

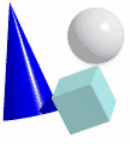
Ralf Gitzel  
Matthias Merz

SCI 2004



# How a Relaxation of the Strictness Definition Can Benefit MDD Approaches With Meta Model Hierarchies

Ralf Gitzel, Matthias Merz  
{gitzel|merz}@wifo3.uni-mannheim.de  
University of Mannheim  
Department of Management Information Systems  
Germany



# Presentation Structure

Ralf  
Gitzel

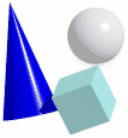
Matthias  
Merz

## Agenda

- Introduction
- Strictness
- Relaxed Definition of Strictness
- Code Generation Sample
- Conclusion

SCI 2004





# Introduction

Ralf  
Gitzel

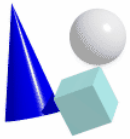
Matthias  
Merz

## Meta modeling

- Mature and established concept
- Essential aspect of many MDD approaches
- Distinguish between
  - ontological and
  - linguistic meta modeling axes
- Separation of ontological and linguistic aspects
  - enables larger hierarchies
  - but also introduces new problems and opportunities

SCI 2004





# Introduction

Ralf Gitzel

Matthias Merz

SCI 2004

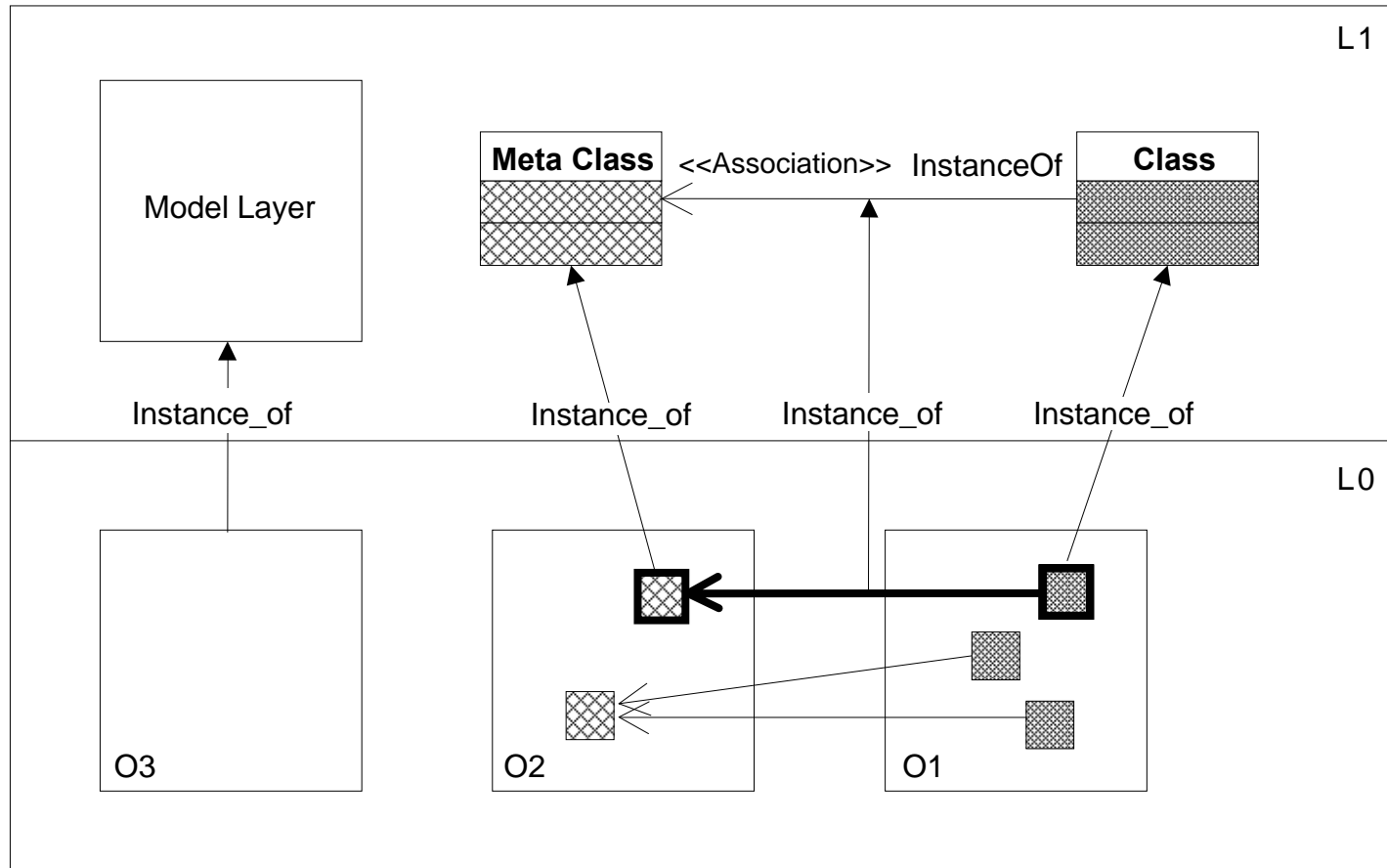
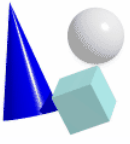


