



## A Roadmap for UEML

Andreas L. Opdahl, Giuseppe Berio

### ► To cite this version:

| Andreas L. Opdahl, Giuseppe Berio. A Roadmap for UEML. I-ESA 2006, 2006, France. pp.2. hal-00488758

HAL Id: hal-00488758

<https://hal.science/hal-00488758>

Submitted on 2 Jun 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 1

---

## A Roadmap for UEML

Andreas L. Opdahl & Giuseppe Berio

The Unified Enterprise Modelling Language (UEML) is an intermediate language that aims to support integrated use of enterprise and information systems (IS) models expressed using different languages. This paper presents a roadmap for the further UEML evolution, organised as ten complementary directions along which development can progress, to some extent in parallel. A prioritisation of near-term developments is also proposed. The aim is to direct and foster debate about how the UEML should evolve.

### 1. Introduction

The Unified Enterprise Modelling Language (UEML) is an ongoing effort to develop an intermediate language for modelling enterprises and related domains, such as information systems. This is an ambitious, long-term goal that will require several years of effort and cooperation between academia and industry. A roadmap for how UEML should evolve is therefore needed to direct and coordinate the work. A first proposal for a roadmap is presented here in order to foster debate.

The roadmap is organised as ten complementary directions along which further UEML development can progress, to some extent in parallel. For example, one direction is broader coverage of relevant modelling languages another, and largely independent one, is extended tool support. Between other directions, however, there are strict dependencies. For example, proper industrial validation of UEML will require a certain level of tool support.

The roadmap is a long-term plan that involves a large amount of work. Clearly, all its research objectives cannot be accomplished inside the INTEROP NoE (INTEROP 2005), the current locus of UEML activity, which ends in 2006. We therefore also propose a prioritisation of UEML development for the near term, i.e., for the final year of INTEROP and its immediate aftermath.

The following Section 2 will present an overview of UEML. Section 3 discusses the dimensions in detail. Section 4 proposes priorities for the near term. Finally, Section 5 concludes the paper and offers paths for further work.

## 2. Overview of UEML

The idea of a Unified Enterprise Modelling Language (UEML) emerged during the ICEIMT'97 conference (Goossenaerts, Gruninger, Nell, Petit & Vernadat 1997), with the aim of providing an underlying formal theory for enterprise modelling languages. A major motivation was enterprise integration in the face of a wide variety of enterprise modelling languages (Vernadat 2002).

The UEML Thematic Network (UEML TN) (2002-2003) was funded by the EU's FP5. It had three main activities (Jochem 2002, Panetto, Berio, Benali, Boudjida & Petit 2004, Mertins, Knothe & Zelm 2004, Berio, Anaya & Ortiz 2004): Requirements collection and analysis, feasibility study and demonstrator development. It developed and demonstrated a common abstract syntax and exchange format, which incorporated three important industrial European enterprise modelling languages: EEML (EXTERNAL 2000), GRAI (Doumeingts, Vallespir, Zanettin & Chen 1992) and IEM (Jochem & Mertins 1999).

UEML development has since continued within the INTEROP Network of Excellence (INTEROP NoE) (2003-2006), funded by the EU's FP6. It also has 3 main activities: Requirements, language selection and definition approach (Berio, Jaekel & Mertins 2005). Starting from high level requirements established in the UEML TN, more focused requirements were collected using a refined method based on a requirement elicitation template. Next, languages were selected from a list. However, the need of developing a well founded way to select languages was identified and originated the need for defining language quality criteria, linked to the collected requirements, that can be evaluated on information collected from INTEROP partners using a language template. Currently, the selected languages are being incorporated construct-by-construct based on information collected from INTEROP partners using a construct template. Collectively, we will refer to the requirement template, the language template and the construct template as the UEML approach in this paper.

