Towards Quality Driven Schema Integration Process Tasks

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

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

***Abstract*—If the structure of information within several departments of an organization has to be integrated, the process of integration must meet quality criteria. In this paper, we address quality in the schema integration process, more specifically, quality driven schema integration process tasks. Therefore we searched the literature for the best practices used for conceptual modeling as such and applied these to integration tasks. We address in particular three tasks within the integration process that should improve the quality of the integrated schema when used with best quality practices. Within each best practice we emphasize the use of knowledge repositories to support the process of creating a high quality schema. The three tasks are: choosing the right integration strategy, choosing the right conflict resolution methods for the chosen level of abstraction and introducing inter-schema properties to improve and clarify dependencies.**

Keywords-Information Management; Organizational Information; Schema Integration; Schema Integration Process; Schema Quality

I. INTRODUCTION

Schema integration has a long research tradition. Nevertheless, it is still ongoing and many tasks of the schema integration process are needed at all times because schemata are not built from scratch anymore. There are a great many schemata available on the Web. Furthermore, if enterprises merge, also the schemata (e.g., enterprise and business process models) available in the enterprises must be merged. Last but not least, if enterprises use available Web Services, then it might be good to know the business process model and at least match the business process models and data models to check the compliance of the Web Service models with the respective enterprise models.

A good quality of results in such contexts is very important. Literature on quality mainly focuses on the quality of the product (i.e., the model). The criteria a model must meet in order to have a certain quality are specified. To achieve this quality, the process and the improvement of process tasks must be considered.

The aim of this paper is to provide a description of what can be done in the integration process of static schemata in order to get a good, integrated model. Since an integrated model is a model too, we analyzed the literature with a focus on static modeling, and on the kind of process tasks that lead to a model with better quality. Then we applied the strategies to the tasks that have to be done in the integration process. Particularly, we addressed three tasks within the integration process that when used with best quality practices should improve the quality of the integrated schema. The three tasks are: choosing the right integration strategy, choosing the right conflict resolution methods for the chosen level of abstraction and introducing inter-schema properties to improve and clarify dependencies.

Since the paper covers schema integration, the integration process and the quality of the integrated schema, and the process, this paper is structured as follows. In Section 2 we give an overview on integration approaches and quality of schemata. In Section 3, we describe the integration process. Section 4 focuses on some best practices for improving schema quality. In Section 5, we describe the influence of best practices mentioned in Section 4 on three of the tasks mentioned in Section 3. The paper closes with a summary and a brief outline of our future project.

II. RELATED WORK

A. Integration

There is a long research history on several aspects of integration. A first substantial study of integration was made by Batini and Lenzerini [2] in the mid-80s. In another work by Batini et al. [3], other approaches on integration were summarized. In the following years, other integration approaches focusing on several aspects of the integration problem were published.

Larsen et al. [24] used attribute equivalence as the most basic concept to explain the integration of structural schemata. Savarese et al. [33] presented operators for deciding on the similarity or dissimilarity of schema construct. On the basis of defined assertions, Johannesson [21] proposed a method to detect equivalent schemata and to automatically integrate two schemata. Bhargava and Beyer [4] concentrated on the automatic detection of naming conflicts. Further algorithms for structural schema integration can be found in Geller et al. [19]. García-Solaco et al. [18], integrated semantically enriched database schemata. Dai [12] presented an object oriented framework for the integration of heterogeneous databases. Metais et al. [28] introduced linguistic knowledge for the integration step. For relationships, for instance, verbs can name relationships. Knowledge of the verbs and their linguistic semantic roles support the integration. Ram and Ramesh [31] described a blackboard architecture for schema integration of existing

 ISBN: 978-1-61208-329-2

databases. With this system, knowledge from designers and end users who feed the system is shared. The impact of similarity measures for schema matching and data integration is discussed in Spaccapietra and Parent [35]. Frank and Eder [16] described the integration of state charts object oriented models. The work of Cheng and Wang [11] is based on the formalization of state chart constructs. Stumptner et al. [36] proposed a meta-class framework on which integration should be based. Raut [32] gave an overview of business process integration. Fan et al. [14] proposed OWL-S ontologies as a support for business process integration. Lee et al. [26] described the integration of use cases on the basis of petri net models. Finally, Winter et al. [37] used a behavior tree approach for integrating requirements.