Figure 1 - Core Principle of Non-Linear Meta Modeling



# Strictness

Ralf  
Gitzel

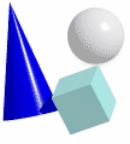
Matthias  
Merz

## Concept of strict meta modeling

- Strictness is often taken for granted, particularly when using architectures such as MOF
- Strictness is a concept that
  - provides the structure of the model layers
  - avoids “illogical” scenarios
- Prevents the meta model hierarchy from collapsing into a single super layer containing all model information

SCI 2004





# Strictness

Ralf  
Gitzel

Matthias  
Merz

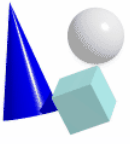
## Definition of Strictness

"In an  $n$ -level modeling architecture,  $M_0, M_1, \dots, M_{n-1}$ , every element of an  $M_m$ -level model must be an *instance-of* exactly one element of an  $M_{m+1}$ -level model, for all  $0 \leq m < n-1$ , and any relationship other than the *instance-of* relationship between two elements  $X$  and  $Y$  implies that  $\text{level}(X) = \text{level}(Y)$ ."

*Atkinson, C. and Kühne, T. (2002):  
Rearchitecting the UML Infrastructure*

SCI 2004





# Strictness

Ralf  
Gitzel

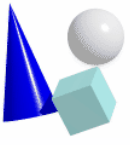
Matthias  
Merz

## Restrictions imposed by Strictness

- Model elements should only have relationships within their own layer and not between layers
- Exception: The *instanceOf* relationship, which represents the connection between the different layers
- One element in a model layer can only instantiate elements of its immediate parent layer

