CSSE 490 Model-Based Software Engineering: Model-Driven Architecture/ Development

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685

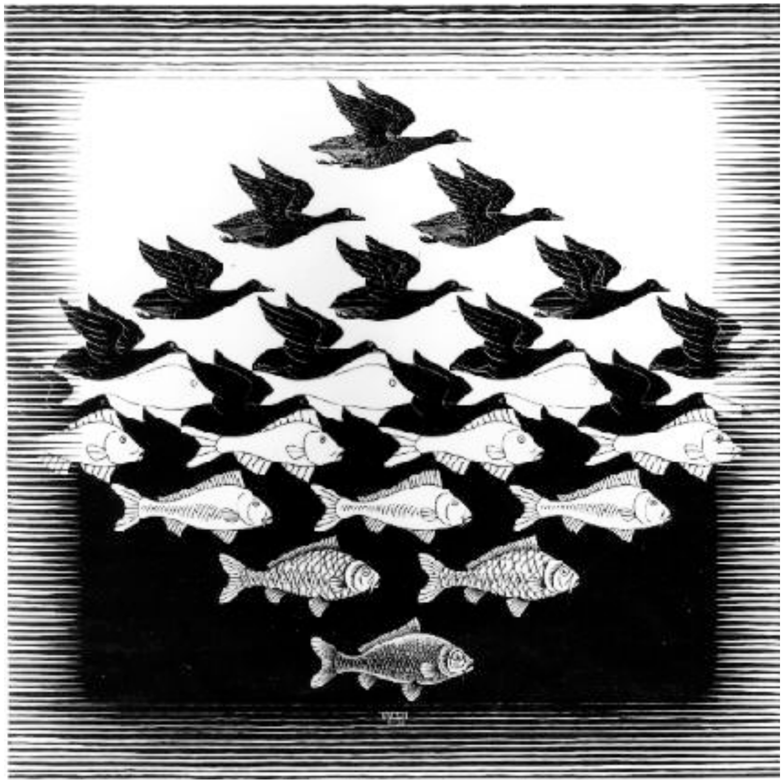
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Learning Outcomes: Transformations

Define transformation rules for abstraction and refinement.

- Discuss paper
- Outline Model-Driven Architecture
- Explore Mappings with MDA Example



Paper Discussion: Feature-Based Transformation Approach Paper

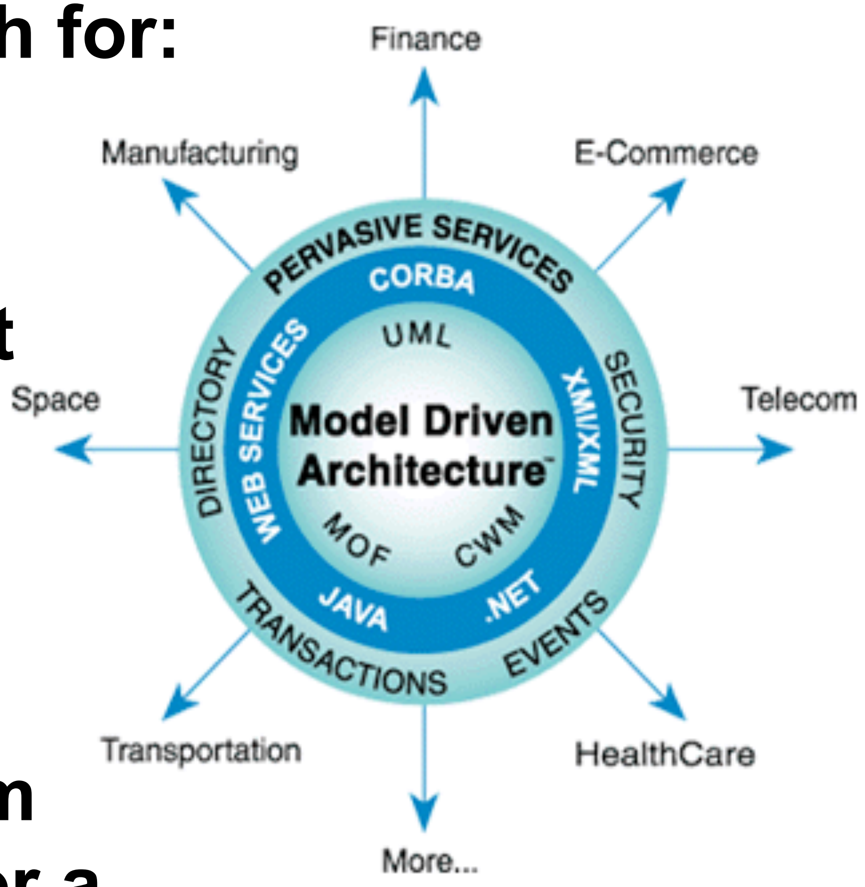
Feature-based survey of model transformation approaches

- What are the main thrusts of the paper?
- What are the controversial points and your positions?
- What did you get out of reading about feature-based transformation approaches?

