CSSE 490 Model-Based Software Engineering: Domain Engineering

Shawn Bohner

Office: Moench Room F212

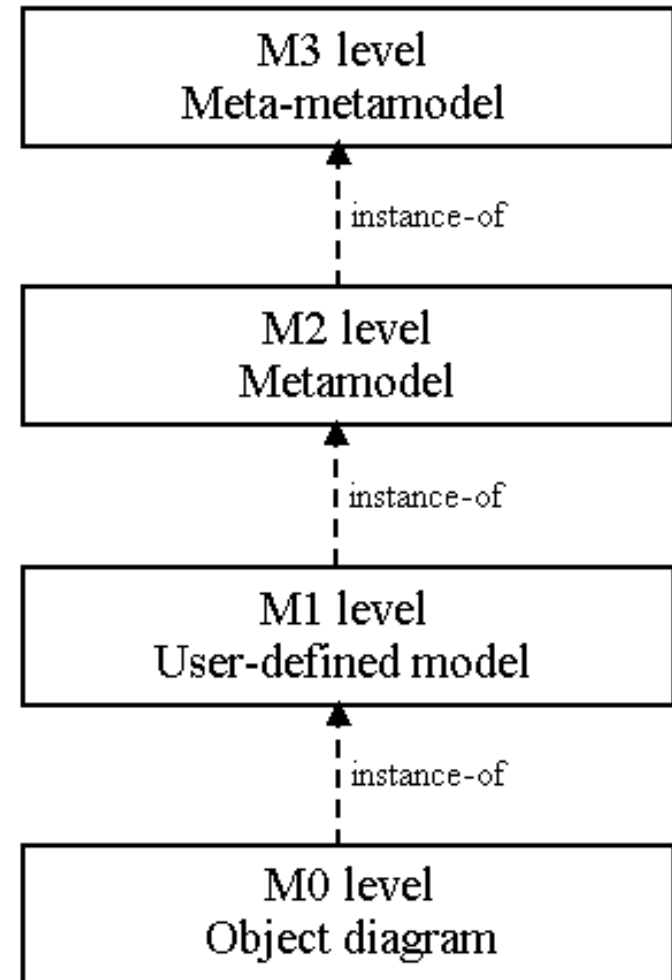**Phone: (812) 877-8685**
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Learning Outcomes: Metamodels

Design a metamodel for a model-based software system.

- **Case Study Discussion**
- **Looking closer at Mapping**
- **Introduce Object Constraint Language (OCL)**
- **Action Semantics (if time)**
- **Introduce Domain Engineering**

M3 level
Meta-metamodel

↑ instance-of

M2 level
Metamodel

↑ instance-of

M1 level
User-defined model

↑ instance-of

M0 level
Object diagram

Case Study/Homework:

"UML 2: A model-driven development tool"
    by Bran Selic

- What are the alternatives?
- How hard are they to implement?
- Is there support from the community?

What is the difference between conventional Software Engineering and Model Based Software Engineering?

- Again, think for 15 seconds…
- Let's talk…

Conventional and MBSE

Conventional Software Engineering	Model-Based Engineering
Requirements Analysis **Produces requirements for one system**	*Domain Analysis* **Produces reusable, configurable requirements for a class of systems**
System Design **Produces design of one system**	*Design* **Produces reusable design for a class of systems and a production plan**
System Implementation **Produces system implementation**	*Implementation* **Produces reusable components, infrastructure and production process**

