University of Duisburg-Essen

In the heart of Europe

https://www.uni-due.de/en/

Essen

Essen

Essen

Essen

Essen

Research Group „Enterprise Modelling“

- part of Institute for Computer Science and Business Information Systems
- 15 research groups in CS and Business Informatics
- one of the largest Business Informatics Groups in Europe
- focus on multi-perspective enterprise modelling
  - language engineering
  - method engineering
  - software engineering
- analysis and assessment of „machine learning“

https://www.umo.wiwi.uni-due.de/
https://www.umo.wiwi.uni-due.de/en/

Personal Background

- Diploma in Business&Administration, Computer Science as a Minor
- Dissertation on Limitations of Artificial Intelligence
- Philosophy of Science as a Subject of Doctoral Programme
- Researcher with the German Research Center for Mathematics and Computer Science
- Senior Scientist at IBM Research in San Jose, Ca.
- Habilitation on Enterprise Modelling

Some Lessons I Learned

- to better understand a system, try to change it .. or study what happens when it is changing
- to get deeper insights, look a principle conflicts related to essential system properties
- study different subjects, from different perspectives and
  - search for commonalities – especially those that are not obvious
  - reflect upon essential differences
- focus on concepts
- recognition implies passion

"Take your time."
Wittgenstein

IS:link – Student Exchange Network

A New Paradigm for Conceptual Modelling and Software Development

Multi-Level Language Engineering

Ulrich Frank

Lehrstuhl für Wirtschaftsinformatik und Unternehmensmodellierung

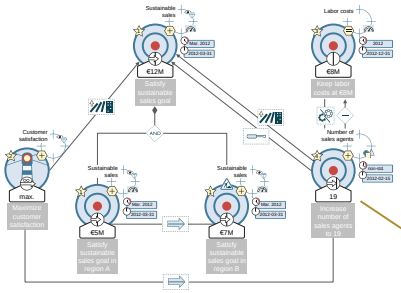
ICB Institut für Informatik und Wirtschaftsinformatik

Institut für Informatik und Wirtschaftsinformatik (ICB)

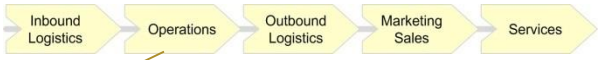
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

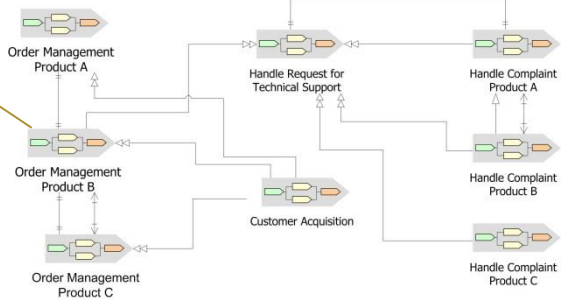
Background: Multi-Perspective Enterprise Modelling