The UEML construct template provides a standard, integrative format for representing modelling constructs (Opdahl & Henderson-Sellers 2004, 2005a, Dallons, Heymans & Pollet 2005, Heymans, Saval, Dallons & Pollet 2005). Entries of the construct template are derived from a UEML meta meta model, inspired by Bunge's ontological model (Bunge 1977, 1979) and on the Bunge-Wand-Weber representation model (the BWW model, Wand & Weber 1988, 1993, 1995). Template entries are filled-in in such a way that the concepts used gradually build a UEML ontology that is rooted in central ontological concepts from Bunge and the BWW model. This ontology grows incrementally as more modelling constructs are added, whether centrally by some UEML management organ or locally within an enterprise that uses UEML. In consequence, when two modelling constructs, from the same or from different languages, have both been described using the UEML-template, the exact correspondences between them can be identified in terms of the common ontology. This paves the way for comparison, consistency checking, update reflection, view synchronisation and, eventually, model-to-model translation across modelling language boundaries (see also Section 3, specifically 3.7 on model management). Therefore UEML can be defined as a web (or family)

of languages (that can also be considered a single language) that are co-existing but at the same time precisely related throughout correspondences. The central concepts (centrally managed by a UEML management organ) in the common ontology may also reveal a language currently called UEML core.

The first version of the construct template was developed in November 2004 and distributed to selected INTEROP partners. Initial attempts were made in the first half of 2005 to describe BPMN, colored Petri nets, GRL, ISO/DIS 19440, UEML 1.0, selected diagram types from UML 2.0, XPDL and YAWL (Berio, Opdahl, Anaya, Dassisti 2005). The template was revised twice during this process. In the autumn of 2005, colored Petri nets, GRL and UML 2.0 class and activity diagrams were chosen for inclusion into the first demonstration version of UEML 2.1. Currently, construct descriptions made using the template are being negotiated between the partners and entered into the prototype tool, contributing towards a first version of the common ontology.

The experiences from the UEML work in INTEROP indicate that the UEML's template-based approach is sufficiently powerful to support integrated use of a broad variety of languages and models. The experiences also indicate that the UEML approach can form the core of a new theory of integration and integrated use of languages and models. For this to happen, however, UEML must be broadened and developed, formalised and documented further, and proof-of-concept prototype tools must be developed. The resulting theory and tools also need to be empirically validated and evaluated in real case studies, while incorporating an increasingly wider selection of modelling languages.

### **3. The Roadmap and its dimensions**

The paper proposes a roadmap for continuing the work on UEML. We have identified the ten directions. The directions are useful because they break UEML evolution into smaller more manageable parts, which are easier to discuss and prioritise against one another and for which clearer goals and sub-goals can be set. The directions are also useful because, to some extent, work can progress in parallel along several directions independently.

#### **3.1 Language breadth**

UEML should be broadened by incorporating more languages, frameworks and standards for representing enterprises, their information systems and other related domains. In particular, more industrial modelling languages, frameworks and standards should be considered (e.g., standards initiatives undertaken in OMG). For several reasons, a systematic approach is needed to determine which languages to include: Firstly, incorporating the right languages first can simplify incorporation of other languages because of reuse of revealed concepts in the common ontology. Secondly, the number of potential languages is high. For example, languages from other areas such as IS, KE and SE are sometimes included in surveys and states of the art of enterprise modelling. A language

quality framework (comprising both quality criteria and quality metrics) has therefore been developed based on the framework of Krogstie, Lindland and Sindre (1995). This framework needs to be fully integrated with the rest of the UEML approach: an embryo of such integration is already available in term of a integrated meta meta model.

Within the limits of the framework, we continue to work on languages mentioned in section 2 and some additional ones (e.g., GRL and KAOS).

### 3.2 Ontological depth

Initial experience with UEML versions 2.0 and 2.1 shows that the common ontology can be described at different levels of precision. For example, many modelling constructs that have been encountered represent active things (things that transform themselves from certain states to certain other states). But these construct do not all represent the same subtype of active things:

- Some constructs represent active things with only one input and one output; others allow many inputs and outputs.
- Some constructs represent active things that consume all their inputs and produce all their outputs synchronously, other behave asynchronously.
- Some constructs have typed inputs and outputs, others not etc.

