An analysis and characterisation of publicly available conceptual models

C. Maria Keet[1], Pablo Rubén Fillottrani[2,3]

[1] Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za
[2] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, prf@cs.uns.edu.ar
[3] Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina

Abstract. Multiple conceptual data modelling languages exist, with newer version typically having more features to model the universe of discourse more precisely. The question arises, however, to what extent those features are actually used in extant models, and whether characteristic profiles can be discerned. We quantitatively evaluated this with a set of 105 UML Class Diagrams, ER and EER models, and ORM and ORM2 diagrams. When more features are available, they are used, but few times. Only 64% of the entities are the kind of entities that appear in all three language families. Different profiles are identified that characterise how a typical UML, (E)ER and ORM diagram looks like.

Keywords: UML Class Diagram, EER, ORM, Quantitative Analysis, Language Feature

1 Introduction

Many conceptual data modelling languages (CDMLs) exist, due to, among others, having originated from different communities (e.g., relational databases vs. OO software), addressing a specific modelling issue (e.g., spatial, temporal), and design decisions for leanness or expressiveness. There is a general trend toward more modelling features in CDMLs over time; compare, e.g., the early UML with the latest v2.4.1 [9] (a.o., with identifiers), ORM vs ORM 2 (a.o., more ring constraints, role values) [5], and ER vs EER (a.o., subsumption, disjointness) [12, 13]. There are also many opinions on what is the 'best' approach to this feature richness, its relation to model quality [8] and fidelity of representing the customer's needs. But: *which features are actually used in conceptual data models 'out there' in the field?* An answer to this question on actual usage will be useful in general for language development, affordances and usability of modelling tools, modelling methodologies, and can feed into the teaching of conceptual modelling. We could find only one *quantitative* analysis of feature usage in conceptual models [11], which used 168 ORM models developed by one modeller with a proprietary tool. The GenMyModel UML diagram repository [https://repository.genmymodel.com] does show some counts of model elements in its web interface, so it will have the statistics of its models, but aggregate data are not available. There are many works on the use of conceptual models as a whole [2], however, to the best of our knowledge, a quantitative study into actual use of CDML features and across CDML families has not occurred.

Besides the general usefulness of such insights into the actual usage of CDML features, we also seek to practically apply this in the development of inter-model assertions and model transformations [3], as, in theory, very many rules and patterns would be needed to cover all the features across CDM languages. This also has the advantage of having a unifying metamodel [7] of the three main conceptual data modelling language families (UML Class Diagrams, EER, and ORM), so that it allows a comparison of the data across models represented in the different languages. This brings us to the following hypotheses to test:

A. When more features are available in a language, they are used in the models.
B. Following the "80-20 principle", about 80% of the entities present in the models are from the set of entities that appear in all three language families.
C. Given the different (initial) purposes of UML class diagrams, (E)ER, and ORM, models in each language still have a different characteristic 'profile'.

To falsify the hypotheses, we collected 105 public available UML Class Diagrams (henceforth UML CD), ER and EER (abbreviated as (E)ER), and ORM and ORM2 (abbreviated as ORM/2) models and categorised all entities in terms of their unifying metamodel [7] for data analysis and comparisons. Although only 64% of the entities are within the exact intersection, this is almost 90% with a suitable transformation between attributes and value types. Most modelling features are indeed used, though some very sparingly. Each CDML family has a profile how a typical model in such language looks like, using characteristics such as entity type to relationship ratios and subsumption.

We first describe the materials and methods (Section 2), which is followed by the results (Section 3), discussion (Section 4), and conclusions (Section 5).

2 Materials and methods

The experimental design is summarised as follows:

1. Collect UML CDs, (E)ER, and ORM/2 diagram, 35 each, sourced from: GenMyModel, scientific articles (e.g., ER'13), textbooks, and online diagrams.
2. For each static, structural entity (i.e., not behavioural, implementation, or organisational element)
   (a) Classify it in terms of the metamodel entities of [4], for a named or unnamed element, including constraints;
   (b) Add any comments in the comment field of the spreadsheet;
   (c) Note any violations (syntax mistakes, not semantic ones);
3. Data analysis: Compute mean and median of element/model, percentage the entity is present in the model at all (presence) and as percentage of the total amount of entities (prevalence), for each family and aggregated; examine other characteristics, such as attribute:class ratio and binaries:n-aries.

