ORM Introduction

Marijke Keet
27-4-2004

Overview topics

- Object Role Modeling
  - The symbols used
  - Implications (attribute free, elementary facts, ...)
  - Domain expert friendly
- Comparison with UML (class diagram), ER
  - Subtyping, qualified associations, objectification, ...
- DOGMA
  - Onto base & commitment layers
- General requirements conceptual modelling language
  - Expressibility, formal foundation, semantic stability, ...
- Larger examples and/or software demo VisioModeler
  - DTO, bacteriociams, MD/disease [dogma]
- ...

ORM: Entity-, value- and relationship/fact types

- Entity type (non-lexical objects): the oval
- Value type: the dashed oval
- Relationship type: the rectangle (here with two roles)
- Predicate names (the 'has' and 'is of')
- A "well-defined reference scheme" (from a to b)
- Role- and uniqueness constraints (blobs and arrows)

ORM: more constraints

- Mandatory role (prev. page)
- Disjunctive mandatory (incl. and excl.)
- Uniqueness (1:1, 1:n, m:n)
- Equality
- Frequency (that each dept. has two budget)
- Ring constraints (reflexive, symmetric, transitive, irreflexive, asymmetric, antisymmetric and intransitive)
ORM: more constraints (cont.)

- OR (a - 'no person wrote and reviewed a book')
- Pair exclusion (b - 'no person wrote and reviewed the same book')

ORM: subtypes, subsets

- Subtypes
- Subsets
  - Join-subsets (e.g. "if Person has a Title that is restricted to Sex, then Person is of Sex", or as in the figure: "if a Room at a Time is used for an Activity that requires a Facility then that Room provides that Facility")

ORM: domain expert friendly

- Near natural language questions to determine mandatory and uniqueness
- Population/examples check
- Verbaliser to confirm

ORM: attributes

'each Employee has a SSN or has a PPN, or both' graph/text, unary predicate 'smokes' and Boolean, Attribute multiplicity [0..1], Country/birthplace, sexcode
ORM: attributes (cont.)

- Attribute-free models...
  - are more stable
  - are easy to populate with multiple instances
  - facilitate verbalization in sentences
  - highlight connectedness through semantic domains
  - are simpler and more uniform
  - make it easier to specify constraints
  - avoid arbitrary modeling decisions
  - may be used to derive attribute views when desired

ORM/UML: qualified associations

Qualifier in UML is a set of one or more attributes, whose values can be used to partition the class thus partitioning the bank into different accounts!

ORM uses an external uniqueness constraint instead and separates bank form account and account number.

ORM/UML: objectification

In ORM only 1:1 and m:n, but not 1:n fact types. UML permits both...

Split fact instead:

ORM/UML: summary differences

<table>
<thead>
<tr>
<th>Data elements/attributes</th>
<th>ORM</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Object</td>
<td>Internal uniqueness</td>
</tr>
<tr>
<td>Value</td>
<td>Data value</td>
<td>External uniqueness</td>
</tr>
<tr>
<td>Object</td>
<td>Object or data value</td>
<td>Subtype mandatory role</td>
</tr>
<tr>
<td>Entity type</td>
<td>Class</td>
<td>Disjoint mandatory role</td>
</tr>
<tr>
<td>Value type</td>
<td>Data type</td>
<td>Frequency, internal, external</td>
</tr>
<tr>
<td>Object type</td>
<td>Class or data type</td>
<td>Value</td>
</tr>
<tr>
<td>— use relationship type</td>
<td>Attribute</td>
<td>Subtype and Equality</td>
</tr>
<tr>
<td>Unary relationship type</td>
<td>— 1:1 use domain attribute</td>
<td>Exclusion</td>
</tr>
<tr>
<td>Binary relationship type</td>
<td>Association</td>
<td>Subtype link and definition</td>
</tr>
<tr>
<td>N-ary relationship type</td>
<td>Link</td>
<td>Super constraints</td>
</tr>
<tr>
<td>N-ary object type</td>
<td>Association class</td>
<td>Sub constraints</td>
</tr>
<tr>
<td>Covariance</td>
<td>Quantitative association</td>
<td>Object community</td>
</tr>
<tr>
<td>— use unique and manual</td>
<td>— use mandatory</td>
<td>Association/composition</td>
</tr>
</tbody>
</table>

§ = incomplete coverage of corresponding concept
**ORM/ER - ‘messy’ subtypes**

A traditional representation of ORM:

- [Image of ORM/ER diagram]

**ORM/UML: more comparisons**

- ORM's elementary facts; ORM no specific notation for aggregation
- Both have formal foundation
- UML's OCL, ORM less textual thus easier to reason with and ConQuer
- UML's other diagrams
- Differences in abstraction/modularization
- ORM maps to UML, IDEFX, logical model and back
- ORM for microsofties

**ORM: formal foundation**

- No nice standard document, but can be extracted from multiple (outdated?) papers
- ORM metamodel
- Conceptual Schema, CS, and the basic Information Structure, IS (and some more...)

$$\mathcal{CS} = \langle \text{CB, VC, RC, RO, Roles}, \text{PolN, SeKOf, Players} \rangle$$

$$\mathcal{IS} = \langle \text{IS, Mand, Uniqe, } \text{PUniqe, Frequency, Subset, Equality, Exclusion} \rangle$$

