Building Metadata Bridges from the Enterprise World to the Cloud Platforms

Dippy Aggarwal

Supervised by: Karen C. Davis

University of Cincinnati, Cincinnati, Ohio, USA
aggarwdy@mail.uc.edu

Abstract. The emerging paradigm of cloud computing presents a new alternative for deploying large-scale, business analytics applications. However, the newer data structures supported by cloud platforms contribute to the proliferation of data models in popular usage. Model-based heterogeneity raises challenges in facilitating exchange and integration of information distributed across multiple data sources expressed in varied formats. Leveraging the capabilities of the new cloud storage models for data warehousing also becomes a tedious job due to the schema mapping issues. We present the concept of developing a generic mapping framework at the schema level to address the problem of schema interoperability across as well as within three different families of data models: operational, data warehouse, and cloud-based data models. This bridges the gap between enterprise and cloud platforms by providing formalism for mapping multidimensional constructs to cloud platforms.

Keywords: cloud databases, model management, meta-model, multidimensional schema, schema interoperability.

1 Introduction

With an increasing number of companies embracing the cloud computing paradigm and analytical data management applications speculated as potential candidates for deployment on the cloud [A09], the field of data warehousing is confronted with a new set of challenges [K12]. Data warehousing's growing need for fast analysis can benefit from a cloud environment's scalability. However, cloud deployment poses a challenge for data warehouse design for restructuring existing multidimensional designs to non-traditional cloud data stores.

The need for developing a mapping framework arises from the different models adopted by different sources in order to cater to the respective domains and applications of the systems they are designed to support. While on one hand having different model representations is an advantage for the flexibility they offer to the user in selecting the right representation based on their requirements, they also introduce model-based heterogeneity that raises challenges for schema interoperability and integration.

Schema interoperability refers to the process of transforming a schema expressed in one model to an equivalent schema in another model. Schema integration involves bringing multiple schemas together by reconciling their differences and capturing their information collectively in an integrated global schema.

There is an extensive body of research on the problem of ensuring schema interoperability [CB12, EER12, FGM+13, ACK+11, K05, BMC06, HFM06] and integration [BLN86, LSS93, B00, CHK+07]. The solution space for these problems is impacted by the proliferation of data models in practice, thus creating a demand for a flexible framework to map the constructs from one model to another. The novelty of our work lies in providing a generic mapping approach to address schema interoperability and integration in a flexible and extensible manner across as well as within three families of data models.

We organize models into three different families, namely operational, data warehouse, and cloud models, based on the needs they serve. While operational models are geared towards designing transaction-based (OLTP) applications, utilizing normalized table structures, data warehouse models are designed for processing analytical workloads (OLAP). Cloud platforms form emerging data storage and processing paradigm for hosting both OLTP and OLAP applications. We consider them as a separate family of models because of the non-traditional data model that they follow. With each family capturing its requirements in a different data model, a mapping framework is required to ensure interoperability across applications executing in these different families.

We adopt the concept of having a metamodel proposed by Atzeni et al. [ACB05] in order to address the proliferation of data models in a flexible manner. A metamodel is designed to serve as a generic representation of all its representative models. Given a set comprising of *n* models, each individual model needs to have a translation with respect to the metamodel only, rather than having one translation for each of the other models in the set. This leads to a linear (2n) rather than quadratic (n^2) number of translations among them [ACT+08]. Apart from serving as a basis for flexible integration of new sources into an existing data store, adopting a metamodeling approach also facilitates formulation of queries on a global, unified representation, thus helping a query writer to avoid being familiar with the schemas of each source [CHM03]. We consider a metamodel for each of the three different families of models (operational, data warehouse, and cloud). Having a different metamodel for each representative class of models enables capturing the specific semantics and constructs of each individual category in a precise and comprehensible manner.

