ORM Introduction

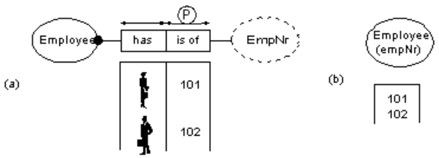
Marijke Keet

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, bacteriocins, MO/disease [dogma]**
- **…**

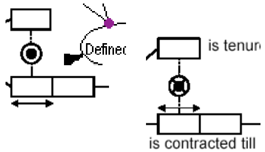
ORM: Entity-, value- and relationship/fact types

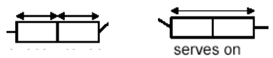


- **Entity type (non-lexical objects): the oval**
- **Value type: the dashed oval**
- **Relationship type: the rectangle (here with two roles)**
- **Predicate names (the 'has' and 'is of')**
- **A "well-defined reference scheme" (from a to b)**
- **Role- and uniqueness constraints (blobs and arrows)**

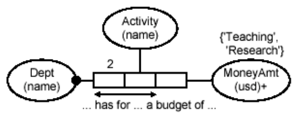
ORM: more constraints

- **Mandatory role** **(prev. page)**
- **Disjunctive mandatory** **(incl. and exl.)**
- **Uniqueness (1:1, 1:n, m:n)**

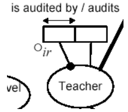




- **Equality**
- **Frequency** **(that each dept. has two budge**

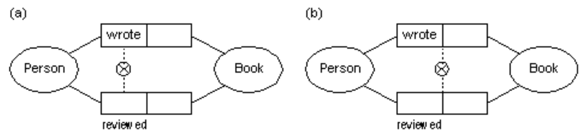


- **Ring constraints** **(reflexive, symmetric, transitive, irreflexive, asymmetric, antisymmetric and intransitive)**



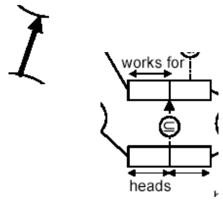
ORM: more constraints (cont.)

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

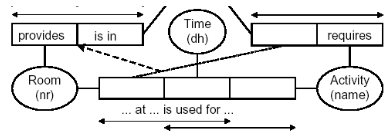


ORM: subtypes, subsets

- **Subtypes**
- **Subsets**

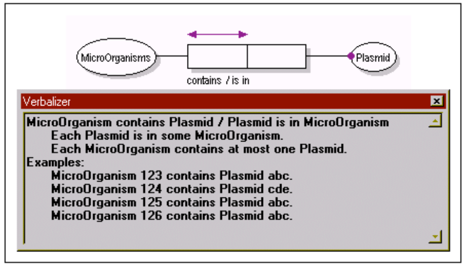


- **Join-subsets** (e.g. “**if** Person$_1$ has **a** Title **that** is restricted to Sex$_1$ **then** Person1 is of Sex$_1$”, or as in the figure: “**if a** Room at **a** Time is used for **an** Activity **that** requires **a** Facility **then that** Room provides **that** Facility”)

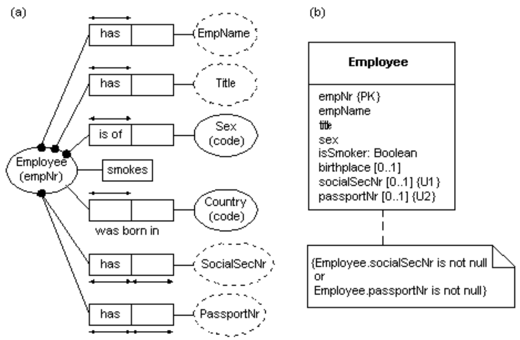


ORM: domain expert friendly

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



ORM: attributes



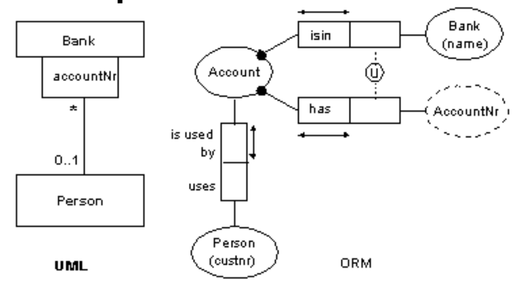
‘each Employee has a SSN or has a PPN, or both’
graph/text, unary predicate ‘smokes’ and Boolean,
Attribute multiplicity [0..1], Country/birthplace, sexcode