B. Schema Quality

A great deal of work has also been written on the quality of conceptual schemata (models). Although quality is a feature of a product or artifact (e.g., a schema), it is also necessary to think about the quality of the process of generating the product to support the quality of the product.

Batini et al. [1] listed eight schema quality characteristics. Lindland et al. [27] proposed a framework consisting of the three dimensions: "syntax", "semantic" and "pragmatics". The syntax-dimension reflects the vocabulary and grammar (i.e., meta-model) of a schema. The semantic dimension relates the used terms and notions to the domain context. The chosen notions modeled by modeling elements must be legal and relevant in the domain, and they must be relevant and legal to the purpose for which the schema has been built. Finally, the pragmatic dimension is achieved if the audience can understand and follow the schema.

Moody [30] concluded that there is still a need for standards, which are also accepted by the industry.

In Moody and Shanks [29], the authors focused on process quality for the development of data schemata (ER diagrams). Their approach was evaluated in a large Australian bank. In the empirical study, it was also important, that the quality was checked throughout the schema development process. In particular, quality-checking was not only made at the end of a phase but before, during and after the schema development phases. Furthermore it turned out that an information architect, who checks the model with respect to enterprise terms can support quality.

In Cherfi et al. [10] the authors presented a framework of four quality characteristics for the ER modeling language.

Becker et al. [5] described the "Guidelines of Modeling (GoM)". Six principles of modeling are introduced in this framework: correctness, relevance, economic efficiency, clarity, comparability, and systematic design. These principles can be seen as general strategic and objective definitions for modeling. Based on these goals, the concluded modeling process consisted of the following steps: goal definition, construction of an overall navigation and structural framework, modeling as such, and completion and consolidation.

With the **se**miotic **qual**ity framework (SEQUAL), Krogstie [23] explains quality of models with model externalization, goals of modeling, modeling domain, explicit knowledge of social actors, interpretation of the social actors and technical actors as well as with languages extension.

C. Summary of the Literature

We adopted the integration process as described in Batini et al. [3] since this is a well-established process. They divided the integration process into four phases: pre-integration, comparison of the schemata, conforming the schemata and merging and restructuring. In Section 3, we describe this process in more detail.

In Section 4, we continue the description about schema quality according to some selected best practices out of the list of schema quality approaches. We have chosen these approaches since they have been shown in practice to improve schema quality. In Section 5, we will then take specific best practices and combine them with three tasks of the integration process steps described in Section 3.

III. INTEGRATION PROCESS

This section should be viewed as a reference point for the following sections in which we describe and discuss best practices in the schema integration process. The integration process starts with a set of schemata, often referred to as views. These views are integrated in order to evolve the global schema. The schema evolvement takes place in four phases proposed by Batini et al. [3]. The output of one phase is used as the input of the next phase.

A. Pre-Integration

Several tasks should be carried out in this phase. Song [34] mentioned that: translating all schemata to the chosen modeling language, checking for differences and similarities in each schema and selecting the integration strategy are all tasks to be performed in pre-integration. Three additional tasks to perform in pre-integration were proposed in Bellström and Vöhringer [7] as follows: schema element name adoption, schema element disambiguation and introduction of missing relationships.

The output from this phase is a set of revised schemata, the definitions of schema elements and the chosen integration strategy.

B. Comparison of the Schemata

This phase has been researched a great deal and has been called an important [34] and difficult phase [13][25]. Several authors [3][22][34] assigned the following tasks to this phase: recognition of name conflicts, recognition of structural conflicts and recognition of inter-schema properties.

The output from this phase is a description of schema element similarities and a description of differences and a description of inter-schema properties.

C. Conforming the Schemata

Also conforming the schemata has received some attention by other researches. For instance, Lee and Ling

 ISBN: 978-1-61208-329-2

[25] called it the most critical phase and Spaccapietra and Parent [35] the key issue in schema integration.

In conforming the schemata, the recognized similarities and differences are resolved by adjusting the input schemata.