Overview Model-Driven Architecture

MDA provides an approach for:

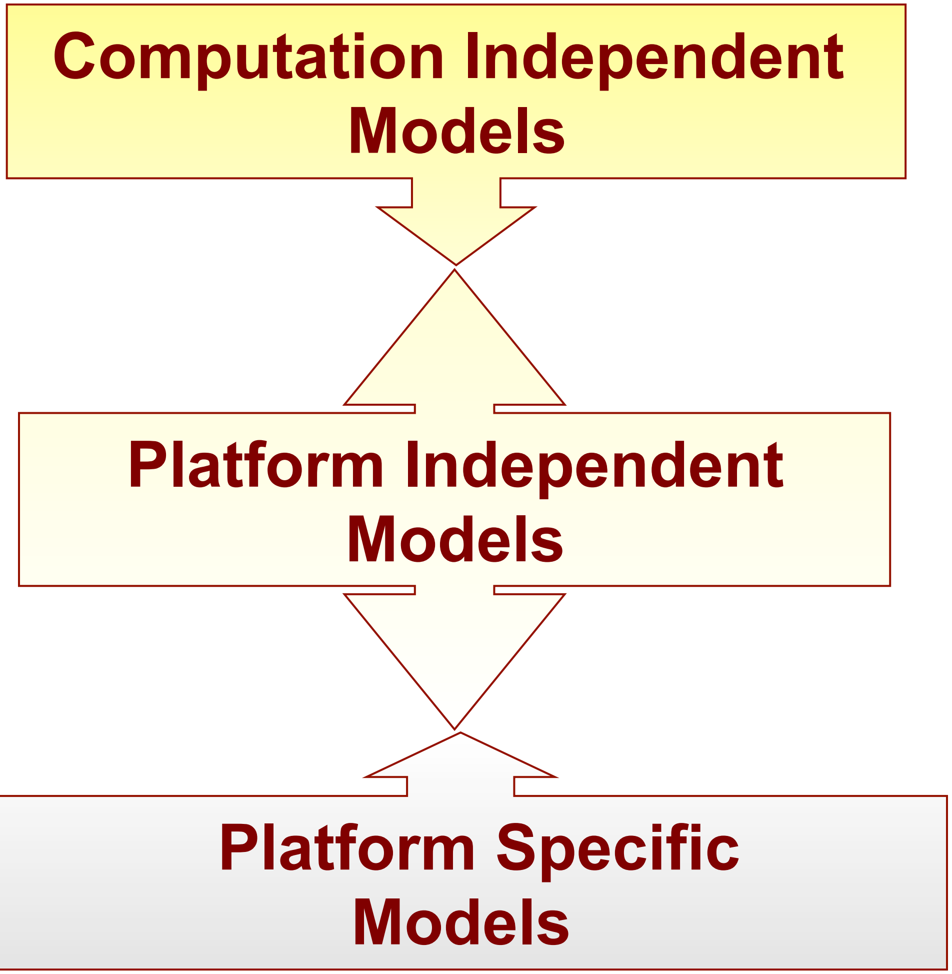
1. Specifying a system independently of the platform that supports it
2. Specifying platforms
3. Choosing a particular platform for the system
4. Transforming the system specification into one for a particular platform



Q1

Recall: Basic Model Layers

- One or model levels per layer
- Mappings and Transforms between models
  - □ Mappings
  - □ Transforms