Abstraction Gaps Bain of Mapping
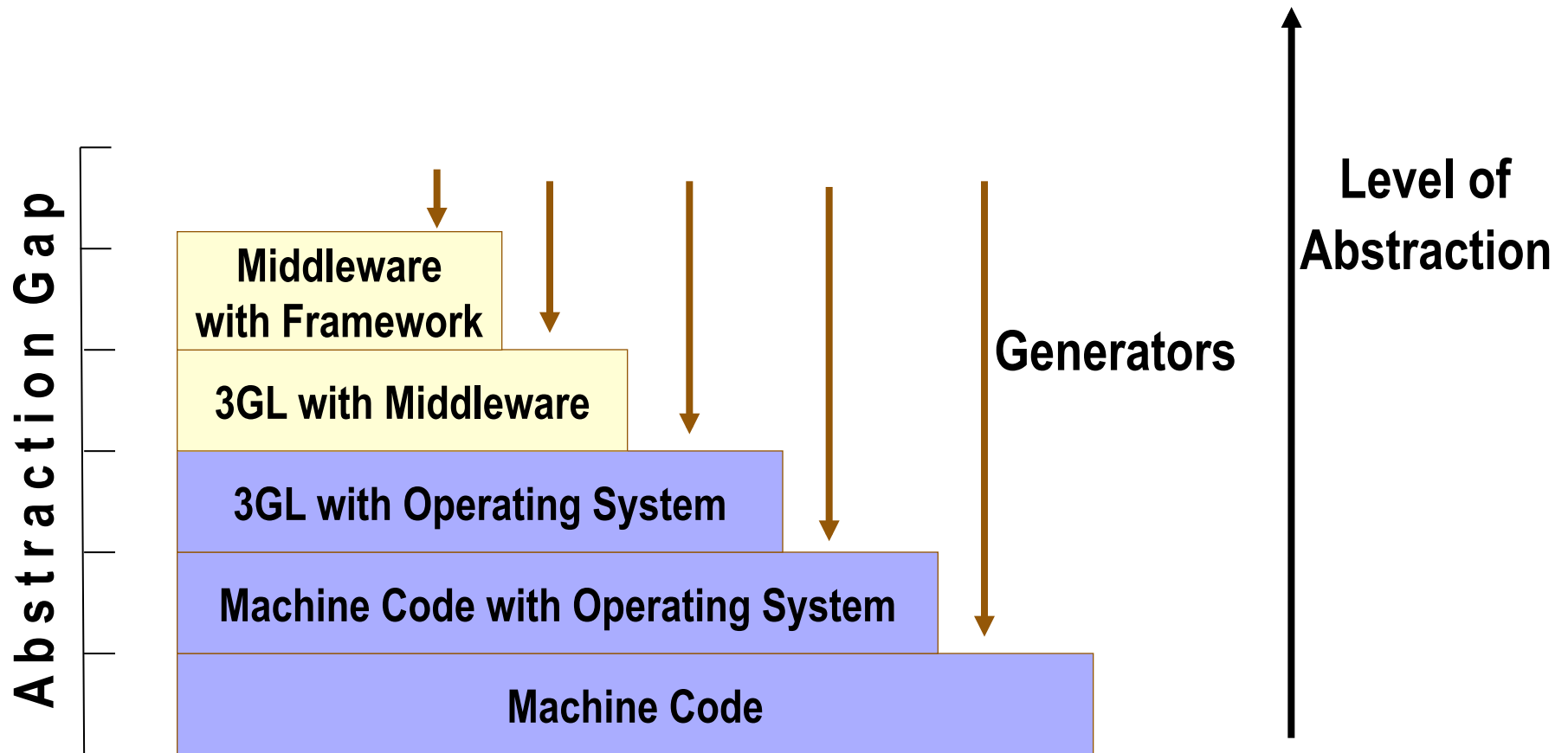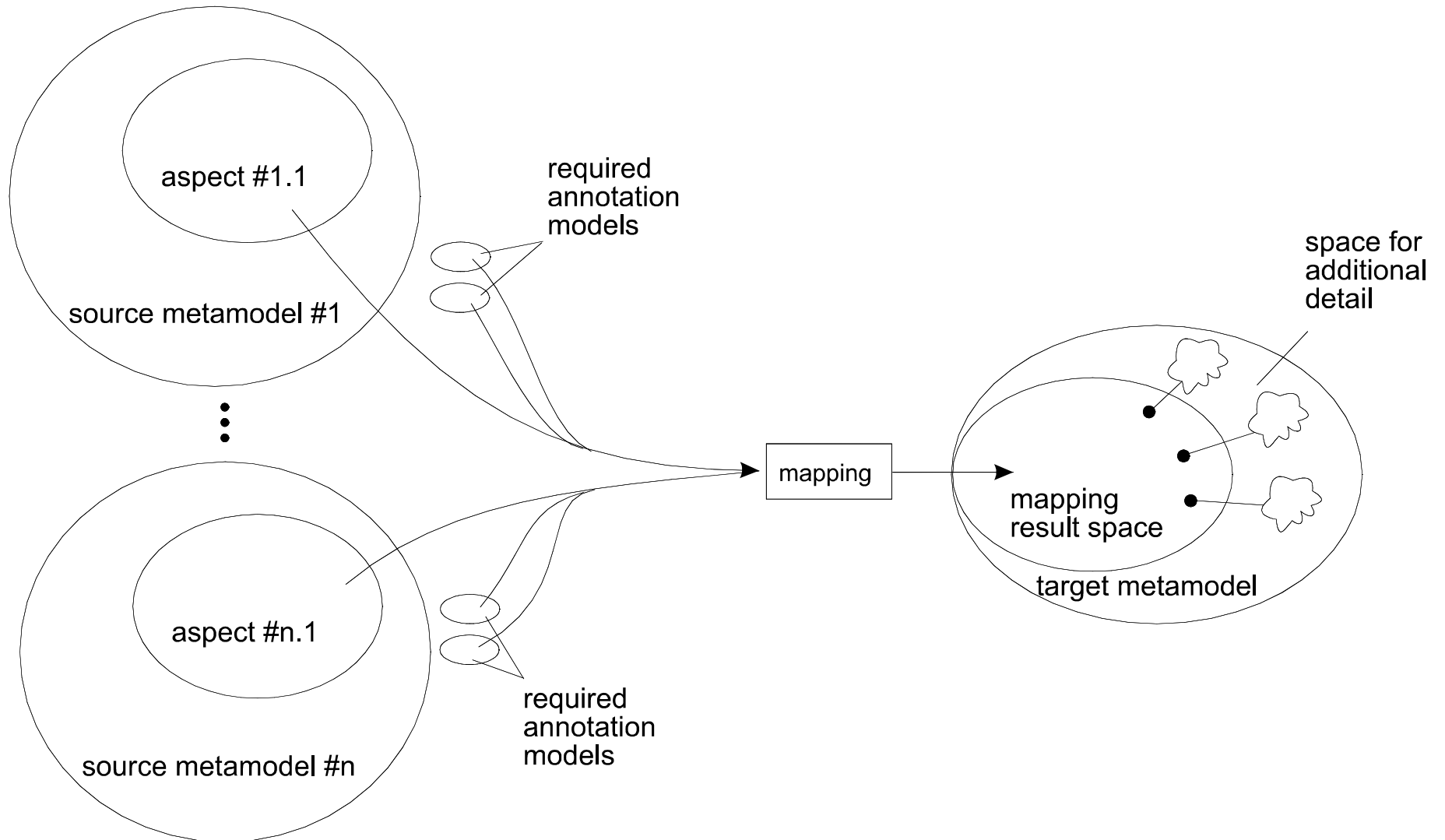
Abstraction or Refinement?