The dataset of 105 models and spreadsheet (xls) with analyses are available from `http://www.meteck.org/SAAR.html`.

The unifying metamodel was introduced in [7] and formalised in [4]. It has Entity as top-type with four main subclasses: Relationship with 11 subclasses, Role, Entity type with 9 subclasses, and Constraint with 49 subclasses.

3 Results and analysis of the classification of entities

Four models had to be discarded; 101 models were classified manually and used in the analysis. Their average 'model size' (more precisely: vocabulary), calculated as (Object Types + Relationships + Subsumption + [Attributive property or Value Type]), is similar for each family: UML CD 51.1, ORM/2 46.6, and (E)ER 50.8.

3.1 Common features in the language families

There was a total of 8037 entities, of which 5191 (64.6%) appear in all three families (UML CD, (E)ER, ORM/2), 1737 in two of the three (21.6%), and 1109 (13.8%) in one. **Thus, Hypothesis B is falsified**. Only if one relaxes CDML feature overlap to also include the obvious transformation rules between UML and (E)ER's attributes and ORM/2's value types (as described in [3]), then it reaches 87.6%. These values remain similar (±2%) regardless whether 20, 25, or 35 models of each family were classified.

The common entities across the model families were one or more Object type, Relationship, Object type cardinality, Subsumption (object type), Single identification, Disjoint and Complete object types. Single identification, however, was very rare in UML (7 occurrences in two models). The metamodel's Object type cardinality does not distinguish between computationally important differences between a 1..* participation and other number restrictions (e.g., ≥ 2, 3..5). We enumerated the presence/absence, whose aggregate data is shown in Table 1. Other number constraints appear in at most 20% of the models, and while optional and mandatory constraints occur in roughly the same amount of models, 0..1 and 1 appear in notably fewer (E)ER models.

Table 1. Presence/absence of cardinality constraints in the models, aggregated.

	no. of models	* or 0..* (optional)	0..1 (functional)	1..* (mandatory)	1 (exactly 1)	other nr constraint
UML CD	34	22	21	17	25	7
ORM/2	33	20	28	14	29	6
(E)ER	34	19	16	19	11	2

3.2 Usage of features in the language families

Prevalence—as percent of total for that family—is included in Table 2 for both the top-5 and bottom-5 for each family. In the top-5 list, note that for UML and (E)ER, there are substantially more attributes than object types, whereas this is roughly the same for ORM's object types and value types. Dimensional

Table 2. Prevalence of particular entity in the models, as percent of total number of entities for that family, aggregated by model family and rounded off to one decimal. OT: Object type; VT: Value type; Rel.: Relationship; Int. Unique.: Internal uniqueness constraint; ID: Identifier.

Top-5		
UML CD	ORM/2	(E)ER
Attribute (31.2%)	OT cardinality (29.0%)	Attribute (39.5%)
OT (21.2%)	OT (14.5%)	OT cardinality (22.1%)
OT cardinality (17.5%)	2-ary Rel. (14.4%)	2-ary Rel. (11.6%)
2-ary Rel. (12.4%)	Int. unique. (13.1%)	OT (11.5%)
OT subsumption (9.6%)	VT (10.4%)	single ID (7.7%)
Bottom-5		
UML CD	ORM/2	(E)ER
3-ary Rel., Subsumption (Rel.), Disjoint OT (0.1%)	Compound cardinality, Equality (Rel.), Join equality (0.0%)	Attribute cardinality (0.0%)
SingleID (0.3%)	Disjoint Rel., Join subset, Role value constraint, Disjoint OT, Completeness, Subsumption (Rel.), 4-ary Rel. (0.1%)	4-ary Rel., Nested OT (0.1%)
Completeness (0.4%)	Role equality, 5-ary Rel. (0.2%)	Multivalued attribute, Disjoint OT (0.3%)
Attribute value constraint (0.7%)	Disjoint roles, External ID, Subsumption (role), Disjunctive mandatory (0.4%)	Completeness (0.4%)
Attribute card. (0.9%)	Dimensional VT (in ref mode) (0.5%)	3-ary Rel. (0.5%)

