Conceptual Modelling in Education: a Position Paper

Robert Andrei Buchmann[1], Ana-Maria Ghiran[1], Victoria Döller[2] and Dimitris Karagiannis[2]

[1] Business Informatics Research Centre, Faculty of Economics and Business Administration, Babeș-Bolyai University, Cluj-Napoca, Romania,
{robert.buchmann, anamaria.ghiran}@econ.ubbcluj.ro

[2] Research Group Knowledge Engineering, Faculty of Computer Science, University of Vienna, Vienna, Austria,
{victoria.doeller, dimitris.karagiannis}@univie.ac.at

Abstract. This position paper introduces a particular angle to address some student preconceptions regarding Conceptual Modelling, by presenting it as a standalone discipline that has a value proposition for any domain. The proposed thesis is that modelling languages should be primarily understood as purposeful knowledge schemas that can be subjected to agile adaptations in support of model-driven systems or decision processes. This thesis is supported by enablers such as the Open Models Laboratory and the Agile Modelling Method Engineering framework, which are also briefly presented.

Keywords: Conceptual modelling languages, Agile modelling method engineering, OMiLAB, Knowledge schema

1 Introduction

The perception on Conceptual Modelling methods has shifted in time – from seeing them as ways of expressing mental constructs in graphical form, to employing them for complexity management, or for building formal specifications in support of model-driven engineering. The literature discusses extensively the nature and categories of Conceptual Modelling – e.g., contrasting between "general-purpose" and "domain-specific". This heterogeneity reflects the multitude of angles from which stakeholders can employ Conceptual Modelling, but it also raises confusion among junior researchers who debut with certain oversimplified preconceptions – e.g., that Conceptual Modelling is a chapter of other disciplines (typically Software Engineering).

This paper formulates a position with respect to how we teach Conceptual Modelling - a position derived from recent discussions and lectures in the NEMO (Next Generation Enterprise Modelling) summer school series [1]. We believe that Conceptual Modelling has its own compelling value proposition in research, practice and education (which is the focus here), suggested by the NEMO summer school slogan: "We use abstraction to reduce complexity in a domain, for a specific purpose". However, this slogan needs to be operationalised in order to remove entry barriers for novices who want to assimilate Conceptual Modelling as part of their complexity management and digitisation skillset.

The angle advocated by the paper aims to defuse certain inertia and ambiguity in how Conceptual Modelling languages are understood - by students, by some practitioners, as well as by junior researchers who do not have an engineering perspective on the nature and constituents of a modelling language or method [2]. Below we provide an exemplary list of dilemmas collected from students debuting with junior research work and/or dissertation theses on topics related to Conceptual Modelling:

My thesis is on Marketing - specifically Service Design and Service-Dominant Logic – how can Conceptual Modelling help me, since it is a Software Engineering technique (this typically being the first contact of students with modelling tasks)? *Why are there so many modelling languages? Why not use Powerpoint, since I have many more shapes available in a single tool? Isn't it possible to model everything with a single language or standard? How can I combine parts of different modelling languages in an integrated way? How could I represent "this"* (domain-specific thing) *with my preferred standard*?

Answers to these questions are well understood and considered implicit by experienced researchers, but not easily available in explicit form to debutants. However, when students start doing research work, they find themselves pushed towards different paradigms – Design Science, Knowledge Management, Enterprise Modelling, Business Process Management etc. A learning curve must be facilitated to help them operationalise model value and accommodate such perspective shifts.

In the next Section we introduce some position statements that have helped students expand their understanding and get involved in productive research work. In Section 3 we also refer to the key enabler for these position statements - the OMiLAB (Open Models Laboratory) digital ecosystem [3][4] that successfully supports a holistic understanding on "model value" through an open community approach.