In the common ontology, it is possible to attempt to represent all these variants and their combinations separately (an ambitious, fine-grained approach), or it is possible to represent only the class of active things (a simple, coarse approach). A fine-grained ontology is harder to use and manage but offers to support a tighter integration and is preferable when precise modelling is required. A less detailed ontology is smaller and friendlier, and is needed when describing less ontologically precise languages.

It should be investigated whether the common ontology should in the end offer both coarse and fine-grained concepts. Note that this is subtly different from arranging, e.g., ontological classes in a generalisation hierarchy: Let C1 be an ontological class, with C2 and C3 as subclasses. A fine-grained construct description might state that construct C' represents the exact class C1. A coarse-grained description might state that construct C'' represents classes C1, C2 and/or C3. In practice, model elements of constructs C' and C'' can represent the same class of things, i.e., that of C1. But in the case of C' we know the description is precise. In the case of C'', we know it is coarse.

In addition, a UEML core should be established based on the results from incorporating a variety of languages. A first version of the core should focus on concrete domains, where most work has been put so far. Later versions can also include conceptual and social domains, if it turns out that they require a different or a modified core. For these purposes, the Bunge/BWW-model ontological concepts are not equally clear. Many central ontological concepts used in the template, such as thing, property, mutual property and class are also accounted for in models from the cognitive disciplines and from ontological models other than Bunge's and the BWW model, e.g., Chisholm (1996). In the future, it may even be possible to use a similar approach to languages for and models of software designs and

implementations besides enterprises (Opdahl & Henderson-Sellers 2005b). This might require yet other theoretical platforms from fields such as computer science and software engineering.

### **3.3 Ontological clarity**

The ontological grounding of UEML should be elaborated. Whereas the previous point (ontological depth) dealt with the number and precision of phenomena in the common ontology, this point deals with the (“upper”) ontological concepts used to describe the common ontology and how well they are defined. Because the common ontology is based on Bunge’s ontological model and the Bunge-Wand-Weber (BWW) representation model, it is already more elaborately described than most competing ontologies, but its description can always be elaborated. A possibility is to explicitly define an ontology language based on the part of the UEML meta-meta model that covers the common ontology. Such an ontology language would improve the current standard web ontology language (OWL) with behavioural constructs. An ontology language that accounts for modalities would be a next step. Of course, this ontology language should eventually support both concrete, conceptual and social domains.

### **3.4 Presentation**

The UEML approach should allow managing how models are presented visually not only what the models represent. The UEML-template already has both a representation part (previously called the “semantics part”), which describes what each modelling construct is intended to represent, and a presentation part (“syntax part”), which describes how each modelling construct is intended to be visualised. But because the representation part is the most innovative and challenging part of the UEML approach, the presentation part has been less focused on so far. In consequence, UEML 2 deals more thoroughly with representation.

Future versions of UEML should support both presentation and representation equally thoroughly. It should also support verifying that the two are consistent. For example, every presentational (or syntactical) relationship between two modelling constructs must be matched by a corresponding representational relationship (i.e., by a mutual property in the common ontology) and vice versa.

Future versions of UEML should also support the widest possible variety of presentation types. So far, the template has focused on conventional boxes-and-arrows diagrams. But the future of enterprise modelling lies beyond static boxes-and-arrows diagrams: Today’s dynamic simulation and animation of behavioural models are only first steps towards immersive simulation/animation of 3-dimensional worlds generated from models. The UEML must take these possibilities into account already now, and must be designed to accommodate yet further future presentation types.