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**

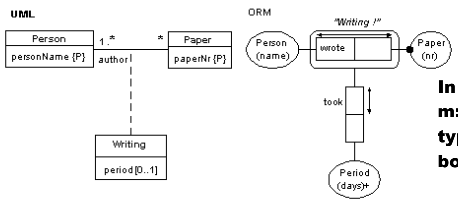
ORM/UML: qualified associations



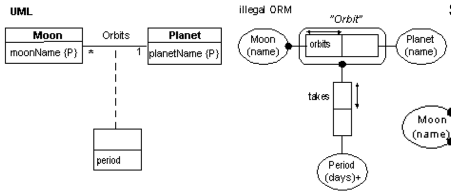
Qualifier in UML is a set of one or more attributes, whose values can be used to partition the class
thus partitioning the bank into different accounts!

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

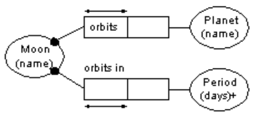
ORM/UML: objectification



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



Split fact instead:

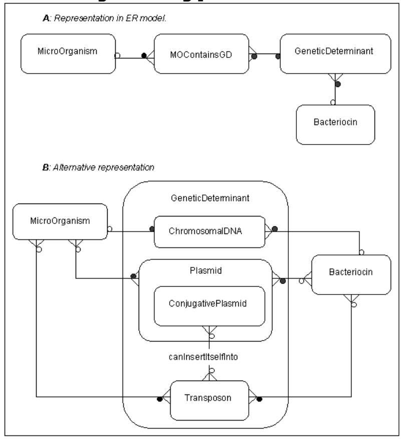


ORM/UML: summary differences

Data instances/structures		*Constraints*	
ORM	*UML*	*ORM*	*UML*
Entity	Object	Internal uniqueness	Multiplicity of ..1 §
Value	Data value	External uniqueness	— { use qualified assoc. § }
Object	Object or Data value	Simple mandatory role	Multiplicity of 1..
Entity type	Class	Disjunctive Mandatory role	—
Value type	Data type	Frequency: internal; external	Multiplicity §; —
Object type	Class or Data type	Value	Enumeration, and textual
— { use relationship type }	Attribute	Subset and Equality	Subset §
Unary relationship type	— { use Boolean attribute }	Exclusion	Or-constraint §
2+-ary relationship type	Association	Subtype link and definition	Subclass discriminator etc. §
2+-ary relationship instance	Link	Ring constraints	—
Nested object type	Association class	Join constraints	—
Co-reference	Qualified association §	Object cardinality	Class multiplicity
		— { use unique and mand. §}	Aggregation/composition
		—	Defaults/changeability
		Textual constraints	Textual constraints

§ = incomplete coverage of corresponding concept

ORM/ER - ‘messy’ subtypes



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…)

$$\mathcal{IS} = \langle \mathcal{OB}, \mathcal{VL}, \mathcal{RL}, \mathcal{RO}, \mathsf{Roles}, \mathsf{PosN}, \mathsf{SubOf}, \mathsf{Player} \rangle$$

$$\mathcal{CS} = \langle \mathcal{IS}, \mathsf{Mand}, \mathsf{Unique}, \mathsf{PUnique}, \mathsf{Frequency}, \mathsf{Subset}, \mathsf{Equality}, \mathsf{Exclusion} \rangle$$