SCI 2004



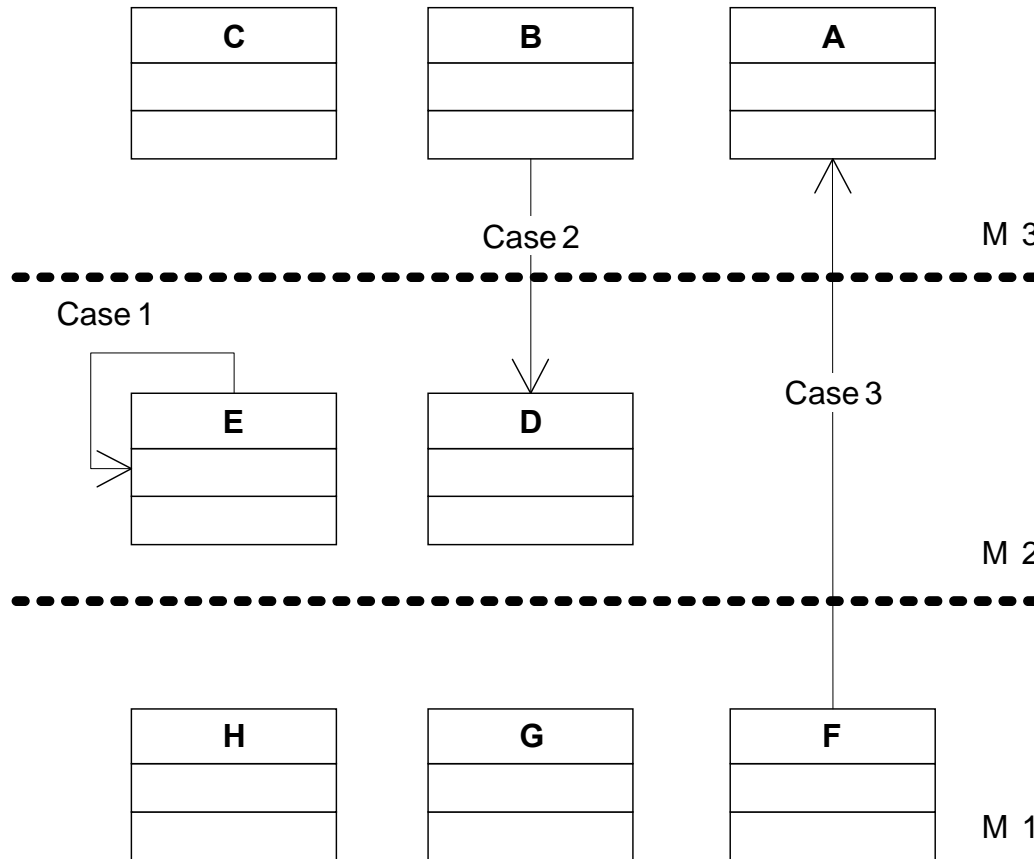


# Strictness

Ralf  
Gitzel

Matthias  
Merz

SCI 2004



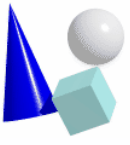
Case 1:  
A class is an  
instance of itself.

Case 2:  
An instantiation  
in the wrong  
direction.

Case 3:  
An Instantiation  
of a model  
element which is  
not in the layer  
directly above.

Figure 2 - Violations of Strictness





# Relaxed Definition of Strictness

Ralf  
Gitzel

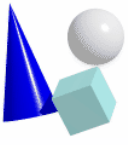
Matthias  
Merz

## Relaxed Definition

In an  $n$ -level modeling architecture,  $M_0, M_1, \dots, M_{n-1}$ , every element of an  $M_m$ -level model must be an instance-of exactly one element of an  $M_o$ -level model, for all  $0 \leq m < n-1$ ,  $m < o \leq n-1$ , and any relationship other than the instance-of relationship between two elements  $X$  and  $Y$  implies that  $\text{level}(X) = \text{level}(Y)$ .

