# An Implementation of Multi-Level Modelling in F-Logic

Muzaffar Igamberdiev, Georg Grossmann, and Markus Stumptner

Advanced Computing Research Centre
School of IT and Mathematical Sciences
University of South Australia, Mawson Lakes, SA 5095, Australia
{firstname.lastname}@unisa.edu.au

**Abstract.** Multi-level modelling is currently regaining attention in the database and software engineering community with different emerging proposals and implementations. One driver behind this trend is to reduce model complexity, a crucial aspect in a time of *big data* research in which more and more data from different sources are required to be integrated. From our experience, multi-level modelling also improves understanding of complex specifications, simplify their management and evolution, and facilitate interoperability between them. This paper focuses on the requirement of reasoning for interoperability. Although there exist formalisation approaches for multi-level modelling, only few have the implementation for three fundamental aspects: formalisation, querying and validation of multi-level models. We propose an F-Logic framework to implement these aspects. In addition, we believe this approach is more likely to be adapted in real-world use cases because of its simple object-oriented declarative nature.

**Keywords:** Multi-level modelling, interoperability, multi-level model reasoning, F-Logic, multi-level model querying, multi-level model validation

## 1 Introduction

Multi-level modelling (also called *deep meta-modelling/instantiation*) is currently regaining attention in the database and software engineering community with different emerging proposals and implementations. Recently there have been multiple works published which enrich multi-level modelling with new features [2,19], propose a formalisation for multi-level modelling [21] or demonstrate the practical application of it [8,13]. The most often mentioned arguments for multi-level modelling are *increased expressiveness*, by introducing multiple classifications [5], and *reduced complexity* [6,20]. This seems to be a contradiction, because one might expect increasing expressiveness may lead to increased complexity, but multiple classification allows to brake down a complex specification into smaller and simpler layers. Apart from the above mentioned advantages we have seen a further three advantages in a use case from the oil and gas industry [13]: (1) Simplification of the standards' specifications by classifying elements into ontological and linguistic elements, (2) simpler management and evolution of standards by structuring them into multiple ontological *instance-of* levels, and most importantly (3) checking specifications for consistency according to software engineering modelling principles.

The third advantage facilitates the interoperability between software systems. During the development life cycle of an interoperability solution, the matching, transformation, and synchronization of models and data rely on querying source and target specifications and checking for consistency to ensure a correct end result (i.e. validation). It only becomes possible with a formal specification that can be executed.

For multi-level modelling there exist some formalisation approaches. Neumayr et al. [19] proposed ConceptBase as the underlying formalisation framework. ConceptBase is a metamodeling system based on Datalog and the Telos data model [18]. Rossini et al. [21] implemented the semantics using the Diagram Predicate Framework (DPF) and Golra et al. [11] used a graph algebra.

We propose a novel approach, namely Multi-level Modelling in F-Logic (MiF), for the implementation, querying and validation of the multi-level models. We propose to use F-Logic as an alternative implementation for the following reasons: in comparison to DPF and graph-based approaches, F-Logic is object-oriented and represents an integrated framework which allows the specification of the semantics and ontological models as well as perform reasoning. ConceptBase also aims at an integrated approach but is not as widely accepted as F-Logic. In particular in the Ontology and Semantic Web community, e.g., there exist commercial and open-source implementations such as OntoBroker [1] and Flora [2], and it has been used for interoperability, such as the Rule Interchange Format [14][3] and model transformation [15].
In the next section we describe our motivating use case from the oil and gas industry, followed by a description of the implementation of multi-level modelling and related work.