2 Position Statements

From past teaching experience we have extracted several "oversimplifications" by which students and junior researchers limit their own understanding when dealing with complex questions related to Conceptual Modelling. We try to address them through corresponding position statements to generate insights and stimulate lateral thinking that can expand the understanding of "model value" for novices:

Oversimplification 1. Conceptual Modelling is a form of graphical documentation – i.e., it produces visual representations that convey some meaning. This interpretation is confirmed in the literature that advocates Conceptual Modelling for the purposes of "understanding and communication" [5] and is propagated among students by the common task of having their theses documented in diagrammatic form. However, these documentations employ quite often semantically poor drawing software rather than modelling tools. Our position statement, aiming to compensate for this perception, is that *Conceptual Modelling produces knowledge structures that can have a visual manifestation*. With the term "knowledge structure" we point to two defining qualities of conceptual models: (i) to be conformant to a "knowledge (representation) schema" - i.e., each model element is an instance of some prescribed concept in a semantically consistent way (e.g., a dotted arrow does not change meaning); (ii) this further enables "model queries" – a term we use for model content retrieval (as basis for the development of model-based functionality, reporting etc.). Examples of model queries can be formulated by analogy with the more familiar "data queries" - e.g., *in a BPMN diagram, give me all tasks following this particular decision made in my department*. Thus, we emphasise the argument that *a modelling language provides a schema for a model repository* – an analogy with traditional databases that students easily grasp, and can be further extrapolated by the next point:

Oversimplification 2. Modelling languages are vocabularies fixed to serve some consensus. This interpretation is supported by the availability of standards – however even standards enable some level of customisation (e.g., UML stereotypes [6]). Moreover, the diversity of standards, showing both conceptual overlapping and purposeful specialisations, suggests that a "one size fits all purposes" vocabulary is not realistic. Therefore, our position statement is that *modelling languages are knowledge schemas that can be tailored to satisfy purposeful (possibly evolving) requirements*. Going back the database analogy, a database schema may be taken for granted, sufficiently stable for a large community of users who interact with it on content (data) level; however, requirements will occasionally trigger schema changes in order to support the evolution of information systems or decision processes. A modelling language ("knowledge schema") can be perceived through a similar lens, as suggested in Fig. 1, where a model repository is presented as complementing a traditional database.

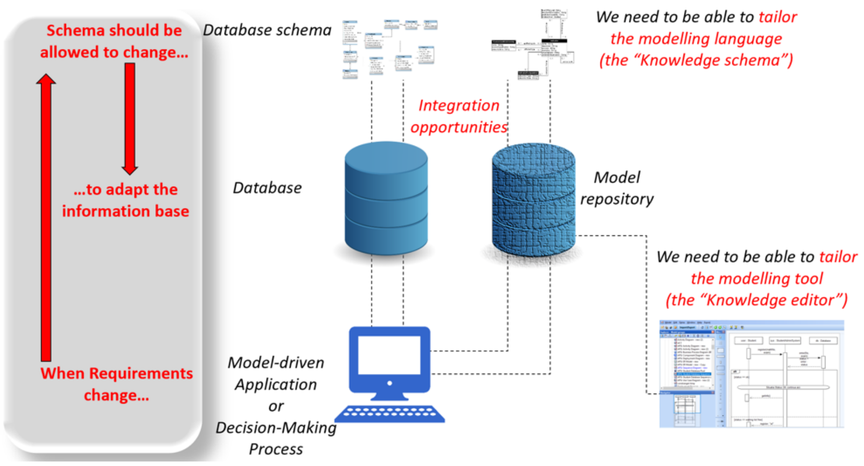

Fig. 1. The Database-Model repository analogy as rationale for agility in modelling languages

Oversimplification 3. Conceptual Modelling is a set of techniques subordinated to Software Engineering (or another discipline that provides initial contact with a modelling language). Our position statement is that *Conceptual Modelling can be applied to any domain* where complexity must be managed through abstraction and structuring. We encourage students who develop theses having no explicit relation to Software

Engineering to apply a modelling lens to their work, to reflect on the value proposition that Conceptual Modelling brings to their domain. For example, students with a background on Marketing may adopt open modelling tools available for their field (e.g., Product-Service system modelling [7]); or, by taking a Design Science approach, they may propose their own abstractions relevant to their field (e.g., Service, Customer). To connect this with the previous points, such abstractions can be guided by model queries as means of information retrieval and model analysis.