**UML with OCL and ORM with ConQuer**

**UML's OCL:**

- Dot-notation, this expressions more succinctly, but reliance on functional attributes
- Textual
- Model-based constraint language

**ORM's ConQuer:**

- Classical logic with set theory
- Standard maths
- Modern user interface
- Derivation rules
- ...
With the conceptually 'most important' entity and relationship types as computed with the weighting algorithm.
ORM: Conceptual Schema Design Procedure

1. Transform familiar information examples into elementary facts, and apply quality checks.
2. Draw the fact types, and apply a population check.
3. Check for entity types that should be combined, and note any arithmetic derivations.
4. Add uniqueness constraints, and check arity of fact types.
5. Add mandatory role constraints, and check for logical derivations.
6. Add value, set comparison, and subtyping constraints.
7. Add other constraints and perform final checks.

DOGMA approach to ontologies

- Ontology base ("entity types and fact types"), composed of set of context-specific binary conceptual relations, called lexons, represented as \(<g,Term\_1, Role, Term\_2>\), \(g\) as context identifier, defining \((g,T)\) as a concept
- Ontology commitment layers ("add the rules"), with ontology view referring to the relevant lexons and additional rules, where each ontological commitment corresponds to an explicit instance of an (intensional) first order interpretation of the domain knowledge in the ontology base. In other words, it is the role of the commitments to provide the formal interpretation(s) of the lexons.
- Resulting: a conceptual schema can be seen as an ontological commitment defined in terms of the domain knowledge

DOGMA's ontology base, an example

<table>
<thead>
<tr>
<th>Context</th>
<th>Term 1</th>
<th>Role</th>
<th>Term 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Life</td>
<td>Organism</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Has</td>
<td>LatinName</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Has</td>
<td>GenNumber</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Has</td>
<td>CultureCollection</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Has</td>
<td>Price</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Price</td>
<td>Has</td>
<td>Value</td>
</tr>
<tr>
<td>Microorganism</td>
<td>LatinName</td>
<td>Has</td>
<td>LatinNameFamily</td>
</tr>
<tr>
<td>Microorganism</td>
<td>LatinName</td>
<td>Has</td>
<td>LatinNameSub</td>
</tr>
<tr>
<td>Microorganism</td>
<td>LatinName</td>
<td>Has</td>
<td>LatinNameSubPart</td>
</tr>
<tr>
<td>Microorganism</td>
<td>LatinName</td>
<td>Has</td>
<td>Designate</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>SuperTypeOf</td>
<td>Bacterium</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>SuperTypeOf</td>
<td>Fungi</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>SuperTypeOf</td>
<td>Archaea</td>
</tr>
<tr>
<td>Microorganism</td>
<td>Microorganism</td>
<td>Has</td>
<td>Morphology</td>
</tr>
<tr>
<td>Diseases</td>
<td>Disease</td>
<td>Has</td>
<td>DiseaseName</td>
</tr>
<tr>
<td>Diseases</td>
<td>Disease</td>
<td>Has</td>
<td>WHO-ID</td>
</tr>
<tr>
<td>Diseases</td>
<td>Disease</td>
<td>CausedBy</td>
<td>Cause</td>
</tr>
<tr>
<td>Diseases</td>
<td>Cause</td>
<td>Has</td>
<td>CauseAgent</td>
</tr>
<tr>
<td>Diseases</td>
<td>CauseAgent</td>
<td>SuperTypeOf</td>
<td>Infection</td>
</tr>
<tr>
<td>Diseases</td>
<td>CauseAgent</td>
<td>SuperTypeOf</td>
<td>Polarity</td>
</tr>
<tr>
<td>Diseases</td>
<td>Disease</td>
<td>Has</td>
<td>Symptoms</td>
</tr>
<tr>
<td>Diseases</td>
<td>Infection</td>
<td>Has</td>
<td>Microorganism</td>
</tr>
<tr>
<td>Diseases</td>
<td>Infection</td>
<td>Has</td>
<td>Virus</td>
</tr>
<tr>
<td>Diseases</td>
<td>Infection</td>
<td>Has</td>
<td>Worm</td>
</tr>
<tr>
<td>Diseases</td>
<td>Strain</td>
<td>Has</td>
<td>Microorganism</td>
</tr>
</tbody>
</table>

DOGMA commitment layers, example (cont.)

"Microbiology department"
DOGMA commitment (cont.)

“Culture collection”

DOGMA commitment (cont.)

“Diseases”

DOGMA screenshots software

Base & commitment in text

Base & commitment graphical

The Descriptive Terms Ontology in UML

Fig 4. Concepts and relationships in the descriptive term ontology. All terms are types of Defined Term. Structures can be parts of other structures recursively, and may have attributes. Terms are composed into groups, which may be restricted to (applies to) certain structures. Therefore these state groups may represent state properties, which may include a structural context. Terms describe a given property, which may be applicable to only certain structures.
The Descriptive Terms Ontology in ORM

The Descriptive Terms Ontology in ER

Bacteriocins

ORM resources

- Object Role Modeling: http://www.orm.net/
- UML homepage: http://www.uml.org/
- DOGMA home page: http://www.starlab.vub.ac.be/research/dogma.htm