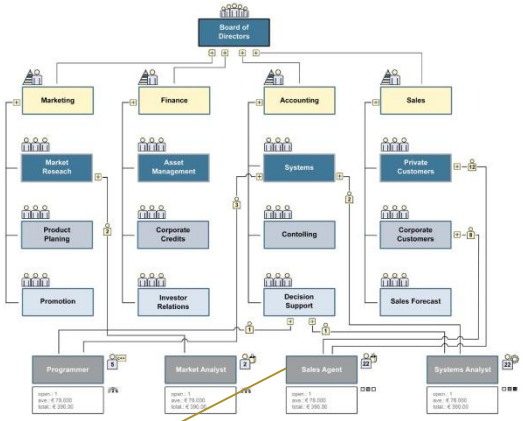
Goal Diagram



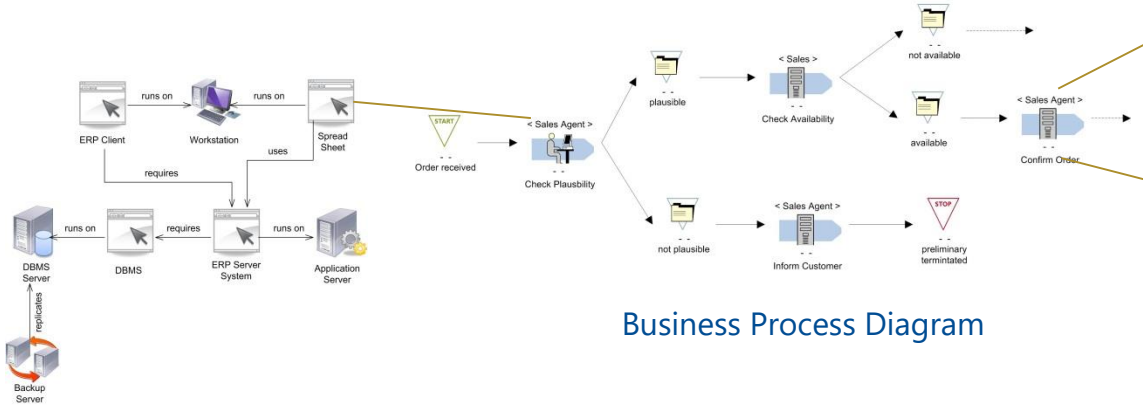
Value Chain Diagram



Business Process Map

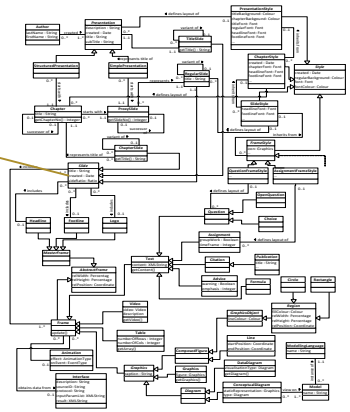


Organisational Chart



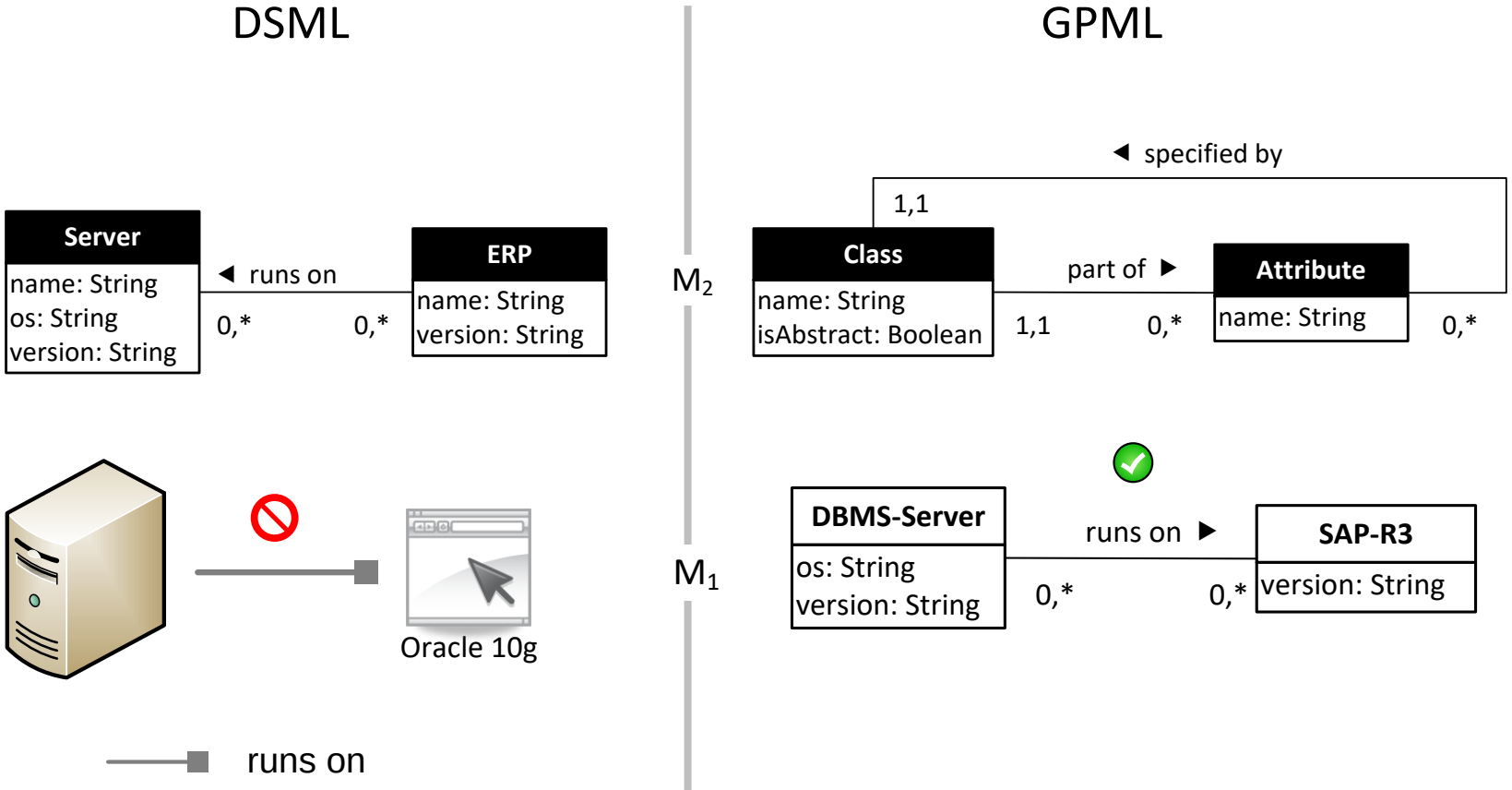
Business Process Diagram

IT Resource Diagram



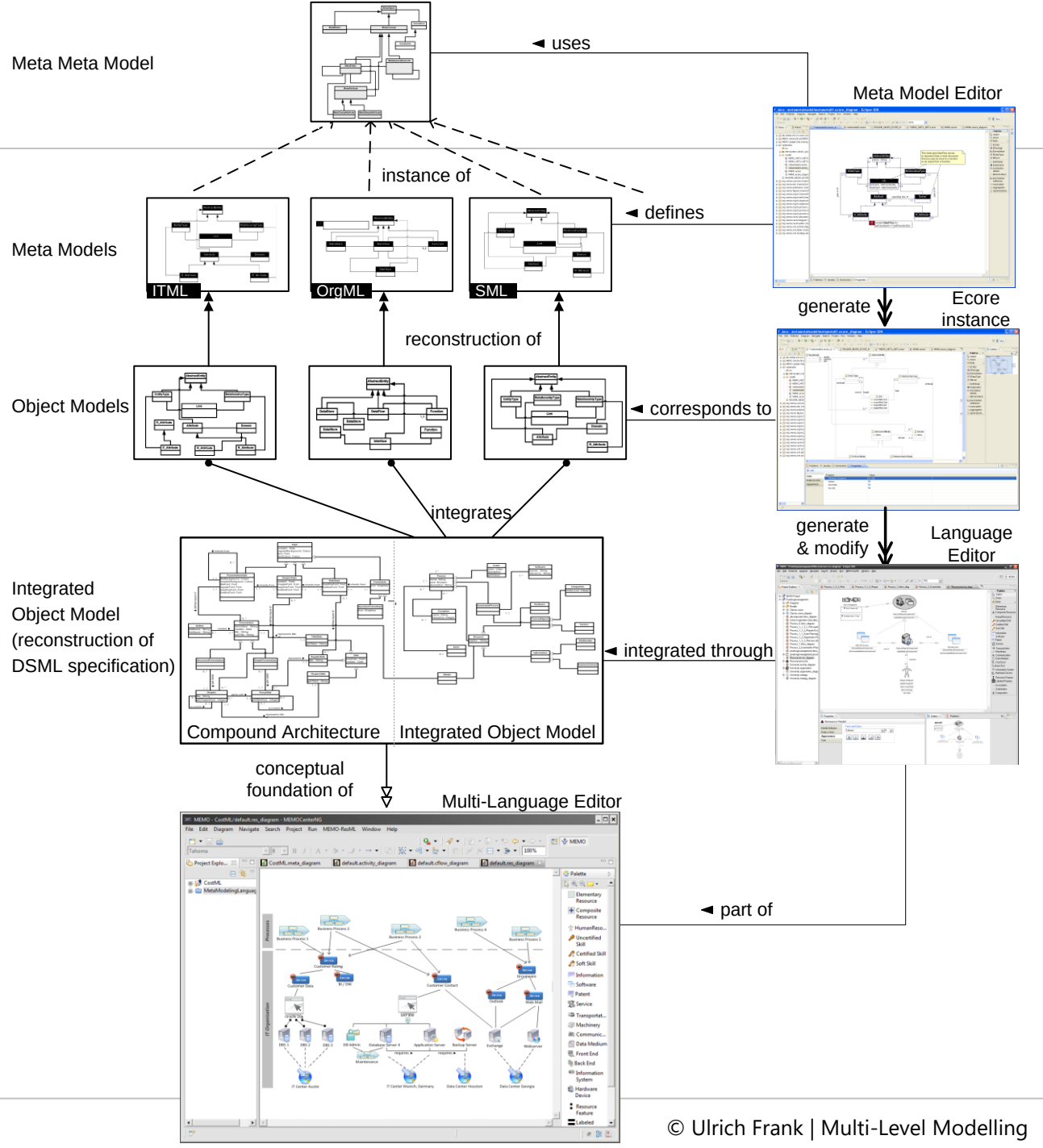
Object Model

DSML: Illustration of Benefits



Design of DSML challenging!

Language Architecture & Tools

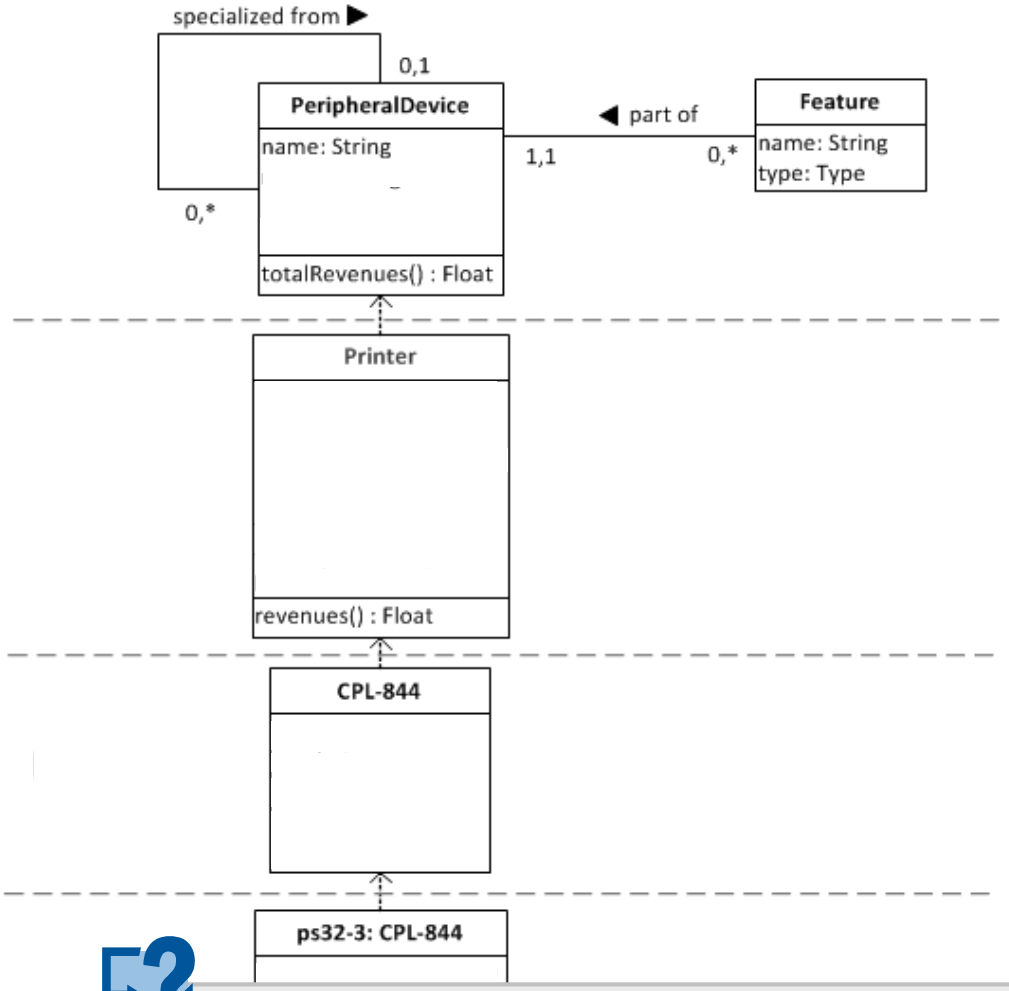


Problems, Limitations, Frustration

- current paradigm restricted to one or two levels of classification – sometimes not enough
- distinction between Instantiation and Specialisation
- distinction between language and language application
- conflict between range of reuse and productivity gain through reuse
- conflict between flexibility and integrity
- synchronisation of models and code as threat to integrity

Serious lack of abstraction!

Instantiation or Specialization?



Sometime no clear distinction between instantiation or specialisation – both make sense.

Instantiation sometimes deferred to lower levels.

(Meta) classes may have relevant state.

(Meta) classes may have features that need to be derived or calculated.

(Meta) classes should be conceptualized as objects.

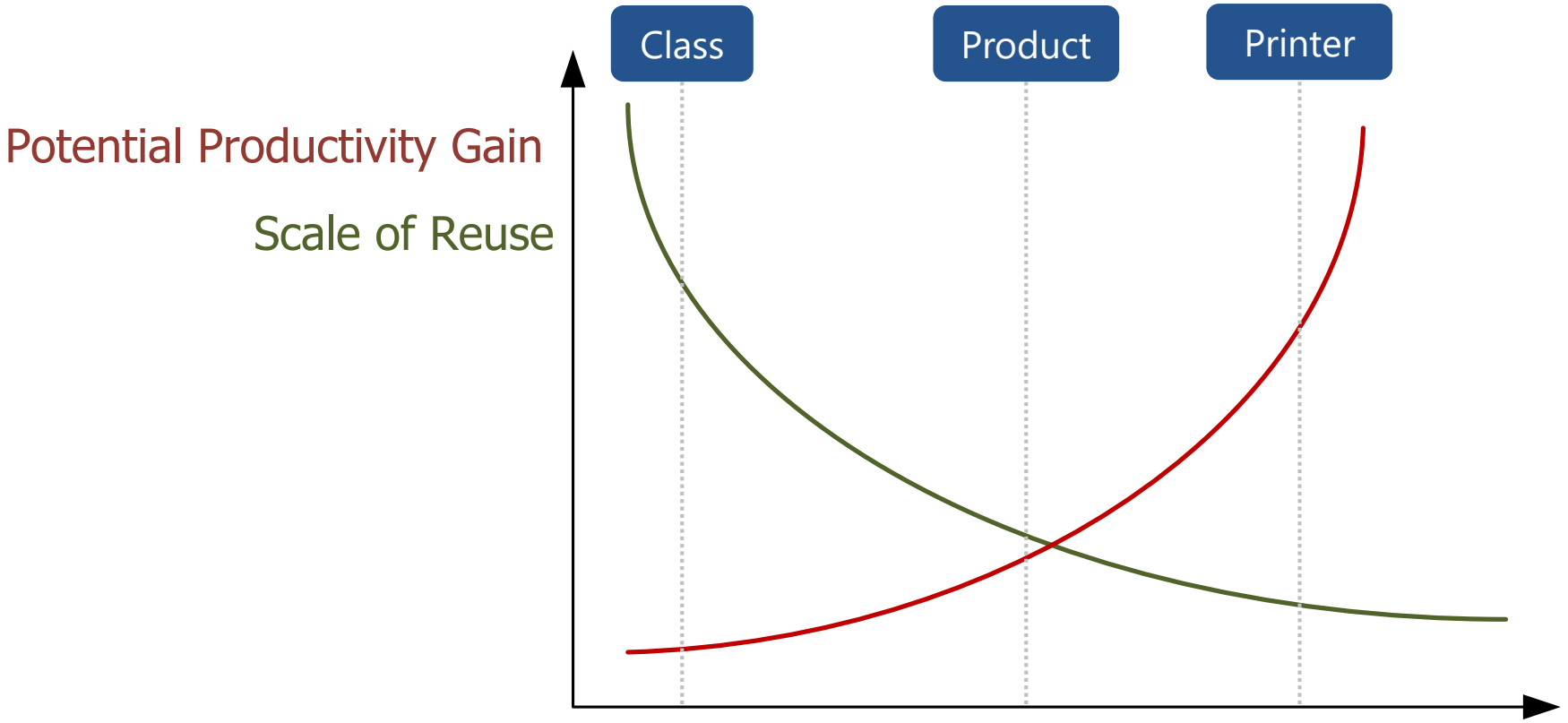
Why is this constellation challenging the current paradigm?