Oversimplification 4. Whatever needs to be modelled, I can do it with language X (no need for other languages). Our position statement is that the claim "I can model everything" commonly means "Whatever I cannot model, *I will compensate by (i) squeezing unstructured information into labels/annotations; or by (ii) hacking semantics*". Fig. 2 indicates such cases for a solution given by students who were asked to use BPMN to model a cooking recipe – see the two prominent ways in which they deal with the absence of domain-specific concepts (*Ingredient*) or properties (*Quantity*). The examples generate obvious complications when resorting to "model queries" (e.g., AQL queries in BEE-UP [8]) and the solution of "language redesign" by adding missing concepts can be proposed as a form of agile schema adaptation with the help of fast prototyping support (i.e., metamodeling platforms).

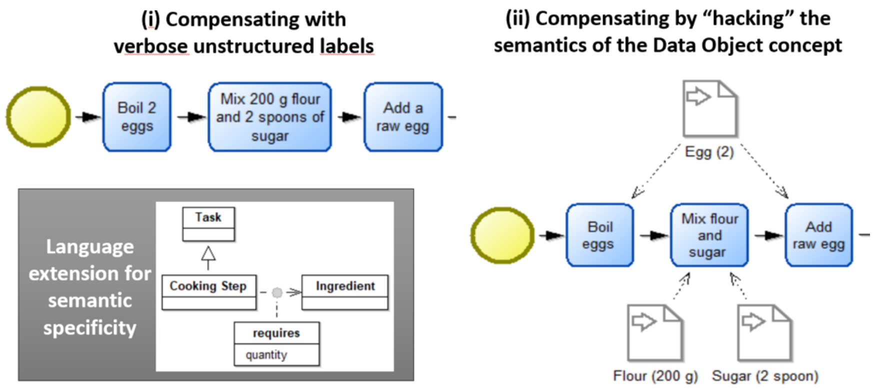

Fig. 2. Compensating lack of domain-specific expressivity - examples

Oversimplification 5. Model value is created solely by modellers. This interpretation finds confirmations in Business Analyst jobs where modelling methods are taken for granted together with established best practices (see BABOK [9]). Our position statement is that *value is co-created, during the lifecycle of a model, by at least a modeller and a modelling method engineer* – the latter being responsible with agilely capturing the right abstraction in order to satisfy the former's requirements and modelling use cases. Other stakeholders (e.g., domain experts) may also be involved.

Oversimplification 6. Modelling languages are of two kinds: general-purpose and domain-specific. Our position statement is that *both domain-specificity and modelling purpose are orthogonal dimensions*, as suggested in Fig. 3: (i) the *purpose* axis ranging between "general purpose" and "narrow purpose"; (ii) the *specificity* axis ranging between "domain agnostic" and "system specific", with various intermediate degrees of specificity. The notion of "language agility" emerging from the previous points allows languages to shift within this Purpose-Domain space.

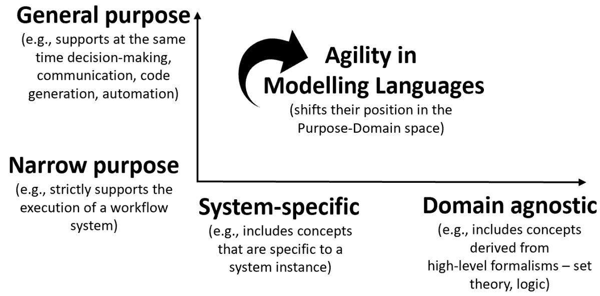

Fig. 3. Expanding the general-purpose/domain-specific dichotomy

3 OMiLAB: the Value Proposition for Conceptual Modelling

The position statements introduced in the previous Section require certain enablers to support them – not only on a principle level, but also for building corresponding proofs-of-concepts. Such enablers are available in the Open Models Laboratory (OMiLAB) [3] – a digital ecosystem built around a holistic "model value" proposition. Junior researchers adopting the hereby presented position statements can benefit from OMiLAB resources in the following ways:

- by tweaking open source modelling tools to shift their domain-specificity and purposefulness (e.g., BPMN for DevOps, ER for Knowledge Graphs);
- by implementing novel modelling methods as proofs-of-concepts created for a selected domain / purpose, including certain types of model-enabled evaluation (via reasoning, model analysis etc.); this can be achieved with the help of the Agile Modelling Method Engineering (AMME) framework [10], which established the conceptualisation process underlying the position statements hereby presented;
- by creating and evaluating model-driven artefacts with the help of interoperability features that can be agilely adapted for any modelling language (e.g., RDF export, XML export, Model-as-a-service);
- by snowballing literature reviews starting from the rich corpus of publications reported by various projects hosted by OMiLAB [11];
- or, by employing modelling tools that are already available for open use, for a variety of languages (e.g., BEE-UP [8] supports in the same tool BPMN, EPC, UML, ER, Petri Nets, model queries and RDF export for any of these model types).

One key resource for the conceptualisation and operationalisation of this value proposition is ADOxx [12] – a metamodelling platform for the fast prototyping of modelling tools, i.e., for tailoring their "knowledge schema" for a selected purpose or desired specificity. Another key resource is the Digital Product lab instance demonstrating the use of models as an intermediate knowledge layer between Design Thinking weakly structured scenes and model-driven systems [4]. International OMiLAB nodes make such resources and infrastructures available to regional communities for both research and education purposes – see the works of OMiLAB Korea [13].

4 Summary

The paper introduced several position statements to encourage a comprehensive perception on model value and on the quality of modelling languages as knowledge schemas that reduce complexity and operationalise semantics. Arguments are targeted to junior researchers who need to cross the expertise gap between how Conceptual Modelling is perceived in bachelor studies and the value proposition it brings for design research and innovation engineering. Tutorials and teaching cases in support of these arguments have been published recently [14] - we take this opportunity to further call for teaching experiences and artefacts that can contribute to a holistic value of models or to the further refinement of the *Purpose-Domain space* where modelling languages can be positioned.

References

1. OMiLAB, NEMO Summer School – the official page, http://nemo.omilab.org. [Accessed: 28/01/ 2019].
2. D. Karagiannis and H. Kühn, "Metamodelling Platforms", in Proceedings of EC-Web 2002 – DEXA 2002, LNCS 2455, Springer, 2002, pp. 182.
3. OMiLAB, Open Models Laboratory Portal – official website. http://omilab.org.
4. D. Bork, R. Buchmann, D. Karagiannis, M. Lee, and E. T. Miron, "An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem", Communications of the Association for Information Systems 44, pp. 673-697, 2019.
5. J. Mylopoulos, "Conceptual modeling and Telos1", in Conceptual Modeling, Databases, and Case - an integrated view of information systems development. Wiley, 1992, pp 49–68.
6. UML Stereotypes, https://www.uml-diagrams.org/stereotype.htmlSd
7. X. Boucher, K. Medini, H. G. Fill, "Product-Service-System Modeling Method", in Domain-specific Conceptual Modeling, Springer, 2016, pp. 455-482.
8. OMiLAB. Bee-Up page in OMiLAB. http://austria.omilab.org/psm/content/bee-up/info.
9. IIBA, A Guide to the Business Analysis Body of Knowledge, https://www.iiba.org/standards-and-resources/babok/
10. D. Karagiannis, Conceptual modelling methods: The AMME Agile Engineering approach, Proceedings of the 15th Int. Conf. Informatics in Economy (IE), LNBIP 273, Springer, 2018, pp. 3-19.
11. D. Karagiannis, H. Mayr and J. Mylopoulos (eds.), Domain-Specific Conceptual Modeling, Springer, 2016.
12. BOC, AdoXX official website, https://www.adoxx.org/live/home.
13. Open Models Laboratory Korea, http://omilab-korea.org.
14. A. M. Ghiran, C. C. Osman, R. A. Buchmann, "A Metamodeling Approach to Teaching Conceptual Modeling at Large", Proceedings of ISD 2019, Association for Information Systems, 2019, in press.