Applied use of Ehrenfeucht-Fraisse Games in Conceptual Model Management and Model Transformation

Author: Victor Morgante
(victormorgante@viev.com)

Abstract

We investigate the expressive power of Object-Role Modelling (ORM) by applied use of an adapted Ehrenfeucht-Fraisse Game (EFG), ranging over theorems of ORM stored within a meta-model of ORM under non-standard analysis and where the original ORM syntax was formalised as isomorphic with theorems of a theory under Finite-Model Theory (FMT). Focus is on the practical benefits of differential interpretation of theorems of a theory under Finite Model Theory, including structure and behaviour modelling within a single meta-model with applied use of EFGs to achieve model transformation. The investigation extends to contemporary trends in conceptual modelling, model management and model driven engineering. Application of the approach is presented as a research update of the Boston computer software developed to implement the hypothesis that model transformation may be viewed as differentiated interpretation by the duplicator of an Ehrenfeucht Fraisse Game.

Introduction

Ehrenfeucht-Fraisse Games (EFGs) have found little practical application outside the study of logic. Indeed, from a practical standpoint, demonstration that theorems of a theory of under Finite Model Theory (FMT) may have multiple interpretations may seem problematic and somewhat unwanted.

Where the bijective nature of isomorphism and homomorphism is given consideration, an alternate view of EFGs may be established. That view holds that multiple structures may usefully, desirably and deliberately map to structures under one other theory.

In a contemporary setting, the problem of managing interrelated conceptual models ranging over a single domain has attracted considerable academic and commercial focus. For example, the Object Management Group (OMG) refines the UML (Unified Modeling Language) under the notion that various conceptual modeling languages (CMLs) may be unified to range over a single domain in what is called a Platform Independent Model (PIM) (Melnik 2004, p. 207). This effort has extended to the OMG's Meta-Object Facility (MOF) (Object Management Group 2016) and XML Metadata Interchange (XMI) standards under a paradigm that recognises the practical problems associated with maintaining a PIM.

By application of an adapted Ehrenfeucht-Fraisse Game ranging over theorems of the theory of Object-Role Modeling (ORM) by way of non-standard analysis of those theorems under finite-model theory, we investigate how an ORM based PIM may be established such that theorems of ORM may quickly and efficiently be transformed into theorems of a conceptual modelling language other than ORM by way of differentiated interpretation. Object-Role Modeling is a graphical theory synthesised as a morphism of Knowledge Language (KL), a symbolic theory synthesised as a theory under FMT in (Halpin 1989). An example of such application of EFGs would be theorems of an ORM model defining a structure that may be interpreted as structures of a UML Class Diagram (Doesburg & Balsters 2012), a UML Use Case Diagram, a Sequence Diagram or a Data Flow Diagram, where those are non-standard interpretations of theorems of ORM.

The hypothesis extends existing theory that any CML with a meta-model that can be reduced to the relations of a first-order logic (FOL) may be modelled within ORM (Lemmens, Nijssen & Nijssen 2007,

p. 881, p. 353); and established as a conceptual meta-model language (CMML) stored within the ORM meta-model. Fagin et al long ago entertained the idea of a central 'core' model from which other models were derived and exchanged (Fagin, Lolaitis & Popa 2005).

Developments such as the acceptance of the fact based modeling standard, SBVR (Semantics for Business Vocabulary and Business Rules) (Object Management Group 2008), and a challenging academic paper by Lemmens, Nijssen and Nijssen add support to research that positions fact-based modelling (FBM) approaches as important at a meta-modelling level (e.g. ORM and NIAM2007) (Lemmens, Nijssen & Nijssen 2007).

Background to the problem space – Why Ehrenfeucht Fraisse Games and Why ORM?

Managing interrelated modelling languages ranging over a single Universe of Discourse (UoD) is what defines the problem space.

Currently favoured architectures dealing with multiple models (e.g. the UML), take three major forms. The first is to use a four-layer metamodel architecture, as in the OMG's Meta Object Facility (MOF). The second relies on model transformation to translate diagrams of one CML to another, while ostensibly operating within one meta-model. For instance, the OMG has introduced the 'Query View Transform' (QVT) Standard to fulfil this role with standards such as the UML. The third contemporary trend is to work with multiple meta-models. For example, the OMG's Business Process Definition Meta-Model (BPDM) standard, addressing 'behaviour' type models, is a separate meta-model from the UML meta-model.