Computation Independent Models

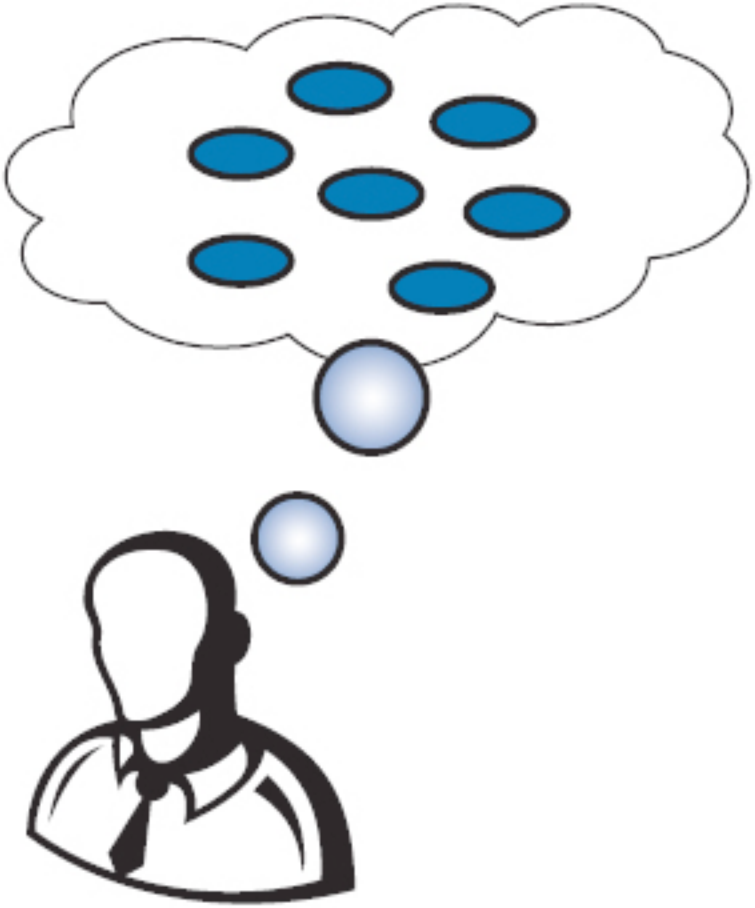
Platform Independent Models

Platform Specific Models

Concepts: Model-Driven

MDA is model-driven because it provides a means for using models to direct the course of:

- Understanding
- Design
- Construction
- Deployment
- Operation
- Maintenance/evolution!

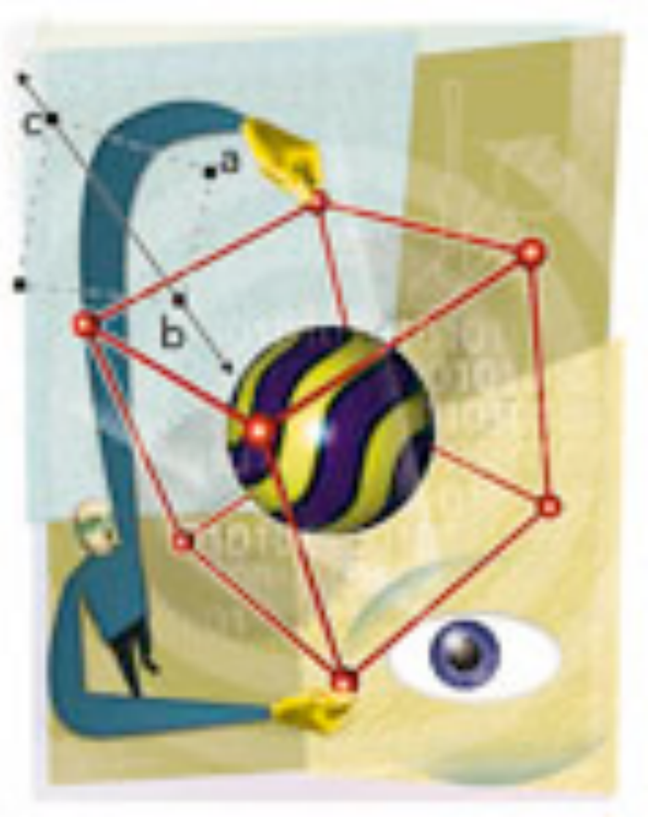


Q2

Concepts: Architecture

- A specification of the components and connectors of the system and rules for interactions between parts using the connectors

- MDA prescribes models to be used, how those models may be prepared and the relationships of the different kinds of models