### **3.5 Mathematical formality**

The UEML-template should be defined formally. The formal definition should include at least: Definitions of the “upper” ontology concepts from Bunge/the BWW-model that are used to define the common ontology. Provisions should be made for defining the phenomena in the common ontology by refining the definitions of the “upper” ontology concepts. The representation and presentation parts of the template should also be formalised, and provisions should be made for describing modelling constructs formally using the template. For example, in the current version of the UEML template (version 1.2), state and transformation laws are described informally using natural language or an arbitrary formal notation. Future versions of the template should offer a standardised formal notation for describing laws. Provisions should be made for ensuring that the representation part of a construct definition is self-consistent, e.g., that the formal law descriptions have as unbound parameters only other ontology properties of the thing that possesses the law. It should also be possible to ensure that the presentation part of a construct description is consistent with itself and with the representation part. Further provisions should be made to ensure presentational and representational consistency between all the constructs in a language and between languages. A particularly challenging task is ensuring that the various laws implied by different modelling constructs are consistent with one another, in particular when those constructs belong to the same language. Although this problem cannot be fully solved formally, automated reasoning may help. Further work should also address the feasibility and importance of supporting other formal properties than consistency for UEML.

### **3.6 Tool support**

Tools should be developed that make the UEML template approach easier to manage and use. Currently, the representation part of UEML construct template is supported by a prototype repository implemented in Protégé-OWL. A first priority is to provide a more suitable user interface to this tool. A second priority is to provide simple analysis features for validating construct descriptions, e.g., checking that the descriptions adhere to the cardinality constraints expressed in the meta-meta model and enforcing other constraints presented in (Opdahl & Henderson-Sellers, 2005a). Generating Prolog code is one viable implementation path. Further work can take several directions. One direction is supporting the presentation part in addition to the representation part, along with the corresponding validation mechanisms. Another direction is to support fully formal construct descriptions, i.e., with standard notations for defining state and transformation laws. Expressing and validating the consistency of fully formal construct descriptions will likely require stronger formal support than offered by Prolog. The current plan is envisioning to use Alloy (<http://alloy.mit.edu/>) to express and reason about such descriptions.

### **3.7 Model management**

Model management enables to operate on models (and data or instances of associated to these models) throughout generic operators such as compose, compare, merge and so on (Bernstein 2003). In the context of our work, we have reformulated the notion of model management by saying that model management should enable integration and integrated use of models. In this sense, UEML, the UEML approach and supporting tools should address management of enterprise models in addition to the management of languages: this is really a key point because it probably constitutes the most direct and common application of UEML. Therefore, the focus of model-management support should be, as first step, utilising the language coordination and integration facilities provided by UEML, in order to support better coordination and integration of the models expressed in those languages. Accordingly, without supporting this model management level, UEML and the UEML approach will remain a preparation for very important functions on models such as automatic model update reflection and model translation.

### **3.8 Validation**

The UEML approach should be validated empirically using a variety of research methods. Current language analysis and description work indicates that the UEML template approach is indeed sufficiently powerful to incorporate a wide range of industrial, standardised or academic enterprise modelling languages. This work must be continued and extended. As soon as UEML has been extended to support model management in addition to language management, it will be necessary to show that the approach is indeed sufficiently powerful to support functions such as cross-language consistency checking, automatic update reflection, cross-language model translation etc. A possible starting point is to manually replicate existing results provided by other approaches. When a tool support for model management becomes available, more elaborate case studies could be undertaken using real industrial models, preparing for eventual action research studies of the UEML core and of model management in real organisations.

### **3.9 Dissemination**

The UEML approach should be disseminated widely to both industry and academia. Dissemination tasks include developing written, oral and interactive (internet) tutorials and providing high-quality examples. Technical documentation must be produced. Research results must be presented in both academic and professional journals and conferences. The management of the UEML core should be placed under the responsibility of a research group or standardisation committee.

In the longer term, the UEML approach should also be prepared and promoted as the international standard for exchanging models of enterprises and related domains. The appropriate standard organisations for achieving this must be sought.

The appropriate positions relative to existing and emerging standards must also be defined. For example, there is clearly a need for a more specific and powerful exchange standard than what is provided by the OMG's XML Model Interchange (MXI) format.

### **3.10 Community**

UEML should be anchored in a cohesive and dedicated community which continues to promote and evolve both UEMl and its approach.