Forgoing the political and economic pressure to set standards in a timely fashion, a process which has arguably influenced the type, quantity and quality of metamodels established, the problems associated with managing metamodels has certainly led to heightened awareness of the science of model and metamodel management (Manzano 1999, Lemmens, Nijssen & Nijssen 2007). Active research looks to merge the BPDM and UML metamodels (White 2004), for instance, in recognition of similarities between metamodels and the benefits of reducing the number of metamodels under management. The OMG also proactively maintains an issue register, modifying standards as a result of analysis of the issues (Object Management Group n.d.).

Listed as an important contemporary issue within model management is the notion of maintaining the consistent identity of model elements across CML meta-data (Object Management Group 2016, p. 23). That is, the identity of model elements within the metamodel of one CML may require synchronisation with the same model elements within the metamodel of another CML, where both model elements have the same logical identity within a UoD.

It is *identity management* of model elements between CML models and metamodels that is the problem most directly refined by the hypothesis. Simplification of model transformation and model interchange is a targeted by-product of the software architecture introduced to implement the hypothesis.

Addressing the Identity Management Problem

An intuitive leap is taken to establish a paradigm where *data* (structure[1]) and *process* (behaviour) model development is performed within one meta-model such that model elements are declared only once, and

[1] We omit *event* modelling in this treatment; however envisage no particular impediment to event modeling within the architecture developed under the hypothesis.

the diagrams of various CMLs projected as standard and non-standard interpretations of theorems under a base theory under FMT, using Ehrenfeucht-Fraisse Games. The intent of this approach is to significantly reduce the effort associated with model transformation, model interchange and identity management, with the software effectively interpreting theorems of a theory of FMT (ORM) as first-order or higher-order theorems by way of non-standard analysis of those theorems and the theory of ORM as a whole under an adapted Ehrenfeucht Fraisse Game.

In essence, the thesis is that the Object-Management Groups *4-Layer Architecture* for metamodel and model management can be achieved by applied use of Ehrenfeucht-Fraisse Games ranging over theorems of one singular base theory, and using one common metamodel, the CMML.

That the management of interrelated conceptual models ranging over a single domain (the intent of the UML and the MOF) is the logical problem of working with multiple interpretations of theorems of a single theory, while perhaps surprising, should not be controversial or altogether unexpected under considered reasoning. The proposal that a meta-model of first-order logic is sufficient to map any CML that can be reduced to facts (relations) enjoys contemporary support and research (Halpin & Morgan 2008, p. 881, p. 353, 5). Extant research sees model transformation as one use of an ORM metamodel (Doesburg & Balster 2012).

That is, research is already leading toward the notion of a single theory investigated under EFGs when speaking of conceptual modeling languages that are *unified*. The grounding of CMLs, such as ORM, under finite model theory, further spurs the hypothesis development under EFGs. An approach that sees model elements declared *once* within a single meta-model and used *many times* (within multiple CMLs) seems to best fit the declared objective of the UML [2].

Under a paradigm that reduces multiple meta-models to one meta-model, the identity of model elements are established once, with model interchange effort reduced to the serialisation of one meta-model and model transformation realised as *model interpretation*.

Applying EFGs to KL and theorems of ORM under non-standard analysis

Suitably, to exploit EFGs in the manner proposed, sentences of ORM must at least be isomorphic with a theory of FOL with an infinite model[3] under non-standard analysis while accepting that ORM is formalised and generally interpreted under Finite-Model Theory. That is, we adopt non-standard analysis of theorems of Object-Role Modeling, as the base theory, and interpret theorems of ORM as any other isomorphic language, or homomorphic language where some ORM theorems are considered as adding no semantics to the interpretation. In this manner we argue that the Lowenheim Skolem Theorem (LS), which necessitates ranging over a theory of FOL with and infinite model, may be applied to theorems of ORM where ORM theorems, under FMT, are interpreted using Ehrenfeucht Fraisse Games extended to the higher order. I.e. it is well known that theorems of FMT may be interpreted as theorems under higher-order logic (Väänänen n.d.) using Ehrenfeucht Fraisse Games, providing a logical game theoretic backbone for the approach and all the while accepting that LS does not apply to the standard interpretation of a theory under FMT.