- **Mapping techniques between two metamodels often formulate**

    1. **An abstraction (leading to more abstract metamodels)   or**

    2. **A refinement (leading to more detailed metamodels)**

- **Hence, one metamodel is sometimes called an *abstraction* or a *refinement* of the other**
    - **When do we call a mapping a refinement?**
    - **When do we call it an abstraction?**

Definitions: Refinement

- Let A and B be two metamodels

- B is said to be a **refinement** of A if

  a "reasonable" (semantic-preserving)

  "surjective" mapping technique
  (or mapping in the algebraic sense)

  from A to B <u>cannot</u> be provided

Refinement Mapping

aspect #1.1

source metamodel #1

required
annotation
models

aspect #n.1

source metamodel #n

required
annotation
models

mapping

mapping
result space

target metamodel

space for
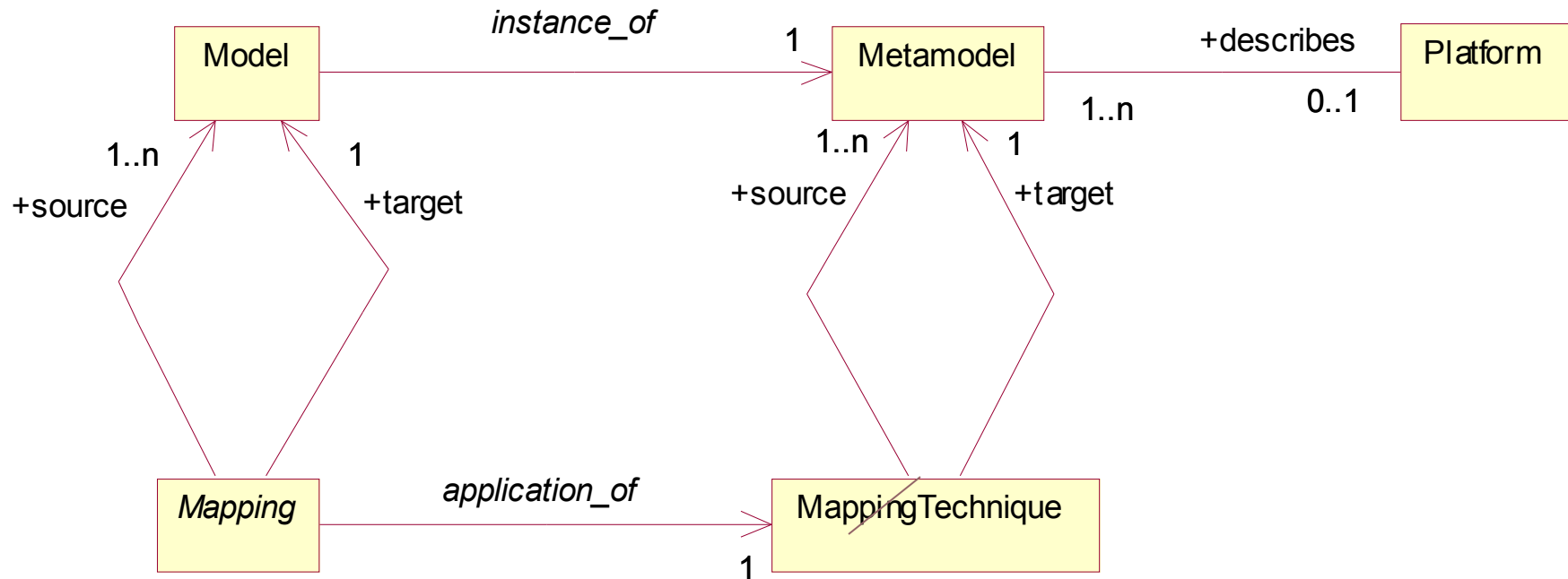additional
detail

Definitions: Abstraction

- Let A and B be two metamodels

- B is said to be an **abstraction** of A if

  a "reasonable" (semantic-preserving) <u>surjective</u> and <u>non-injective</u> mapping technique (or mapping in the algebraic sense)