value types were observed (n=16 scattered in 29% of the models), of which 9 are about date and time and the others with measurements, such as Blood pressure (Pa) in `InfoModelerDiagram2` and Area (sq.m) in `campbellAbs`. Several types of entities did not appear at all. For UML, they were: Nested Object Type, Qualified relationship, Attribute value constraint, and Disjoint roles; for (E)ER, they were: Subsumption on roles or relationships, Inclusive mandatory, and Disjoint roles; for ORM/2: Join-disjointness and Value comparison constraint.

Overall, while some features are used little, the vast majority are, and **thus, hypothesis A is validated**.

Analysing the uncommon entities, one can look at the percent present in the model, and the median; see online material for details. What is immediately clear in the general case, is that many ORM constraints are hardly used, but most of them are used at least in some models. From a computational complexity viewpoint, there are several noteworthy observations, of which we touch upon disjointness and completeness constraints and ORM's ring constraints.

Disjointness and completeness constraints for class hierarchies feature prominently in ontologies, and they are useful for automated reasoning. Object types in CDMs are assumed to be disjoint, but it has to be declared explicitly for subsumption. There were only 13 disjointness and 21 completeness constraints in 11.8% of the full set of models, each family has some, and they were observed only in models for teaching conceptual data modelling. (E)ER had the most disjointness and completeness constraint (8 and 9, respectively). An additional 13 disjoint roles and 2 disjoint relationships were present in the ORM models.

Ring constraints are not computationally well-behaved, yet ORM has 6 different ones and ORM2 has 11. Of the 33 ORM/2 models, there were 23 relationship constraints in total (median: 0), and 32% of the ORM models had at least one of them. Of the 23 relationship constraints, 11 were irreflexive, 4 acyclic, 3 symmetric, 3 intransitive, 1 asymmetric, and 1 probably purely reflexive (ambiguous icon). 16 models had at least one recursive relationship; e.g., ERmodelROMULUS's hasModule between Ontology clearly could have been declared transitive and acyclic (and thus also asymmetric, antisymmetric, and irreflexive) if it had been available in EER. The reason for a low incidence of relationship constraints is unclear. Perhaps it is not perceived to be needed either semantically or in the implementation, or both, or is unknown or too hard for an 'average' modeller.

3.3 Salient aggregate characteristics for each family

We first consider a set of ratios that contribute to formulating a 'characteristic profile' of a family; they are included in Table 3 and elaborated on in the remainder of this section. While the average model sizes are fairly similar, in ratio to the overall amount of entities, UML has relatively few constraints compared to ORM. This also matches the notion that ORM has more constraint types: if you have them, they are used some time somewhere.

Table 3. A selection of ratios of entities aggregated by family and combined.

Ratio	**UML**	**ORM/2**	**(E)ER**	**combined**
model size:total entities	0.8	0.5	0.7	0.6
Attribute or Value type:Object type	1.5	0.7	3.5	1.7
binaries:n-aries	180.5	12.4	20.9	20.4
Subsumption(class):Object type	0.5	0.3	0.2	0.3
Relationship (non isa):Object type	0.8	1.1	1.1	1.0
Object type cardinality:other constraint	7.4	1.2	2.2	1.8
Single identification:other ID	–	17.3	5.4	8.4
role:relationship naming	4.3	(readings, mostly)	0.1	N/A

There are large differences among UML CDs, ORM/2 and (E)ER in their use of attributes or value types per object type. This may be due in part to the sharability of an ORM value types among object types, whereas attributes are exclusive to the class in UML and (E)ER so that multiple attributes have to be modelled for what is semantically the same attribute (e.g., Age).