[2] That no current research mentions EFGs as most applicable to achieving multiple interpretations of a single domain in conceptual modelling, is probably to be expected, when given that multiple interpretations of theorems of a theory has often been to be avoided. From one perspective, EFGs intimate ambiguity, while from another (the approach adopted here) EFGs intimate ultimate clarity as to the identity of model elements within a model under a theory.

[3] The Upward Lowenheim Skolem Theorem is sufficient for our purposes. Researches wishing to explore the Downward Lowenheim Skolem Theorem in ORM may look to reproducing Cantor's Diagonal Theorem within Sample Population Sets (Fact Tables) within ORM as evidence of transfinite structure within theorems of KL and as isomorphic with theorems of ORM.

Current research already sees the application of non-standard analysis of ORM to realise CML transformation (Doesburg & Balster 2012). Of course, to extend Doesburg and Balsters hypothesis to realise the interpretation of UML Powertypes within an interpreted UML Class Diagram would necessitate the higher-order logical interpretation of an ORM metamodel (Halpin 2005).

We address here the non-standard analysis of the theory, Knowledge Language (KL), synthesised in (Halpin 1989) that allows the duplicator of an EFG to adopt higher-order logic in the interpretation of theorems of ORM, and define the approach:

1. That ORM version 1 (NIAM) has a model is implicit by design. ORM's basis language, NIAM, is isomorphic with a theory of symbolic logical, 'Knowledge Language' (KL) (Halpin 1989);

    a. The standard interpretation of ORM is formalised as isomorphic to the theory, Knowledge Language, a theory with finite structures under finite-model theory, with sentences of ORM limited to finite structures (Halpin 1989)[4];

    b. For every finite ORM model, there exists a set of KL theorems isomorphic with the ORM model (Halpin 1989);

2. Under finite-model theory, higher-order interpretation of ORM diagrams does not generally apply [5] (Bollen 2007, pp 16,23);

3. Any conceptual modelling language with a meta-model that can be reduced to relations of FOL, can be modelled in ORM (Halpin & Morgan 2008, p. 881, p. 353, 5);

4. When interpreting the theorems of a CML other than ORM, as stored within an ORM metamodel, we remove the requirement that sentences of ORM and KL have finite structure;

    a. Theorems of KL may then be interpreted as theorems under Zermelo–Fraenkel set theory (ZFC) [6] without the use of negation, which are then interpretable as higher order theorems under Ehrenfeucht Fraisse Games extended to the higher order;

    b. Theorems of ZFC may be investigated under the Lowenheim Skolem Theorem to include first-order and higher-order interpretations because we deal with a theory with an infinite model. We limit ourselves here, however, to EFGs extended to the higher order ranging over theorems of FMT achieving the same aim of having a base theory having alternate interpretations as when having an infinite model.

5. Under our interpretation, ORM and KL otherwise have an infinite model and we content ourselves that under the formalisation of KL (Halpin 1989), various structures and well formed formulas of KL have structures that may be extended to the infinite by removing the upper limit of 'n' in the formalisations defining the structure of sentences of ORM/KL. In particular, by removing the

[4] It is arguable that the current version of ORM, ORM2 is not isomorphic with KL, but we limit ourselves to NIAM/ORM here as synthesised in (Halpin 2989).
[5] For the purposes of this work, we ignore contemporary arguments that theorems of finite-model theory cannot be finitely contained as for each finite-model there exists at least one finite-model with one more theory than the first; leading to the notion of infinite models under an otherwise finite-model theory. ORM models are considered here as individually unique finite-models under their standard interpretation.
[6] Space restrictions here prevent the syntactic mapping of KL to ZFC;

upper limit of ‘n’ in the formalisation, leaving:

i. Fact Types that may have infinite Roles with associated Object Types (Entity Types, Value Types, Objectified Fact Types);
ii. No limit to the number of Fact Types within any one ORM model;
iii. Fact Tables, as Sample Population Sets representing data within ORM, that may range over boundless data sets [7];
iv. The Labels of Object Types in ORM as boundless in size and number, such that the interpretation of the Labels under an adapted EFG and combinatorics may be as symbols of the interpreted model[8].