Distinction of Language and Model

Language concept or language application?

Paradigm requires distinction, but how?

Conflict between Range of Reuse and Productivity



Semantics promotes reuse (productivity).
Semantics compromises reuse (scale).

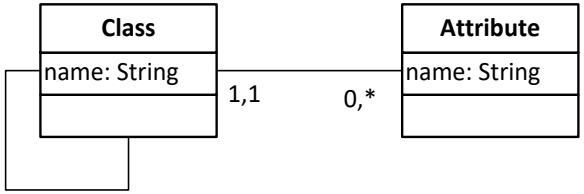
Corresponds to conflict between integrity and flexibility!

Obstacles of Model-Driven Software Development

Modelling Environment

Progamming Environment

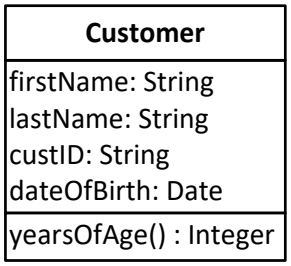
Meta-Model



M0

Model

M0



generate

Code

```
class Customer
{
    String firstName;
    String lastName;
    Date dateOfBirth;

    public int yearsOfAge()
           ....
}
```

M1

Program instance

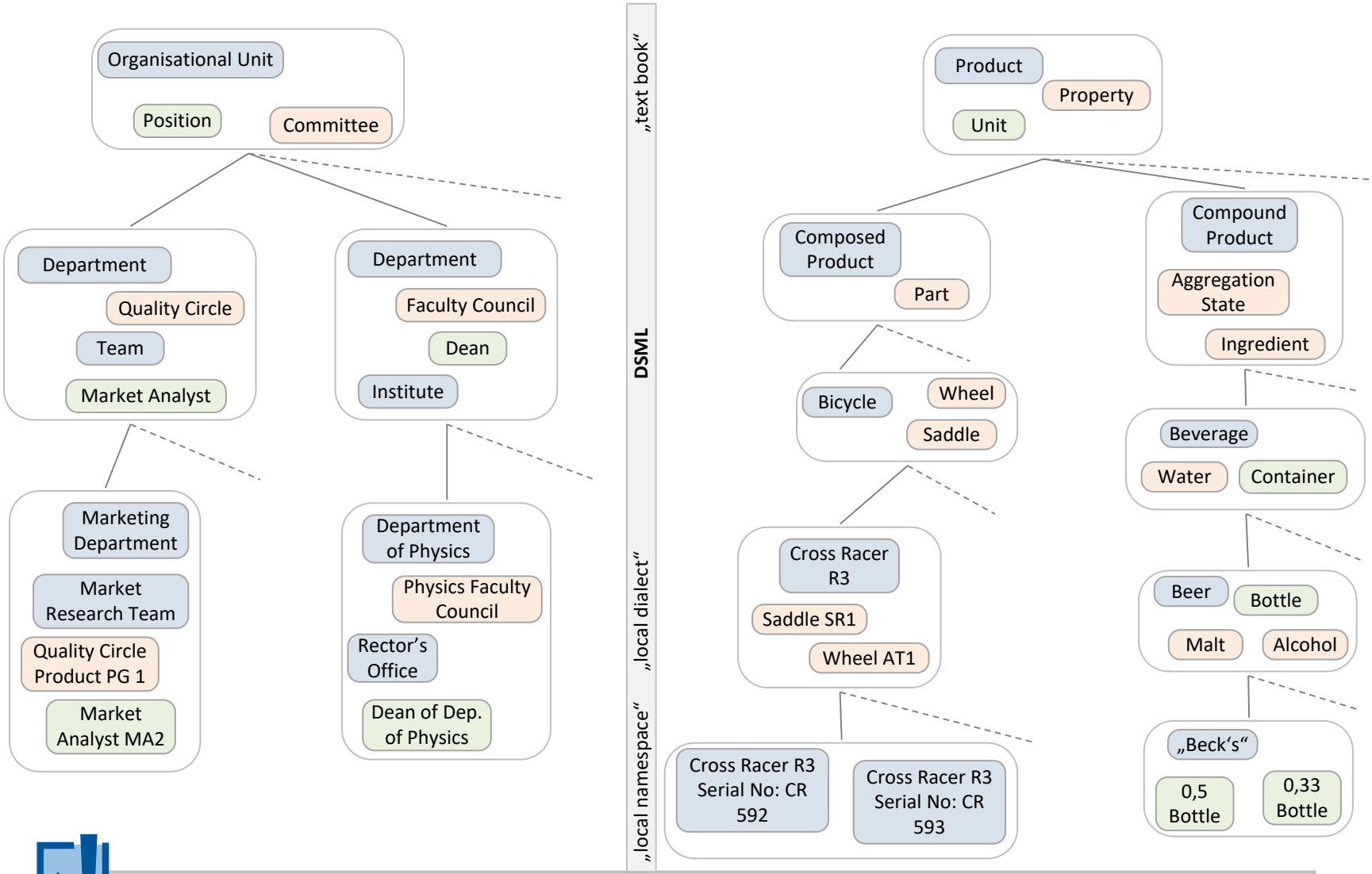
M1

Creates serious synchronisation problem

Multi-Level Modelling to the Rescue

- allows for an arbitrary number of classification levels
- motivated by the lack of abstraction in traditional, MOF-like language architectures
  - creates avoidable complexity
  - contributes to lack of flexibility
- first introduced in 2001 by Atkinson and Kühne
- various approaches developed since then
- focus on modelling, not on programming languages