Figure 1 shows a high level architecture for our metamodeling approach encompassing the three different families of data models defined above. The architecture consists of three main subsystems. Subsystem 1 represents the operational data family. The goal of this system is to capture specific operational models in a unified format defined by the operational metamodel. The outcome of Subsystem 1 is a set of operational metamodel instances derived from mapping schemas expressed in specific models to the metamodel. Subsystem 2 represents the data warehouse family including the dictionary-based structure for the data warehouse metamodel and the representative models of the family. The schema merging algorithms developed over the data warehouse metamodel belongs to this subsystem. Subsystem 3 represents the cloud model

that serves as the implementation platform for the data warehouse. With the differences in the extent of features supported by each model, we recognize the possibility of information loss during the transformation process. The component, *non-mapped schema elements* address this concern by storing any elements that could not be mapped during the transformation process.

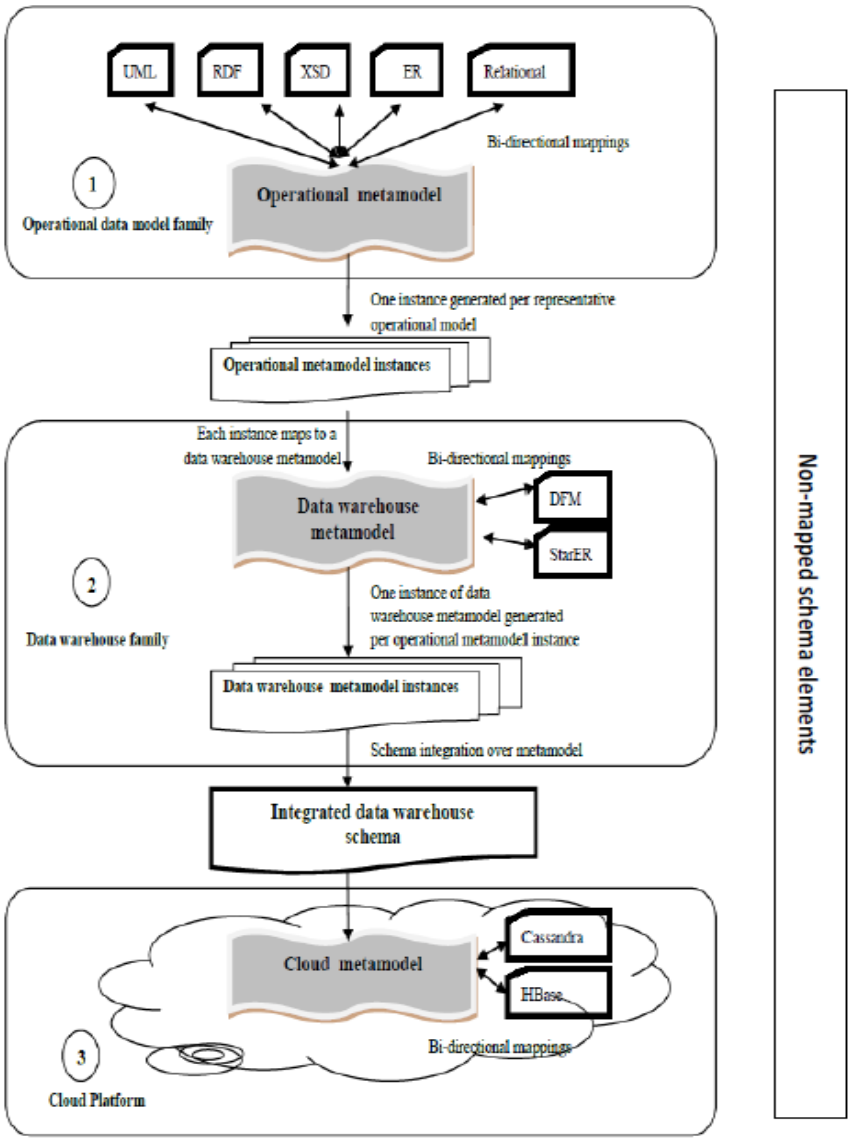

Fig. 1. High-level Architecture of our approach

The contribution of our work lies in achieving the following primary goals:

1. Developing a single comprehensive solution to address the problem of schema interoperability across as well as within three different families of data models: operational, data warehouse, and cloud-based data models.
2. Providing a formalism for developing a generic, extensible, and semi-automated mapping framework by devising algorithms over metamodels, instead of individual specific models.
3. With the operational sources contributing data to the data warehouse, their schemas need to be merged. This requires synthesizing the proposed schema merging algorithms and applying them to our data warehouse metamodel to generate an integrated multidimensional schema.

