CSSE 490 Model-Based Software Engineering: More MBSD

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

Learning Outcomes: MBE Discipline

Relate Model-Based Engineering as an engineering discipline.

- Software development
- More on transition from traditional to model-based development
- Elements of MBSE
- MBSysE (if time)

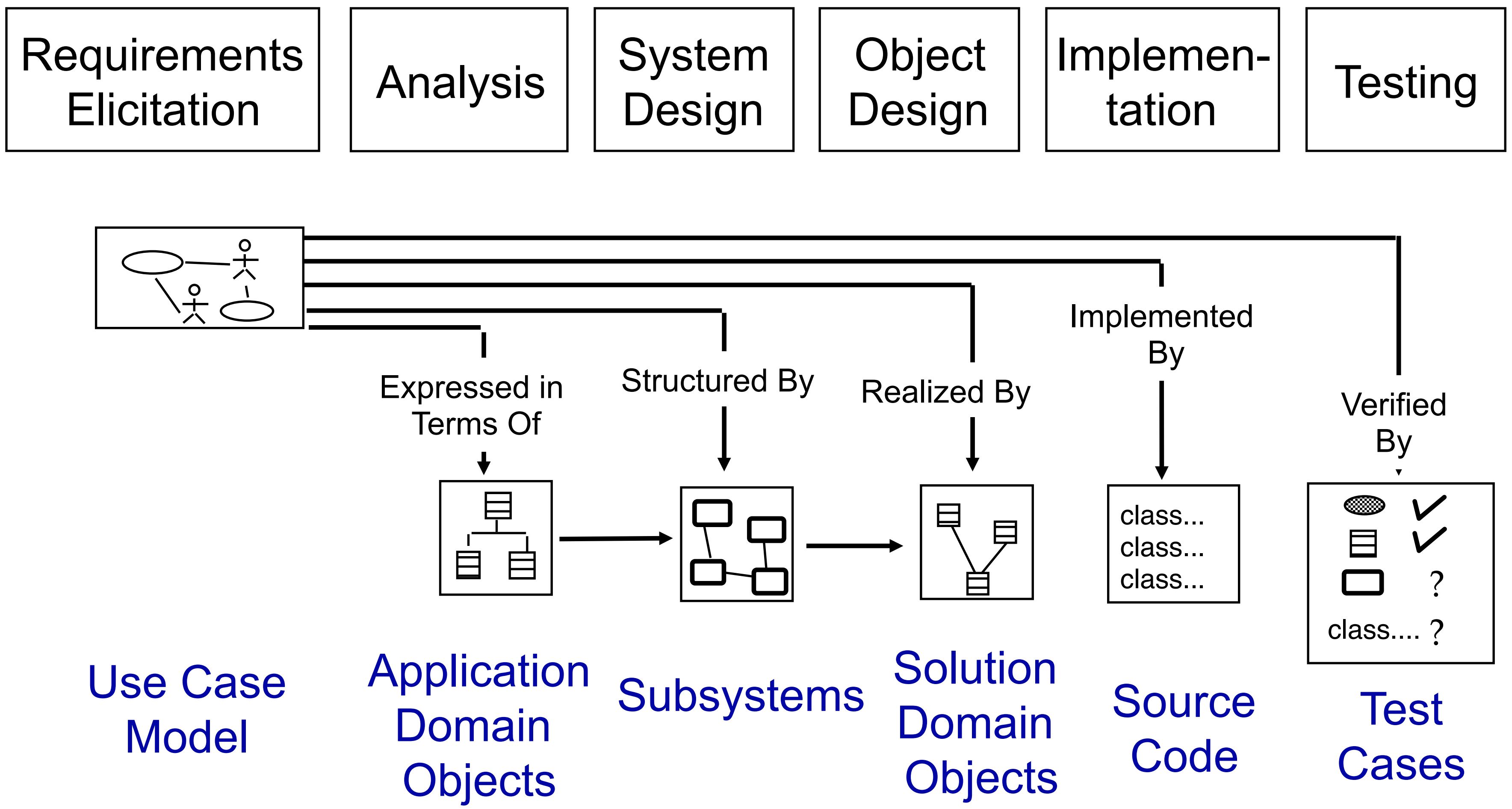
What are some of the most important models you use in developing software?
How do they relate to each other?

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

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Software Lifecycle Activities