Inspired by Actual Use of Technical Languages



Language hierarchies with variable number of levels

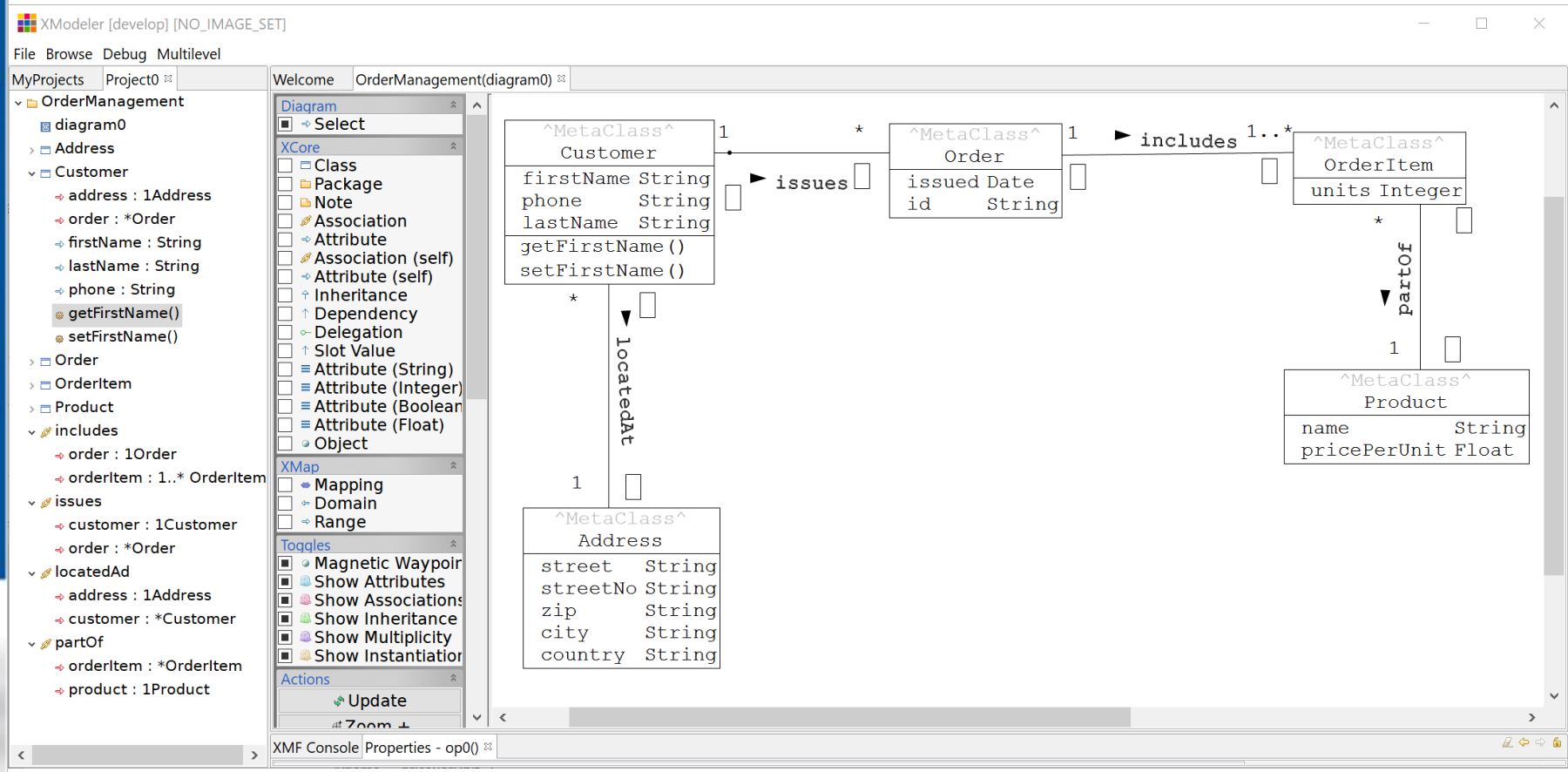
The LE4MM Project

- "Language Engineering for Multi-Level Modelling"
- conjoint project with Prof Tony Clark, Ashton University, UK
- aims at extending the **Xmodeler** to become a more powerful multi-level language engineering environment
  - allow for explicit classification level
  - support for delayed instantiation
  - support for change management
  - ..
- … and the development of prototypical application systems to demonstrate the power of multi-level software architectures

Facets of the Approach

- multilevel language architecture
  - arbitrary number of classifications
  - intrinsic features (deferred instantiation)
- **common representation** of models and code
- no strict distinction between language and language application
  - modeling and language engineering intertwined activities
  - integration of modelling and meta-modelling tool

No Need for Code-Generation and Synchronisation



XModeler [develop] [NO_IMAGE_SET]

File Browse Debug Multilevel

MyProjects

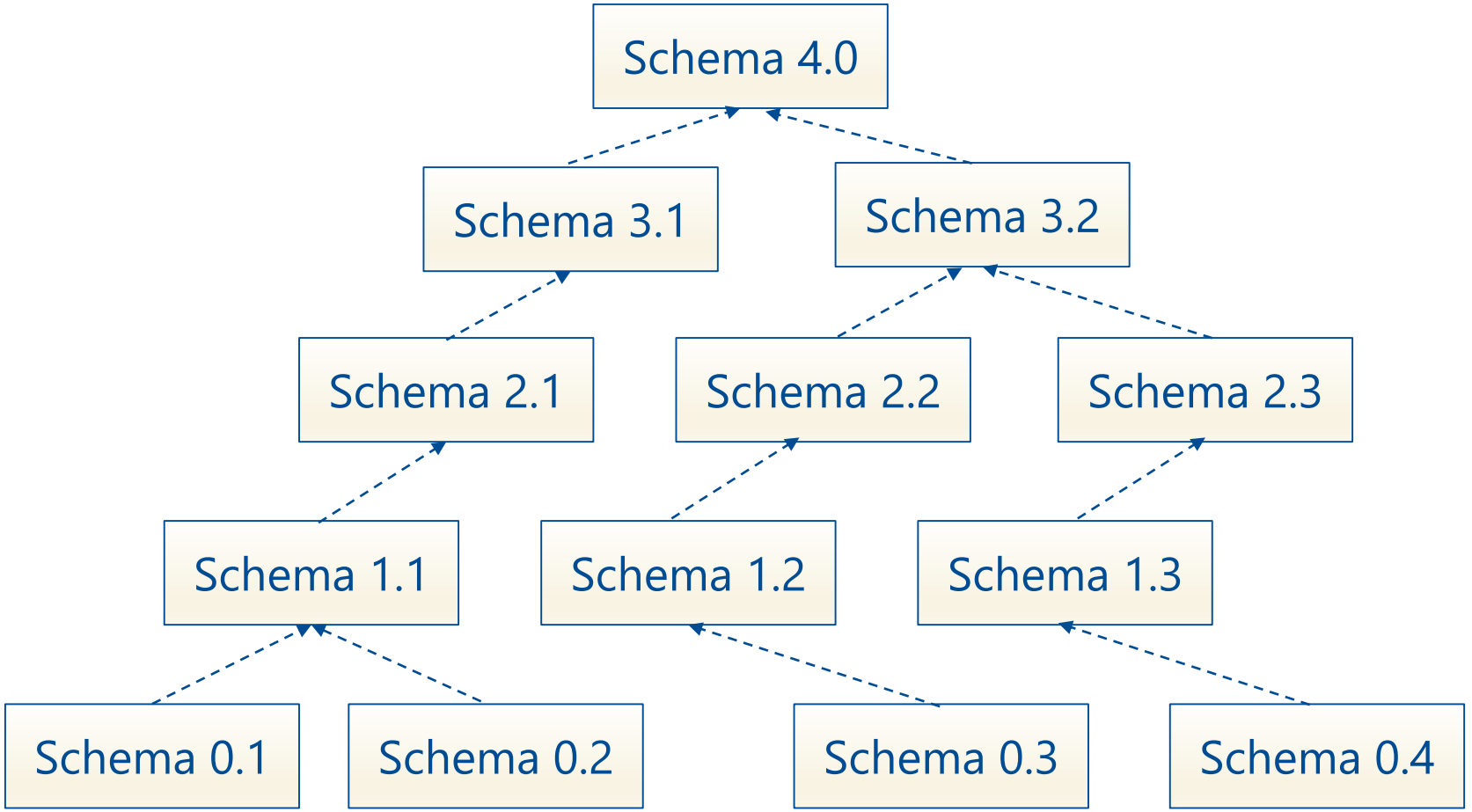
Welcome C:\Users\Bjoern\git\XMF\com.ceteva.xmf.system\xmf-addons\Test\OrderManagement.x...

```
1  parserImport XOCL;
2  parserImport Parser::BNF;
3
4  import OCL;
5  import XOCL;
6
7  context Root
8
9  @Package OrderManagement metapackage Root::XCore extends Root::XCore
10
11   @Class Product
12       @Attribute pricePerUnit : Float end
13       @Attribute name : String end
14   end
15
16   @Class OrderItem
17       @Attribute units : Integer end
18       @Attribute product : Set(Product) end
19   end
20
21   @Class Order
22       @Attribute orderItem : Set(OrderItem) end
23       @Attribute issued : AuxiliaryClasses::Date end
24       @Attribute id : String end
25   end
26
27   @Class Address
28       @Attribute street : String end
29       @Attribute streerNo : String end
30       @Attribute zip : String end
31       @Attribute city : String end
32       @Attribute country : String end
33   end
34
35   @Class Customer
36       @Attribute order : Set(Order) end
37       @Attribute firstName : String end
38       @Attribute phone : String end
39       @Attribute lastName : String end
40       @Attribute address : Set(Address) end
41
42       @Operation getFirstname():String
43           self.firstName
44       end
45
46       @Operation setFirstname(firstName: String)
47           self.firstName := firstName
48       end
49
```

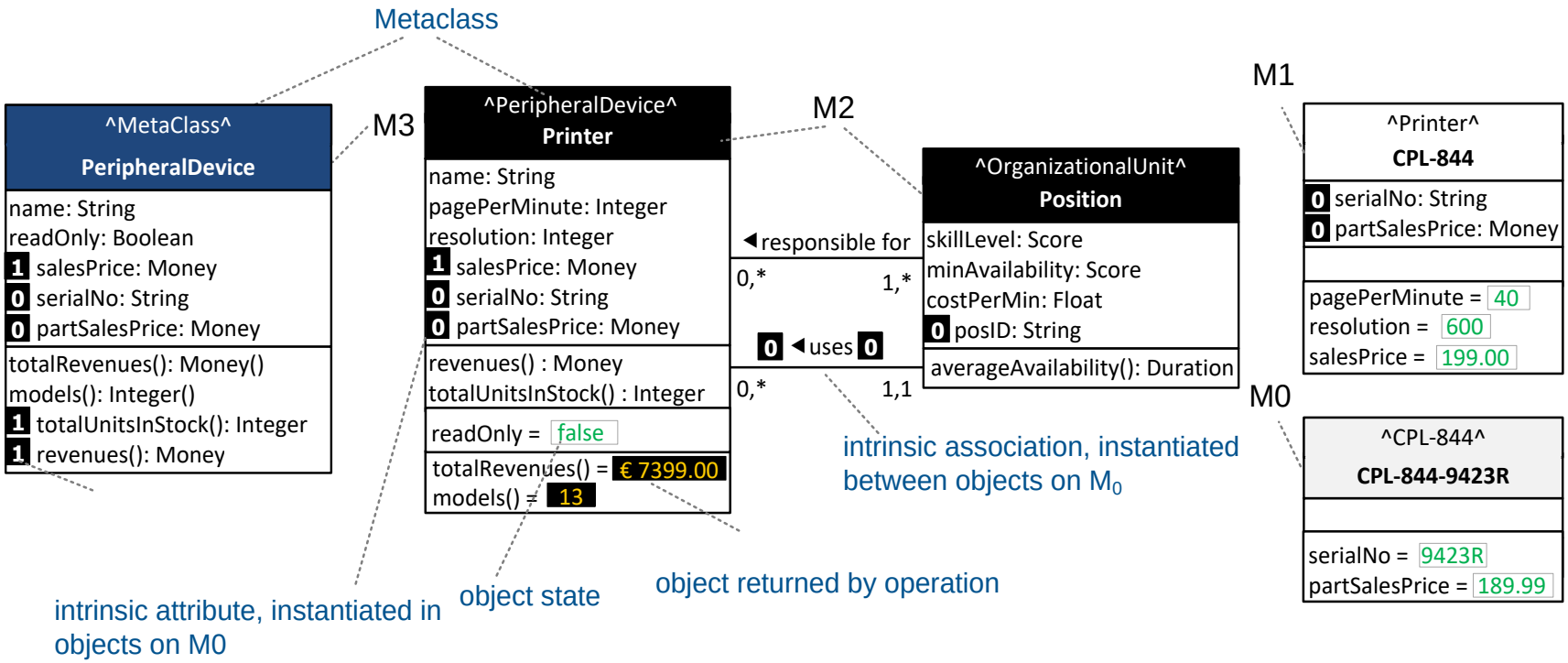
Common representation of model and code!