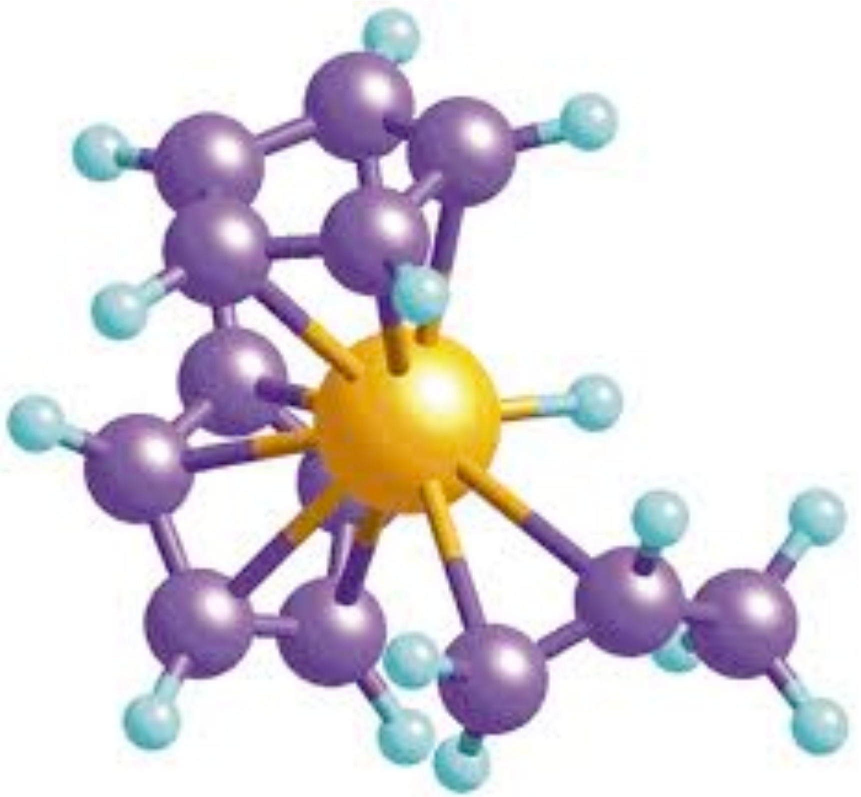
Concepts: Viewpoint

- An abstraction using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within that system
- Concepts and rules may be considered to form a viewpoint language

Q4

Examples of Viewpoint

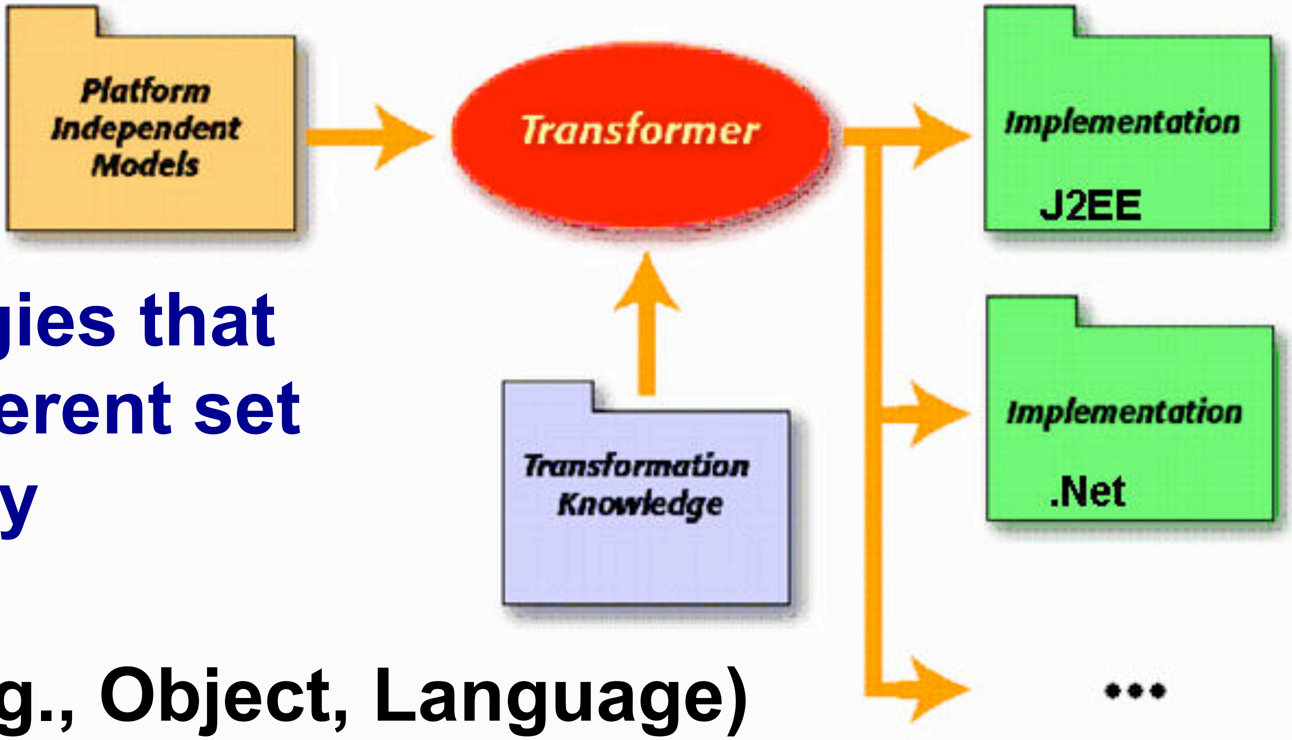
Conceptual viewpoint, a specification (logical) viewpoint, and an implementation (physical) viewpoint, specifying in detail the construction of that system