## 2 Oil & Gas Interoperability Pilot

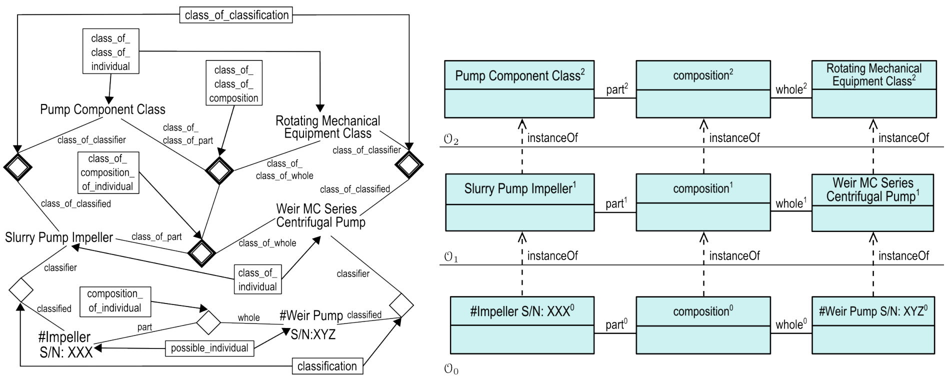
A large-scale standard-based interoperability is one of the main challenges in the oil and gas industry. Some comparable figures of how much an inadequate interoperability costs came from the US Capital Facilities Industry and the construction and engineering domain with an estimate of about $15.8 billion per year [9,10].

A lot of effort has been invested into data standards to overcome the interoperability issue in the oil and gas industry. One effort is the joint academic-industry project *Oil & Gas Interoperability (OGI) Pilot* hosted by MIMOSA[4] and supported by the ISO TC 184 OGI Technical Specification project. The

[1] OntoBroker: http://www.semafora-systems.com/
[2] Flora-2: http://flora.sourceforge.net/
[3] RIF: http://www.w3.org/2001/sw/wiki/RIF
[4] MIMOSA: http://www.mimosa.org

goal of the OGI Pilot is increased automation in the digital hand-over of design information of very large physical assets to the operation and maintenance side. This requires identifying commonalities and differences as well as open gaps in the specifications of the major standards in the area: ISO 15926 [12] and MIMOSA's Open Systems Architecture for Enterprise Application Integration (OSA-EAI) [17]. Within the OGI Pilot we have identified some challenges [16] of which we will focus on ISO 15926 standard in this paper. The ISO standard relies on a 4-dimensional information model specified in STEP/EXPRESS, RDF, OWL, and first order logic.

**Example:** Our motivating example is taken from an engineering diagram which specifies that "*The impeller with serial number XXX is part of the Weir Pump with serial number XYZ*". Figure 1a displays the flat model of this example using the *instance diagram* notation appearing throughout the ISO 15926 documentation [12].




(a) Instance diagram (b) Multi-level representation

Fig. 1: Two representations of the same example: *relating an impeller to a pump.*

In ISO 15926 terminology each box represents a *class* which is part of the specification and identified by its label. A diamond represents a *relationship* where a diamond with a thick line represents a *class of relationship.* A (class of) relationship has *roles* which are displayed by labelled arcs connected to the diamond, e.g., "part" and "whole" are the two roles of "composition_of_individual". A symbol with prefix # is a *possible individual with a temporal part*, e.g., "#Impeller S/N: XXX" is a possible temporal part with identifier "Impeller S/N: XXX". Remaining elements are *classes* identified by its label, e.g., "Pump Component Class".

Jordan et al. [13] applied rules on the ISO 15926 specification for the transformation of the flat model into a multi-level model. For example, one of those rules assigns a model level according to the prefix of a class label, e.g., "class_of_composition" is an instance of "class_of_class_of_composition" and removes the prefix "class_of_" to simplify the notation. Figure 1b shows the result of applying those rules on the example introduced above.

Some of the advantages of a multi-level model representation over the flat model are: (1) explicit *instance-of* relationships, (2) separation of concerns through multiple levels making it easier for users to focus on particular aspects of the model, (3) reduced complexity, and (4) clarified terminology which improves understandability.

In order to verify the multi-level models and perform queries, for example, to support matching with other standards for model transformation, we require a formal framework. We propose to use F-Logic because of its object-oriented semantics and its wide acceptance in the ontology, business rules and model transformation communities[1].

## 3 Implementation of Multi-level Modelling in F-Logic

This section introduces the semantics of multi-level modelling, its implementation, querying and validation in F-Logic.

### 3.1 Multi-level Modelling Semantics

The semantics of multi-level modelling involves the characterization of concepts and the definition of relationships between them. Similarly to semantics definition in two-level modelling, which is designated by its meta-model, the multi-level modelling semantics have been described by its meta-model that includes a *linguistic meta-model* and an *ontological stack* [3,21].