Interim Result/Research Update: The Boston software (developed to implement the hypothesis) is currently at the state where Entity Relationship Diagrams, Property Graph Schemas and State Transition Diagrams are interpreted from theorems of a CMML injected within the ORM meta-model. The identity of Model Elements in Boston are established primarily once, under a regime that implements the recommendation under the OMG’s SBVR standard where the surrogate key of model elements match the *name* or *label* of the model elements. E.g. an Entity Type labelled, ‘Person’, in the Boston software, has a surrogate key at the meta-model level of ‘Person’ (Object Management Group 2008, p. 228).

The software adopts the classic 4-layer architecture, but reuses the same metamodel for various conceptual modelling languages, rather than having one metamodel for each language. For example, in the Boston software Entity Relationship Diagrams and Property Graph Schema use the same metamodel.

Interpretation of theorems stored in an ORM Meta-model Using an adapted Ehrenfeucht Fraisse Game

Refining the approach, multiple model interpretation under EFGs and over the meta-model of ORM has been achieved by use of Pages s (Halpin & Morgan, pp. 852-53) with each Page providing a fixed and limited subset of all possible sentences of ORM. Each Page is flagged as to the language of interpretation and as presented to the user of the software. ORM models/diagrams within Pages are realised as injections of ORM models within the ORM meta-model, and when a Page is retrieved from the database the relations may be interpreted as theorems of ORM or theorems of KL, or further as other languages and as need be.

An adapted Ehrenfeucht-Fraisse Game is applied to achieve differential analysis of theorems of ORM and/or KL to display each CML other than ORM.

1. Each Page-displaying-form within the graphical user interface in the software has a fixed interpretation of theorems of ORM and acts as a Duplicator in an EF Game (Bollen 2007, p. 26).

[7] This approach is utilised throughout the Boston software developed to implement the hypothesis, where Facts (data) within the ORM meta-model are interpreted as the symbols of theorems of CMLs other than ORM. While the approach of exploiting isomorphism of *data* was made famous by Gödel (Gödel 1992), Gödel Numbering is neither needed nor utilised to support the hypothesis; i.e. The Boston software interprets both the *symbols* of theorems of ORM and the *data* within theorems of ORM in combination as either standard or non-standard points of structure under EFGs.

[8] While boundless in an open world, existing natural languages reduce the set of Object Type Labels to thousands, rather than an infinite cardinality. Of course, in an open world, the cardinality of Labels is infinite.

2. The software retrieves theorems of ORM from the data-store and sends the theorems to the Page/Duplicator with the software, retrieving the theorems from the data-store and sending them to the Duplicator as acting as the Spoiler in an EF Game (Bollen 2007, p. 26).

3. The Page/Duplicator interprets the limited theorems received from the data-store.

4. Rather than interpreting the theorems as an ORM Model, the Page interprets the theorems as the theorems of the language specific to the Page type (e.g. an Entity Relationship Diagram) and displays the theorems to the user as under the language specific to the Page type.

   In effect, while the software supplies theorems of ORM to the Page, the Page *wins* the EF Game by using a theory homomorphic with the finite set of theorems provided, and displays the theorems under a diagrammatic language other than ORM.

5. The applied EF Game necessitates that the Duplicator effectively adopts a differentiated interpretation of theorems of ORM/KL to find an interpretation homomorphic with ORM/KL but which represents a different model. With each page only providing limited subsets of all possible theorems of ORM/KL, the theory of interpretation of the Duplicator (e.g. State Transition Diagram) considers all other possible sentences of ORM as providing no semantics. I.e. we are only interested in homomorphic subsets of theorems of ORM/KL as isomorphic interpretation of all possible theorems of ORM would merely result in the interpretation of each Page as an ORM diagram or set of KL theorems. EF Games are also, by definition, limited to a fixed set of rounds/theorems.

Interim Result/Research Update: In practice, the software does not first translate ORM theorems to KL theorems and then propose a theory isomorphic/homomorphic to KL. The software is written in acknowledgement that KL has multiple interpretations under EFGs when each sentence of KL is viewed as a theorem of ZFC ranging over an infinite model. Each interpretation is investigated in advance mapping a homomorphism from theorems of the CMML and hard-coded for each Page type. The existence of structures in the CMML with interpretations of than an ORM model demonstrates that there exists at least one substructure of ORM isomorphic with theorems of a theory other than ORM, as in the approach described by Doesburg & Bolsters (Doesburg & Bolsters 2012). The result is expected, as each CML chosen for the software has a model reducible to theorems of FOL/ZFC.

