Toward a Unified Conception of Multi-Level Modelling

Ulrich Frank

Information Systems and Enterprise Modelling

ICB Institute for Computer Science and Business Information Systems
- prepare for the specification of a new version of the FMML\textsuperscript{x}
- .. and a new version of the XModeler\textsuperscript{ML}
- contribute to a discourse on requirements to foster a consolidation of the field
FMML*: Notation Overview

---

^MetaClass^  
PeripheralDevice

- name: String
- salesPrice: Float
- serialNo: String
- partSalesPrice: Money

- totalRevenues() : Money
- models() : Integer
- totalUnitsInStock() : Integer
- revenues() : Money

---

^PeripheralDevice^  
Printer

- name: String
- pagePerMinute: Integer
- resolution: Integer
- serialNo: String
- partSalesPrice: Money

- salesPrice = false
- race = true
- totalRevenues() = €7399.00
- models() = 13

---

^OrganizationalUnit^  
Position

- skillLevel: Score
- minAvailability: Score
- costPerMin: Float
- posID: String

- averageAvailability(): Duration

---

^Printer^  
CPL-844

- serialNo: String
- partSalesPrice: Money

- pagePerMinute = 40
- resolution = 600
- salesPrice = 199.00

---

M1

M2

M3

---

^MetaClass^ intrinsic operation, instantiated in classes on M1

^PeripheralDevice^ intrinsic attribute, instantiated in objects on M0

^OrganizationalUnit^ intrinsic association, instantiated between objects on M0

---

object returned by operation

object state

---

© Ulrich Frank | Requirements
RC1: Contingent Level Classes
In addition to having classes with a definite level, it should be possible to define classes with a contingent level. A contingent level allows for adapting the concrete level of a class to different contexts of use.

Priority: high

De Lara et al. (2014) “leap potency”, “phantom instance”
It should be possible to define intrinsic properties (attributes, operations, associations) with a contingent instantiation level.

Priority: medium
RC3: Deferred Specification of Multiplicities of Intrinsic Associations

It should be possible to defer the specification of multiplicities of intrinsic associations.

Priority: high

It should be possible to define further constraints on multiplicities of intrinsic associations -> principle of *maximum expressiveness*. 
RC4: “Classless” Classes

It should be possible to define preliminary classes without a meta-class.

Priority: high

Atkinson, Kühne “exploratory” vs. “constructive” modelling
RC5: Intrinsic Classification of Intrinsic Attributes

^Component^
BicycleFork

1 color: Color
1 material: Material
1 weight: Float
2 mudMount: Boolean
0 stockNo: String

^Product^
Material

synthetic: Boolean
1 specificWeight: Float
0 rigidity: Float

^Material^
Metal

corrodes: Boolean
alloy: String
1 specificWeight: Float
0 rigidity: Float

synthetic = false

^Metal^
Aluminum

availability: Level
0 specificWeight: Float

alloy = "2011"
corrodes = false

^Aluminum^
Alum2011: Aluminum

availability = #high
specificWeight = 2.78
It should be possible to specify intrinsic attributes through classes on levels higher than M1 together with an intended instantiation level.

Priority: medium
RC6: Meta-Association

<table>
<thead>
<tr>
<th>MetaClass</th>
<th>Association</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Association</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Association</th>
<th>Documentation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Association</th>
<th>Aggregation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Association</th>
<th>Delegation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Married_to</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Holds</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Employs</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Documentation</th>
<th>Archives</th>
</tr>
</thead>
</table>

© Ulrich Frank | Requirements
It should be possible to define domain-specific association types. In order to cover the variance of domain-specific association types, it could be helpful, if a language provides meta-association types on a higher level, that would allow the instantiation of meta-association types.
It should be possible to indicate that certain properties of a class are to be inherited to its instances.
RC8: Semantic Enrichment of Properties

^Product^  
PeripheralDevice

- minQualityLevel: Level
- qualityLevel: Level
- serialNum: String

constraint on values of instances of instances

^PeripheralDevice^  
Printer

- pagesPerMinute: Boolean
- color: Boolean
- resolution: Integer
- serialNum: String

serves representing values that are shared by all instances of instances

minQualityLevel = #med

^Printer^  
XP-600C

- serialNum: String

serves representing particular instance property values

color = #true
pagesPerMinute = 90
resolution = 600
qualityLevel = #high

serves representing particular instance property values
RC8: Semantic Enrichment of Properties

It should be possible to clearly distinguish between attributes that are regularly instantiated into individual values of instances, that are instantiated into values of instances that actually represent common values of their instances, and those that serve the specification of constraints on attribute values. At the same time, corresponding access operations should be qualified accordingly.

Priority: **high**
Conclusions

- Multi-Level Modelling – very promising, however
  - need for common grounds
  - .. and for competition

- focus on requirements as an option to foster collaboration and coherence – without giving up on competition

- essential part of such a joint effort: development of a unified terminology

- diversity of notations not that problematic at present time, but evaluations and further development required