There are relatively few class subsumption hierarchies in general, though UML CD's ratio is twice as high as that of (E)ER and a third higher than in ORM/2. The ratio Relationship (non isa):Object type are similar for ORM/2 an (E)ER, and they have about 35% more relationships than the UML models. Together with the Subsumption (of object types):Object type ratio, it shows UML is much more object type-oriented. This may be expected, as it comes from OO history, and is thus also recognisable from our dataset.

Among the relationships, noteworthy are UML CD's aggregation and binaries vs. n-aries ($n > 2$).The ratio of 'plain' association:aggregate is 2.6. Whether they are modelled correctly and whether they are implicitly present in ORM/2 and (E)ER through the name of the role or relationship, and whether they would be used in the latter if there were an icon for it, is an interesting avenue for further work. The ratio of binaries to n-aries (also in Table 3) differs greatly between UML CDs vs ORM and (E)ER. The data does not explain why UML models have mostly just binaries (261); it may be an avoidance strategy or lack of affordance in the tool, and it has been shown that modelling n-aries in UML is problematic due to notation, whereas ER does not have this problem [10].

The ratio of object type cardinality constraints to other constraints are as one may expect, with UML 3 to 6 times higher than (E)ER and ORM/2, respectively, as there are not many other constraints to add in UML CDs. The much lower values for ORM/2 and (E)ER is largely due to the manifold identification constraints in ORM and EER, which UML does not require.

3.4 Characteristic features of a family

Using the data and analysis, we construct the following feature-based characteristic profiles of the three families, therewith **validating Hypothesis C.**

The analysed UML diagrams are characterized by expressing more 'object-oriented' features: mostly classes and binary associations, naming of association ends, relatively high use of class subsumption, cardinalities, attribute value constraints, and aggregative composition of object types. More than 99% of all the elements in studied UML class diagrams are one of these features.

(E)ER diagrams are characterised by those features that describe relationships: extended use of binary and ternary relationships that are named, complex identification schemes (multiattribute and weak entity types), composite and multivalued attributes, and associative object types. These features together entity types and attributes, also obtains more than 99% of all the elements in the studied (E)ER diagrams.

Despite that ORM/2 is the most expressive language family, we can also obtain a characteristic profile based on relatively few features, although with a less high coverage than for UML and (E)ER (more than 98% of all the elements in the analysed ORM/2 models). In general, distinguished features of ORM diagrams are: fact type-orientedness, constraints over arbitrary n-ary fact types, subsumption constraints between object types and between roles, nested object types, disjointness between roles, internal and external uniqueness constraints, value constraints, and internal and external identification constraints.

3.5 Data analysis of potential interfering factors

Other observations regarding the current scope concern mainly syntax and tooling. Especially the (E)ER models had a mix of notational styles, but the real syntax errors from a modelling viewpoint were the identifiers; e.g., giving subtypes new identifiers (e.g., `ABM`), object types without one (e.g., `BritellER14-er`), a wrong Weak identification with underlined attribute instead of dashed (Bill in ER_no_2big), and adding identifiers to a relationship's attribute (malli1). This has been noted as early as 1993 (Batra in [2]), yet easily could be checked by a metamodel or logic-based syntax checker, which NORMA [1] already does for the first two types of mistake. More problematic to detect are oddities of the wide use of the graphical notations beyond its original scope, notably as graphical notation of a relational model (fewer constraints) and ontology visualisation (no identifiers). Further, the language version and, with that, the set of available language features, is difficult to discern, hampering differentiation between what is 'available, but not needed' and what is 'needed, but not available'. In most cases, it is unclear which CASE or drawing tool has been used to develop the model. For instance, ORMmulti...all is drawn in DOGMA Modeler [6] that permits only binary relations and no value types. The free GenMyModel (v0.32.2) allows only binary associations and 0..1, 1, 1..*, or * multiplicity, therewith pushing down their respective counts, whereas the (for payment) Edraw (v7.8) allows for n-aries and arbitrary multiplicities with easy drag-'n'-drop icons.

The size of the dataset does influence the obtained aggregates and ratios, but not much and many features remained at very similar or the same averages and median regardless the number of models classified (see dataset for analysis).

