A Design for Coordinated and Logics-mediated Conceptual Modelling

Pablo Rubén Fillottrani[1,2] and C. Maria Keet[3]

[1] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, prf@cs.uns.edu.ar
[2] Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina
[3] Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

Abstract. The past 20 years have seen many efforts to provide logic-based reconstructions of various conceptual modelling languages so as to automatically reason over them, using a myriad of logics and those in the DL family of languages in particular. Subtle differences in the languages as well as modellers' preferences make it difficult to put the pieces together to provide one unified view and system. Therefore, we propose a mechanism to unify the back-end in the tool yet showing linkable ORM, UML, and ER diagrams in the interface. This is achieved by taking a two-pronged approach: 1) supporting full expressiveness of these languages and relying on a unifying metamodel, and 2) a logics back-end for their evidence-driven profiles in DL. Both have a set of rules for inter-model assertions. These profiles are tractable. As a result, we propose an architectural design for a tool that would help to integrate heterogeneous conceptual models and link them with ontologies, based on clear semantic specifications and with tractable algorithms.

1 Introduction

Data analysis in conceptual data modelling for database and information system development has a long history in computer science, dating back to its introduction in the 1970s. Conceptual data models represent what data will have to be stored and processed in a particular system, and the constraints among them in that 'universe of discourse' of the application. These models are almost exclusively represented graphically, using lines, boxes, diamonds, or ellipses, and the various adornments for the constraints. Many conceptual modelling languages are being used, such as UML Class Diagrams [25], EER [7, 31], ORM [17], MADS [27], and Telos [24], with as many tools to support them or, more often, a subset of their features, e.g., SmartDraw, MS Visio, GenMyModel, OmniGraffle, ERwin, NORMA, and ArgoUML. Conceptual modelling is used in industry, with experiments showing that it is being used notably in small (< 100) and large (> 1000) organisations, increasingly by modellers up to 10 years of experience, using modelling and CASE tools [8].

With increasingly complex system development and systems integration, the conceptual models become larger, their languages tend to acquire more features, and have more complex interactions because the components of the systems have to, and it is not unusual to use different modelling languages for the different components. A typical scenario is to use an ER diagram for the planned relational database and a UML Class Diagram for the application layer. Also, modellers have certain preferences for notation. The other main scenario is conceptual model-based data integration. In this case, one or more elements in each model needs to be linked across models that may be represented in different languages. This requires either one representation language or a common metamodel. Metamodel-based approaches by the conceptual modelling community typically analyse differences [15] rather than trying to find commonalities. In the Description Logics (DL) community, on the other hand, some effort has gone into trying to unify them in that all the different graphical depictions would end up as 'syntactic sugar' with a logic-based reconstruction into one suitable DL in the background hidden from the modeller [5, 21]. Most works, however, have focussed on logic-based reconstructions so as to be more precise and for automated reasoning over conceptual models to improve their quality; e.g., [1, 2, 16, 18, 19, 29] (though also other logics are being used, including OCL [28], CLIF [26], and Alloy [4]). Zooming in on DLs, $\mathcal{ALUNI}$ was used for a partial unification [5], whereas others are used for particular modelling language formalisations, such as *DL-Lite* and $\mathcal{DLR}_{\mathit{ifd}}$ [1, 2], or OWL, which may have offer only incomplete coverage of the full modelling language, such as omitting ER's 'keys' (identifiers) [5] or n-aries proper [1], among many variants. Also, multiple formalisations in multiple logics for one conceptual modelling language have been published (e.g., [14, 16, 18, 20] for ORM), and ORM in full is undecidable. Most of these efforts aim to cover most, or all, modelling language features. In contrast, a separate track of works looks at lean fragments so as to use the formalised conceptual data model at runtime. These lean profiles can be used for, among others, scalable test data generation [30], designing [3] and executing [6] queries, or querying databases during the stage of query compilation [32].

Thus, the reality is that there are multiple conceptual modelling languages that neither will go away nor will be superseded by just one language, with many logic-based reconstructions in multiple logics that seek to solve different problems, with more or less tools that are not designed to be interoperable. This raises the need for the integration of model design and their respective compatible formalisations, and to do this in a common tool. We aim to develop such a tool by building upon and extending ICOM [10, 13] that used a tailor-made single graphical language, single logic ($\mathcal{ALCQI}$), and reasoner (Racer). In particular, we are working on the following three major changes:

1. the **graphical interface**, such that modellers can model in UML Class Diagram notation in one model (module), in EER in another, and in ORM notation in yet another, with each model checked against a unifying metamodel for correctness regarding syntax;
2. the **logic-based reconstructions**, targeting a multi-modal approach, i.e., with one as comprehensive as the modelling language, caring little about performance, and, in the first instance, another mode with an *evidence-based*

core profile for the three conceptual data modelling language families that is computationally well-behaved;
3. the support for **inter-model assertions** so as to handle also cross-modelling language assertions, to allow also for attribute links, and to allow for approximations with common 'type' conversions, such as Attribute↔Value Type. Verifying such links relies on rules as well as the metamodel.