The recognized inter-schema properties are also used in this phase. However, its full value is shown in merging and restructuring.

The output of this phase is a set of revised schemata.

D. Merging and Restructuring

The first task performed in this phase is to merge the revised input schemata into one global intermediate schema. The intermediate schema is then restructured e.g., detected inter-schema properties are introduced to semantically enrich the schema. Furthermore, schema elements that are truly redundant are recognized and removed from the schema. Merging the schemata as well as restructuring the schemata results in a new intermediate schema.

Before the integrated schema is handed over to the developers implementing the information system, the schema is again analyzed, meaning that the schema is checked and verified according to several quality criteria [1][3] and/or quality factors [29].

The result of this phase should be a high quality schema that can be passed on to the following phases in which the information system is implemented.

IV. SOME BEST PRACTICES REGARDING SCHEMA QUALITY

Both the Guidelines of Modeling (GoM) [5] and the quality factors explained in Moody and Shanks [29] focus on: improving quality of the modeling process and quality of the resulting product (i.e., the conceptual model).

Both frameworks are a good basis for understanding the quality of the conceptual modeling integration process. The Guidelines of Modeling are a more strategic framework for covering all aspects of enterprise models (e.g., data, organization, processes, and behavior). The work in Moody and Shanks [29] focuses on data schemata more specifically ER data models.

Because of its more operational focus, we adopted the following practices from Moody and Shanks [29] for the integration process in order to fulfill the quality factors and improve the quality of the modeling:

- Introducing a specific kind of stakeholder – the information architect
- Introducing continuous quality checks and reviews.

As well as the general practices:

- Stakeholder participation
- Introducing naming conventions, standards, etc.

We will adopt these practices for the integration process as well.

The information architect (in [29] called data administrator) is a person that was introduced to review a schema with respect to the other data schemata (models) existing in the enterprise.

According to Moody and Shanks [29], who proposed continuous checks and reviews for schema development, reviews must not only be made at the end, but also before and during a development step. Such reviews should support the total quality management aim that the quality checks and reviews should not detect errors, but prevent errors.

The participation of different kind of stakeholders is a successful technique used in Information Systems and Enterprise Engineering. Since the schemata (models) represent the knowledge of ideas of people with different backgrounds, it is necessary that different stakeholders are involved.

The introduction of an information architect also implies the usage and management of standards (e.g., what a schema should look like syntactically, which terms are used and preferred to other terms, etc.).

V. APPLYING BEST PRACTICES TO INTEGRATION TASKS

In general the best practice of "continuous improvement" is a driver for the whole integration process. Although quality is usually considered in or even after the last step of schema integration, we will follow the principle of introducing quality as early as possible here. Therefore we will focus on tasks needed in earlier steps. We will relate them to the best practices in order to improve them. These tasks are: *choosing the right integration strategy*, *choosing the right conflict resolution methods for the chosen level of abstraction* and *introducing inter-model properties to improve and clarify dependencies*. The first is a task that has to be done during pre-integration. The second and the third tasks are at least executed during the 2nd and 3rd steps.

A. Choosing the Right Integration Strategy

In Batini et al. [3], several strategies are proposed for integrating end-user schemata (views). They distinguish between binary and n-ary integration strategies. Among binary strategies a ladder strategy [2] or a balance strategy [3] can be chosen. In the ladder strategy, the stakeholders start with two views. They integrate these two views. Afterwards the first integrated schema is compared and matched with another view, and so on. In the balanced strategy, two views are integrated in an intermediate schema. This intermediate schema is integrated with other intermediate schemata until the global schema is reached. The n-ary strategies are the one-shot strategy (a global schema is generated at once from all views) and the iterative strategy. The iterative strategy uses one shot strategies only to produce intermediate schemata. These schemata are then integrated with each other (two or more). Integrated schema can also be integrated with views. The iterative strategy can be seen as a mixture of the previous three strategies.

1) Continuous Checks and Reviews

For continuous checks and reviews, the integration strategy must prove enough definite points of inspections.