The fundamental concepts of multi-level modelling are characterized by linguistic and ontological perspectives. The linguistic meta-model deals with syntax and grammar, whereas the ontological stack addresses structural hierarchies and classification of an underlying domain. These perspectives make understanding of multi-level modelling easier and clearer. We also define the semantics in the light of the two perspectives.

The link between multi-level modelling perspectives is established by a *linguistic instance-of relationship.* It differs from the relationships within the linguistic meta-model and ontological stack in the sense that it connects the concepts across the perspectives. Every model element and relationship in the ontological stack is a linguistic instance-of the concepts from the linguistic meta-model. An ontological model element may or may not have an ontological type, but the linguistic type (e.g. clabject) is mandatory.

**Meta-model for use case:** The multi-level modelling meta-model for our use case is illustrated in Figure 2. It is organized in linguistic meta-model and ontological stack.

Some of the concepts are inspired by the work on multi-level modelling and the formalisation of deep meta-modelling [4,21]: The root element in the linguistic meta-model is called *instantiable element*, meaning all model elements are instantiable in the ontological stack. Depending on the representation of the

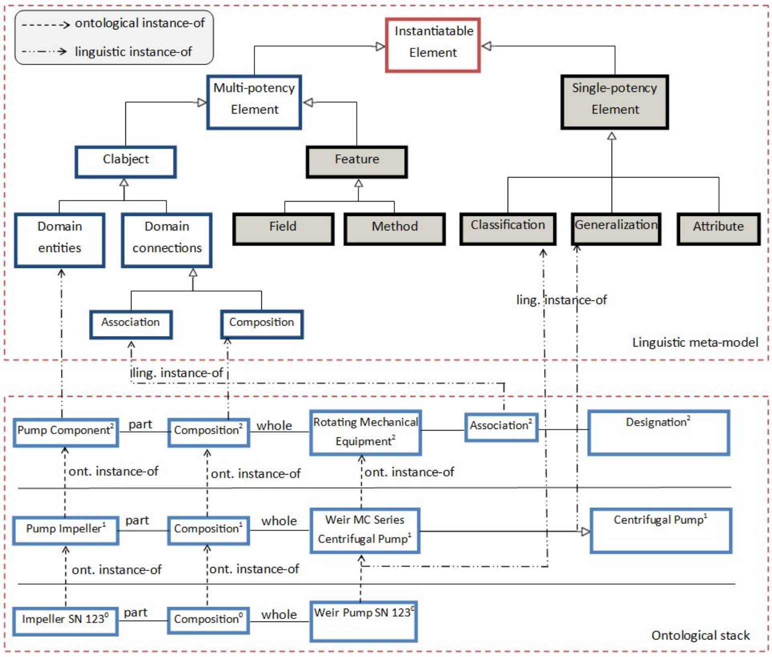



Fig. 2: Multi-level modelling meta-model with the motivational example

instantiated model element across ontological levels, instantiable elements are categorized as *multi-* and *single-potency elements* in Figure 2. While generalization and classification relationships cannot be instantiated across ontological levels, association and composition relationships can. The former ones are the examples for single-potency and latter ones for multi-potency elements. While the single potency elements can be instantiated in any ontological level and will not have further instances, the multi-potency elements can have further instances depending on the value of their potency[21]. The composition relationship is illustrated as a multi-potency relationship between *impeller* and *pump* elements in the example. The same multi-potency semantics is valid for clabject and association relationship.

Generalization relationship and attribute model elements are sub-class of a single-potency element. The difference between attribute and feature is that while the former one is considered as a single-potency, the latter one has multi-potency characteristics. Attribute is a property of the model element that can be instantiated in any ontological level and will not be instantiated in the next levels (single-potency). Alternatively, feature (field and method) is multi-potency element and can be instantiated across ontological levels. Clabject is sub-classed into *domain entity* and *domain connection*. Domain entities are the clabjects that characterize the domain concepts and domain connections address domain specific relationship, e.g., association and composition are domain connections.