The high-level orchestration of the components is depicted in Fig. 1, with at its centre the DLs and their automated reasoners. This feature set is substantially different from ICOM's scope. The theoretical foundations and evidence obtained with experiments are based on a series of papers, which are the outcome of the project "Ontology-driven unification of conceptual data modelling languages" [`http://www.meteck.org/SAAR.html`] that was funded by Argentina's and South Africa's respective Ministries of Science and Technology.

The ontology-driven metamodel unifying UML Class Diagram v2.4.1, EER, and ORM2 and the metamodel's formalisation are described in [23, 9], which the left-hand side of Fig. 1 relies on. The approach for the inter-model link checker and core transformation rules (right-hand side of Fig. 1) are described in [12]. This approach avails of the metamodel to direct the checking of the intermodel assertions. The language profiles—at the centre of Fig. 1—are described in [11], which are based on an experimental evaluation of 101 conceptual models on the language features used in publicly available models [22]. Interestingly, the language profiles are tractable, and thus could be used for run-time usage of ontologies in an ontology-driven information system, such as proposed by [6, 30, 32].

We are currently developing a web-based prototype of the tool that involves the first two components in Fig. 1. A user edits a conceptual model on a web browser, while reasoning services are hosted on the server. For the moment, exclusive access to the model is given to the user. Simultaneously editing of the same model is planned in future versions.

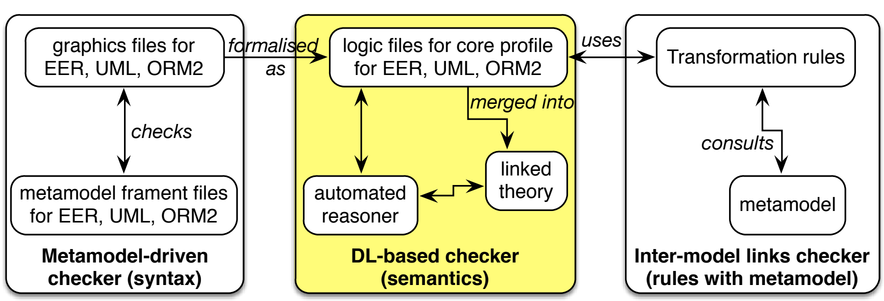

Fig. 1. High-level overview of the main components of the common tool for logic-based conceptual data modelling.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Reasoning over extended ER models. In: Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B. (eds.) Proceedings of the 26th International Conference on Conceptual Modeling (ER'07). LNCS, vol. 4801, pp. 277–292. Springer (2007), auckland, New Zealand, November 5-9, 2007
2. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. Artificial Intelligence 168(1-2), 70–118 (2005)
3. Bloesch, A.C., Halpin, T.A.: Conceptual Queries using ConQuer-II. In: Proceedings of ER'97: 16th International Conference on Conceptual Modeling. LNCS, vol. 1331, pp. 113–126. Springer (1997)
4. Braga, B.F.B., Almeida, J.P.A., Guizzardi, G., Benevides, A.B.: Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal methods. Innovations in Systems and Software Engineering 6(1-2), 55–63 (2010)
5. Calvanese, D., Lenzerini, M., Nardi, D.: Unifying class-based representation formalisms. Journal of Artificial Intelligence Research 11, 199–240 (1999)
6. Calvanese, D., Keet, C.M., Nutt, W., Rodríguez-Muro, M., Stefanoni, G.: Web-based graphical querying of databases through an ontology: the WONDER system. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) Proceedings of ACM Symposium on Applied Computing (ACM SAC'10). pp. 1389–1396. ACM (2010), march 22-26 2010, Sierre, Switzerland
7. Chen, P.P.: The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
8. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58, 358–380 (2006)
9. Fillottrani, P.R., Keet, C.M.: KF metamodel formalization. Technical Report 1078634 (September 2014), arxiv.org, 21p.
10. Fillottrani, P.R., Franconi, E., Tessaris, S.: The ICOM 3.0 intelligent conceptual modelling tool and methodology. Semantic Web Journal 3(3), 293–306 (2012)
11. Fillottrani, P.R., Keet, C.M.: Evidence-based languages for conceptual data modelling profiles. In: Morzy, T., et al. (eds.) 19th Conference on Advances in Databases and Information Systems (ADBIS'15). LNCS, vol. 9282, pp. 215–229. Springer (2015), 8-11 Sept, 2015, Poitiers, France
12. Fillottrani, P., Keet, C.: Conceptual model interoperability: a metamodel-driven approach. In: Bikakis, A., et al. (eds.) Proceedings of the 8th International Web Rule Symposium (RuleML'14). LNCS, vol. 8620, pp. 52–66. Springer (2014), august 18-20, 2014, Prague, Czech Republic
13. Franconi, E., Ng, G.: The ICOM tool for intelligent conceptual modelling. In: 7th Workshop on Knowledge Representation meets Databases (KRDB'00) (2000), berlin, Germany, 2000
14. Franconi, E., Mosca, A., Solomakhin, D.: The formalisation of ORM2 and its encoding in OWL2. KRDB Research Centre Technical Report KRDB12-2, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy (March 2012)
15. Halpin, T.A.: Advanced Topics in Database Research, vol. 3, chap. Comparing Metamodels for ER, ORM and UML Data Models, pp. 23–44. Idea Publishing Group, Hershey PA, USA (2004)
16. Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. Ph.D. thesis, University of Queensland, Australia (1989)

