

Preface to the 4th International Workshop on Multi-Level Modelling (MULTI 2017)

Tony Clark

Sheffield Hallam University, UK

Ulrich Frank

University of Duisburg-Essen, Germany

Manuel Wimmer

TU Wien, Austria

I. INTRODUCTION

Traditional approaches to modelling systems tend to emphasize two-levels: type and instance. Recently, the benefits of meta-level access have been argued to be of importance in terms of language definition, particularly in terms of domain-specific languages. The ability to define elements at the meta-level improves the scope for abstraction, reuse, and ultimately system quality. Providing meta-level access introduces a number of methodological and technological challenges: how, and whether, to introduce strict separation between levels; how to express the relationships between elements at different levels; technologies for expressing multi-level concepts.

The roots of multi-level modelling can be traced back over 15 years to when the first papers on flaws in the OMG's four level modelling architecture were published. From these roots, various flavours of multi-level modelling have emerged over time, many with associated supporting tools. Examples include DPF workbench, GModel, Melanee, MetaDepth, Nivel, OMME and XModeller. Although these technologies share many common ideas, there are significant differences in the fundamental principles they use to support multi-level modelling. As a result, the current tool landscape is highly fragmented with different tools designed to serve very different purposes. There is also a lack of common multi-level modelling guidelines, educational material and even terminology. Without a common foundation and an evidence-base for case studies on the benefits of multi-level modeling, particularly in industry, it will be difficult for the approach to evolve beyond the currently active research community.

The MULTI series of workshops aims to provide a forum for advances in the field to be presented and discussed. The first MULTI workshop (www.miso.es/multi/2014/) was held at MODELS 2014 in Valencia and spawned a special theme issue of SoSyM. The second (www.miso.es/multi/2015/) and third MULTI workshops were held at MODELS 2015 in Ottawa and at MODELS 2016 in St. Malo respectively. The average

number of participants was around 20. MULTI 2015 also spawned a new Wiki for the multi-level modeling research community:

<http://homepages.ecs.vuw.ac.nz/Groups/MultiLevelModeling/>

The scope of the MULTI workshops includes the following topics:

1. The exact nature of elements in a multi-level hierarchy and how best to represent them.
2. The importance and role of potency and its variants such as durability and mutability.
3. The structure and labeling of a multi-level modelling framework.
4. Methods and technique for discovering clabjects, specializations and classification relationships.
5. Formal approaches to multi-level modelling.
6. Experiences and challenges in providing tool support for multi-level modelling.
7. Experiences and challenges in applying multi-level modelling techniques to large and/or real world problems.
8. Model management languages (transformation, code generation etc.) in a multi-level setting.
9. Criteria and approaches for comparing multi-level modelling approaches.
10. Integration of modelling and programming languages.

II. PAPERS

MULTI 2017 attracted 14 submissions and accepted 8 as full papers and an additional poster. The papers covered a wide spectrum of topics related to the multi-level modelling, demonstrating that the field is active and has significant applicability. The workshop introduced the MULTI challenge (described below) and included a presentation addressing its issues.

A configuration is a physical artifact that is composed of components. A component may be composed of other components or of basic parts. There is a difference between the type of a component and its instances. A component has a weight. A bicycle is built of components like frame, a handle bar, two wheels, etc. A bicycle component is a component. A frame, a fork, a wheel, etc. are bicycle components. Frames and forks exist in various colors. Every frame has a unique serial number. Front wheel and rear wheel must have the same size. Each bicycle has a purchase price and a sales price. There are different types of bicycles for different purposes such as race, mountains, city, etc. A mountain bike or a city bike may have a suspension. A mountain bike makes have a rear suspension. That is not the case for city bikes. A racing fork does not have a suspension. It does not have a mud mount either. A racing bike is not suited for tough terrains. A racing bike is suited for races. It can be used in cities, too. Racing frames are specified by top tube, down tube, and seat tube length. A racing bike can be certified by the UCI. A racing frame is made of steel, aluminum, or carbon. A pro race bike is certified by the UCI. A pro race frame is made of aluminum or carbon. A pro racing bike has a minimum weight of 5200 gr. A carbon frame type allows for carbon or aluminum wheel types only. "Challenger A2-XL" is a pro-racer for tall cyclists. The regular sales price is 4999.00. Some exemplars are sold for a lower price. It is equipped with a Rocket-A1-XL pro race frame. The Rocket-A1-XL has a weight of 920.0 gr. A sales manager may be interested in the average sales price of all exemplars of a certain model and may also be interested in the average sales price of all mountain bikes, all racing bikes etc.

Figure 1: The MULTI Challenge (2017)

A key problem faced by the MULTI community is one that arises from the variety of approaches and definitions for concepts and relationships. The paper *Developing an Ontological Sandbox: Investigating Multi-Level Modelling's Possible Metaphysical Structures* by Partridge, de Cesare, Mitchell, Gailly and Khan addresses this issue by proposing a structure within which definitions can be constructed and analyzed.