A one shot strategy can be excluded as a good strategy by applying this best practice. Otherwise, it would mean that a global schema exists without any intermediate results. If intermediate results are missing, then it is impossible to identify definite review milestones. Following the best practice of continued improvement given in literature, an iterative, and balanced or ladder strategy should be applied.

Doing so each time, an intermediate schema is generated, this intermediate schema can be reviewed.

It cannot be determined which of the other three strategies should be chosen since all these strategies have intermediate points where schemata can be reviewed before or during integration. The choice between a balanced, a ladder, or an iterative strategy, is a pragmatic decision of available time for the integration and other environmental factors.

2) Information architect, stakeholder participation and standards

Since integration is part of modeling, an information architect, stakeholder involvement, and standards are also necessary for integration.

The information architect has to assure that a certain intermediate schema as well as the views already integrated is in compliance with existing schemata in the enterprise. Stakeholders check the semantic correctness and completeness with respect to a certain examined section represented by the views (schemata) or intermediate schemata. For both the information architect and stakeholder involvement, strategies that have more intermediate points for discussions and reviews (i.e., ladder, balanced, iterative strategy) are more supportive.

Standards help to check if the schema is syntactically correct and if terms are used in compliance with the enterprise. It is therefore necessary that standards are used. Standards equally drive all the four strategies (one shot, ladder, balanced and iterative). Knowledge repositories, such as stemmers and lemmatizers, could be used to facilitate the task of checking that terms are used in a correct way. Drawing tools might also aid in the modeling process and be used to check that the schema is syntactically correct.

B. Choosing the Right Conflict Resolution Methods for the Chosen Level of Abstraction

In the phase comparison of the schemata two schemata are compared for the purpose of finding similarities as well as differences, often more generally referred to as conflicts. In the phase that follows, conforming the schemata, the conflicts are resolved. However, the same resolution methods are often proposed (and used) for implementation-neutral schemata and implementation-dependent schemata. Using different conflict resolution methods for different levels of abstraction is very important since an implementation-neutral schema is often used in the earlier phases of information systems development while an implementation-dependent schema in the later phases is close to programming and technical issues.

The purpose of the schema under design may also vary. Boman et al. [9] address this in their four schema purposes as follows: "A schema can serve at least four different purposes. First, it can be used for clarifying the language used in an organisation. Secondly, it can be used for making explicit the rules that prevail in an organisation, which helps to criticise them and possibly to draw up new rules. Thirdly, a schema can be useful for reviewing existing information systems. Fourthly, a schema can be used for developing a new information system" (p. 122).

One way of combining the mentioned two levels of abstraction with the four purposes stressed by Boman et al. [9] might be as follows. First, clarifying the language is closely related to the implementation-neutral level since then the designers are interested in concepts and connections between concepts rather the implementation-dependent issues and trying to reduce the number of concepts and connections [6]. Secondly, making explicit the rules is also closely connected to the implementation-neutral level since rules must be expressed so that all stakeholders understand the rules and therefore also can criticize them. Thirdly, using a schema for reviewing an already existing information system is closely related to the implementation-dependent level since the schema describes an already implemented information system. Finally, using a schema during the development of a new information system refers to both levels of abstraction. This is motivated since the designers might use different schemata during the development of the information system. The designers might also use different modeling languages dependent on phase and focus in the information systems development process. If choosing the right conflict resolution methods for the chosen level of abstraction are ignored the integrated schema might not only suffer semantic loss but also being hard to understand.

1) Continuous Checks and Reviews

Having designed the schemata on the chosen level of abstraction and in comparison of the schemata recognized the conflicts between two schemata, it is important that in conforming the schemata the right conflict resolution methods are used. However, this is not always the case. Therefore, while applying the best practice of continued checks and reviews, it is important to check that the right conflict resolution methods have been chosen for the current level of abstraction. If the wrong conflict resolution method has been introduced, it should not only be recognized during continuous checks and reviews but also changed to the right one. This should in the end contribute to an integrated schema with high quality since an additional check and review has been conducted. For instance, if during the comparison of the schemata, we recognize a synonym conflict (e.g., article in schema 1 and product in schema 2), it should during conforming the schemata be resolved. However, if the schemata are designed on an implementation- neutral level it is important that all concept names and dependencies are kept as long as possible since they might be of importance for one or several stakeholders. We should therefore not rename of one or both concept names, which is one of the most ordinary proposed resolution methods for a synonym conflict, but instead introduce a resolution method that keeps both concept names. One way to fulfill this could be to introduce mutual inheritance dependency described as A and B are synonyms if and only if A inherits B and B inherits A [20].