elling

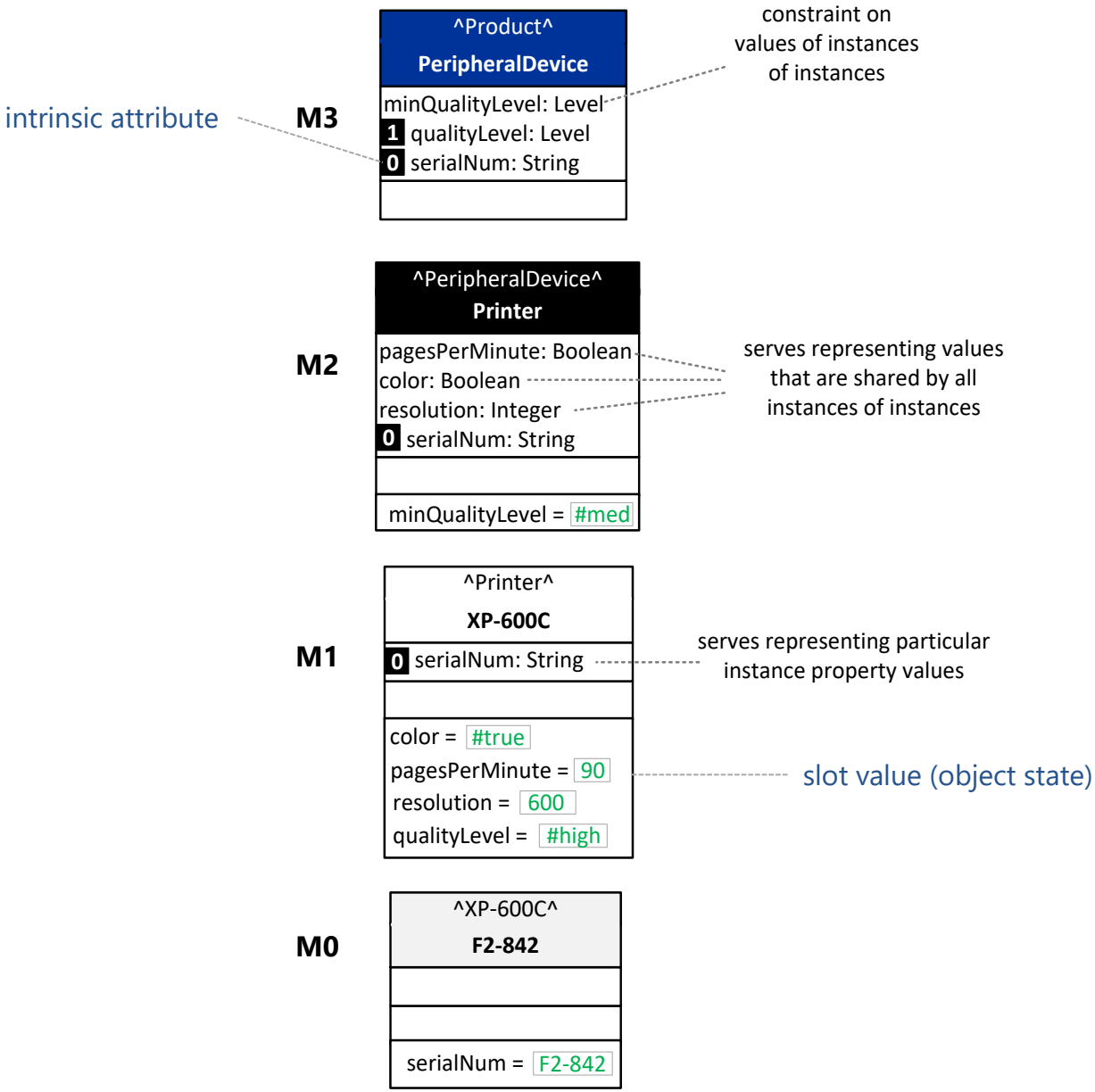
Relaxing the Conflict between Integrity and Flexibility



