When and How to Use Multi-Level Modelling

Authors: J. De Lara, E. Guerra & J. Sánchez Cuadrado
Presenter: Nick Fung
January 29, 2018
Running Example

- Task
  - Create a model-based computer system for managing the products sold in a bookstore
Running Example

- Task
  - Create a model-based computer system for managing the products sold in a bookstore
Standard Metamodelling Architecture

- M0 (Terminal) Model
  - Representation of a system
Standard Metamodelling Architecture

- **M0 (Terminal) Model**
  - Representation of a system

- **M1 Metamodel**
  - Abstract syntax of models
  - Prescription of what can be represented
Standard Metamodelling Architecture

- **M0 (Terminal) Model**
  - Representation of a system

- **M1 Metamodel**
  - Abstract syntax of models
  - Prescription of what can be represented

- **M2 Metametamodel**
  - Metamodelling facilities
  - Auto-descriptive
Standard Metamodelling Architecture

M2

UML Metamodel

M1

Product 1 madeBy Manufacturer

Book

Food

Pub.

Fact.

M0


Limitations of Architecture (1)

- Type-Objects (Clabjects)
  - New types (i.e. classes) cannot be instantiated dynamically
Limitations of Architecture (1)

- Type-Objects (Clabjects)
  - New types (i.e. classes) cannot be instantiated dynamically

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Objects</th>
<th>New types (i.e. classes) cannot be instantiated dynamically</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>Book</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Food</td>
<td></td>
</tr>
<tr>
<td>M0</td>
<td>AI:Book</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pearson:Pub.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manufacturer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pub.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fact.</td>
<td></td>
</tr>
</tbody>
</table>
```
Limitations of Architecture (1)

- Type-Objects (Clabjects)
  - New types (i.e. classes) cannot be instantiated dynamically
Dynamic Features
- New features cannot be added to a type
Limitations of Architecture (2)

- Dynamic Features
  - New features cannot be added to a type
Limitations of Architecture (3)

- Auxiliary Domain Concepts
  - New entities relevant to an existing type cannot be added

![Diagram showing relationships between Product, Manufacturer, Book, Food, Pub., Fact., and AI:Book, Pearson:Pub.](image)
Limitations of Architecture (3)

- Auxiliary Domain Concepts
  - New entities relevant to an existing type cannot be added
Limitations of Architecture (4)

- Relation Configuration
  - Reference types (i.e. relations) cannot be reconfigured
Relation Configuration
- Reference types (i.e. relations) cannot be reconfigured
Element Classification

- New classifications for (new) classes cannot be created
Limitations of Architecture (5)

- Element Classification
  - New classifications for (new) classes cannot be created
Explicit Modelling

- Model dynamic types, relations, etc. at the M0 level
- Flexible, but yields complicated models
Two-Level Solutions (2)

- Promotion
  - Model types at M0 and transform into a meta-model
  - Flexible, but may require complex transformation
Two-Level Solutions (3)

- Powertypes
  - Types whose instances are subtypes of another type
  - Limited to modelling features in the next two levels
Stereotypes
- Extensions of the metamodel
- Limited to modelling domain concepts at two levels
Multi-Level Modelling

- **Potency**
  - Model elements are clabjects
  - All elements are instantiable (for given number of times)
Multi-Level Modelling

- **Potency**
  - Model elements are clabjects
  - All elements are instantiable (for given number of times)

- **Orthogonal Classification Architecture**
  - Metamodelling facilities available at all meta-levels
  - Two kinds of types: ontological & linguistic
Multi-Level Modelling

- Type-Object

```
@2
Product
vat@1: double
price: double

@1
Book: Product
vat=4.0

@0
GoF: Book
price = 35
```
Multi-Level Modelling

- **Type-Object**
  - @2
    - Product
      - vat@1: double
      - price: double
  - @1
    - Book: Product
      - vat=4.0
  - @0
    - GoF: Book
      - price = 35

- **Relation Configurator**
  - @2
    - Product
      - madeBy
    - Manufacturer
  - @1
    - Book: Product
      - pub: madeBy
    - Editorial
      : Manufacturer
  - @0
    - GoF: Book
      :pub
    - AdWesley
      : Editorial
Field Study

- Results
  - Over 400 metamodels surveyed
  - 84 contain at least one type-object
  - 459 occurrences of patterns in total
Field Study

- **Results**
  - Over 400 metamodels surveyed
  - 84 contain at least one type-object
  - 459 occurrences of patterns in total

- **Discussion Points**
  - Which approach is more “natural”?
  - Is multi-level modelling a workaround?
Electronic Health Record (EHR)

- Requirements
  - Capture holistic view of patient
  - Applicable/extendible to all clinical domains
  - Applicable for all clinical environments
openEHR Architecture

ontologies of everything

ontologies of information

domain content models (variable)
openEHR archetypes & templates

information models (stable)
openEHR Reference Model
openEHR Service Model

IM vocab
openEHR terminology

mediated by

ontologies of reality

classifications
ICDx
LOINC
ICPC

process descriptions
guidelines

descriptive terminologies
SNOMED-CT

languages of representation

programming languages
UML

XML schema languages
openEHR Archetype Model

OWL

limit of application software and DB schemas

generic software components
openEHR Architecture

ontologies of everything

ontologies of information

- domain content models (variable)
  - openEHR archetypes & templates

information models (stable)

- openEHR Reference Model
- openEHR Service Model

IM vocab

openEHR terminology

ontologies of reality

- classifications
  - ICDx
  - LOINC
  - ICPC

- process descriptions
  - guidelines

- descriptive terminologies
  - SNOMED-CT

languages of representation

- openEHR Archetype Model
- OWL

- XML schema languages
- UML
- programming languages

Metamodelling Facilities components

limit of application software and DB schemas
openEHR Architecture

ontologies of everything

ontologies of information

domain content models (variable)
openEHR archetypes & templates

information models (stable)
openEHR Reference Model
openEHR Service Model

IM vocab
openEHR terminology

ontologies of reality

classifications
ICDx
LOINC
ICPC

descriptive terminologies
SNOMED-CT

process descriptions
guidelines

languages of representation

programming languages
UML
XML schema languages

Metamodelling Facilities components

Metamodelling

M1
limit of application
M2
and DB schemas
openEHR Architecture

**Domain Knowledge Environment**
- Archetype Library
- Terminology/Ontology
- Domain Specialists
- Template Library
- Drives GUI
  - Used for: Data Capture, Validation, Querying

**Runtime System**
- User
- App
- System
- Information
- Communication
- Data Store

**Technical Development Environment**
- Develop Once
- Implemented in Reference Model
- Archetype Model/Language
- Schema
- Defines
Conclusions

- Multi-Level Modelling
  - Potency
  - Orthogonal Classification Architecture
- Design Patterns
  - Type-Object Pattern
  - Dynamic Features
  - Dynamic Auxiliary Domain Concepts
  - Relation Configurator Pattern
  - Element Classification
- Discussion
  - When to use multi-level modelling?