The generalization is demonstrated with a relationship between *Weir MC Series Centrifugal Pump* and more general concept *Centrifugal Pump* in the ontological stack. The association is illustrated with a relationship between *Rotating Mechanical Equipment* and *Designation* model elements, that represents that rotating mechanical equipment has a designation.

### 3.2 Implementation in Flora-2

In this section, we first introduce F-Logic briefly, then discuss built-in features of F-Logic which directly support part of the meta-model and finally describe how we add new semantics to fully support the meta-model. Due to lack of space, we provide the excerpts of the semantics implementation.

F-Logic stands for Frame Logic, frame-based, object-oriented knowledge representation and reasoning language. It has a declarative, compact, simple and expressive syntax with well-defined semantics. These characteristics makes it attractive to apply on integration of information, semantic search, intelligent agents, semantic web and other areas.

In this paper, we use one of F-Logic implementations: Flora-2 [22]. Flora-2 is a dialect of F-Logic with numerous extensions and it supports extensibility, flexibility and modularity through dynamic modules. It is more suitable for a knowledge representation and reasoning in a way that multi-model semantics can be compactly expressed and the constrains can be checked, validation rules can be applied and more importantly the multi-model concepts and relationships can be easily queried. The source code presented in this paper is based on Flora-2 syntax. We now continue with the overlapping and distinct features of F-Logic and multi-level modelling.
**Direct support:** Some of the multi-level modelling concepts can be mapped directly to F-Logic. Concepts which are supported directly are represented with in grey in Figure 2. The mapping between modelling concepts and F-Logic elements are illustrated in Table 1.

| **MLM Concept** | **F-Logic Concept** | **Flora-2 Example** |
|---|---|---|
| Generalization | Subclass | `A::B. subclass::class.` |
| Classification | "IS A" relationship | `A:B. object:class.` |
| Attribute | If $M$ in `O[M->V]` is a constant it is dealt as an attribute | `Pump[component->'impeller'].` |

Table 1: Multi-level modelling concepts and equivalent representation in F-Logic and its implementation in Flora-2.

**Multi-level modelling aware extensions:** The linguistic meta-model elements with white background in Figure 2 represent model elements and relationships that have not been addressed by F-Logic yet. Due to space limitation we provide its the implementation only for the linguistic and ontological instance-of relationship:

**Linguistic instance-of relationship:** It is similar to the `IS-A`(instance-of) relationship implemented by the colon (`:`) operator in F-Logic. We introduced a new operator, (`<:`), for the linguistic instance-of relationship with the following short excerpt of validation:

```
1. :- _op(400,xfx,<:).
2. linguistic_instance_of_validation(?X, ?Y) :-
3.    ?X <:: ?Y,
4.    ?Y \= clabject,
5.    writeln(?X, ' can only be an instance of CLABJECT') @ _plg.
```

The operator is defined by `_op(400,xfx,<:)` statement. The first argument (`400`) defines precedence order to follow when statement contains other elements. We define precedence as the same as F-logic's instance-of relationship precedence.
The Lines 2-5 represent a rule in F-Logic to validate the linguistic instance-of relationship. If the model element in ontological stack is not a linguistic instance of clabject (`?Y\=clabject`), then it prints a validation fail message on the screen (Line 5).

**Ontological instance-of relationship:** We introduced new operator '`<::`', which specifies the semantics of the ontological instance-of relationship. The rule (Lines 2-5) checks the potency before ontologically instantiating the model element.

```
1. :- _op(400,xfx,<::).
2. ?X <:: ?Y :-
3.      ?Y[potency -> ?_P], ?_P == 0,
4.      \+(?X[potency -> ?_P - 1]),
5.      writeln('cannot instantiate! It should be P > 0.') @ _plg.
```

The second argument of the `_op()` operator, `xfx` is used to define the type where `f` stands for the operator, and `x` and `y` stand for the arguments. The negation operator is denoted by `\+` symbol in Flora-2.