SCI 2004





# Relaxed Definition of Strictness

Ralf Gitzel

Matthias Merz

## Example

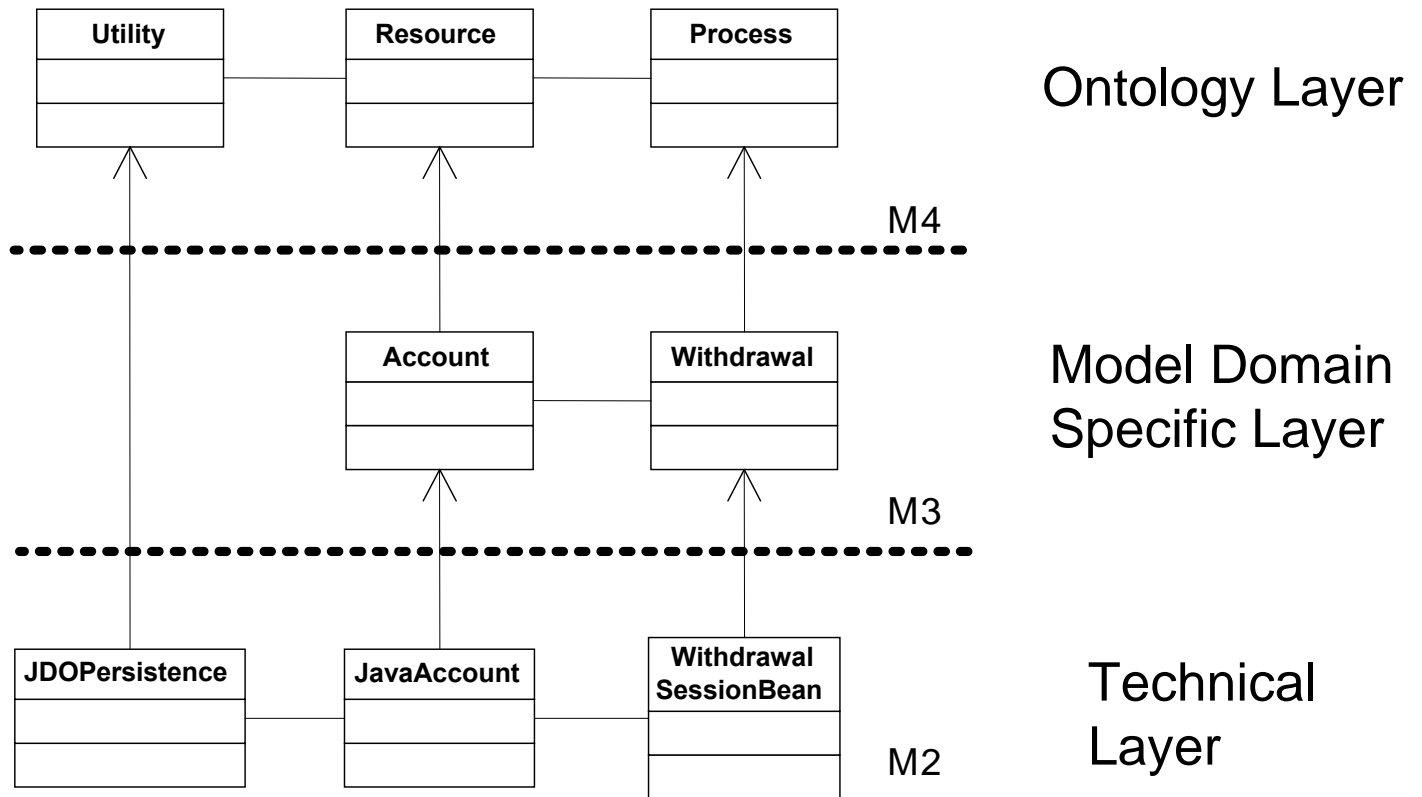
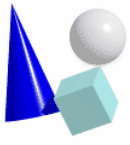