17. Halpin, T., Morgan, T.: Information modeling and relational databases. Morgan Kaufmann, 2nd edn. (2008)
18. Hofstede, A.H.M.t., Proper, H.A.: How to formalize it? formalization principles for information systems development methods. Information and Software Technology 40(10), 519–540 (1998)
19. Kaneiwa, K., Satoh, K.: Consistency checking algorithms for restricted UML class diagrams. In: Proceedings of the 4th International Symposium on Foundations of Information and Knowledge Systems (FoIKS'06). Springer Verlag (2006)
20. Keet, C.M.: Mapping the Object-Role Modeling language ORM2 into Description Logic language $\mathcal{DLR}_{ifd}$. Tech. Rep. 0702089v2, KRDB Research Centre, Free University of Bozen-Bolzano, Italy (April 2009), arXiv:cs.LO/0702089v2
21. Keet, C.M.: Ontology-driven formal conceptual data modeling for biological data analysis. In: Elloumi, M., Zomaya, A.Y. (eds.) Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data, chap. 6, pp. 129–154. Wiley (2013)
22. Keet, C.M., Fillottrani, P.R.: An analysis and characterisation of publicly available conceptual models. In: Johannesson, P., Lee, M.L., Liddle, S., Opdahl, A.L., Pastor López, O. (eds.) Proceedings of the 34th International Conference on Conceptual Modeling (ER'15). LNCS, vol. 9381, pp. 585–593. Springer (2015), 19-22 Oct, Stockholm, Sweden
23. Keet, C.M., Fillottrani, P.R.: An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. Data & Knowledge Engineering 98, 30–53 (2015)
24. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. ACM Transactions on Information Systems 8(4), 325–362 (1990)
25. Object Management Group: Superstructure specification. Standard 2.4.1, Object Management Group (2012), http://www.omg.org/spec/UML/2.4.1/
26. Pan, W.L., Liu, D.x.: Mapping object role modeling into common logic interchange format. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE'10). vol. 2, pp. 104–109. IEEE Computer Society (2010)
27. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual modeling for traditional and spatio-temporal applications—the MADS approach. Berlin Heidelberg: Springer Verlag (2006)
28. Queralt, A., Artale, A., Calvanese, D., Teniente, E.: OCL-Lite: Finite reasoning on UML/OCL conceptual schemas. Data & Knowledge Engineering 73, 1–22 (2012)
29. Queralt, A., Teniente, E.: Decidable reasoning in UML schemas with constraints. In: Bellahsene, Z., Léonard, M. (eds.) Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08). LNCS, vol. 5074, pp. 281–295. Springer (2008), montpellier, France, June 16-20, 2008
30. Smaragdakis, Y., Csallner, C., Subramanian, R.: Scalable satisfiability checking and test data generation from modeling diagrams. Automation in Software Engineering 16, 73–99 (2009)
31. Thalheim, B.: Extended entity relationship model. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, vol. 1, pp. 1083–1091. Springer (2009)
32. Toman, D., Weddell, G.E.: Fundamentals of Physical Design and Query Compilation. Synthesis Lectures on Data Management, Morgan & Claypool (2011)