### 3.3 Querying and Validation

An essential feature of F-logic is reasoning. This paper focuses on the querying and validation aspects of reasoning. A knowledge base is built based on the facts (e.g. like the ones in the previous subsection) and can be easily queried. The following facts illustrate ontological instance-of relationships on the motivational example.

```
weir_mc_series_centrifugal_pump<::rotating_mechanical_equipment.
weir_pump_sn_123<::weir_mc_series_centrifugal_pump.
```

Validation rules can be introduced to check certain properties of the concepts and relationships. For example, the following predicate, validation rule checks for the condition that a potency of a property should be equal or less than the potency of an object.

```
validate_property_potency(?Prop):-
   ?X[property->?Prop],
   ?X[potency->?XPot],
   ?Prop[potency(?X)->?PropPot],
   ?PropPot=<?XPot.
```

Further, the knowledge base can be queried. For example, to determine the potency or all instances of a particular pump:

```
?- weir_pump_sn_123[potency->?Potency].
?- writeln('Give me all instances of the pump') @ _plg,
   instances(?X, weir_mc_series_centrifugal_pump).
```

## 4 Related work

Even though multi-level modelling was introduced more than ten years ago, the formalisation of its semantics has only recently been addressed. Research on multi-level modelling started to get momentum, and some formalization or implementation attempts were made. In this section we compare related work using six comparison criteria: *(1) Linguistic extension and open semantics*: support to extend the linguistic meta-model with new concepts and relationships. E.g. specifying semantics of "membership" relationship, *(2) object-oriented semantics*: the framework is based or supports object-oriented modelling principles, *(3) integrated framework*: a single framework for formalization, querying and validation, *(4) relationship across levels*: a support for the relationships across levels in the ontological stack, *(5) mediation of relationship*: a need of intermediate relationship to instantiate the relationship in not-immediate ontological/instantiation level, and *(6) single and multi-potency semantics*: a support of single and multi-potency concepts in the ontological stack (see Figure 2).

The evaluation of some multi-level modelling approaches is illustrated in Table 2.

The linguistic and semantic extension criteria are covered by almost all of approaches. The object-oriented criterion is supported by most of approaches including this paper. We benefit from object-orientation in two ways: (1) It already covers part of multi-level modelling and (2) it is built-in paradigm of F-Logic. In the context of the integrated framework criterion, ConceptBase [19] (based on Datalog and Telos) addresses formalism and validation, and METADEPTH [21] deals with all components of the integrated framework:

| Criteria/Approaches | [4] | [11] | [19] | [21] | [7] | MiF |
|---|---|---|---|---|---|---|
| Linguistic extension and open semantics | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| Object-oriented | ✓ | – | ✓ | – | ✓ | ✓ |
| Integrated framework | ✓ | ✓ | – | – | ✓ | ✓ |
| Relationship across levels | – | – | ✓ | ✓ | – | – |
| Mediation of relationship | – | – | ✓ | ✓ | – | ✓ |
| Single and multi-potency semantics | ✓ | – | ✓ | ✓ | ✓ | ✓ |

Table 2: Evaluation of multi-level modelling approaches.

formalization(METADEPTH), querying(Epsilon Object Language (EOL)) and validation (Epsilon Validation Language (EVL)). The Multi-level modelling in F-Logic (MiF) approach uses F-Logic / Flora-2 to implement the formalism, querying and validation. The relationship across levels is used between models with different ontological structures and in different domains/spaces [19,21]. MiF approach could also support relationship across levels as well, however we did not come across a use case in the OGI Pilot so far. All DPF[21], DDI[19] and MiF support mediation of relationship, additionally MiF supports mediation of a composition relationship as well. Almost all approaches support the criterion of single and multi-potency semantics. F-OML [7] approach behaves as the same as MiF approach, except in mediation of relationship criterion.

## 5 Conclusion

In this paper we introduced an alternative implementation of multi-level modelling using F-Logic and Flora-2. We have applied the implementation on a subset of the OGI Pilot use case, which dealt with the specification of a pump using the ISO 15926 standard. The main benefit behind this proposal is the integrated approach of F-Logic which allows the specification of the semantics, ontological stack and reasoning capabilities (i.e. querying and validation) in a single framework, its object-oriented semantics and its wide acceptance in the ontology modelling community. In the future we plan to implement the semantics of another standard used in the OGI Pilot, the MIMOSA standard and use the reasoning capabilities of F-Logic to automate the matching and transformation between ISO 15926 and MIMOSA.