- Computation Independent Viewpoint
- Platform Independent Viewpoint
- Platform Specific Viewpoint

Concepts: Platform

A set of subsystems and technologies that provide a coherent set of functionality

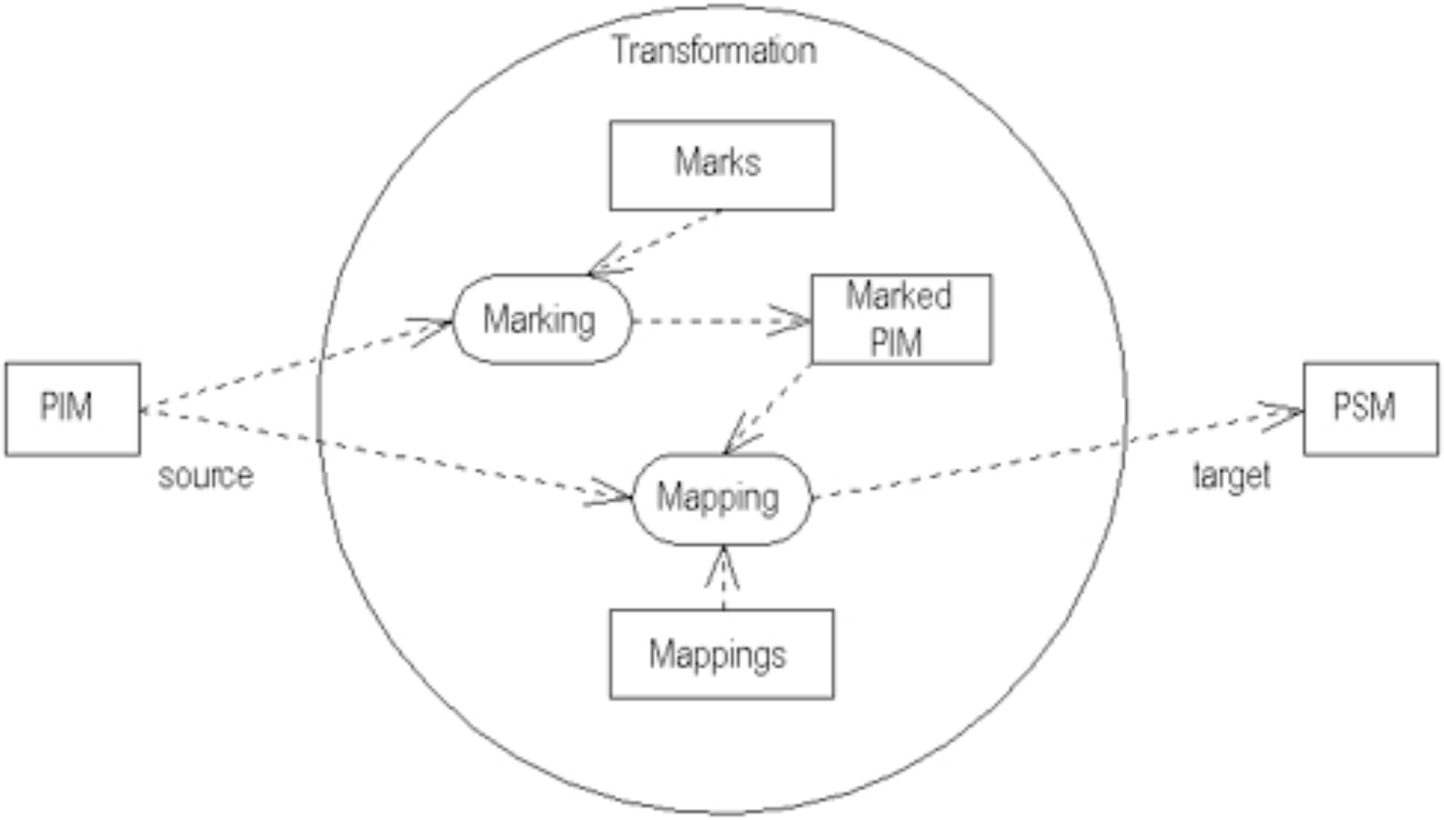


1. Generic (e.g., Object, Language)
2. Specific (e.g., CORBA)
3. Vendor (e.g., .Net)

Mapping

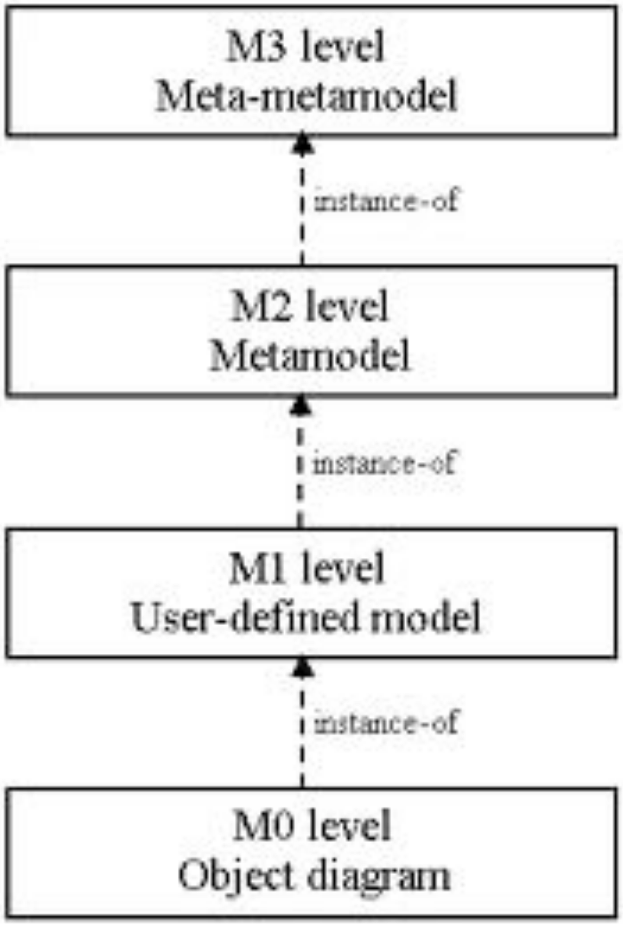
MDA mapping provides specifications for transformation of a PIM into a PSM for a platform