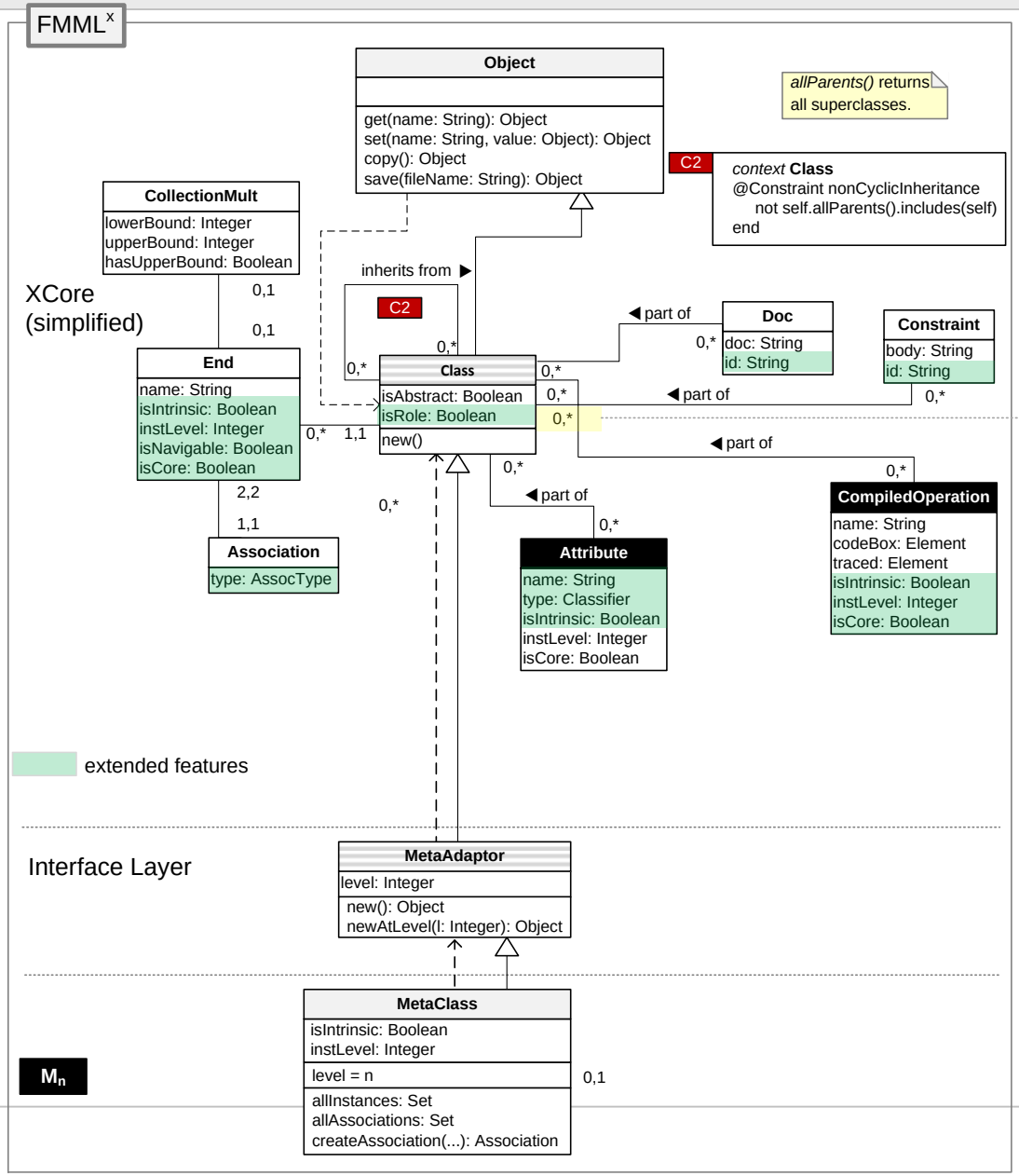
Multi-Level Language: The FMML^x



A Solution to the Problem



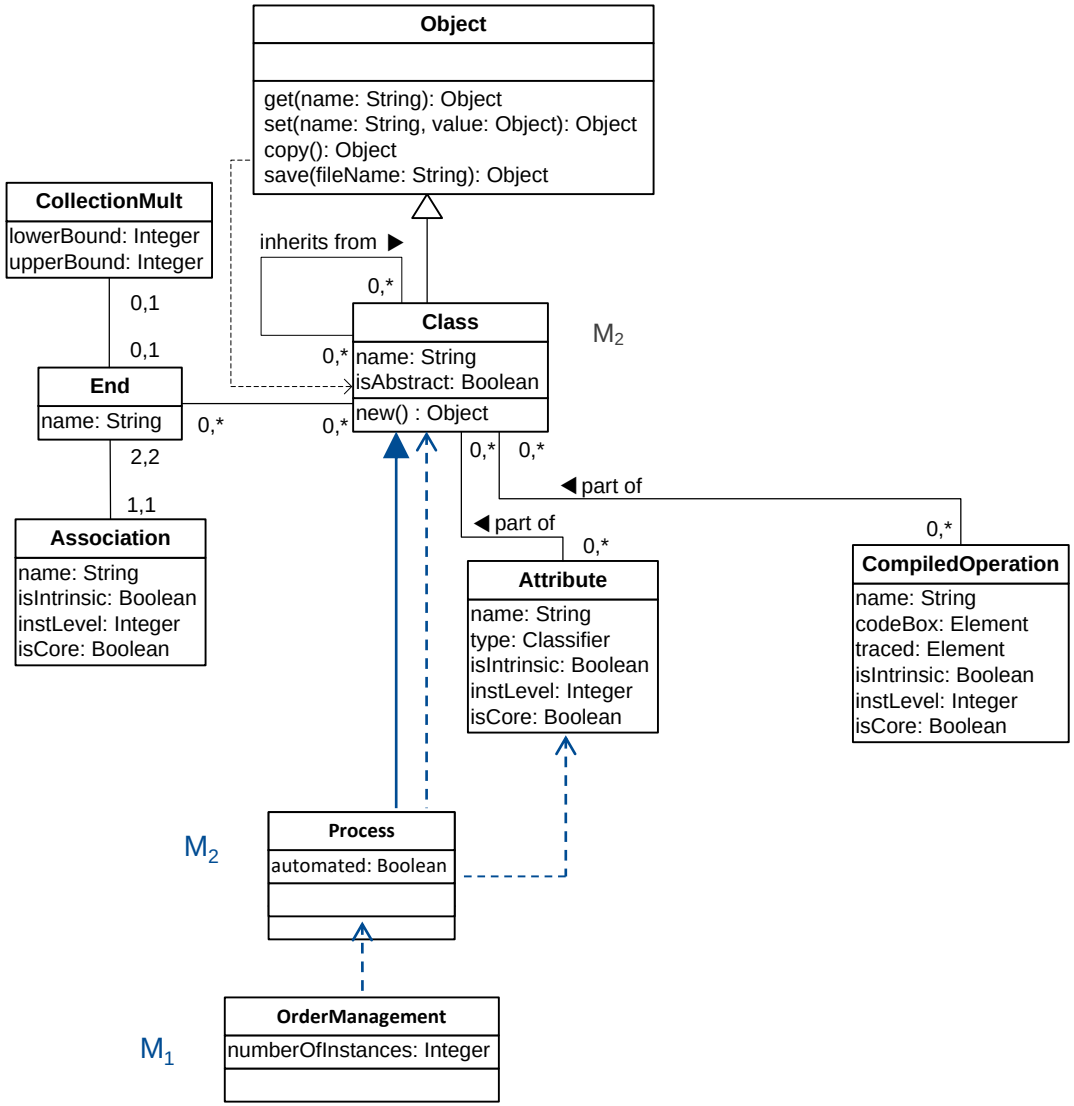
FMML^x Metamodel



Foundation: (Meta) Programming Language XMF

- similarities to Smalltalk, Lisp and OCL
- based on recursive/reflexive metamodel („golden braid“) , **XCore**
- language-execution engine
- all languages that are mapped to XCore can be executed by XMF
- allows for an arbitrary number of classification levels

Background: Raising the Level of Classification



XModeler: Language Engineering Environment

- allows for convenient specification of languages
- all languages that are defined with Xcore are executable
- includes a diagram editor to present and edit conceptual models that are represented in the same way as code
- Graphical notations can be specified with inbuilt tool.

XModeler 1.1.6 [NO_IMAGE_SET]

File Browse Multilevel

MyProjects ×

MyProjects

Welcome ×

\+ - URL: file:/C:/Users/frank/Entwic Go

XModeler

www.wi-inf.uni-due.de/LE4MM

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Sheffield Hallam University

Pattern: Search

Web Source Projects Models

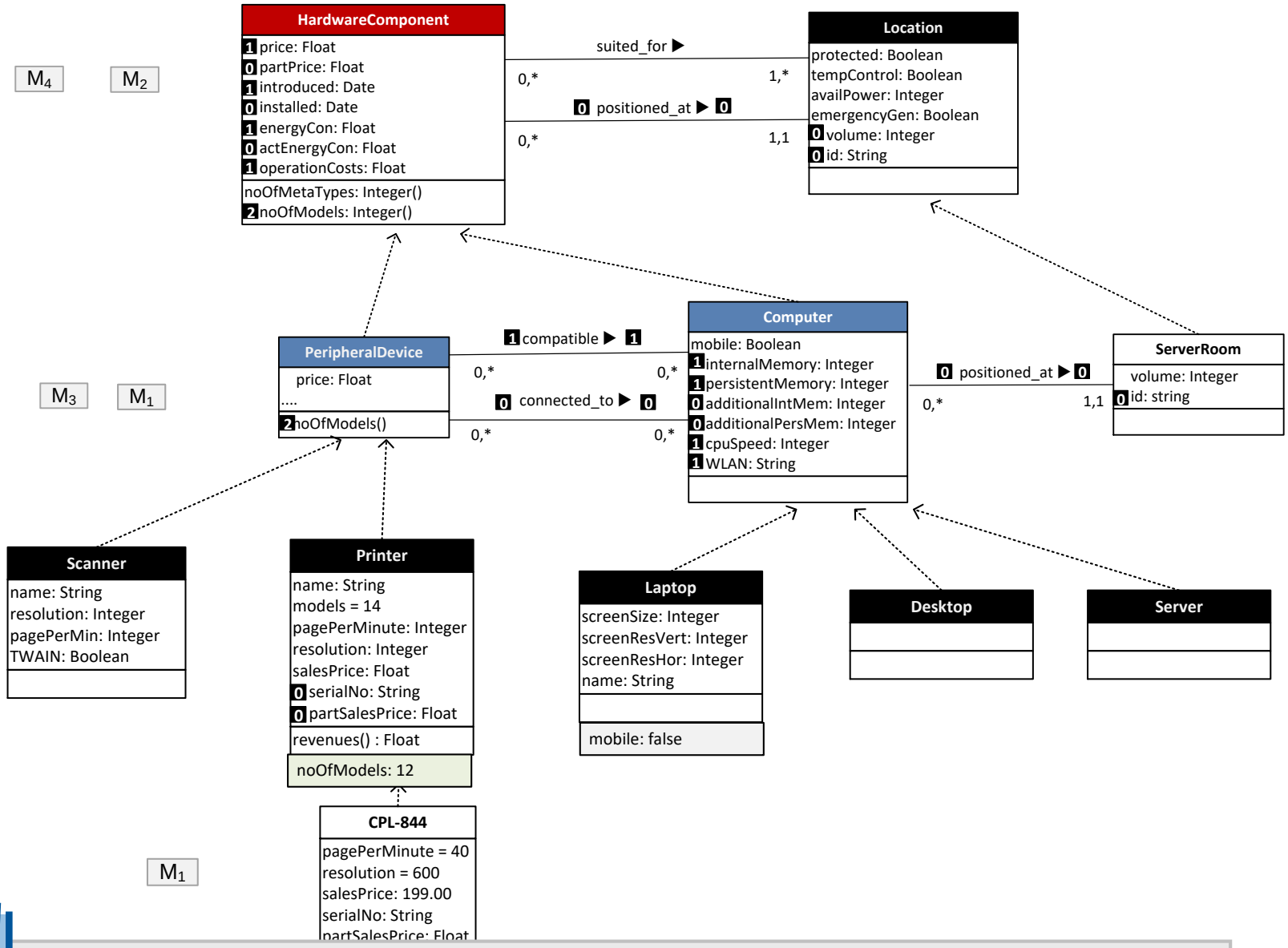
Team Contribution

The following people have contributed to the design and development of XModeler:

Tony Clark
Ulrich Frank
Jens Gulden
Daniel Töpel
Björn Benner
Paul Sammut

XMF Console ×

```
[ Loading .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/Types2.o 0:0:0:1   ms ]
[ .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/ComposedTypes.o is up to date. ]
[ Loading .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/ComposedTypes.o 0:0:0:73  ms ]
[ .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/ComposedTypes2.o is up to date. ]
[ Loading .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/ComposedTypes2.o 0:0:0:0   ms ]
[ .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/Injection.o is up to date. ]
[ Loading .\xmf-src/../xmf-addons/Multilevel-Editor/auxiliaryTypes/Injection.o 0:0:0:1   ms ]
0:0:0:359 ms ]
0:0:-49:-122 ms ]
```



Clear contribution to reuse, flexibility, integration and integrity!

Models@Runtime: Self-Referential Enterprise Systems

An SRES integrates enterprise software with corresponding enterprise modelling environment.