The rest of the paper is organized as follows. In Section 2 we enumerate our research objectives. Section 3 presents the concept that provides the foundation for the proposed research. Section 4 discusses the background and related work. In Section 5 we present a sketch of our approach and Section 6 concludes the paper, highlighting our expected contributions.

2 Research Objectives

In order to develop a generic and extensible framework for mapping schemas expressed in heterogeneous models, we have identified the following research challenges:

1. There has been a proliferation of data models in popular usage, increasing the one-to-one mappings to support interoperability between each pair of models. Develop a metamodel for each data model family in order to support interoperability in a flexible and extensible manner, thus avoiding pairwise mappings.
2. With the formalisms defined above and in order to support intra-family and inter-family mappings, the instances of each family need to be expressed in the unified representation, i.e. metamodel, of their respective family. Thus, develop a set of algorithms to support mappings between (a) native models to the family metamodel, and (b) metamodel of different families.
3. Develop a set of integration algorithms over the data warehouse metamodel to support schema merging over multidimensional models.
4. Develop a software prototype demonstrating the effectiveness and automation of our proposed algorithms.

3 Research Foundations

This section presents an overview of the dictionary-based metamodeling approach that provides the foundation for the proposed research.

The dictionary-based metamodeling approach [ACB05] provides an implementation of the *ModelGen* operator proposed by Bernstein [B03] which involves translating a schema from one model to another.

The first step in model management involves generating a dictionary for each model. A dictionary consists of a table for each of the constructs in a model. The tuples in the dictionary tables correspond to the elements in the schema [V08]. To illustrate this translation, consider the ER diagram represented in Figure 2.

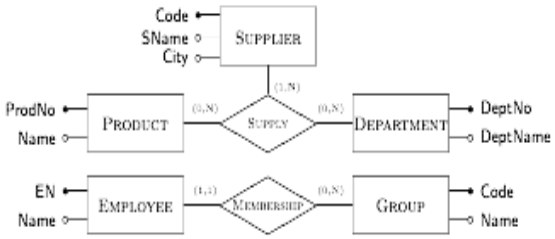

Fig. 2. Two Schemas in ER model [ACB05]

Transforming the two ER schemas to a dictionary-based format (as an example, representing constructs Entity and Relationship) results in the tables depicted in Figure 3.

SCHEMA

OID	Name
s1	1st ER Schema
s2	2nd ER Schema

RELATIONSHIP

OID	Name	Schema
r1	Supply	s1
r2	Membership	s2

ENTITY

OID	Name	Schema
e1	Supplier	s1
e2	Product	s1
e3	Department	s1
e4	Employee	s2
e5	Group	s2

Fig. 3. Dictionary for ER Model [ACB05]

Similarly, a dictionary model needs to be defined for each representative model. Once we have model specific dictionaries, the next step is to generate a dictionary that maps model-specific constructs to model-independent constructs, termed as *metaconstructs*. Atzeni et al. describe this approach in detail [ACB05].

We contribute to this metamodeling approach in two ways, (1) by augmenting the dictionary structures for models in the operational data family, and (2) creating dictionary-based formalisms for data warehouse and cloud models. The former includes creating new tables for incorporating additional models or adding columns in the existing tables defined by the authors. The second contribution is providing an enhancement of the application of their approach to multidimensional and cloud model schemas.