- **Platform model determines the nature of the mapping**



Model Type Mappings: Metamodel Mappings

- A metamodel mapping is a specific example of a model type mapping, where the types of model elements in the PIM and the PSM are both specified as metamodels



Since we are talking meta…

Example of Mapping

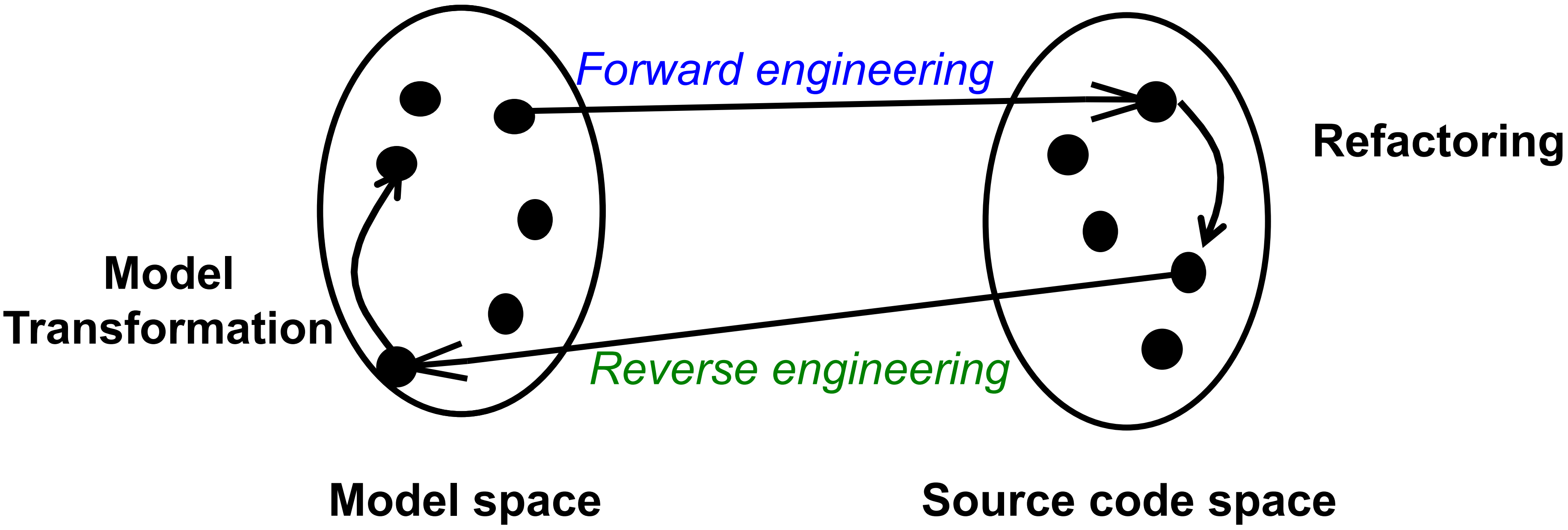
A platform model for EJB includes the Home and RemoteInterface as well as Bean classes and Container Managed Persistence