Common representation of model and code

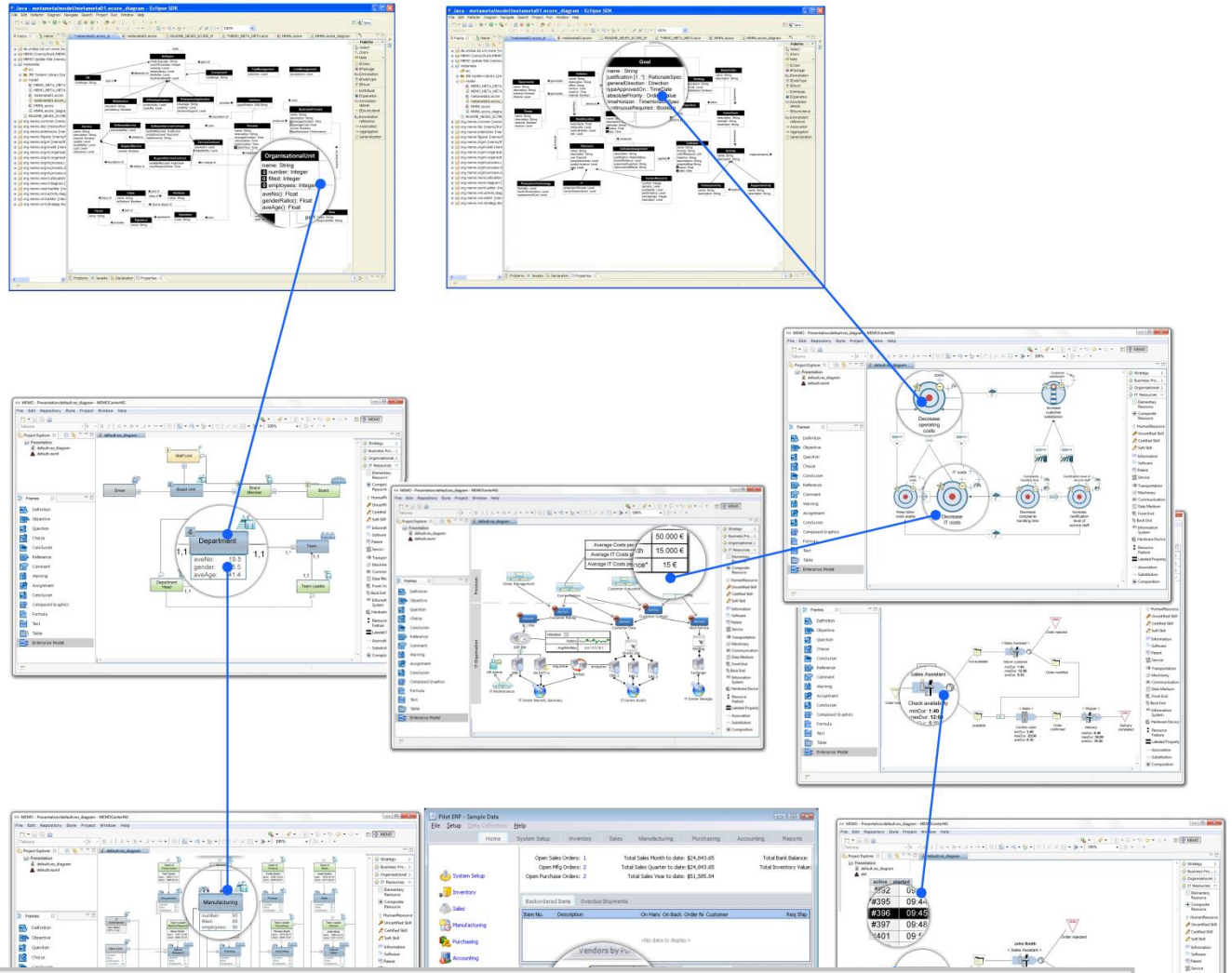
It is possible to navigate between operational level representations provided by the enterprise software and conceptual representations provided by the modelling environment.

Users get access to those parts of the model that fits their expertise and authorization.

Authorized users can change the system on any level they prefer.

Enterprise model and the respective enterprise software are automatically synchronized.

Illustration (1)



Integration of enterprise modelling environment and enterprise software at runtime

Illustration (2)

M_1

M_0

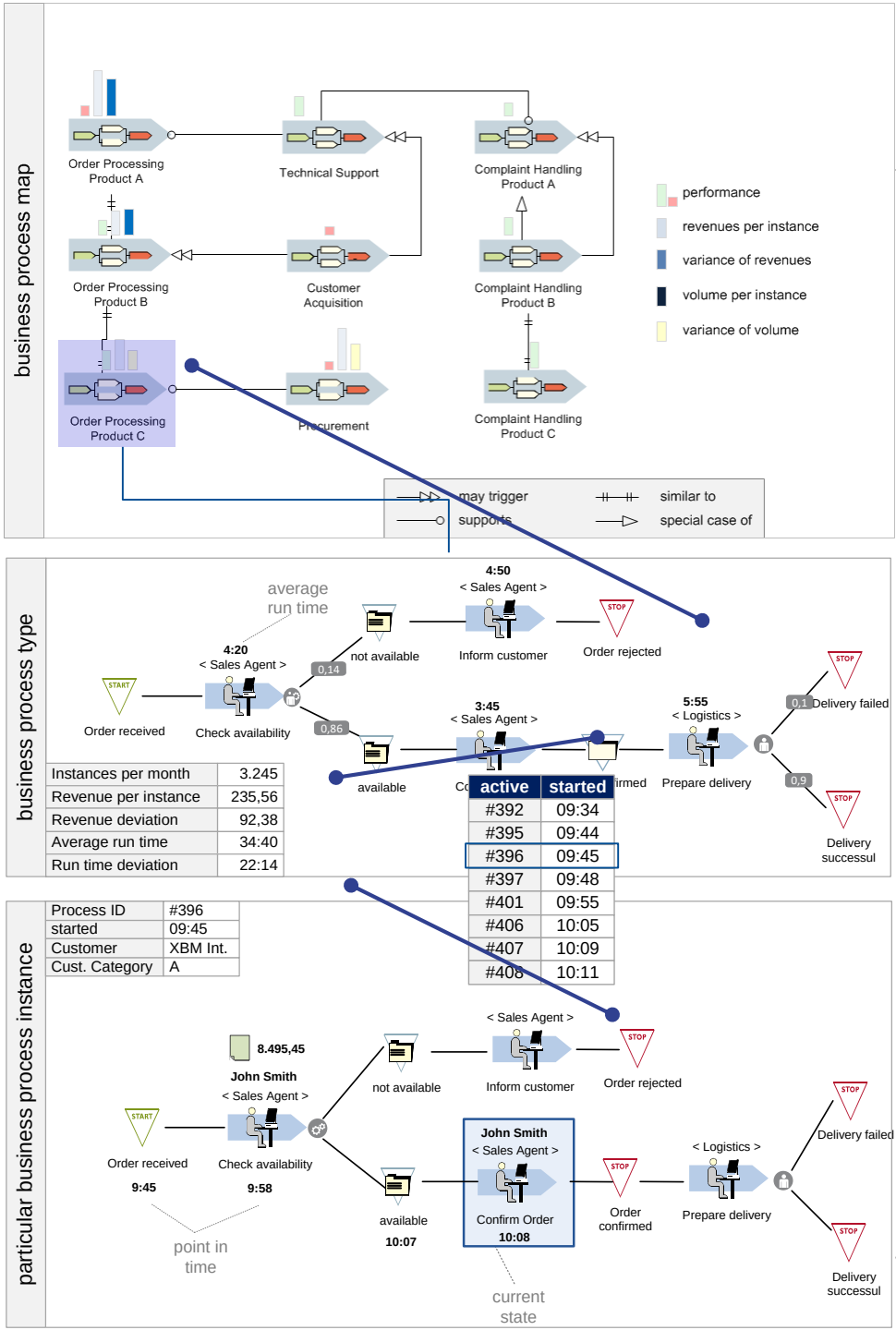
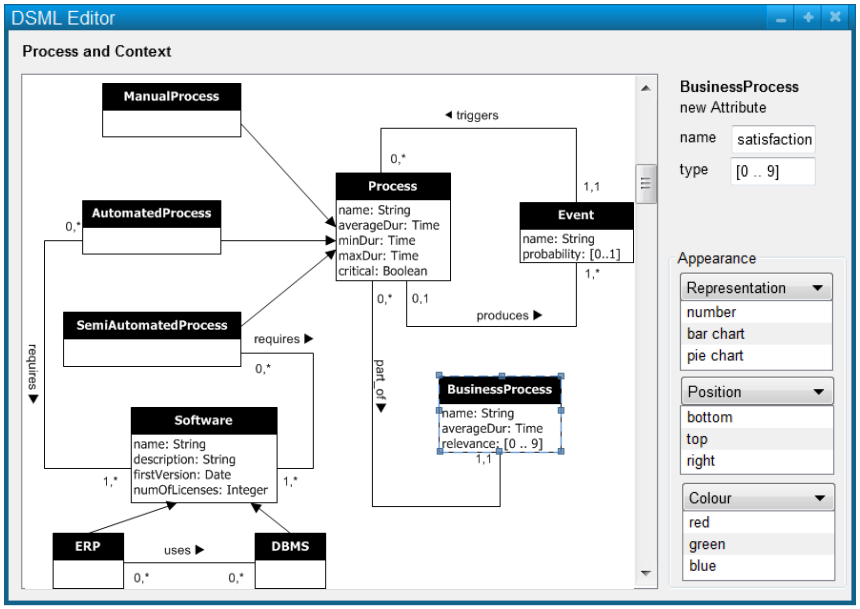
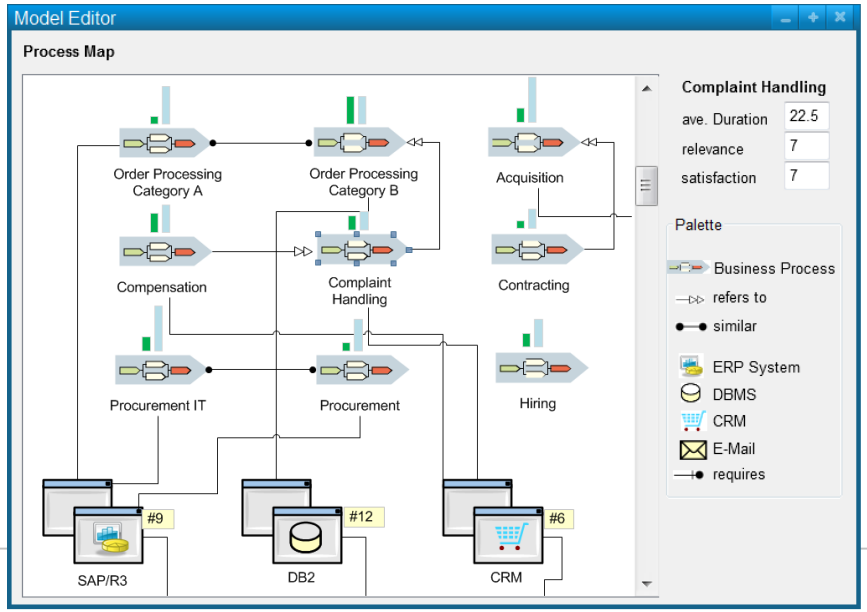