4 Related Work

Several surveys [RB01, BMR11, MP06, SE05] focus on techniques for resolving model-agnostic schematic heterogeneities. The model-agnostic schemas can differ in terms of linguistics, constraints, granularity and the structure of the captured information, while being represented in the same model. Rahm et al. [RB01] classify approaches as instance or schema level mappings and linguistic or constraint based approaches. Bernstein et al. [BMR11] summarize advancements in adopting a generic schema matching approach. Manakanatas et al. [MP06] provide a tabular comparative study of the various tools implemented in the literature for schema matching, including Cupid [MBR01], Clio [HMH01], Protoplasm [BMP+04] and COMA++ [AMD+05].

There have been several research efforts [HL01, PMI+03, BKK04, LL06, TC07, N09] that recognize the significance of addressing model-based schematic heterogeneities. The source-driven design methodology for creating a data warehouse design addresses mapping between data model families, namely the operational and multidimensional data model families [BS12, GRG05, JRS+12, CMR10]. As another example, consider the two dominant standards of representing information on the web, XML and RDF. While XML schemas provide support for representing structural information, RDF excels at expressing the semantics in a machine-understandable syntax. Hunter and Lagoze [HL01] highlight the advantage of bringing them together in order to exploit their complementary features and that requires identification and resolution of the heterogeneity in the two modeling schemes.

In the context of existing solutions for bridging schema-based, relational stores to cloud models, representative works include Google's megastore [BBC+11]. Its core idea is to blend the scalability provided by cloud models with the structure and ACID guarantees supported by relational, schema-based models. The focus of our work is in providing formalization for transforming information captured in schema based models to cloud's models.

Although these approaches make novel contributions in the context of ensuring schema interoperability, we recognize the lack of a unified framework that incorporates models from all the three data model families identified in our work, under one comprehensive approach. We leverage the concept of metamodeling in developing such a framework.

There have been contributions that have proposed metamodeling based solutions towards ensuring schema interoperability [BOA+13, JD13, JCF03, ZC05]. We drive inspiration from them in adopting the metamodeling paradigm in our work thus aligning our efforts with the Model Driven Engineering methodology. Bazhar et al. [BOA+13] propose a model sharing framework, Persistent Meta-Modeling Systems (PMMS), that stores meta-models, models, and instances in a single database but also supports model management operations. Judson et al. [JCF03] and Zepeda et al. [ZC05] specify model transformations at the metamodel layer. In another work, Janga et al. [JD13] introduce the Schema Extended Context Free Grammar (SECFG) to ensure interoperability between XSD and DTD which represent two modeling schemes for XML documents. The grammar serves to provide a uniform representation to the disparate XML formats.

5 Proposed Approach

In this section, we present architecture for our proposed research encompassing three families of data models.