UML with OCL and ORM with ConQuer

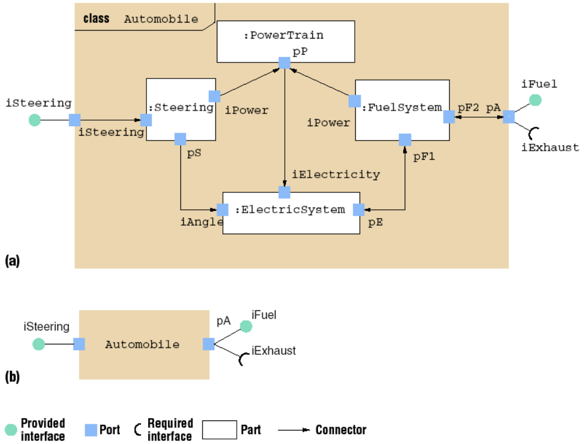
UML’s OCL:

- **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**
- **…**

UML modularization



ORM's abstractions (1)

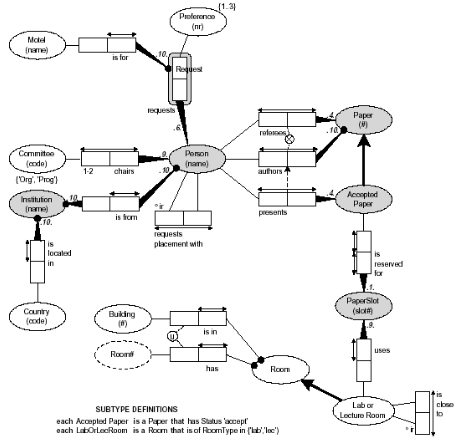


Figure 31: WeightSchema($\mathcal{CS}_2$): Abstraction Level 2

ORM's abstractions (2)

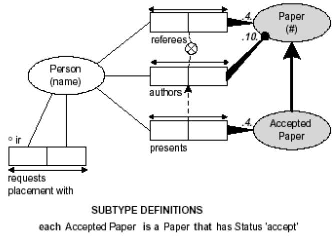


Figure 35: WeightSchema($\mathcal{CS}_4$): Abstraction Level 4

With the conceptually 'most important' entity and relationship types as computed with the weighting algorithm

Conceptual modeling language criteria

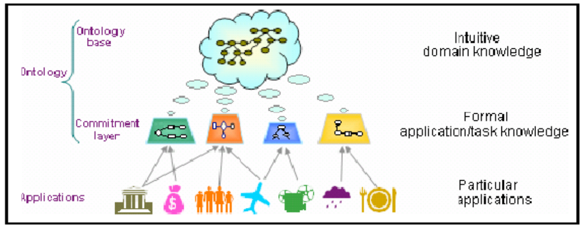
- ***Expressibility*: the "100% principle"**
- ***Clarity*: ease of understanding and use**
- ***Semantic stability*: how well models or queries expressed in the language retain their original intent in the face of changes to the application [the more changes one is forced to make in order to cope with an application change, the less stable]**
- ***Semantic relevance*: only conceptually relevant details need to be modeled, the "conceptualization principle"**
- ***Validation mechanisms*: the ways by which the domain expert can check if the model represents what s/he thinks the application does/has to do**
- ***Abstraction mechanisms*: ways by which unwanted details may be removed from immediate consideration. Modularization, show/hide toggles, zoom…**
- ***Formal foundation*: obviously…**
- **Other?**

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

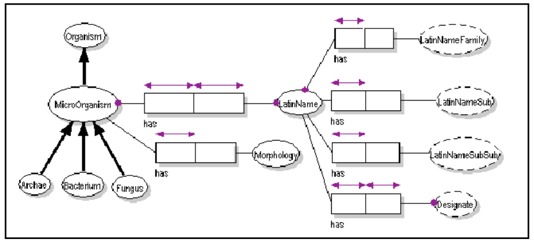
- **Ontology base ("entity types and fact types"),** composed of set of context-specific binary conceptual relations, called *lexons*, represented as <□ $Term_1$, Role, $Term_2$>, □as context identifier, defining (□T) as a concept
- **Ontology commitment layers ("add the rules"),** with ontology *view* 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