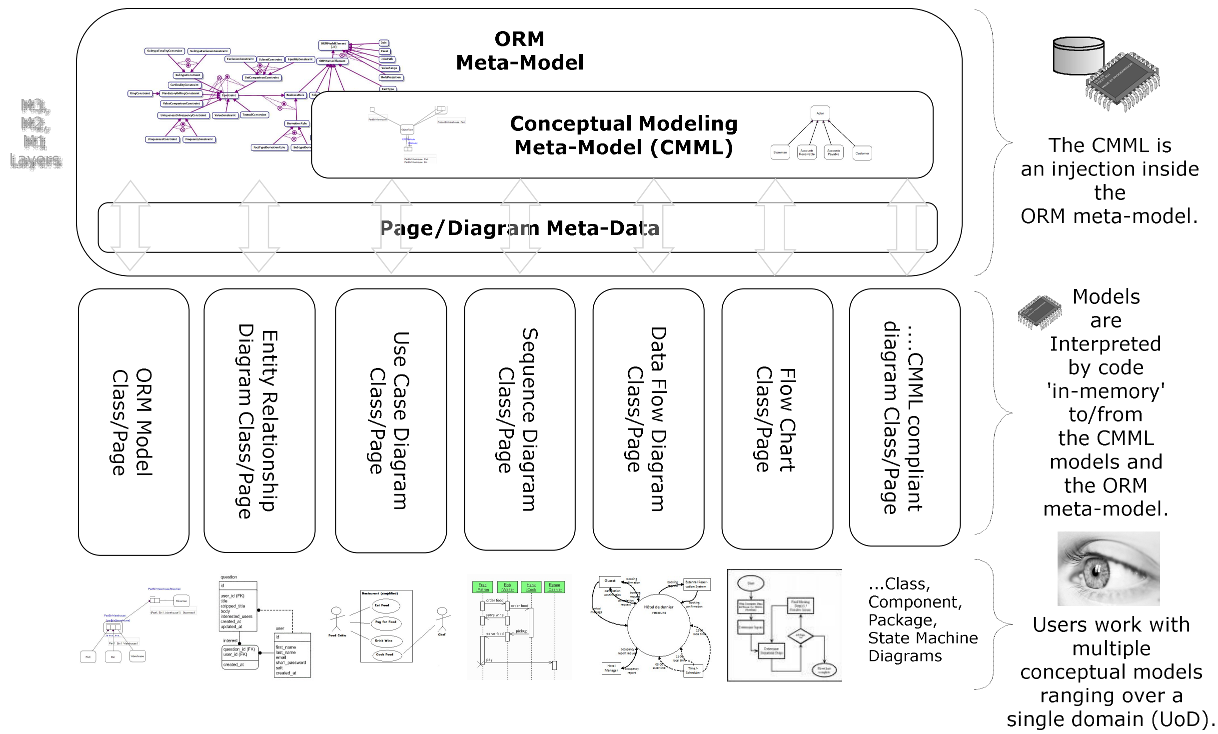


Figure 1. Overview of the Boston Architecture

By way of example, if a Page represents a State Transition Diagram, the software interprets the respective model elements within the CMML/meta-model (expressed in ORM) as a State Transition Diagram, and that is what in turn is diagrammatically presented to the user of the software. In like manner, when the user creates a diagram within a CML other than ORM, the Boston software injects the KL equivalent theorems of that CML back within the physical realisation of the ORM meta-model (database or in-memory data-store) as an ORM model. In this way, although the user is visually working with a State Transition Diagrams, Data Flow Diagram, Sequence Diagram etc, in reality they are constructing and interpreting well formed theorems of ORM (see Figure 1 and the example of use that follows).

Collections of ORM models, as Pages under various CMLs, refer to one UoD and may include data and process modelling within the ORM meta-model (Halpin & Morgan 2008, p. 832)[9]. Pages only share model elements under the UoD where the intended meaning of the identity of the model element is the same (e.g. An Entity Type, 'Customer', within a UoD has the same identity whether it is an Entity Type within an ORM Diagram or an Node within a Property Graph Schema). The intent of the Boston architecture is realised in this way to dispense with duplication of the identity of model elements.