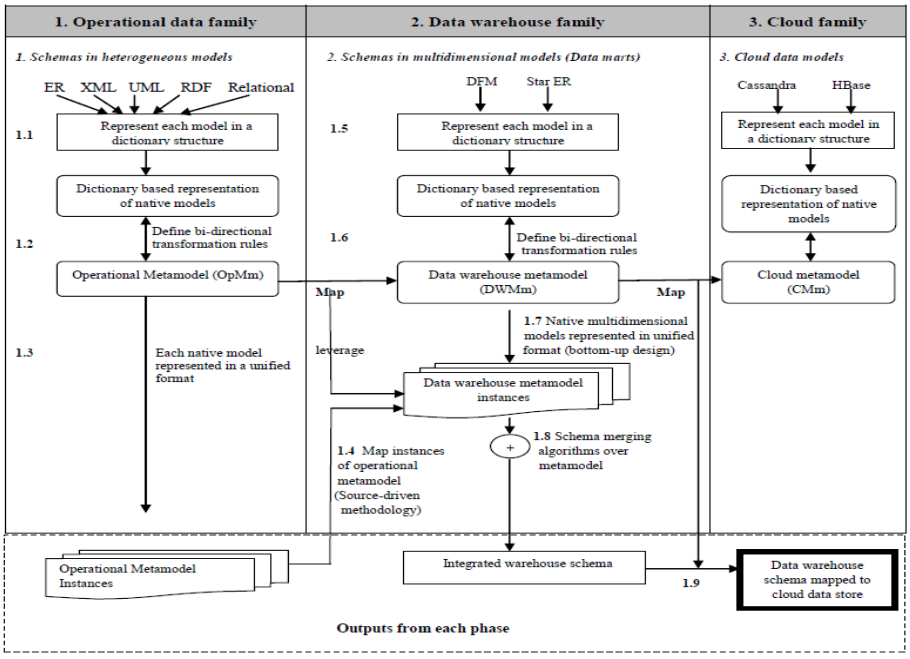

Fig. 4. Our Proposed Approach

The architecture in Figure 4 consists of three main subsystems. The first subsystem (leftmost box) represents the operational data family. The goal of this system is to capture all the specific operational models in a unified format defined by the operational metamodel. The outcome of subsystem is a set of operational metamodel instances derived from mapping schemas expressed in specific models to the metamodel. The subsystem represents the data warehouse family including the dictionary-based structure for the data warehouse metamodel and the representative models of the family. The schema merging algorithms developed over the data warehouse metamodel belongs to this subsystem. The third subsystem represents the cloud platform that serves as the implementation platform for the data warehouse design. The dictionary-based meta-modeling approach [ACB05] is depicted as a two-step process across all the three families.

a) *Creating a model specific dictionary:* The native representative models in each family (such as, ER, RDF etc. in operational models, DFM and StarER in data warehouses and column-family and key-value stores in cloud models) are represented

in a tabular, dictionary based format. A dictionary consists of a table for each of the constructs in the model. As an example, for the ER model, dictionary items are entities, attributes, and relationships.

b) *Constructing a metamodel*: A generic model is defined for each family consisting of metaconstructs. Metaconstructs define the limited generic set of model-independent constructs. The second step involves drawing a correspondence between each model's specific constructs to the metaconstructs. Algorithms are developed for formalizing and automating transformation rules.

6 Expected Contributions

We have presented a high-level architecture for addressing the problem of model-based schema heterogeneity. In summary, our research will contribute a framework for the data warehouse design to address the problem of interoperability between the enterprise world and the cloud environment. Our work is still in an early stage and clearly a lot of work remains to be done in order to achieve the identified objectives. In the near future, we aim to enhance Atzeni's approach [ACB05] to enable its application to data warehouse schemas and cloud models. We also aim to enrich their metamodel by incorporating additional models in the family of operational models. We envision the application of our work in providing an end-to-end solution for data warehouse design with operational data serving as its source and investigating cloud platforms as the target implementation.

References

[A09] Abadi, Daniel J, "Data Management in the Cloud: Limitations and Opportunities," Proceedings of the IEEE Data Engineering Bulletin, Vol.2, No. 1, 2009, pp. 3-12.

[ACB05] Atzeni, Paolo, Paolo Cappellari, and Philip A. Bernstein., "A Multilevel Dictionary for Model Management," Proceedings of the International Conference on Conceptual Modeling, Klagenfurt,Austria, October 24-28, 2005, pp. 160-175.

[ACT+08] Atzeni, Paolo, Paolo Cappellari, Riccardo Torlone, Philip A. Bernstein, and Giorgio Gianforme. "Model-independent schema translation." The VLDB Journal—The Interna tional Journal on Very Large Data Bases 17, no. 6 (2008): 1347-1370.

[AMD+05] Aumueller, David, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. "Schema and ontology matching with COMA++." In Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 906-908. ACM, 2005.

[ACK+11] Alexe, Bogdan, Balder TEN Cate, Phokion G. Kolaitis, and Wang-Chiew Tan. Characterizing schema mappings via data examples." *ACM Transactions on Database Systems (TODS)* 36, no. 4 (2011): 23.

[B00] Behrens, Ralf. "A grammar based model for XML schema integration." In Advances in Databases, pp. 172-190. Springer Berlin Heidelberg, 2000.

[B03] Bernstein, Philip A. "Applying Model Management to Classical Meta Data Problems." In CIDR, vol. 2003, pp. 209-220. 2003.