Figure 3 - A Scenario With Relaxed Strictness

SCI 2004





# Relaxed Definition of Strictness

Ralf Gitzel

Matthias Merz

## Other Solutions?

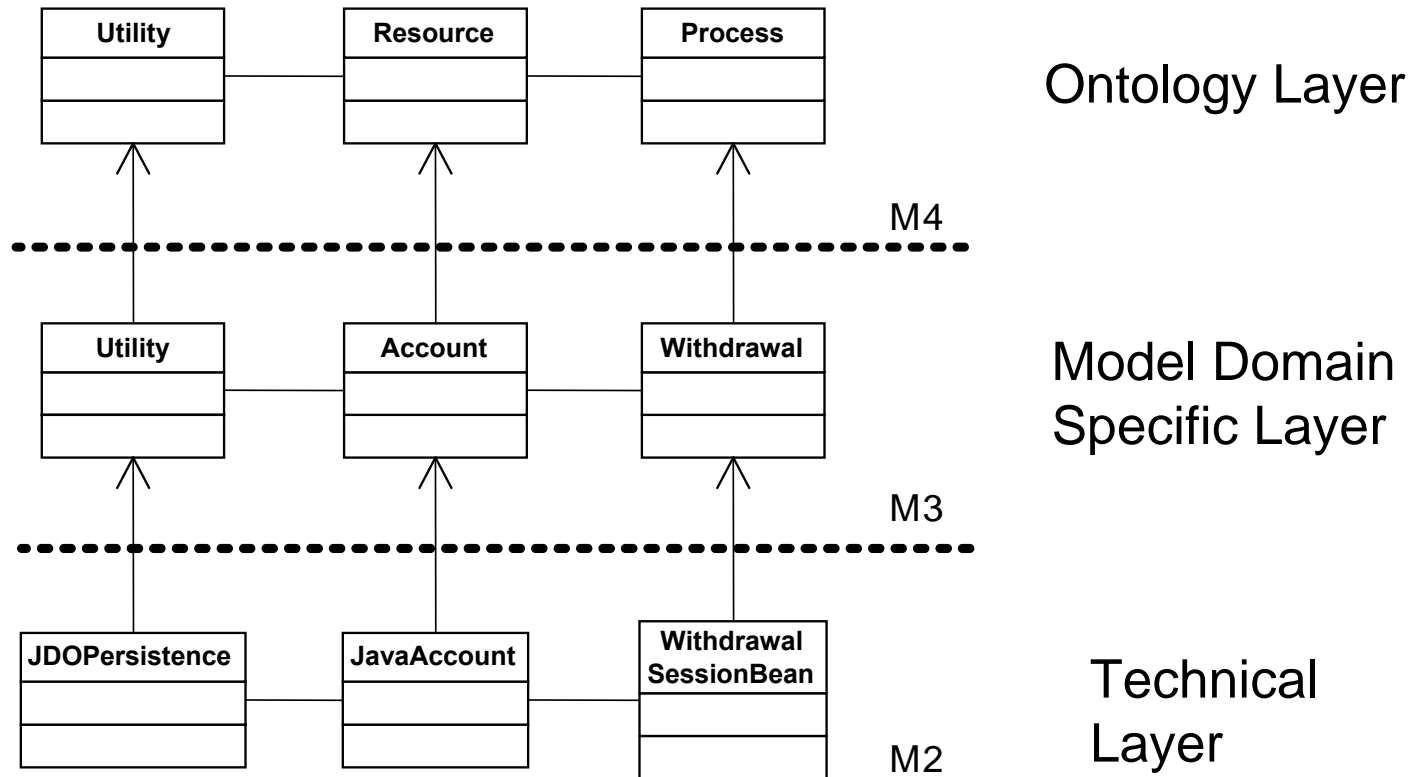
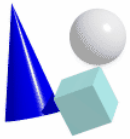