System engineering tooling is an area that is highly appropriate for the application of multi-level modelling since tools must manage type-level information as data while at the same time allowing tool-users to manipulate tool data as types. The benefits of the approach together with a proof of concept implementation is investigated in the context of model-based user interface development in the paper *A Multi-level Approach for Model-Based User Interface Development* by Bjorn Benner. In addition, tooling must offer a range of user functionality that is an extension of that supported by traditional tools, addressing issues such as: *what happens to models and their instances when type-levels are changed*. The paper *Maintenance of Multi-Level Models – An Analysis of Elementary Change Operations* by Toepel and Benner presents a framework within which multi-level model change functionality can be defined and studied.

Programming languages have provided meta-level features since the early days of Lisp and Smalltalk. Lisp has run-time types and a self-defined interpreter. Smalltalk has meta-classes. More recent languages, such as Java, have a reflective library to inspect the program at run-time. However, the motivation in most cases relates to implementation issues rather than modelling. The paper *DeepRuby: Extending Ruby with Dual Deep Instantiation* by Neumayr, Schuetz, Horner and Schrefl shows how principled multilevel modelling can be implemented in the Ruby programming language.

Multi-level modelling introduces the notion of user-defined types together with the constraints that classify the instances of the type. The question of how to check the user-defined type constraints is addressed by the paper *Validated Multi-Layer Meta-modeling via Intrinsically Modeled Operations* by Mezei,

Urbán and Theisz who propose a modular approach that offers a validation framework.

Multi-level modelling offers the potential for increased abstraction and reuse when representing data. Such abstraction is key when aiming to achieve data integration through the definition of new language features whose property constraints apply across type levels. The paper *Applying Multi-Level Modeling to Data Integration in Product Line Engineering* by Nestic and Nyberg describes how power-types can be used in product-lines to succinctly capture constraints such as disjointness and completeness.

Working with multi-level models poses an interesting challenge since there is no agreed approach to visually present and manipulate the many different type levels and their relationships. The paper *An Example Application of a Multi-Level Concrete Syntax Specification with Copy-and-Complete Semantics* by Jens Gulden describes the application of a new language (the Topology Type Language) to this problem in terms of the step-by-step construction of a multi-level model.

The field of multi-level modelling has recently expanded to produce several approaches and technologies. Some address different problem domains as described by the papers in this workshop; some agree on key features and others take opposing views. There is a need to provide a way of evaluating the application of multi-level approaches as described in the paper *On Evaluating Multi-Level Modeling* by Atkinson and Kühne.

The poster *Extending a UML and OCL Tool for Multi-Levels: Applications towards Model Quality Assessment* by Doan and Gogolla shows how the USE tool can be extended to support multi-level modelling.

III. MULTI CHALLENGE

Multi-level modelling is an active field with potential to provide significant improvements to system engineering. However, it is a challenging area and to date the multi-level community has tended to focus on technology issues making the benefits difficult to appreciate for those outside the field. The MULTI 2017 organizers proposed and presented a

challenge based on a real-world scenario, and invited researchers to use the challenge as the basis of concrete contributions that can be used to raise the profile of the field. The challenge was discussed at the workshop whose delegates agreed to refine the details for MULTI 2018 with the aim of attracting as many contributions as possible. The 2017 version of the challenge is shown in figure 1. The challenge is intended as the basis for any multi-level modelling approach providing that concrete benefits can be demonstrated.

Multi-level features of any submission to the challenge may include any of the following: knowledge about the domain should be represented at the highest level possible; the model can be a foundation for a software system suited for a wide range of general bicycle stores including specialization to, for example, a dealer of professional racing bikes; associations and constraints should cross levels where appropriate; the integrity of lower levels of the model should be consistent with any changes applied to higher levels; mechanisms should synchronize MLM-based models with code.

The following are examples of application-level features of any submission to the challenge: multi-level modelling can be used as a basis for configuration for example *every bicycle type except for racing bikes may be equipped with an electric motor and electric bikes need enforced brakes and a battery*; as a basis for advanced business analytics, for example *find every bicycle type that has an electric motor and that has the least number of sales in 2017*; representing business processes in multi-level models, for example *order management, such as Customer, Order*; addressing behaviour abstraction within multi-level models, for example *most dealerships favour their own type of order management process, a multi-level model of an order management process should support the reuse of common aspects of order management and extend/refine them to satisfy particular requirements*; generating a bicycle product management system from a multi-level model that uses models@run-time to support the addition of new types of bicycle.

The challenge is used as the basis of exemplifying multi-modelling using MultEcore as demonstrated in the MULTI 2017 paper *Multilevel Modelling with MultEcore: A Contribution to the MULTI 2017 Challenge* by Macías, Rutle and Stolz.