## **4. Near-term Priorities**

The roadmap is an ambitious, long-term plan that clearly cannot be accomplished inside the INTEROP NoE, which ends in 2006. A prioritisation of UEMl development for the near term, i.e., in the final year of INTEROP and its immediate aftermath, is also needed. We propose the following priority tasks:

1. Simple tool support, including a user interface and simple consistency checking. This is the first priority, because it is becoming hard to continue UEMl development without tool support.
2. A second priority is dissemination through documentation, tutorials and presentations. This activity is already started and an initial tutorial on the BWW-model has been delivered to the involved participants.
3. A third priority is formalisation of the UEMl approach. This formalisation should be performed to a certain extent before it can support model management and be realised by a tool.

Additionally, at least embryonic activities about the following points should be performed:

4. Model management as well as language management, supported by the tool. Model management must be in place to a certain extent before UEMl can be meaningfully validated empirically.
5. Empirical validation of tool-supported language and model management. Empirical validation is obviously necessary to ensure industrial uptake of UEMl. Empirical validation is also necessary of UEMl work shall contribute to theory.

In addition, the present roadmap itself is only a draft, which we expect to evolve in the coming months and years based on ideas and recommendations of the project partners.

## **5. Conclusion and Further Work**

The paper has presented a first proposal for a roadmap for evolving UEMl in order to foster debate about its further development. The roadmap is organised into ten complementary directions along which further UEMl development can progress, to some extent in parallel. Each dimension was discussed and elaborated with

subgoals. A prioritisation of work for the near term was only proposed. This is only a first draft of the roadmap, which will hopefully evolve in parallel with our continued UEML work.

The most important limitation of the current roadmap is that it is solidly anchored in the current UEML approach, based on templates, separation of reference and a common UEML ontology. The UEML principles discussed at the start of Interop opened up for a multi-approach UEML, which could be defined in multiple ways – some of them perhaps experimental – as long as great care was taken to ensure that the definitions were consistent. Such a multi-approach UEML might be even more widely applicable than the currently evolving UEML. It also might serve as a laboratory for comparing language and model definition and management approaches. This is a both challenging and interesting research task for the longer term. Work in the near term should focus on delivering a solid and agreed-on UEML version 2.

### Acknowledgement

This work is partially funded by the Commission of the European Communities under the sixth framework programme: the Interop Network of Excellence, IST-508011. The authors thank all the partners involved in WP5/DEM in the Interop network.

### References

- Berio G., Anaya V., and Ortiz A. Supporting Enterprise Integration through a Unified Enterprise Modeling Language. In Proc. of EMOI 2004 (Enterprise Modelling and Ontologies for Interoperability) (Janis Grundspenkis, Marite Kirikova Eds.), joint with CAiSE04, Riga Technical University, 3: 165-176.
- Berio G., Jaekel Frank-Walter and Mertins K. (2005). Common Enterprise Modelling Framework fro Distributed Organisations. In Proc. of the 16th IFAC World Congress, Prague, Czech Republic, July.
- Berio, G., Opdahl, A., Anaya, V. and Dassisti, M. (2005). Deliverable DEM1. [www.interop-noe.org](http://www.interop-noe.org).
- Berio, G. and Petit, M. (2003). Enterprise Modelling and the UML: (sometimes) a conflict without a case, in Proc. 10th ISPE International Conf. on Concurrent Engineering: Research and applications, Madeira, Portugal.
- Bernstein, P.A. (2003) Applying Model Management to Classical Meta Data Problems. In Proc. CIDR 2003, pp. 209-220
- Bunge, M. (1977). Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World. Boston:Reidel.
- Bunge, M. (1979). Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. Boston:Reidel.
- Chisholm, R. M. (1996). A Realistic Theory of Categories: An Essay on Ontology, Cambridge University Press.
- Dallons, G., Heymans, P. and Pollet, I. (2005). A Template-based Analysis of GRL, in Proc. of EMMSAD'05 (CAiSE\*05), Tenth International Workshop on Exploring Modeling Methods in Systems Analysis and Design, pp. 493-504.