2) Information architect, stakeholder participation and standards

While doing schema integration, it is important that both the information architect as well as the stakeholders are very much involved. By involving these actors several of the mentioned pitfalls should be recognized and addressed as

early as possible in the integration process (the current iteration cycle) and not included into the global integrated schema. This is the case since it is the stakeholder and the information architect that possess the knowledge of how their concepts should be named and which concepts should be connected to each other. However, the information architect also has to take into account already existing data schemata within the enterprise and therefore should have a holistic perspective. A stakeholder might instead focus on integrating a schema of a specific department.

Naming conventions, standards and ontologies, so called knowledge repositories, might also exist in the enterprise that need to be taken into account in the integration process. However, it is important that these naming conventions and standards do not restrict the naming of concepts which impoverish the language used in the schema but instead are used as a tool to facilitate the integration process. Therefore standards should not enforce the usage of one concept name but instead give guidelines on how concepts names should be used such as name concepts in singular.

C. Introducing Inter-Schema Properties to Improve and Clarify Dependencies

Another task in comparison of the schema is the recognition of inter-schema properties. An inter-schema property is not really a conflict, but instead it describes a specific link between two concepts. The two most common inter-schema properties described in the literature are hypernym-hyponym dependencies (often referred to as "is-a") and holonym-meronym dependencies (often referred to as "part-of"). When an inter-schema property has been recognized it is documented and passed to the next phase in the schema integration process in which it is used. However, its full value is shown in the last phase of the schema integration process where the inter-schema properties are used as guidance while merging and restructuring the global integrated schema. Introducing inter-schema properties in the schema integration process is of great importance since an inter-schema property has a clear meaning and should therefore also be used not only to clarify and improve a specific meaning between two concepts but also to reduce the number of concepts in the integrated schema if possible. However, reducing the number of concepts should be done carefully. Deleting a concept might reduce the quality of the integrated schema instead of improving its quality. In the worst case, it violates the completeness quality factor addressed in [29].

Finally, it should be noted that a holonym-meronym dependency might be of two types: aggregation and composition in which composition is the stronger.

1) Continuous Checks and Reviews

In the comparison of the schemata, the binary strategy (or n-ary iterative) is used while recognizing similarities and differences, e.g., inter-schema properties, between two schemata. When an inter-schema property has been recognized, it should be documented and passed on to the following phases in the integration process. At the end, the inter-schema property should not only in merging and restructuring be treated as a source of semantic improvement but also be used as guidance, a knowledge repository, while merging and restructuring the integrated schema.

However, since an inter-schema property is used in at least two phases in the integration process, it is substantially important that the inter-schema property is used in a right way and not instead polluting the input schemata and/or the integrated schema. An even worse scenario could be that the inter-schema property is used in a wrong way causing semantic errors. Applying the best practices of continuous checks and reviews is therefore of great importance to improving not only the quality of the integrated schema as such but also to verifying that the inter-schema property is used in a correct way.

For instance, if we in comparing the schemata have recognized not only a hypernym-hyponym dependency between concept A and B in schema 1 but also a hypernym-hyponym dependency between concept B and A in schema 2, problems might later on be introduced into the integrated schema. The inter-schema dependencies are documented and passed on to the following phase in which the schemata are adjusted to solve the recognized conflicts and inter-schema properties. Having done that, the schemata (and some extra information resources) are passed to the last phase in which the schemata are integrated. In the worst case, both hypernym-hyponym dependencies described above are introduced to the integrated schema causing what is sometimes called reverse subset relationship [1] or cyclic generalization [34]. However, applying the best practice of continuous checks and reviews, this problem should be recognized and resolved in the current iteration cycle and not be left to later iterations in the integration process.