[BBC+11] Baker, Jason, Chris Bond, James Corbett, J. J. Furman, Andrey Khorlin, James Larson, Jean-Michel Léon, Yawei Li, Alexander Lloyd, and Vadim Yushprakh. Megastore: Providing Scalable, Highly Available Storage for Interactive Services."

In CIDR, vol. 11, pp. 223-234. 2011.
[BLN86] Batini, Carlo, Maurizio Lenzerini, and Shamkant B. Navathe. "A comparative analysis of methodologies for database schema integration." ACM computing surveys (CSUR) 18, no. 4 (1986): 323-364.
[BOA+13] Bazhar, Youness, Yassine Ouhammou, Yamine Aït-Ameur, Emmanuel Grolleau, and Stéphane Jean. "Persistent meta-modeling systems as heterogeneous model reposito ries." In Model and Data Engineering, pp. 25-37. Springer Berlin Heidelberg, 2013.
[BKK04] Bernauer, Martin, Gerti Kappel, and Gerhard Kramler. Representing XML schema in UML–A Comparison of Approaches. Springer Berlin Heidelberg, 2004.
[BMP+04] Bernstein, Philip A., Sergey Melnik, Michalis Petropoulos, and Christoph Quix. Industrial-strength schema matching." ACM SIGMOD Record 33, no. 4 (2004): 38-43.
[BMC06] Bernstein, Philip A., Sergey Melnik, and John E. Churchill. "Incremental schema matching." Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006.
[BMR11] Bernstein, Philip A., Jayant Madhavan, and Erhard Rahm. "Generic schema matching, ten years later." Proceedings of the VLDB Endowment 4, no. 11 (2011): 695-701.
[BS12] Berger, Stefan, and Michael Schrefl. "FedDW global schema architect: UML-based design tool for the integration of data mart schemas." In Proceedings of the fifteenth international workshop on Data warehousing and OLAP, pp. 33-40. ACM, 2012.
[CHM03] Camillo, Sandro Daniel, Carlos Alberto Heuser, and Ronaldo dos Santos Mello. Querying heterogeneous XML sources through a conceptual schema." In Conceptual Modeling-ER 2003, pp. 186-199. Springer Berlin Heidelberg, 2003.
[CHK+07] Chiticariu, Laura, Mauricio A. Hernández, Phokion G. Kolaitis, and Lucian Popa. Semi-automatic schema integration in clio." In Proceedings of the 33rd international conference on Very large data bases, pp. 1326-1329. VLDB Endowment, 2007.
[CMR10] Carmè, Andrea, Jose-Norberto Mazón, and Stefano Rizzi. "A model-driven heuristic approach for detecting multidimensional facts in relational data sources." In Data Warehousing and Knowledge Discovery, pp. 13-24. Springer Berlin Heidelberg, 2010.
[CB12] Chen, Liang Jeff, Philip A. Bernstein, Peter Carlin, Dimitrije Filipovic, Michael Rys, Nikita Shamgunov, James F. Terwilliger, Milos Todic, Sasa Tomasevic, and Dragan Tomic. "Mapping XML to a wide sparse table." In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 630-641. IEEE, 2012.
[EER12] Peukert, Eric, Julian Eberius, and Erhard Rahm. "A self-configuring schema matching system." Data Engineering (ICDE), 2012 IEEE 28th International Conference.
[FGM+13] Franceschet, Massimo, Donatella Gubiani, Angelo Montanari, and Carla Piazza. "A graph-theoretic approach to map conceptual designs to XML schemas." ACM Transac tions on Database Systems (TODS) 38, no. 1 (2013): 6.
[GRG05] Giorgini, Paolo, Stefano Rizzi, and Maddalena Garzetti. "Goal-oriented requirement analysis for data warehouse design." In Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, pp. 47-56. ACM, 2005.
[HL01] Hunter, Jane, and Carl Lagoze. "Combining RDF and XML schemas to enhance interoperability between metadata application profiles." In Proceedings of the 10th international conference on World Wide Web, pp. 457-466. ACM, 2001.
[HFM06] Halevy, Alon, Michael Franklin, and David Maier. “Principles of dataspace systems.” In Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS). ACM, 2006, New York, NY, USA, pp. 1-9.
[HMH01] Hernández, Mauricio A., Renée J. Miller, and Laura M. Haas. "Clio: A semi-automatic tool for schema mapping." ACM SIGMOD Record 30, no. 2 (2001): 607.
[JCF03] Judson, Sheena R., Doris L. Carver, and Robert B. France. "A metamodeling approach to model transformation." In Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pp. 326-327.