  from A to B <u>can</u> be provided
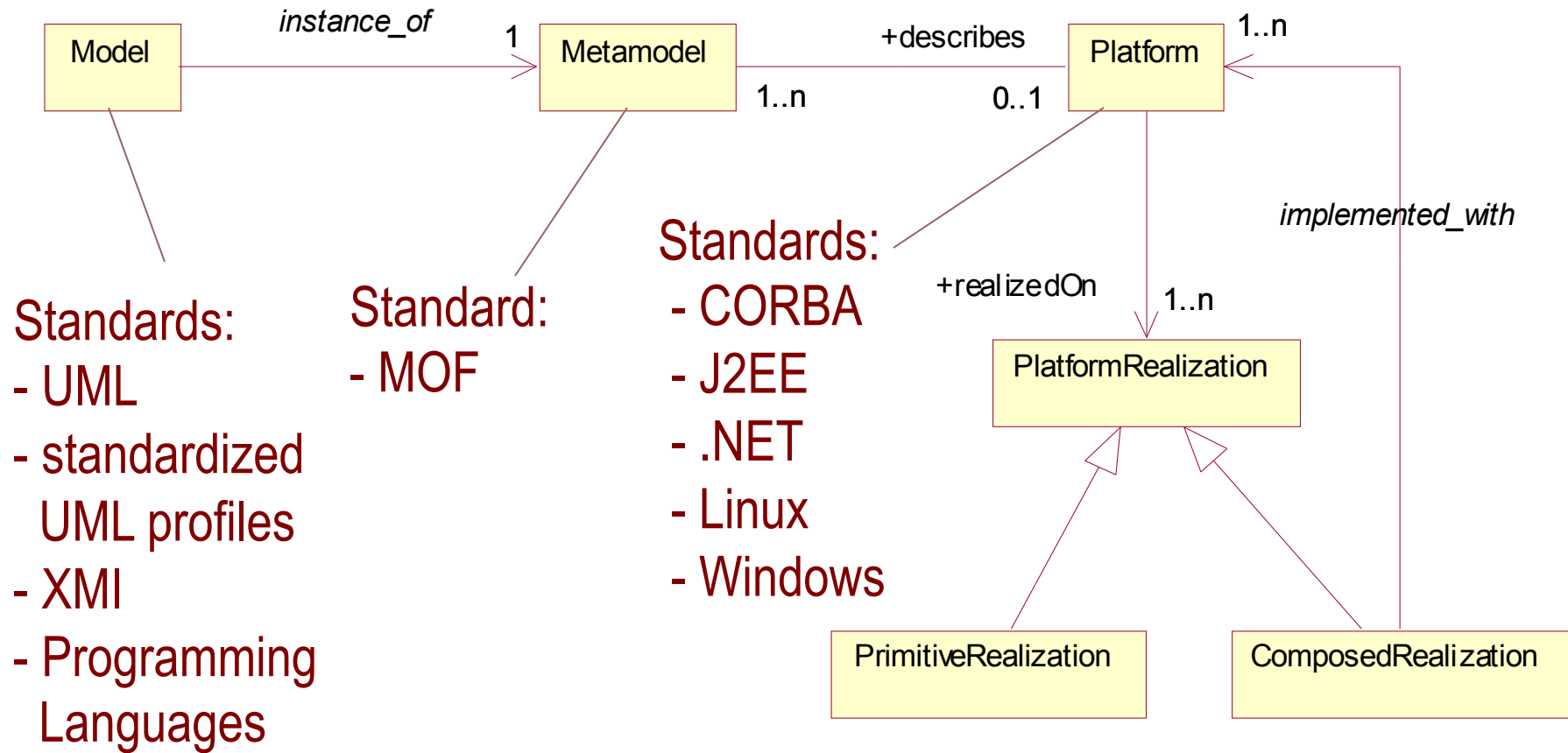
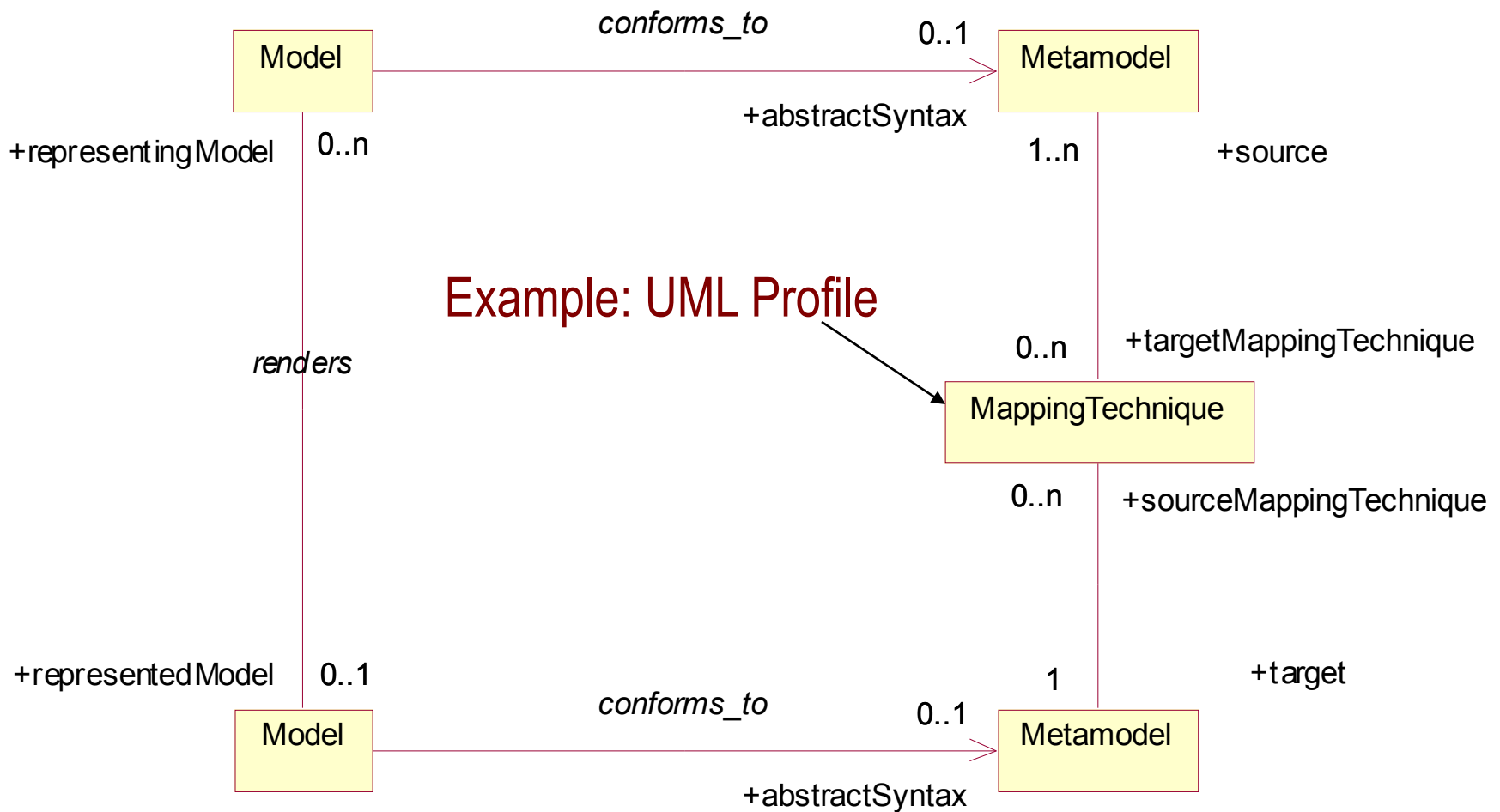# A matter of perspective ☺