- A UML PIM to EJB mapping provides marks to be used to guide the PIM to PSM transformation
  - It may also include templates or patterns for code generation and for configuration of a server
- Marking a UML class with the Session mark results in the transformation of that class according to the mapping into a session bean and other supporting classes

What are some ways that you map software models to code?

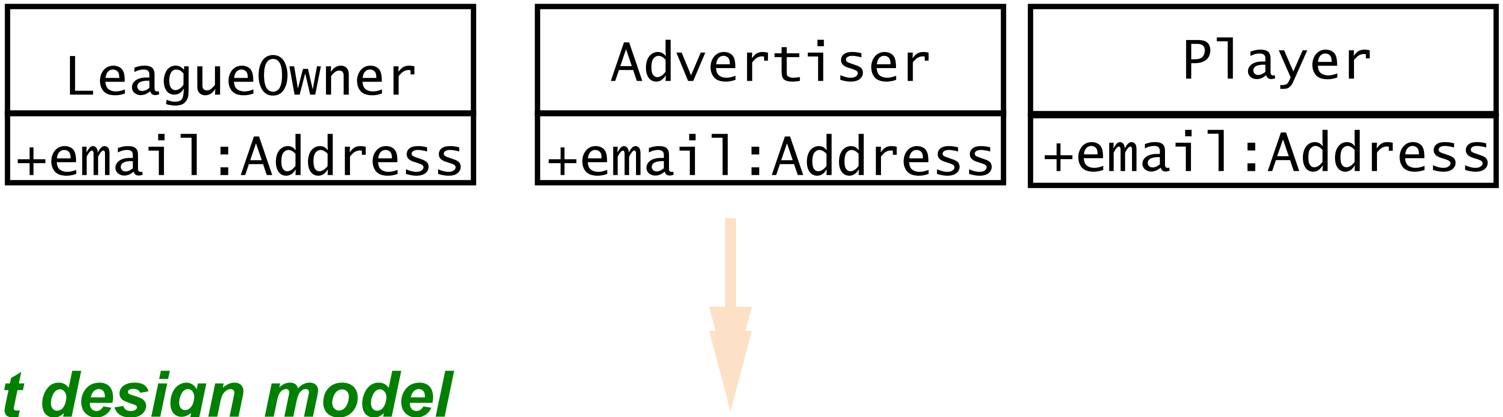
- Think for a minute…
- Turn to a neighbor
and discuss it for a minute