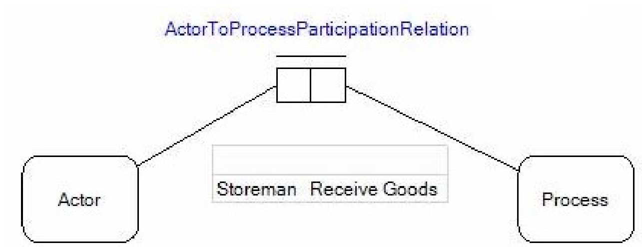
Example Use of the Boston Architecture/Software and CMML

[9] 9 Within the one work, alternate views of the usefulness of ORM to model behaviour, beyond structure, is addressed (Halpin & Morgan 2008, pp. 780-781, p. 832) with ORM listed as a suitable candidate for Process/Resource modeling (Halpin & Morgan 2008, pp. 780-785) and not suitable (Halpin & Morgan 2008, pp. 780-781, p. 832). For the purposes of this paper, we read the advice of (Halpin & Morgan 2008, p. 832) as 'state of the art' as it stood in 2008

In the space left available, a very simple example of how the architecture is implemented is provided. We limit ourselves to a single theorem of the following conceptual modelling languages[10]:

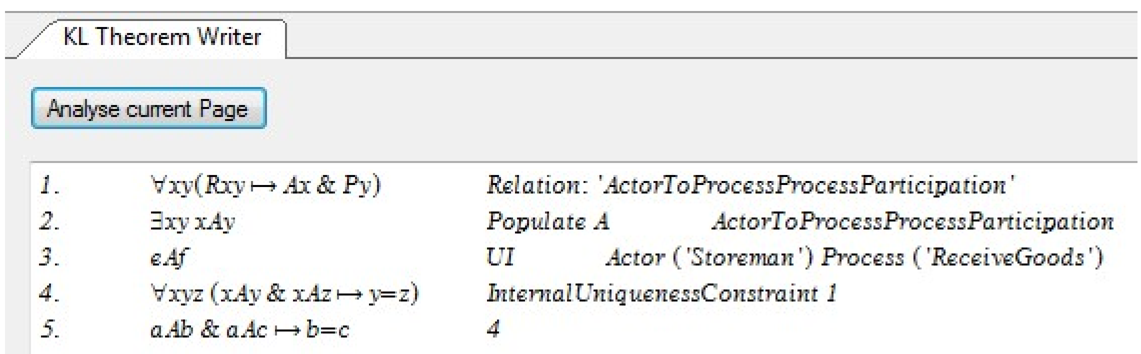
1. Object-Role Modeling;
2. Use Case Diagrams; and
3. Sequence Diagrams;

Central to the formalisation of ORM is the notion of Facts, or predicates, equitable with the relations (1)
of first-order logic (e.g. $F(x,y)$, where F is a function with free variables, x, and y). The following is a typical example of a simple ORM Model:



* This ORM Diagram is stored as a (M2 equivalent) Page as a CMML (meta-model) injection within the ORM meta-model upon which Boston operates. Users of the software have the option to visualise the contents of the meta-model directly within Boston, because to the software, the meta-model is just another ORM Diagram.

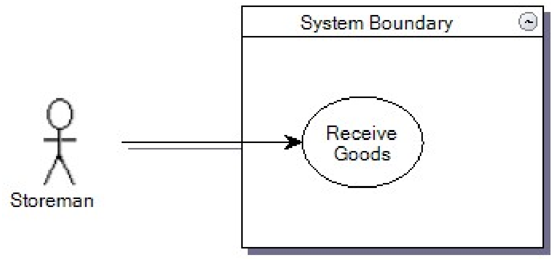
Trivially, under ORM the diagram in (1) maps to a formula of FOL as Rxy where R has an (2)
interpretation of 'ActorProcessParticipationRelation', x an interpretation of 'Actor' and y an interpretation of 'Process', or simply a function: ActorProcessParticipationRelation(Actor, Process). A KL Theorem writer is developed in the software to analyse ORM Models as theorems of KL. Output of the analysis of the ORM Model in (1) is shown below:



The same theorems may be interpreted syntactically as theorems of ZFC.

An isomorphic mapping of the fact, {'Storeman', 'Receive Goods'} of the Fact Type, (3)
'ActorToProcessParticipationRelation', in (1) would yield the following Use Case Diagram:

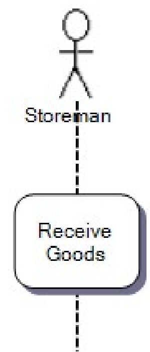
[10] The generalised result is assured under EFGs over a tailor made CMML. Given more space any number of examples could be provided.



