to carry out to facilitate the resulting time consuming and error-prone integration process.

In this paper, it has been shown that in all presented tasks, some type of electronic knowledge source could and should be used to assist the domain experts and designers, contributing to a semi-automatic approach to schema pre-processing.

In future research, we will continue our work on developing a semi-automatic method for the integration of modeling language independent behavioral schemata. In doing so, we will amongst other things research how to use knowledge sources, such as ontologies, taxonomies, dictionaries and lexicons, in the entire integration process.

 ISBN: 978-1-61208-181-6

REFERENCES

[1] 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.

[2] BPMN – Business Process Modeling Notation, [Electronic], Available: http://www.bpmn.org/ [20111108].

[3] A.W. Scheer, ARIS – Business Process Frameworks, Heidelberg: Springer, 1999.

[4] P. Bellström, J. Vöhringer, and C. Kop, "Toward Modeling Language Independent Integration of Dynamic Schemata," Information Systems Development Toward a Service Provision Socity, Heidelberg: Springer, 2009, pp. 21-29.

[5] C. Batini and M.Lenzerini, "A Methodology for Data Schema Integration in the Entity-Relationship Model," IEEE Transactions on Software Engineering, 10 (6), 1984, pp. 650-664.

[6] J.A. Larson, S.B. Navathe, and R. Emasri, "A Theory of Attribute Equivalence in Databases with Application to Schema Integration," Transactions on Software Engineering, Vol. 15 (4), 449-463.

[7] A. Savasere, A. Sheth, and S. Gala, "On Applying Classification to Schema Integration," Proc. First International Workshop on Interoperability in Multidatabase Systems (IMS'91), IEEE Press, 1991, pp. 258-261.

[8] P. Johannesson, "A Logical Basis for Schema Integration," Third International Workshop on Research Issues on Data Engineering (RIDE-IMS'93), IEEE Press, 1993, pp. 86-95.

[9] H. K. Bhargava and R.M. Beyer, "Automated Detection of Naming Conflicts in Schema Integration: Experiments with Quiddities," Proc. 25th Hawaii International Conference on System Sciences, IEEE Press, 1992, pp. 300-310.

[10] J. Geller, A. Mehta, Y. Perl, E. Neuhold, and A. Sheth, "Algorithms for Structural Schema Integration," Proc. Second International Conference on Systems Integration (ICSI'92), IEEEP Press, 1992, pp. 604-614.

[11] M. García-Solaco, F. Saltor, and M. Castellanos, "A Structure Based Schema Integration Methodology," Proc. Eleventh International Conference on Data Engineering, IEEE Press, 1995, pp. 505-512.

[12] H. Dai, "An Object-Oriented Approach to Schema Integration and Data Mining in Multiple Databases," Proc. Technology of Object-Oriented Languages (TOOLS), IEEE Press, 1997, pp. 294-303.

[13] E. Métais, Z Kedad, I. Comyn-Wattiau, and M. Bouzeghoub, "Using Linguistic Knowledge in View Integration: Toward a Third Generation of Tools," Data & Knowledge Engineering 23(1), 1997, 59-78.

[14] S. Ram and V. Ramesh, "A Blackboard-Based Cooperative System for Schema Integration," IEEE Expert, June 1995, 56-62.

[15] A. Gal, "Interpreting Similarity Measures: Bridging the Gap between Schema Matching and Data Integration," Data Engineering Workshop of ICDEW 2008, IEEE Press, 2008, pp. 278-285.

[16] H. Frank and J. Eder, "Towards an Automatic Integration of Statecharts," International Conference on Conceptual Modeling (ER 1999), Heidelberg: Springer, 1999, pp. 430-444.

[17] H. Frank and J. Eder, "Integration of Behavioral Models," Proc. ER'97 Workshop on Behavioral Models and Design Transformations: Issues and Opportunities in Conceptual Modeling, [Electronic], Available: http://osm7.cs.byu.edu/ER97/workshop4/fe.html [20111108], 1997.

[18] B.H.C. Cheng. and E. Y. Wang, "Formalizing and Integrating the Dynamic Model for Object Oriented Modeling," IEEE Transactions on Software Engineering, Vol 28 (8), August 2002, 747-762.

[19] M. Stumptner, M. Schrefl, and G. Grossmann, "On the Road to Behavior-Based Integration," Proc. 1st APCCM Conference, 2004, pp. 15-22.

[20] A. Raut, "Enterprise Business Process Integration," Conference on Convergent Technologies for Asia-Pacific Region, IEEE Press, 2003, pp. 1549-1553.

[21] S. Fan, L. Zhang, and Z. Sung, "An Ontology Based Method for Business Process Integration," International Conference on Interoperability for Enterprise Software and Applications in China, IEEE Press, 2008, pp. 135-139.

[22] W. J. Lee, S. D. Cha, and Y. R. Kwon, "Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering," IEEE Transaction of Software Engineering, Vol. 24 (12), December 1998, 1115-1130.

[23] K. Winter, I.J. Hayes, and R. Colvin, "Integrating Requirements: The Behavior Tree Philosophy," 8th IEEE International Conference on Conference on Software Engineering and Formal Methods (SEFM), IEEE Press, 2010, pp. 41-50.

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

[25] P. Bellström and J. Vöhringer, "A Semi-Automatic Method for Matching Schema Elements in the Integration of Structural Pre-Design Schemata," unpublished.

[26] P. Johannesson, Schema Integration, Schema Translation, and Interoperability in Federated Information Systems, Dissertation, Stockholm University, 1993.

[27] P. Bellström, J. Vöhringer, and C. Kop, "Guidelines for Modeling Language Independent Integration of Dynamic Schemata," Proc. IASTED International Conference on Software Engineering, 2008, pp. 112-117.

[28] J. Vöhringer, Schema Integration on the Predesign Level, Dissertation, Alpen-Adria-Universität Klagenfurt, 2010.

[29] G. Fliedl, C. Kop, and J. Vöhringer, "From OWL class and property labels to human understandable natural language," Proc. NLDB'07, 2007, pp. 156-167.

[30] G. A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, 38 (11), 1995, 39-41.

[31] P. Bellström, and J. Vöhringer, "Towards the Automation of Modeling Language Independent Schema Integration," International Conference on Information, Process, and Knowledge Management (eKNOW 2009), IEEE computer society, 2009, pp. 110-115.

[32] P. Bellström and J. Vöhringer, "A Three-Tier Matching Strategy for Predesign Schema Elements," The Third International Conference on Information, Process, and Knowledge Management (eKNOW 2011), 2011, pp. 24-29.

[33] 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.

[34] D. Kensche, C. Quix, X. Li, and Y. Li, "GeRoMeSuite: A System for Holistic Generic Model Mangement," Proc. 33rd International Conference on Very Large Data, 2007, pp. 1322-1325.

[35] S. B. Navathe and S.U. Gadgil, "A Methodology for View Integration in Logical Database Design," Proc. Eighth International Conference on Large Data Bases, Morgan Kaufmann, 1982, pp. 142–164.

[36] G. Fliedl, C. Kop, H.C. Mayr, W. Mayerthaler, and C. Winkler, "Linguistically based requirements engineering – The NIBA project," Data & Knowledge Engineering, 35, 2000,111-120.