Figure 4 - Replicate Utility in the M3 layer

SCI 2004





# Relaxed Definition of Strictness

Ralf Gitzel

Matthias Merz

## Other Solutions?

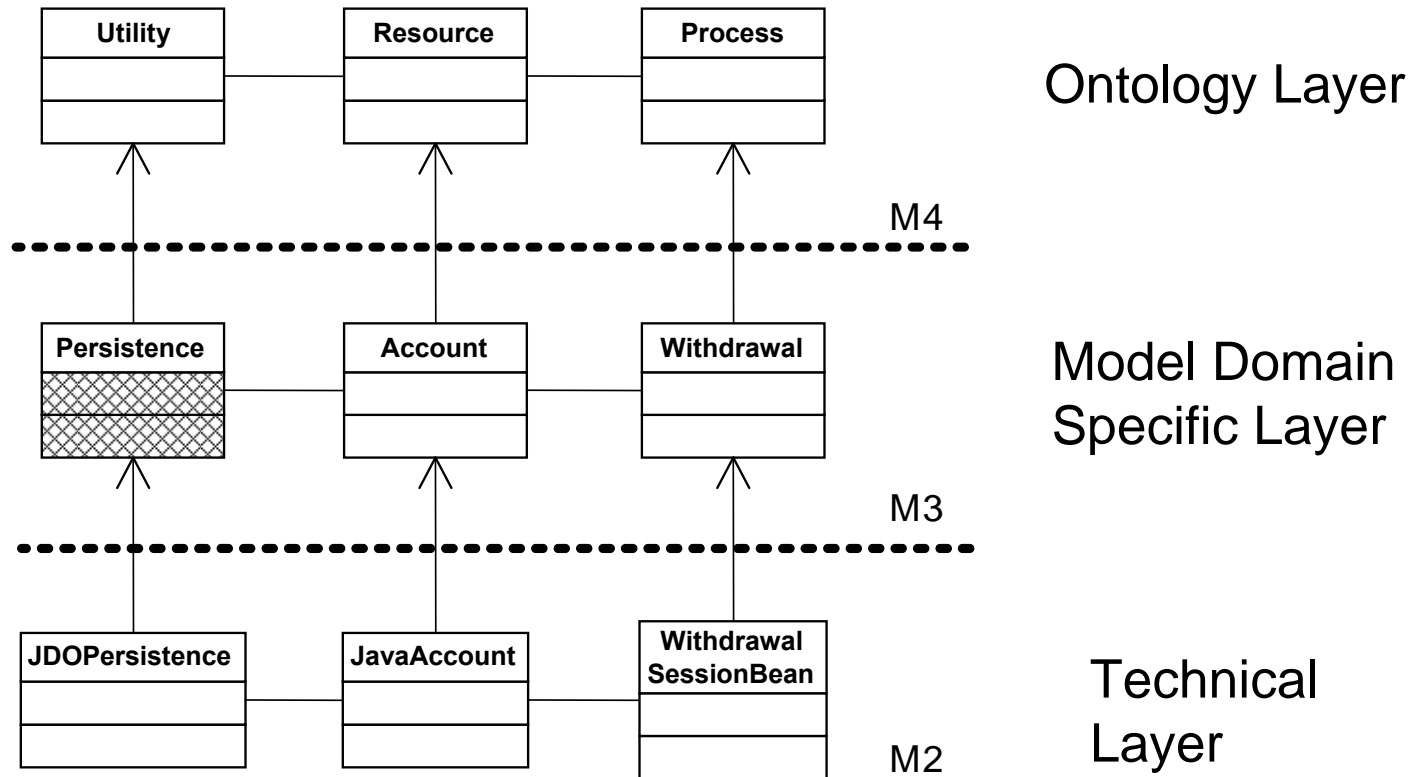
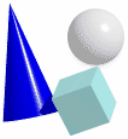


Figure 5 - Introducing a Persistence Meta Class in M3

SCI 2004





# Relaxed Definition of Strictness

Ralf  
Gitzel

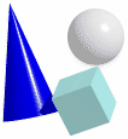
Matthias  
Merz

## Conclusion:

- There are scenarios where a relaxed strictness definition can be useful
- While there is often a technical way to avoid the need for the relaxation this can lead to problems with regard to the semantics of the model layers
- Besides the theoretical aspects - the question remaining now is if the relaxed Definition of Strictness benefits for the code generation process

SCI 2004





# Code Generation Sample

Ralf  
Gitzel

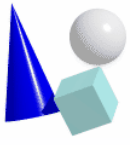
Matthias  
Merz

## Overview:

- Small prototypical code generation script, which reads in an XML file containing the model information
- Ant script is used to generate, compile, and deploy the examples
- Templates (written in XSLT) which cover business logic and technological mapping
- Based on
  - XDoclet 1.2
  - JBoss 3.2.3 application server
  - JDO-implementation IntelliBo 3.61
  - MySQL-Database 4.0.15 with InnoDB extension