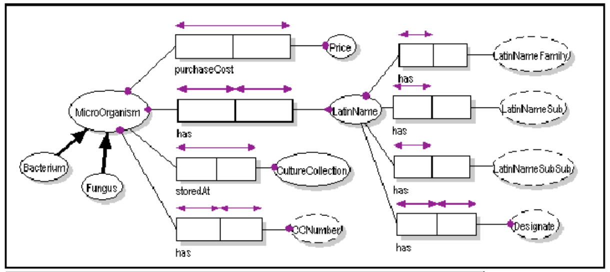
Context	$Term_1$	Role	$Term_2$
Microorganisms	Microorganism	IsAn	Organism
Microorganisms	Microorganism	Has	LatinName
Microorganisms	Microorganism	Has	CCNumber
Microorganisms	Microorganism	StoredAt	CultureCollection
Microorganisms	Microorganism	PurchaseCost	Price
Microorganisms	Price	Has	Value
Microorganisms	Price	Has	Currency
Microorganisms	LatinName	Has	LatinNameFamily
Microorganisms	LatinName	Has	LatinNameSub
Microorganisms	LatinName	Has	LatinNameSubSub
Microorganisms	LatinName	Has	Designate
Microorganisms	Microorganism	SupertypeOf	Bacterium
Microorganisms	Microorganism	SupertypeOf	Fungus
Microorganisms	Microorganism	SupertypeOf	Archae
Microorganisms	Microorganism	Has	Morphology
Diseases	Disease	Has	DiseaseName
Diseases	Disease	IdentifiedBy	WHO_ID
Diseases	Disease	CausedBy	Cause
Diseases	CausativeAgent	SupertypeOf	Infection
Diseases	CausativeAgent	SupertypeOf	Poisoning
Diseases	Disease	Has	Symptoms
Diseases	Infection	By	Microorganism
Diseases	Infection	By	Virus
Diseases	Infection	By	Worm
Diseases	Poisoning	By	Microorganism

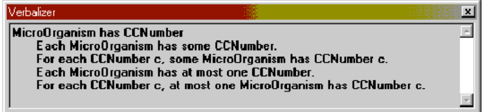
DOGMA commitment layers, example (cont.)



"Microbiology department"

DOGMA commitment (cont.)

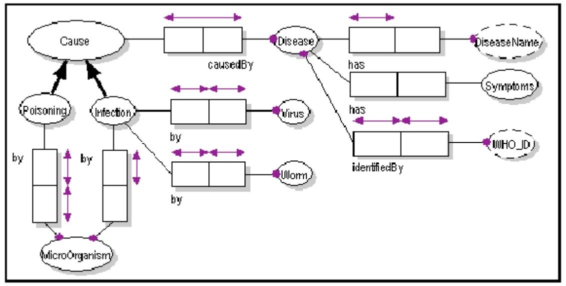




"Culture collection"

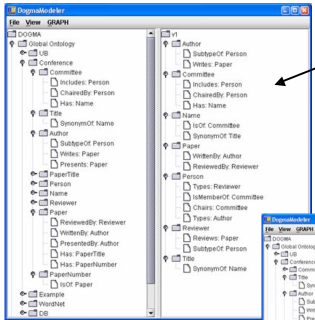
Figure 2.14 (Near) natural language of the ontological commitment rules between MicroOrganism *and* CCNumber.

DOGMA commitment (cont.)



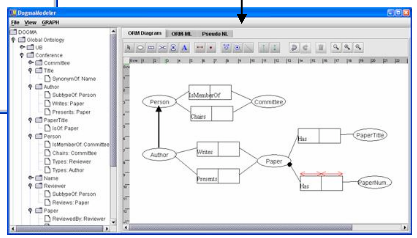
"Diseases"