## References

1. Juergen Angele, Michael Kifer, and Georg Lausen. Ontologies in F-Logic. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 45–70. Springer Berlin Heidelberg, 2009.
2. Colin Atkinson and Ralph Gerbig. Level-Agnostic Designation of Model Elements. In *Proc. of ECMFA 2014*, volume LNCS 8569, pages 18–34. Springer, 2014.
3. Colin Atkinson, Bastian Kennel, and Björn Goß. The level-agnostic modeling language. In *Software Language Engineering*, pages 266–275. Springer, 2011.

4. Colin Atkinson and Thomas Kühne. The essence of multilevel metamodeling. In *UML 2001*, pages 19–33. Springer, 2001.
5. Colin Atkinson and Thomas Kühne. Rearchitecting the UML Infrastructure. *ACM TOMCATS*, 12(4):290–321, 2002.
6. Colin Atkinson and Thomas Kühne. Reducing accidental complexity in domain models. *Software and System Modeling*, 7(3):345–359, 2008.
7. Mira Balaban and Michael Kifer. Logic-based model-level software development with F-OML. In *Model Driven Engineering Languages and Systems*, pages 517–532. Springer, 2011.
8. Juan de Lara, Esther Guerra, Ruth Cobos, and Jaime Moreno Llorena. Extending deep meta-modelling for practical model-driven engineering. *The Computer Journal*, 57(1):36–58, 2014.
9. Fiatech. Advancing Interoperability for the Capital Projects Industry: A Vision Paper. Technical report, Fiatech, February 2012.
10. M. P. Gallaher, A. C. O'Connor, Jr. Dettbarn, J. L., and L. T Gilday. Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. Technical report, NIST, 2004.
11. Fahad R. Golra and Fabien Dagnat. The Lazy Initialization Multilayered Modeling Framework. In *Proc. of ICSE 2011*, pages 924–927. ACM, 2011.
12. ISO. ISO 15926: Industrial automation systems and integration Integration of life-cycle data for process plants including oil and gas production facilities. Technical report, ISO, 2004.
13. Andreas Jordan, Georg Grossmann, Wolfgang Mayer, Matt Selway, and Markus Stumptner. On the application of software modelling principles on ISO 15926. In *Proc. of the Modelling of the Physical World (MOTPW) Workshop at MODELS 2012*. ACM, 2012.
14. Michael Kifer. Rule Interchange Format: The Framework. In *Prof. of RR 2008*, LNCS 5341, pages 1–11. Springer, 2008.
15. Michael Lawley and Jim Steel. Practical Declarative Model Transformation with Tefkat. In *MoDELS Satellite Events*, LNCS 3844, pages 139–150. Springer, 2005.
16. Wolfgang Mayer, Markus Stumptner, Georg Grossmann, and Andreas Jordan. Semantic Interoperability in the Oil and Gas Industry: A ChallengingTestbed for Semantic Technologies. In *AAAI 2013 Fall Symposium on Semantics for Big Data*, 2013.
17. MIMOSA. Open systems architecture for enterprise application integration (osa-eai) 3.2.3. Technical report, MIMOSA, 2012.
18. John Mylopoulos, Alexander Borgida, Matthias Jarke, and Manolis Koubarakis. Telos: Representing Knowledge About Information Systems. *ACM TOIS*, 8(4):325–362, 1990.
19. Bernd Neumayr, Manfred A. Jeusfeld, Michael Schrefl, and Christoph Schätz. Dual Deep Instantiation and Its ConceptBase Implementation. In *Proc. of CAiSE 2014*, LNCS 8484, pages 503–517. Springer, 2014.
20. Bernd Neumayr, Michael Schrefl, and Bernhard Thalheim. Modeling techniques for multi-level abstraction. In *The Evolution of Conceptual Modeling*, pages 68–92. Springer, 2011.
21. Alessandro Rossini, Juan de Lara, Esther Guerra, Adrian Rutle, and Uwe Wolter. A formalisation of deep metamodelling. *Formal Aspects of Computing*, in press(in press):1–41, 2014.
22. Guizhen Yang, Michael Kifer, and Chang Zhao. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *Proc. of OTM 2003*.