Mapping Models

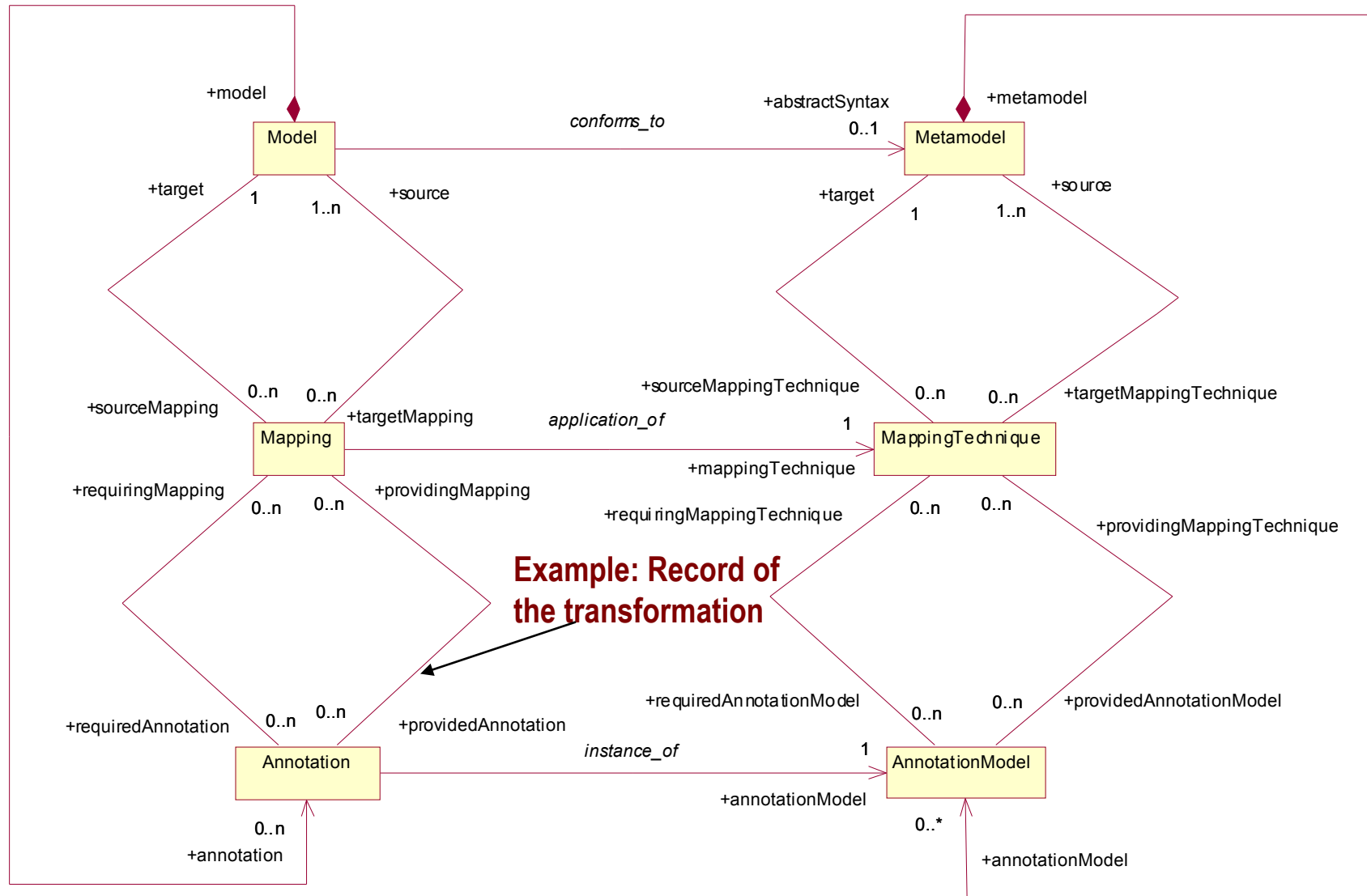**Multiple mappings may be applied successively in a chain**