DOGMA screenshots software



Base & commitment in text

Base & commitment graphical



The Descriptive Terms Ontology in UML

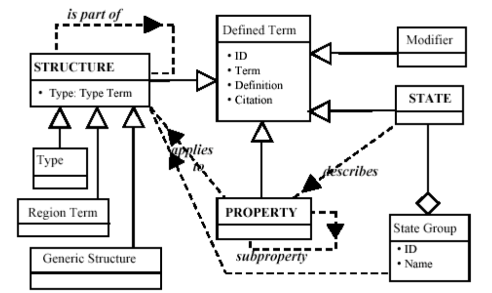
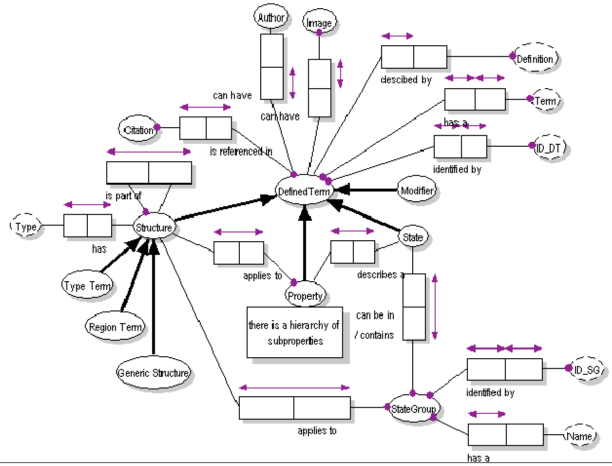
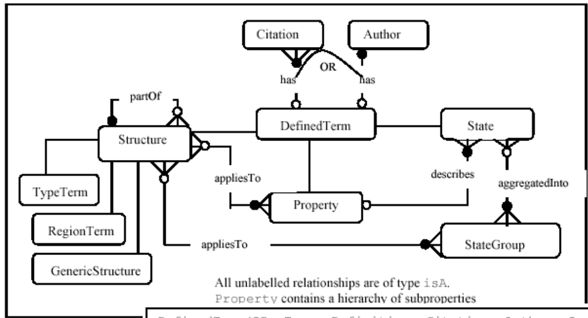


Fig 4. Concepts and relationships in the descriptive term ontology. All terms are types of Defined Term. Structures can be 'part-of' other structures recursively, and may have attribute: Type. States are composed into groups, which may be restricted to ('applies-to') certain structures. Therefore these state groups may represent 'de facto' properties, which may include a structural context. States describe a given property, which may be applicable to only certain structures.

The Descriptive Terms Ontology in ORM

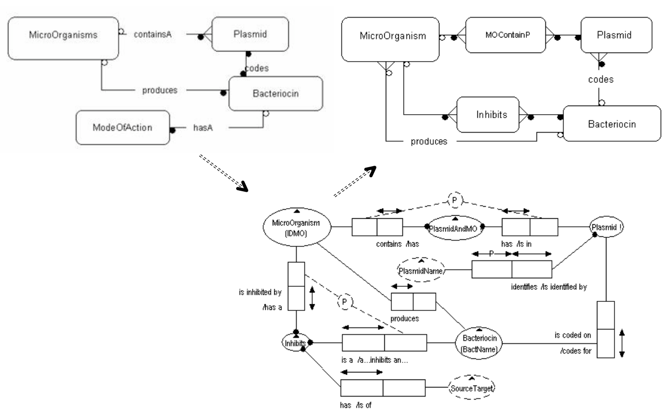


The Descriptive Terms Ontology in ER




```
DefinedTerm(ID, Term, Definition, Citation, Author, Image)
Modifier(ID, Term, Definition, Citation, Author, Image)
TypeTerm(ID, Term, Definition, Citation, Author, Image)
RegionTerm(ID, Term, Definition, Citation, Author, Image)
GenericStructure(ID, Term, Definition, Citation, Author, Image)
Property(ID, Term, Definition, Citation, Author, Image)
Structure(ID, Term, Definition, Citation, Author, Image, Type)
State(ID, Term, Definition, Citation, Author, Image)
StateGroup(ID, Name)
```

Bacteriocins



ORM resources

- **Journal of Conceptual Modeling: http://www.inconcept.com/jcm**
- **Object Role Modeling: http://www.orm.net/**
- **Halpin, T. (2001). *Information modeling and relational databases*. San Francisco: Morgan Kaufmann Publishers.**
- **UML homepage: http://www.uml.org/**
- **DOGMA home page: http://www.starlab.vub.ac.be/research/dogma.htm**