ACM, 2003.
[JD13] Janga, Prudhvi, and Karen C. Davis. "Schema Extraction and Integration of Heterogeneous XML Document Collections." Model and Data Engineering. Springer Berlin Heidelberg, 2013. 176-187.
[JRS+12] Jovanovic, Petar, Oscar Romero, Alkis Simitsis, and Alberto Abelló. "ORE: an iterative approach to the design and evolution of multi-dimensional schemas." In Proceedings of the fifteenth international workshop on Data warehousing and OLAP, pp. 1-8. ACM, 2012.
[K05] Phokion G. Kolaitis. "Schema mappings, data exchange, and metadata management." In Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS). ACM, New York, NY, USA, pp. 61-75
[K12] Kimball, R., "The Evolving Role of the Enterprise Data Warehouse in the Era of Big Data Analytics," White Paper, Cloudera, 2012.
[LL06] Liu, Chengfei, and Jianxin Li. "Designing quality xml schemas from er diagrams." In Advances in Web-Age Information Management, pp. 508-519. Springer Berlin Heidelberg, 2006.
[LSS93] Lakshmanan, Laks VS, Fereidoon Sadri, and Iyer N. Subramanian. "On the logical foundations of schema integration and evolution in heterogeneous database systems." In Deductive and Object-Oriented Databases, pp. 81-100. Springer Berlin Heidelberg, 1993.
[MBR01] Madhavan, Jayant, Philip A. Bernstein, and Erhard Rahm. "Generic schema matching with cupid." In VLDB, vol. 1, pp. 49-58. 2001.
[MP06] Manakanatas, Dimitris, and Dimitris Plexousakis. "A Tool for Semi-Automated Semantic Schema Mapping: Design and Implementation." In DISWEB. 2006.
[N09] Nečaský, Martin. "Reverse engineering of XML schemas to conceptual diagrams." In Proceedings of the Sixth Asia-Pacific Conference on Conceptual Modeling-Volume 96, pp. 117-128. Australian Computer Society, Inc., 2009.
[PMI+03] Della Penna, Giuseppe, Antinisca Di Marco, Benedetto Intrigila, Igor Melatti, and Alfonso Pierantonio. "Xere: Towards a natural interoperability between xml and er diagrams." In Fundamental Approaches to Software Engineering, pp. 356-371. Springer Berlin Heidelberg, 2003.
[RB01] Rahm, Erhard, and Philip A. Bernstein. "A survey of approaches to automatic schema matching." the VLDB Journal 10, no. 4 (2001): 334-350.
[SE05] Shvaiko, Pavel, and Jérôme Euzenat. "A survey of schema-based matching approach es." In Journal on Data Semantics IV, pp. 146-171. Springer Berlin Heidelberg, 2005.
[TC07] Tsinaraki, Chrisa, and Stavros Christodoulakis. "Interoperability of XML schema applications with OWL domain knowledge and semantic web tools." In On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, pp. 850-869. Springer Berlin Heidelberg, 2007.
[V08] Vaidyanathan, V. "A Metamodeling Approach to Merging Data Warehouse Conceptual Schemas," M.S Thesis, 2008, University of Cincinnati, Ohio.
[ZC05] Zepeda, Leopoldo, and Matilde Celma. "Specifying metamodel transformations for data warehouse design." In Proceedings of the 43rd annual Southeast regional conference-Volume 1, pp. 266-267. ACM, 2005.