SCI 2004





# Code Generation Sample

Ralf Gitzel

Matthias Merz

SCI 2004

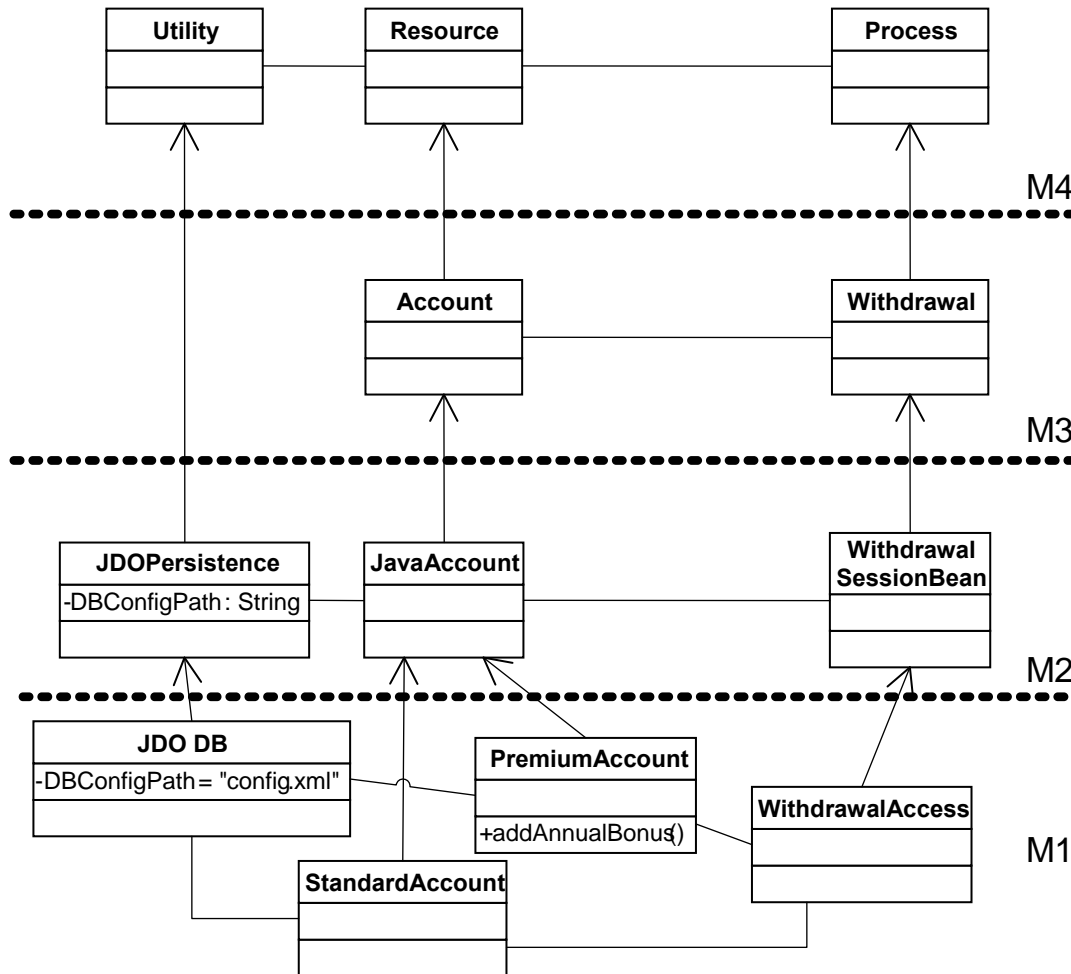
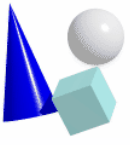