- Doumeingts, G., Vallespir, B., Zanettin, B. and Chen, D. (1992). GIM, GRAI Integrated Methodology – A methodology for designing CIM systems, Version 1.0, Unnumbered report, LAP/GRAI, University of Bordeaux 1, France.
- EXTERNAL (2000). Extended Enterprise Resources, Networks and Learning, EC project, IST- 1999-10091, 2000.
- Goossenaerts, J., Gruninger, M., Nell, J.G., Petit, M. and Vernadat, F. (1997). Formal Semantics of Enterprise Models. In Proc. of ICEIMT'97, K.Kosanke and J.G Nell. (Eds.), Springer- Verlag.
- Heymans, P., Saval, G., Dallons, G. and Pollet, I. (2005). A Template-Based Analysis of GRL: Book chapter, in Advanced Topic in Database Research - Volume 5. Idea Group Publishing.
- INTEROP (2005). Interop Network of Excellence. [www.interop-noe.org](http://www.interop-noe.org), 2005.ISO/DIS (2004). preEN ISO/DIS 19440:2004. Enterprise integration — Constructs for enterprise modelling, ISO, ISO TC 184/SC5/WG1 - CEN TC 310/WG1.
- Jochem, R. (2002). Common representation through UEML – requirement and approach. In Proc. of ICEIMT 2002, Kosanke K., Jochem R., Nell J., Ortiz Bas A. (Eds.), Polytechnic University of Valencia, Valencia, Spain, April 24-26, Kluwer. IFIP TC 5/WG5.12.
- Jochem, R. Mertins, K. (1999). Quality-Oriented Design of Business Processes. Kluwer, Boston.
- Krogstie, J., Lindland, O.I., & Sindre, G. (1995). Defining Quality Aspects for Conceptual Models. In E. D. Falkenberg, W. Hesse, & A. Olive (Eds.). Proc. of the IFIP8.1 working conference on Information Systems Concepts (ISCO3); Towards a consolidation of views, March 28-30 (pp. 216-231), Marburg, Germany.
- Mertins, K., Knothe, T., Zelm, M. (2004). User oriented Enterprise Modeling for Interoperability with UEML. In Proc. of EMMSAD'04 Evaluating Modeling Methods for Systems Analysis and Design pp.25-36, joint with CAiSE04, Riga – Latvia, June 7-8.
- Panetto H., Berio G., Benali K., Boudjilida N. and Petit M. (2004). A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. In Proc. of the 11th IFAC INCOM2004 Symposium, , Bahia, Brazil, April 5-7.
- OMG (2001). OMG Unified Modeling Language Specification, version 1.4. Object Management Group.
- Opdahl, A.L. and Henderson-Sellers, B. (2004). A Template for Defining Enterprise Modelling Constructs. Journal of Database Management 15(2).
- Opdahl, A.L. and Henderson-Sellers, B. (2005a). Template-Based Definition of Information Systems and Enterprise Modelling Constructs. In Ontologies and Business System Analysis, Peter Green and Michael Rosemann (eds.). Idea Group Publishing, 2005.
- Opdahl, A.L. and Henderson-Sellers, B. (2005b). A Unified Modeling Language Without Referential Redundancy, Data and Knowledge Engineering (DKE) 55(3). Elsevier.
- Petit, M. Gossenaert, J. Gruninger, M. Nell, J.G. and Vernadat, F. (1997). Formal Semantics of Enterprise Models. In Proc. of ICEIMT 1997, K.Kosanke and J.G Nell. (Eds.), Springer- Verlag.
- Wand, Y. & Weber, R. (1988). An ontological analysis of some fundamental information systems concepts. In Proc. "Ninth International Conference on Information Systems", (DeGross, J.I. & Olson, M.H. (eds.), Minneapolis/USA, November 30–December 3, 1988, pp. 213–225.
- Wand, Y. & Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. Journal of Information Systems, 3:217–237.
- Wand, Y. & Weber, R. (1995). On the deep structure of information systems. Information Systems Journal, 5:203–223.