4 Discussion

The results reveal some interesting patterns, and help prioritising rules for model transformations and inter-model assertions. There is considerable overlap in which features are being used and how often, but this holds for only two-thirds on the conceptual level, negatively impacting feasibility and success of transformation rules and algorithms and validation of inter-model assertions to effectively link components of a complex software system. It may induce further investigation into various issues, such as why some features are used so little and prevalence of part-whole relations.

There are two main limitations to the data collection and results. First, arguably, 'appropriate' conceptual data models are safely guarded in IT departments in companies and our dataset may neither be the best nor representative. Realistically, there is no way to ascertain that. Notwithstanding, when computer science students and graduates search online for examples and reuse, these are the kind of models they will find and emulate, for better or worse.

Second, great care has been taken in the manual classification, and a selection of models was classified several times (data not included), but automation may enhance precision. However, conceptual models are typically made available as figures, not their XML-serialised version, and even if they were availableas XML file, the problem of multiple XSD format arises, which has yet to be resolved.

That said, comparing our data with the only other dataset, consisting of ORM models [11], there are only minor differences in averages and ratios for their set of models. For example, their 0.2 for subsumption:object type (0.3 in our data), and 3.2 for relationship:object type, which is higher (1.1 in our data) but anyway confirming the importance of relations cf. UML CDs, and also supporting the low value type:object type ratio (0.5 cf. our 0.7).

5 Conclusions

The quantitative evaluation of features in a set of 105 publicly available UML Class Diagrams, ER and EER models, and ORM and ORM2 diagrams is, to the best of our knowledge, the first of its kind, and constitutes a public dataset that can be used for further in-depth analyses. The quantitative evaluation showed that 64% of the entities were classified in those features shared by all three model families, and also that when more features are available, they are mostly used in at least one model. Although graphical notations may not always be strictly according to syntax and purpose, each family still yielded different characteristic profiles that typify how a typical diagram in that family looks like.

The outcomes inform a prioritisation of mapping rules for automated validation of inter-model assertions [3], and raised multiple questions, from UML's aggregation to the affordances and features of modelling tools.

Acknowledgements This work is based upon research supported by the National Research Foundation of South Africa (Project UID: 90041) and the Argentinian Ministry of Science and Technology.

References

1. Curland, M., Halpin, T.: Model driven development with NORMA. In: Proc. of HICSS'40. pp. 286a–286a. IEEE Computer Society (2007), Los Alamitos, Hawaii
2. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? DKE 58, 358–380 (2006)
3. Fillottrani, P., Keet, C.M.: Conceptual model interoperability: a metamodel-driven approach. In: Proc. of RuleML'14. LNCS, vol. 8620, pp. 52–66. Springer (2014), august 18-20, 2014, Prague, Czech Republic
4. Fillottrani, P., Keet, C.M.: KF metamodel formalisation. Technical report 1412.6545v1 (December 2014), arxiv.org
5. Halpin, T., Morgan, T.: Information modeling and relational databases. Morgan Kaufmann, 2nd edn. (2008)
6. Jarrar, M., Demy, J., Meersman, R.: On using conceptual data modeling for ontology engineering. Journal on Data Semantics 1(1), 185–207 (2003)
7. Keet, C.M., Fillottrani, P.R.: Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2. In: Proc. of ER'13. LNCS, vol. 8217, pp. 313–326. Springer (2013), 11-13 November, 2013, Hong Kong
8. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. DKE 55, 243–276 (2005)
9. Object Management Group: Superstructure specification. Standard 2.4.1, Object Management Group (2012), http://www.omg.org/spec/UML/2.4.1/
10. Shoval, P., Shiran, S.: Entity-relationship and object-oriented data modeling—an experimental comparison of design quality. DKE 21, 297–315 (1997)
11. Smaragdakis, Y., Csallner, C., Subramanian, R.: Scalable satisfiability checking and test data generation from modeling diagrams. ASE 16, 73–99 (2009)
12. Song, I.Y., Chen, P.P.: Entity relationship model. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, vol. 1, pp. 1003–1009. Springer (2009)
13. Thalheim, B.: Extended entity relationship model. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, vol. 1, pp. 1083–1091. Springer (2009)