* The Fact, {‘Storeman’, ‘Receive Goods’}, in (1) is established within the CMML meta-model from the Page where this Use Case Diagram was created. I.e. The architecture exploits homomorphism to get data *into*, and *out of*, the CMML meta-model and ORM meta-model underlying the architecture.

Under the interpretation, theorems 4 and 5 of (2) (the KL interpretation of the ORM Diagram uniqueness constraint) are realised in the formalisation of Use Case Diagrams and not explicitly shown on the diagram. The semantics of theorems 4 and 5 still hold for Use Case Diagrams.

The segment of a Sequence Diagram to the right uses the same fact established in (1) such that model transformation (Use Case Diagram (3) to Sequence Diagram) is achieved by non-standard interpretation of theorems of ORM under an adapted Ehrenfeucht Fraisse Game.



(4)

The implication of (1), (2) and (4) collectively is simply that an ORM metamodel is sufficient for the establishment of ORM models and a common conceptual modelling meta-language (such as the MOF) stored within the one metamodel.

A Conceptual Modelling Meta-Language (CMML) within an ORM meta-model

Satisfied that the meta-model of ORM is suitable for modeling multiple CMLs under one metamodel and using EFGs, we move to the development of a common Conceptual Modeling Meta-Language[11].

The CMML is simply a metamodel and called a language here for our purposes. The CMML holds information of a common model investigated under a Universe of Discourse such that Object Types, and the various other model elements of an ORM Model, such as Role Constraints and Facts, may be shared between Pages. The Language (or varied interpretation of the CMML) of each Page determines how

[11] Other than ORM, no consideration is given to the proof theoretic nature any of the other CMLs included within the investigation; neither is any CML, such as the UML favoured. We merely content ourselves that the chosen CMLs have one thing in common; the graphical nature of the theorems of the respective methodologies comprise of symbols and implied relations between symbols, where those symbols and implied relationships have intentional meaning. That is, CMLs such as Use Case Diagrams have found use and acceptance regardless of any proof theoretical foundation. Researchers of combinatorics and model theory may look beyond this paper to the Curry Howard Correspondence for the proof theoretic nature of individual modeling languages, and to where a language such as the UML may be used to generate computer software. i.e. We are not concerned with the politics here of which CML is better or more useful than another, as the architecture of the hypothesis can be used if all the user of the software wants to do is develop models under *one* language (e.g. ORM Diagrams).

model elements are presented as diagrams[12]. The only meta-data beyond an ORM model stored against each Page are the coordinate position and orientation of model elements as they appear on a computer screen or hardcopy printout. That is, the Boston software interprets ORM diagrams and projects them as non standard models of ORM as variable interpretations of the CMML/different languages.

Research Update: The development of a CMML within the Boston software is maturing and there are two approaches available going forward. The first is to develop a carbon copy mapping of existing meta-models for various CMLs (e.g. the meta-models of the CMLs under the UML) within the ORM meta-model, or to adopt a generic approach where the relationship between model elements is defined once only, more akin to the BPDM specification. It is the second approach that seems the most compelling and as described in this paper, as model transformation is rapid where isomorphism/homomorphism can be established between model elements and relations of different CMLs. i.e. The intent is to minimise the effort of model transformation, and an approach where every model draws from a single meta-model is the most compelling.

Brief Comparison of the Boston Architecture to Existing ORM Modeling Tool Architectures

The *Pages* approach implemented within the Boston software has partially, and successfully, been adopted by other existing ORM modelling tools such as NORMA[13], where large ORM models are segmented into Pages (ORM Diagrams). A Page/ORM Diagram in NORMA may be either a complete ORM model or part of a larger ORM model. The interpretation within Boston differentiates from this approach by adopting a position that ORM models ranging over a single domain (UoD) within the ORM meta-model may be interpreted by Pages representing other conceptual modelling languages. E.g. UML Class Diagrams, Entity Relationship Diagrams, Data Flow Diagrams or any diagrammatic CML that may be reduced to a theory based on relations.

Diagrams homomorphic with ORM Diagrams are readily generated from the ORM meta-model within existing ORM modeling tools. One example is the Barker ER Diagram view in the NORMA software. The Boston architecture and approach differs from that taken by NORMA, however, with the projected theorems required to generate ER Diagram Pages stored back within the ORM meta-model itself and where NORMA stores the Barker ER diagrams separate from the core ORM meta-model.