Models, Metamodels, & Platform Stack

Model —instance_of→ 1 Metamodel —+describes→ Platform 1..n

1..n   0..1

Standards:
- UML
- standardized
  UML profiles
- XMI
- Programming
  Languages

Standard:
- MOF

Standards:
- CORBA
- J2EE
- .NET
- Linux
- Windows

+realizedOn   1..n

PlatformRealization

implemented_with

PrimitiveRealization

ComposedRealization

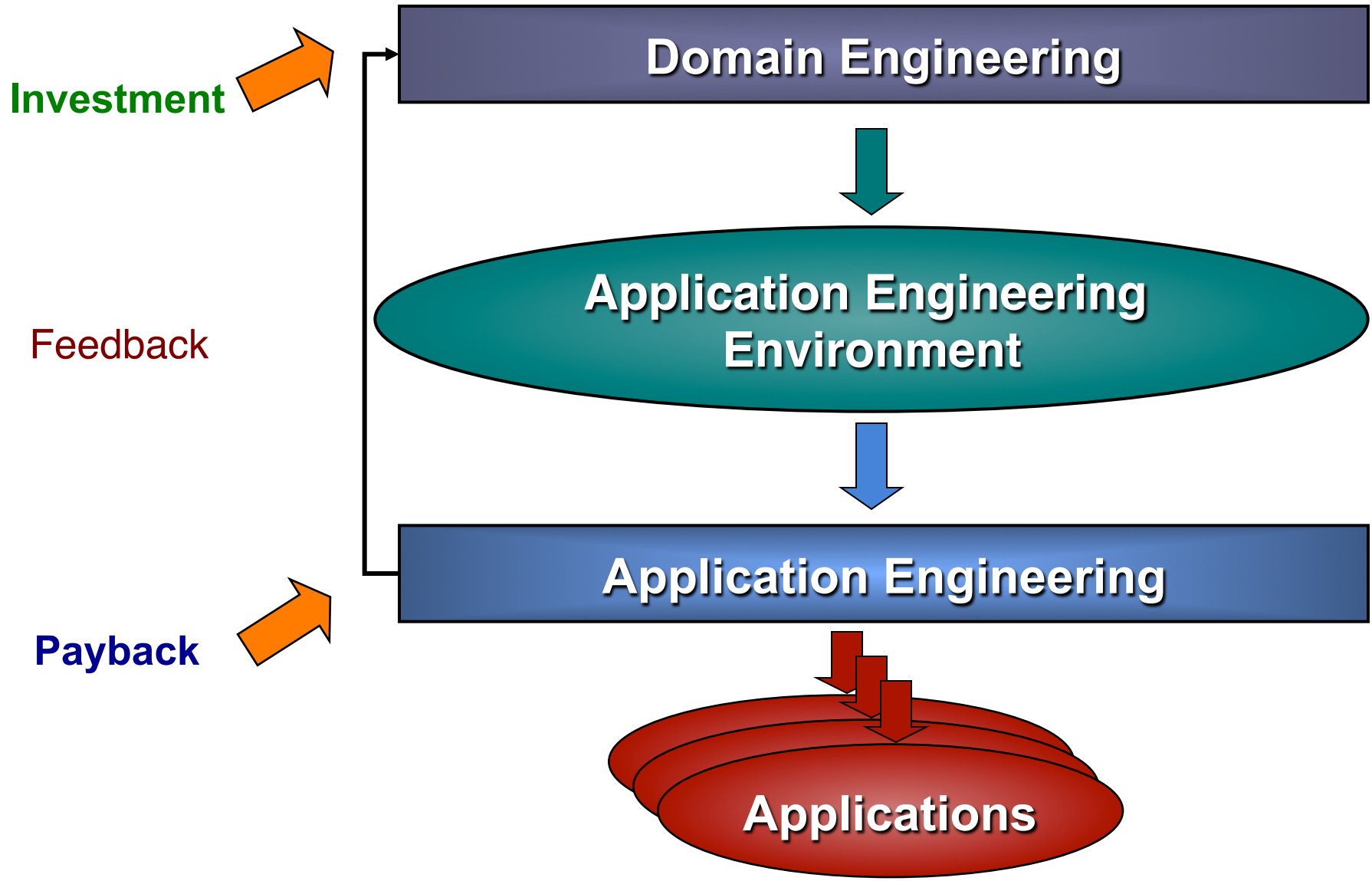# Formal: Mapping Techniques

Annotations for Specific Mapping Techniques

"Analysis" is to "Design" as "Domain" is to _____ ?
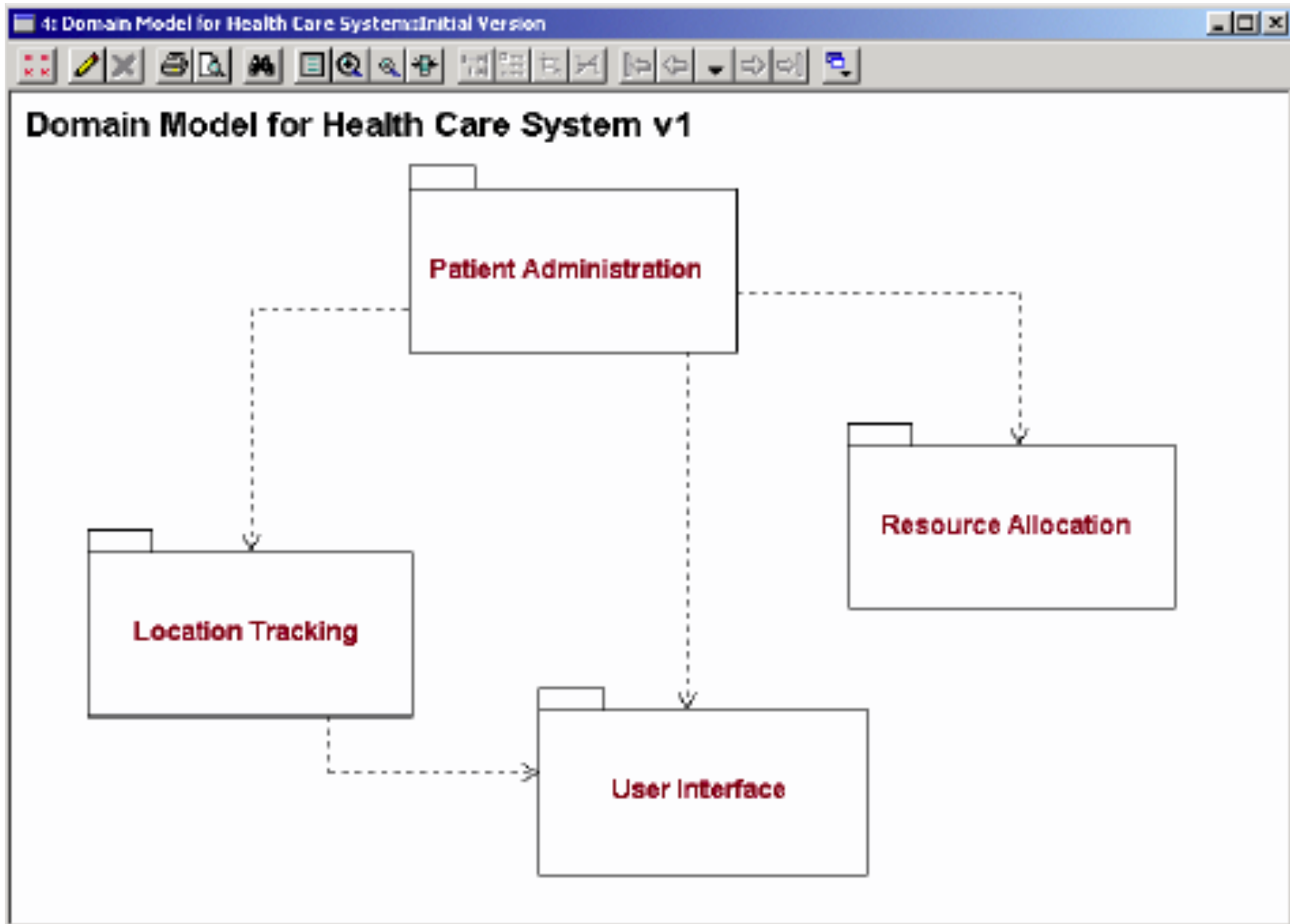
- **Again, think for 15 seconds…**
- **Let's talk…**