Model Transformations



Model Transformation Example

Object design model before transformation



LeagueOwner

+email:Address

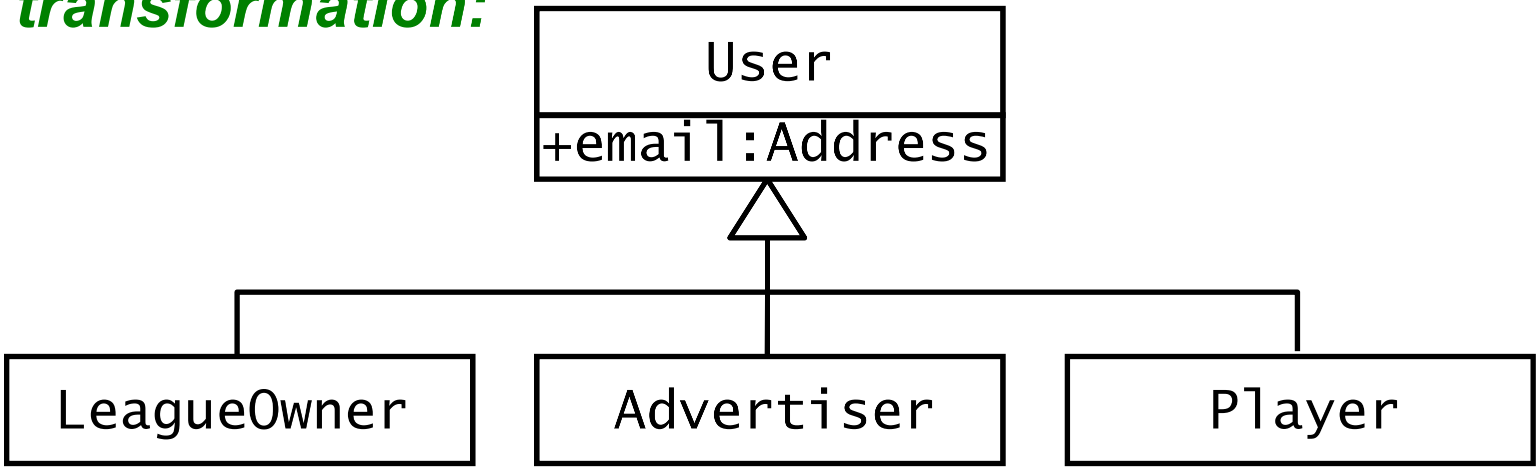
Advertiser

+email:Address

Player

+email:Address

Object design model after transformation:



User

+email:Address

LeagueOwner

Advertiser

Player

Refactoring Example: Pull Up Field

```
public class Player {
   private String email;
   //...
}
public class LeagueOwner {
   private String eMail;
   //...
}
public class Advertiser {
   private String email_address;
   //...
}
```

```
public class User {
   private String email;
}

public class Player extends
   User {
   //...

}
public class LeagueOwner
   extends User {
   //...
}
public class Advertiser extends
   User {
   //...
}
```

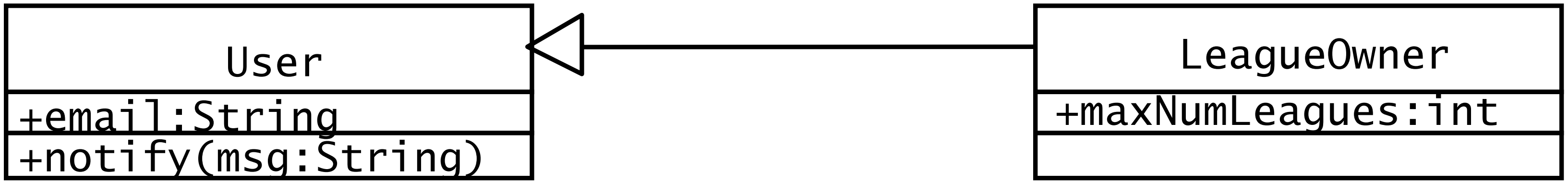
Refactoring Example:
Pull Up Constructor Body

```
public class User {
    private String email;
}


public class Player extends User {
    public Player(String email) {
            this.email = email;
    }
}
public class LeagueOwner extends
    User{
    public LeagueOwner(String email) {
            this.email = email;
    }
}

public class Advertiser extendsUser{
    public Advertiser(String email) {
            this.email = email;
    }
}
```

```
public class User {
    public User(String email) {
            this.email = email;
    }
}
public class Player extends User {
    public Player(String email) {
            super(email);
    }
}
public class LeagueOwner extends User
    {
    public LeagueOwner(String email) {
            super(email);
    }
}
public class Advertiser extends User {
    public Advertiser(String email) {
            super(email);
    }
}
```

PIM to PSM Example

Object design model before transformation



User

+email:String

+notify(msg:String)

LeagueOwner

+maxNumLeagues:int

Source code after transformation

```
public class User {
    private String email;
    public String getEmail() {
          return email;
    }
    public void setEmail(String value){
          email = value;
    }
    public void notify(String msg) {
          // ....
    }
    /* Other methods omitted */
}
```

```
public class LeagueOwner extends User {
    private int maxNumLeagues;
    public int getMaxNumLeagues() {
          return maxNumLeagues;
    }
    public void setMaxNumLeagues
                    (int value) {
          maxNumLeagues = value;
    }
    /* Other methods omitted */
}
```

Other Mapping Activities

- Optimizing the Object Design Model
- Mapping Associations
- Mapping Contracts to Exceptions
- Mapping Object Models to Tables

Homework and Milestone Reminders

- Read Metamodel Paper (will be on Angel tonight)
- Milestone 2: Establish a repository and structure for assembling components for your FacePamphlet application
  - Due by 11:55pm Friday, April 1st, 2011 (no foolin'!)
- Continue to familiarize yourself with material on Eclipse Modeling Project

  http://www.eclipse.org/modeling/

Tomorrow we will talk more about Milestone 2