2) Information architect, stakeholder participation and standards

Introducing inter-schema properties should result in a semantic richer schema since the inter-schema properties should have a clear meaning compared with, for instance, the association dependency with or without specified cardinality. However, introducing new schema constituents might also result in new problems and errors. Involving information architect as well as stakeholders are also of great importance, since these actors possess the knowledge of their specific domain. However, the information architect has to take into account the schemata already existing within the enterprise and make sure that these match the new schema being developed. On the other hand, a stakeholder from one department might instead only focus on his/her part of the schema (model) and therefore argue for his/her point of view in the integration process.

Finally, naming conventions, standards as well as ontologies, so called knowledge repositories, might also exist within the enterprise. Ontology, or even domain ontology, might for instance be useful when deciding how to resolve the cyclic generalization dependency. This is the case since a description on how concept A and concept B are dependent might be stated in ontology.

 ISBN: 978-1-61208-329-2

VI. CONCLUSION AND FUTURE WORK

In this paper, we have addressed schema quality within the schema integration process. In doing so, we have focused on four best practices of quality improvement given in the literature and three specific integration tasks that should increase the quality of the schema being designed. The four best practices addressed are: *continuous checks and reviews*, *information architect*, *stakeholder participation* and *standards*. The three integration tasks addressed are: *choosing the right integration strategy*, *choosing the right conflict resolution methods for the chosen level of abstraction* and *introducing inter-schema properties to improve and clarify dependencies*. Within each integration task we have also addressed how knowledge repositories might be used to aid in the process of producing a high quality schema.

To conclude (see also Table I), the four best practices used for conceptual modeling if addressed in connection to schema integration can improve the three mentioned tasks and hence the integration process. Continuous checks and reviews, information architect and stake holder participation can be drivers for choosing the right integration strategy. Standards do not have an influence on this task. Continuous checks and reviews, standards, information architect and stakeholder participation are essential in the conflict resolution task. The more conflicts are checked and resolved the better. The more the stakeholders and the information architect are involved, the more conflicts can be resolved. Standards support this task as long as they do not restrict the enterprise specific naming of concepts.

For the inter-schema property introduction, which is used in at least two phases of the integration process, continuous checks and reviews can help verify that the inter-schema property is used in the correct way. Stakeholders and the information architect are the ones who possess the domain knowledge and can thus support the aim to get a semantically richer schema with clear meanings. Standards and ontologies are useful to support the detection of inter-schema properties.

In the long run these improved tasks contribute to a high quality integrated schema.

In future, we will continue our work on identifying particular best practices for quality improvement for other tasks of the integration process. Specifically, we will look at other tasks of the phases (e.g., recognition of name conflicts and structural conflicts, merging the revised schema). We will also investigate the process from the perspective of aspects of quality (e.g., the SEQUAL views – physical, empirical, syntactical, semantic quality).

TABLE I. BEST PRACTICES AND INTEGRATION TASKS

Best Practice	Choosing the Right Integration Strategy	Choosing the Right Conflict Resolution Methods for the Chosen Level of Abstraction	Introducing Inter-Schema Properties to Improve and Clarify Dependencies
Continuous Checks and Reviews	are facilitated by the ladder, balanced and iterative integration strategy.	are the enablers to verify that the schemata illustrate the chosen level of abstraction during the whole integration process.	are the enablers to verify that the inter-schema properties are used in a correct way during the whole integration process.
Information Architect	checks that the schemata are in compliance with existing enterprise schemata.	checks that the choosen conflict resolution methods are in compliance with existing enterprise schemata.	checks that the introduced inter-schema properties are in compliance with existing enterprise schemata.
Stakeholder Participation	is the enabler to check the semantic correctness and completeness of the schemata.	is the enabler to check that choosen conflict resolution methods are semantically correct and that the schema is complete.	is the enabler to check that the introduced inter-schema properties are semantically correct and that the schema is complete.
Standards	help in the process of checking that the schemata are syntactically correct and that terms are used in compliance with the enterprise schemata	help in the process of introducing the correct resolution method for not only naming conflicts but also structural conflicts.	help in the process of introducing the correct inter-schema propterty and help in the process of introducing the inter-schema property in a correct way.