Conclusions

Offering no proofs in this work, the conclusions of this paper are that of empirical study under the development of the Boston software.

1. The meta-model of ORM is suitable for combined M1, M2 and M3 layers (model, meta-model, meta-meta model) and can store any number of languages reducible to theorems of FOL;

2. The Boston architecture confirms the approach of a 4-Layer Architecture for multiple model management within one software tool; and

3. The OMG's 4-Layer Architecture for multi-model management is describable as applied use of Ehrenfeucht-Fraisse Games to interpret more than one model from theorems of a base theory stored in a meta-meta/meta-model and as under Finite-Model Theory.

---

[12] We use the terms 'CML', 'Diagram' and 'Model' here interchangeably and not necessarily as a 'Model' as it is interpreted under model theory.
[13] NORMA adopts the concept of Pages (ORM Diagrams) as projections of segments of a larger ORM model, and limits the type of model that can be manipulated within NORMA to ORM Diagrams. Barker ERDs in NORMA are stored in a separate metamodel to the ORM metamodel.

References

Bollen, P. 2007, “Fact-oriented modeling in the data-, process- and event perspectives,” *OTM-WS 2007, Part I. LNCS*, Heidelberg: Springer, pp. 591-602.

Doesburg J., Balsters H. (2012) “Fact-Based Specification of a Data Modeling Kernel of the UML Superstructure”. In: Herrero P., Panetto H., Meersman R., Dillon T. (eds) On the Move to Meaningful Internet Systems: OTM 2012 Workshops. OTM 2012. Lecture Notes in Computer Science, vol 7567. Springer, Berlin, Heidelberg.

Fagin, R., Lolaitis, P.G., Popa, L., (2005), "Data Exchange: Getting to the core". ACM Transactions on Database Systems, 30(1), p.174-210.

Gödel, K. 1992, *On Formally Undecidable Propositions of Principia Mathematica And Related Systems*, New York: Dover Publications.

Halpin T. 1989, *A Logical Analysis of Information Systems: static aspects of the data-oriented perspective*, PhD Thesis, Queensland University.

Halpin, T. 2005, ‘Higher-Order Types and Information Modeling’, Advanced Topics in Database Research, vol. 4, ed. K. Siau, Idea Publishing Group, Hershey, pp. 218-237.

Halpin, T., & Morgan, T. 2008, *Information Modeling and Relational Databases*, Morgan Kaufmann.

Lemmens, I., Nijssen M., & Nijssen S. 2007, “A NIAM2007 Conceptual Analysis of the ISO and OMG MOF Four Layer Metadata Architectures,” *Lecture Notes in Computer Science*, Vilamoura, Algarve, Heidelberg: Springer, pp. 613-23.

Manzano, M. 1999, *Model Theory*, New York: Oxford University Press.

Melnik, S. 2004, *Generic model management : concepts and algorithms*, Berlin ;New York: Springer.

Object Management Group (OMG) 2017, *OMG Unified Modeling Language (OMG UML), Infrastructure, v2.5.1.*, Object Management Group (OMG).

Object Management Group (OMG) 2016, *Meta Object Facility (MOF) Core Specification, v2.5.1*, Object Management Group (OMG).

Object Management Group (OMG) 2008, *Business Process Definition MetaModel (BPDM), v1.0*, Object Management Group (OMG).

Object Management Group (OMG) 2008, *Semantics for Business Vocabulary and Business Rules (SBVR) v1.0*, Object Management Group (OMG).

Object Management Group (OMG) 2016, *Meta Object Facility (MOF) 2.5.1 Query/View/Transformation Specification, v1.3*, Object Management Group (OMG).

Object Management Group (OMG), “OMG Issue Pages.”, Available at: https://issues.omg.org/issues/lists/ , Accessed at December 2021.

White, S. 2004, "Process Modeling Notations and Workflow Patterns", *Workflow Handbook 2004*, Future Strategies Inc. Lighthouse Point, FL, USA.

Väänänen, Jouko, "Second-order and Higher-order Logic", The Stanford Encyclopedia of Philosophy (Fall 2020 Edition), Edward N. Zalta (ed.), forthcoming URL = https://plato.stanford.edu/archives/fall2020/entries/logic-higher-order/, Accessed at 20/07/2020.