Where is Domain Engineering?

Investment

Feedback

Payback

Domain Engineering

Application Engineering Environment

Application Engineering

Applications

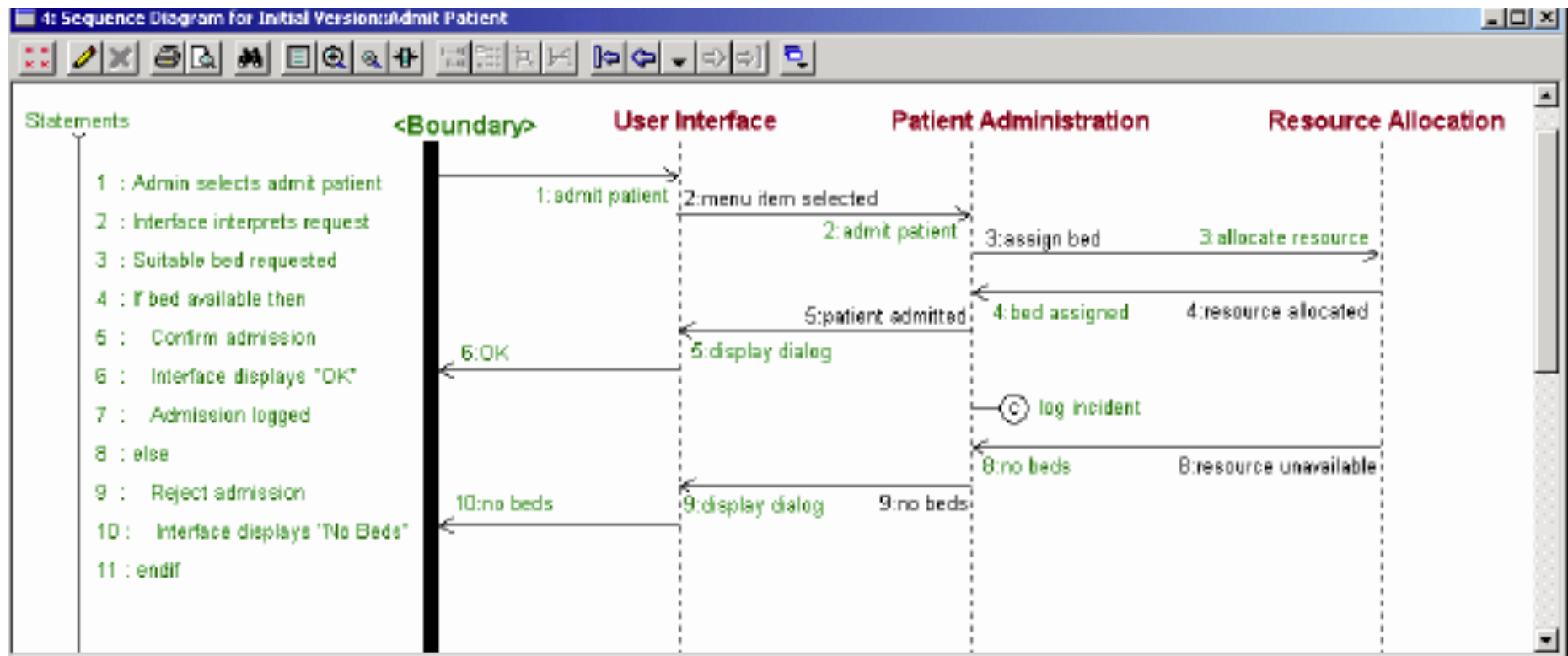# Example: Domain Context

Example: Interactions in Domain

Homework and Milestone Reminders

- Read Chapter 7 on MDSD Capable Target Architectures