Figure 6 - A Code Generation Example



# Code Generation Sample

Ralf Gitzel

Matthias Merz

SCI 2004

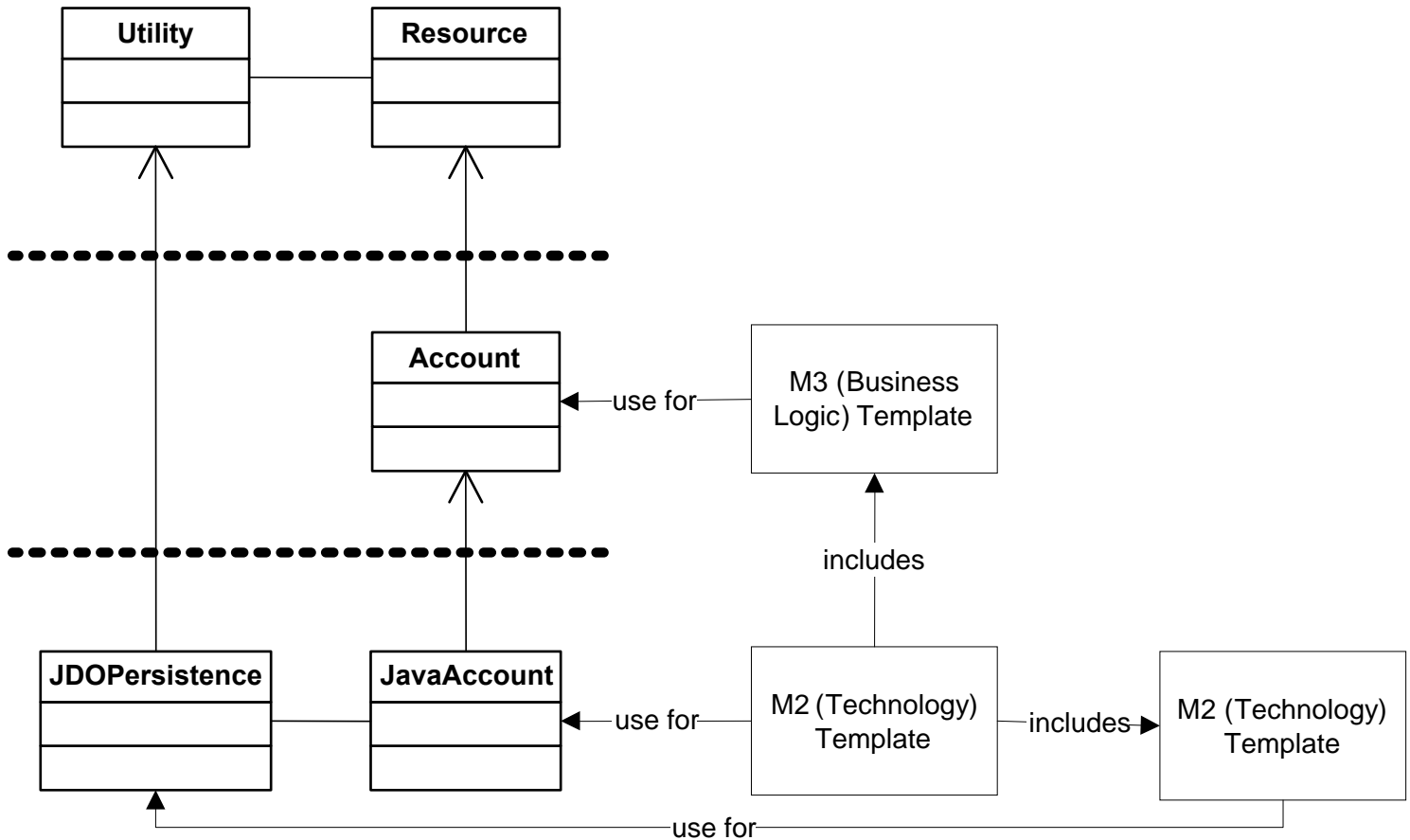
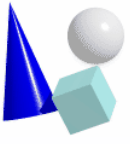


Figure 7 - Templates for Code Generation





# Code Generation Sample

Ralf  
Gitzel

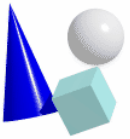
Matthias  
Merz

## Code Generation Summary

- The information provided by the model is sufficient for code generation purposes
- A code generator can handle the “gap” in the M3 layer
- Without the relaxed definition of strictness, a lot of replication in the templates is required

SCI 2004





# Conclusions

Ralf  
Gitzel

Matthias  
Merz

## The Relaxation of the Strictness Definition

- can be beneficial to automated code generation
- improves the quality of the input models
- avoids replication of concepts and therefore reduces the number of templates
- can be also beneficial for concepts which occur on all layers of a model such as instances of primitive data types (e.g. String or Integer)

SCI 2004