REFERENCES

[1] C. Batini, S. Ceri, and S. B. Navathe, Conceptual Database Design an Entity Relationship Approach. Redwood City: Benjamin/Cummings Publishing Company, 1992.

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

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

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

[5] J. Becker, M. Rosemann, C. von Uthman, "Guidelines of Business Process Modeling," W. van der Aalst et al. (Eds.):

Business Process Management, LNCS 1806, Springer-Verlag Berlin Heidelberg, 2000, pp. 30-49.

[6] P. Bellström, "On the Problem of Semantic Loss in View Integration," in Information Systems Development Challenges in Practice, Theory, and Education. Volume 2, C. Barry, K. Conboy, M. Lang, G. Wojtkowski, and W. Wojtkowski, Eds. New York: Springer, 2009, pp. 963-974.

[7] P. Bellström and J. Vöhringer, "A Semi-Automatic Method for Matching Schema Elements in the Integration of Structural Pre-Design Schemata," International Journal on Advances in Intelligent Systems. vol. 4 (3 & 4), 2011, pp. 410-422.

[8] 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 Press, 2009, pp. 110-115.

[9] M. Boman, Jr. J. A. Bubenko, P. Johannesson, and B. Wangler, Conceptual Modelling. London: Prentice Hall, 1997.

[10] S. S. Cherfi, J. Akoka, and I. Comyn-Wattiau, "Perceived vs. Measured Quality of Conceptual Schemas: An Experimental Comparision," Proceedings of Tutorials, Posters, Panels and Industrial Contributionof the Twenty-Sixth International Conference on Conceptual Modeling (ER 2007), vol. 83, 2007, pp. 185-190.

[11] 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), 2002, pp. 747-762.

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

[13] L. Ekenberg and P. Johannesson, "A Formal Basis for Dynamic Schema Integration," Conceptual Modeling – ER'96. LNCS 1157, 1996, pp. 211-226.

[14] S. Fan, L. Zhang, and Z. Sung, Z., "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

[15] G. Fliedl, C. Kop, H. C. Mayr, W. Mayerthaler, and C. Winkler, "Linguistically based requirements engineering – The NIBA project," Data & Knowledge Engineering, vol. 35 (2), 2000, pp. 111-120.

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

[17] 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.

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

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

[20] R. Gustas, Semantic and Pragmatic Dependencies of Information Systems. Kaunas: Kaunas Technologija, 1997.

[21] 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.

[22] P. Johannesson, Schema Integration, Schema Translation, and Interoperability in Federated Information Systems. Dissertation. Stockholm University & Royal Institute of Technology No. 93-010-DSV, 1993.

[23] J. Krogstie, Model based Development and Evolution of Information Systems – A Quality Approach. London: Springer, 2012.

[24] J. A. Larson, S. B. Navathe, and R. Elmasri, "A Theory of Attribute Equivalence in Databases with Application to Schema Integration," Transactions on Software Engineering, vol. 15 (4), 1989, pp. 449-463.

[25] 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.

[26] 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), 1998, 1115-1130.

[27] O. L. Lindland, G. Sindre, and A. Solvberg, "Understanding Quality in Conceptual Modeling," IEEE Software, vol. 11 (2), 1994, pp. 42-49.

[28] 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, vol. 23 (1), 1997, pp. 59-78.

[29] D. L. Moody and G. G. Shanks, "Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework," Information Systems Journal, vol. 28 (2), 2003, pp. 619-650.

[30] D. L. Moody, "Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions," Data & Knowledge Engineering, vol. 55 (3), 2005, pp. 243-276.

[31] S. Ram and V. Ramesh, "A Blackboard-Based Cooperative System for Schema Integration," IEEE Expert, 1995, vol. 10 (3), pp. 56-62.

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

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

[34] W. Song, Schema Integration – Principles, Methods, and Applications. Dissertation. Stockholm: Stockholm University & The Royal Institute of Technology No. 95-019, 1995.

[35] S. Spaccapietra and C. Parent, "View Integration: a Step Forward in Solving Structural Conflicts," IEEE Transactions on Knowledge and Data Engineering, vol. 6 (2), 1994, pp. 258-274.

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

[37] 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.

 ISBN: 978-1-61208-329-2