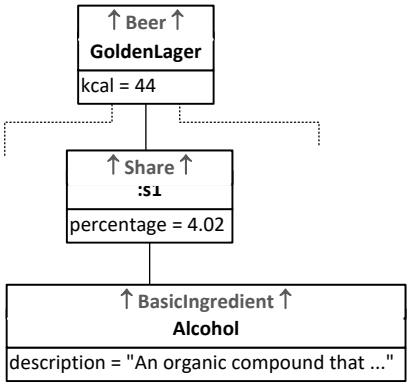
Illustration (3)

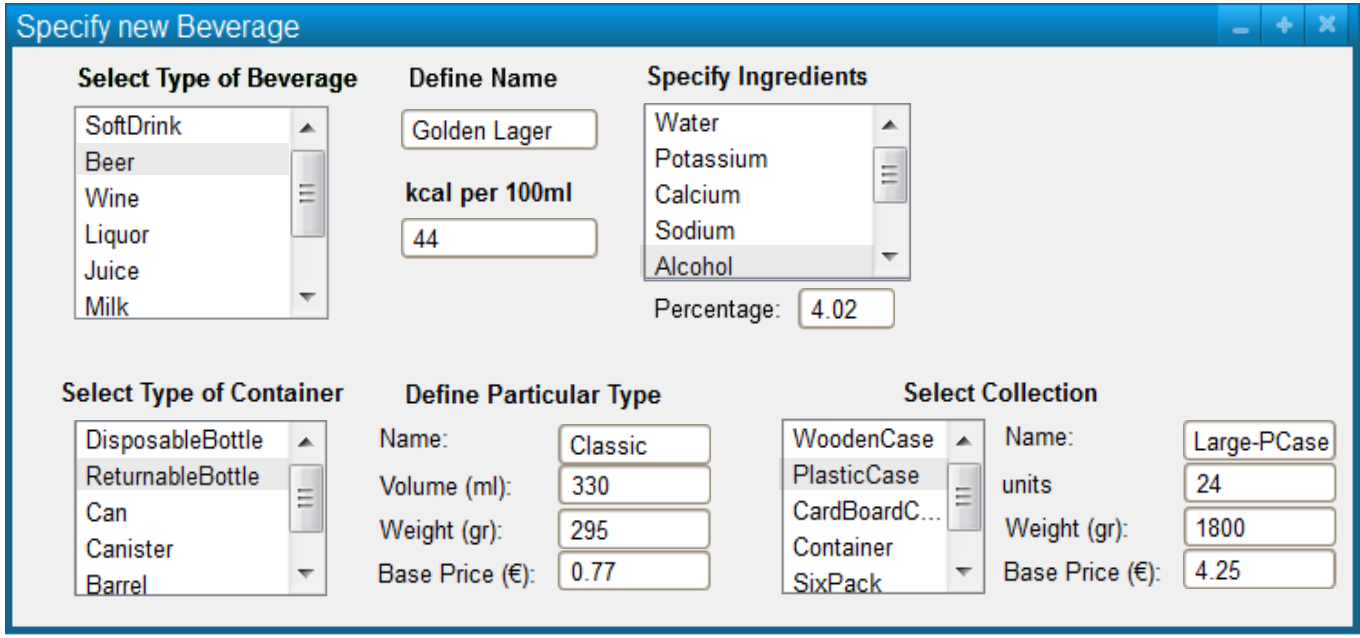




Modelling not Restricted to Diagram Editors

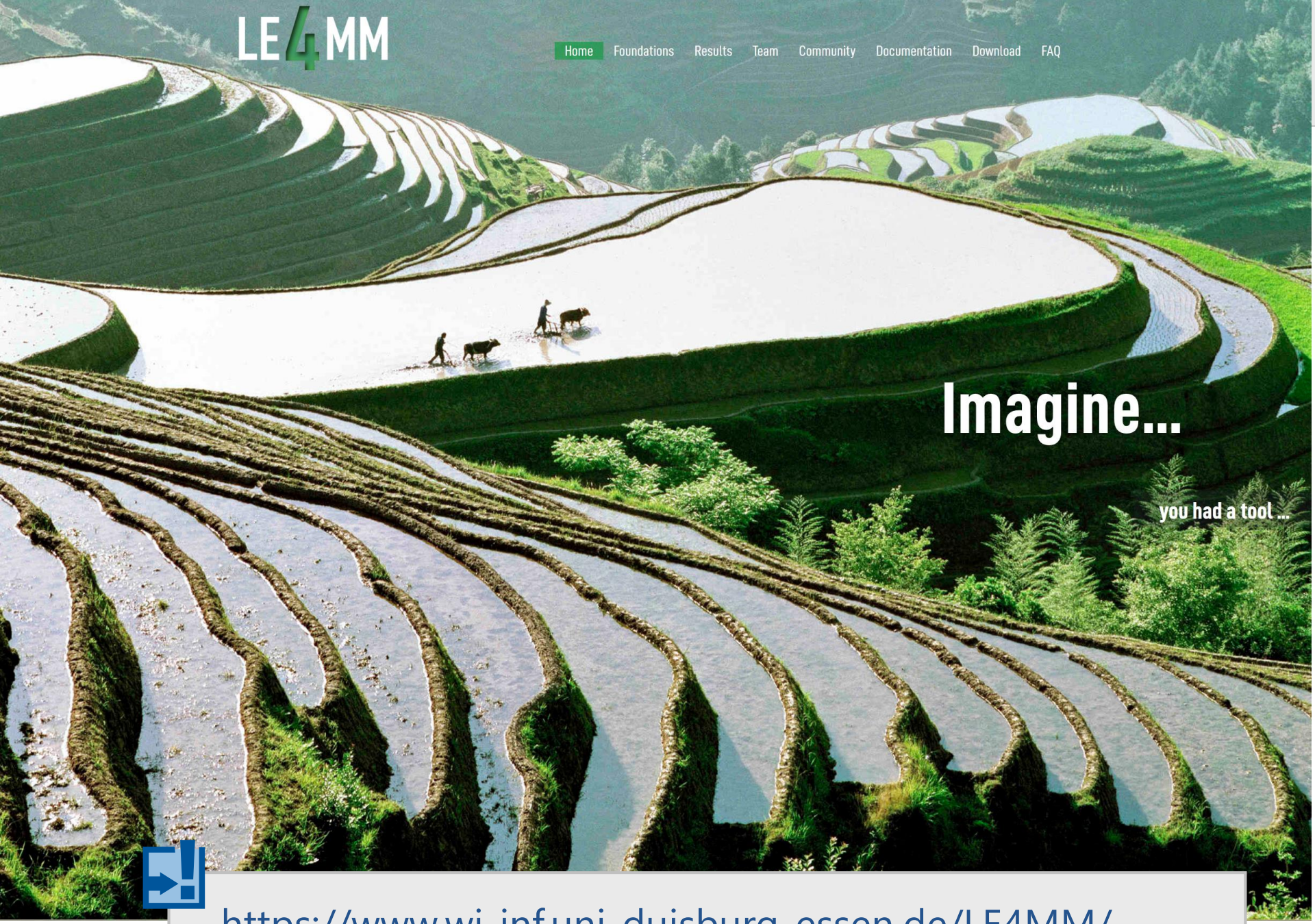
M_2, M_1





Conclusions and Future Research

- traditional language paradigm suffers from various conflicts
- multi-level modelling promising approach to
  - relax design conflicts
  - to enable self-referential software systems
- particular strength of Xmodeler: common representation of models and code
- Various challenges remain for future research
  - fill semantic gaps of current specification
  - so far focus on static abstractions, in future also on dynamic abstractions
  - however, process abstraction is especially challenging
  - construction of versatile user interfaces to hide complexity from users
  - need for more powerful solutions with regard to persistence and distribution



https://www.wi-inf.uni-duisburg-essen.de/LE4MM/

References (1)

Atkinson, C., & Kühne, T. (2001): The Essence of Multilevel Metamodeling. In M. Gorgolla & C. Kobryn (Eds.), Lecture Notes in Computer Science: Vol. 2185. UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools: 4th International Conference, Toronto, Canada, October 1-5, 2001. Proceedings (pp. 19–33). Berlin, London, New York: Springer.

Clark T., Sammut P., Willans J. (2008): Applied metamodelling: a foundation for language driven development. https://eprints.mdx.ac.uk/6060/1/Clark-Applied_Metamodelling_%28Second_Edition%29%5B1%5D.pdf.

Clark T, Sammut P, Willans J (2008): Superlanguages: Developing Languages and Applications with XMF. https://eprints.mdx.ac.uk/6079/.

References (2)

Frank U (2012): Thoughts on Classification / Instantiation and Generalisation / Specialisation. (ICB Research Report No. 53). Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen

Frank U (2014): Multilevel Modeling: Toward a New Paradigm of Conceptual Modeling and Information Systems Design. In: Business and Information Systems Engineering 6(6), pp. 319–337.

Frank, U. (2018). The Flexible Multi-Level Modelling and Execution Language (FMMLx) (ICB Research Report No. 66). Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen

Frank, U. (2019). Specification and Management of Methods: A Case for Multi-Level Modelling. In I. Reinhartz-Berger, J. Zdravkovic, J. Gulden, & R. Schmidt (Eds.): Enterprise, Business-Process and Information Systems Modeling: 24th International Conference EMMSAD 2019. Proceedings (pp. 311–325).