...and their models



Q1

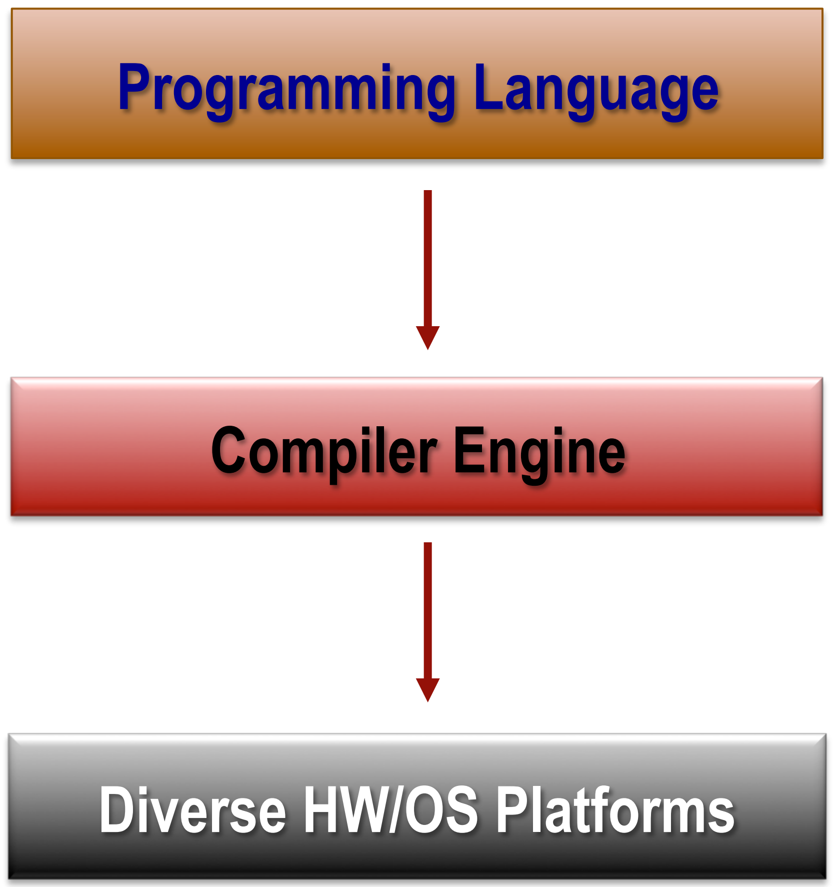
Status Quo of Development

- Integrated Development Environments (IDE)
  - e.g., Eclipse
  - …Rationale Enterprise Edition
  - …Visual Studio…
- Including Modeling tools
  - e.g., UML (e.g., ArgoUML)
  - …Refactoring tools
  - …Design Pattern tools…
- Agile Methods as a Silver Bullet
  - Incrementally faster, with less risk
  - May not play well with other disciplines (e.g. QA)

Level of Automation: Past ~35 Years

Programming IDEs (e.g. Eclipse, Rational, Visual Studio)

Platfom independent model (PIM)



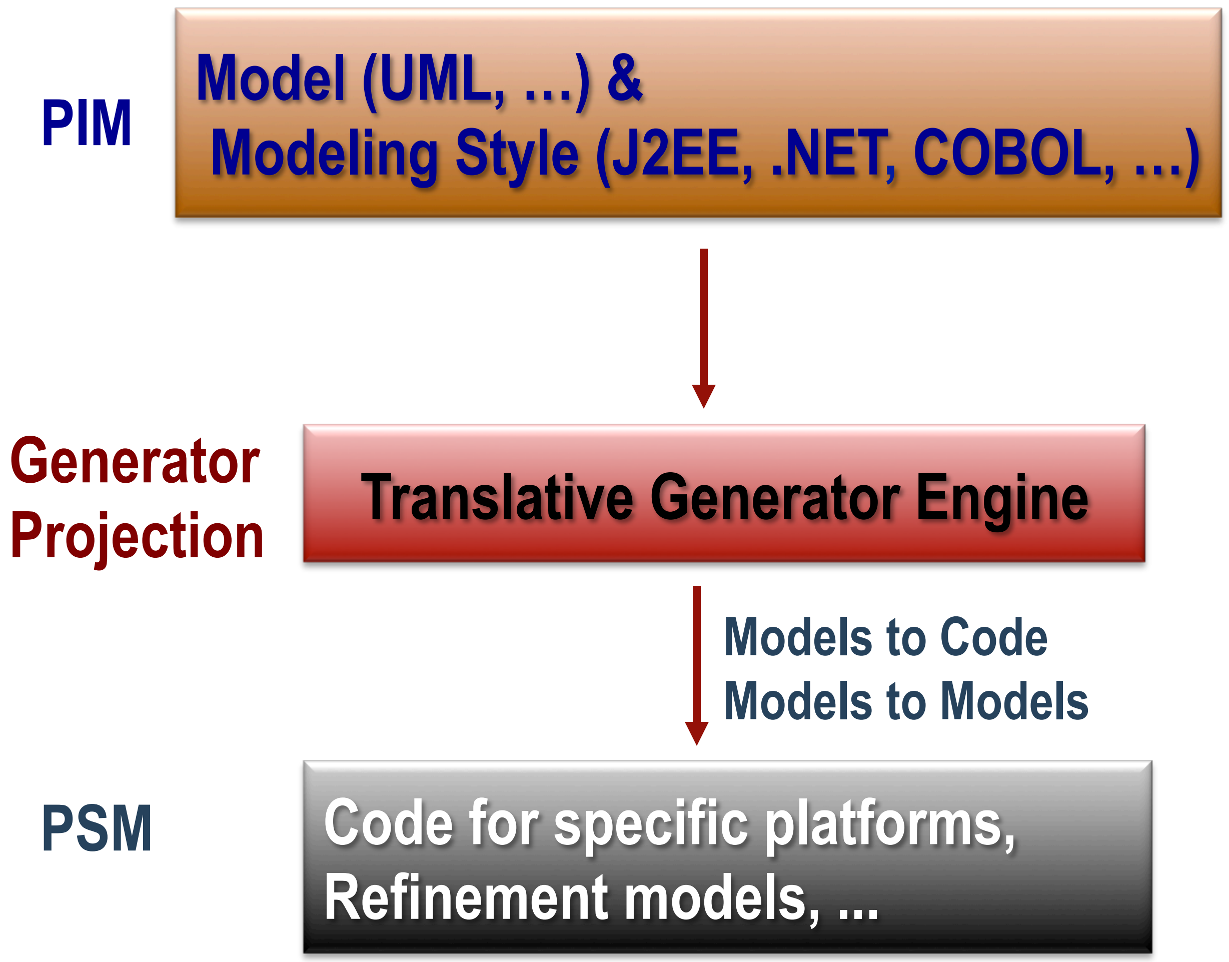
- Higher level of expression
- Easier to understand
- Portable
- Standardized

- Dependable
- Flexible
- Configurable
- Optimizing
- Complete: Linker, Debugger, etc.

Platfom specific model (PSM)

Levels of Automation: MBSE

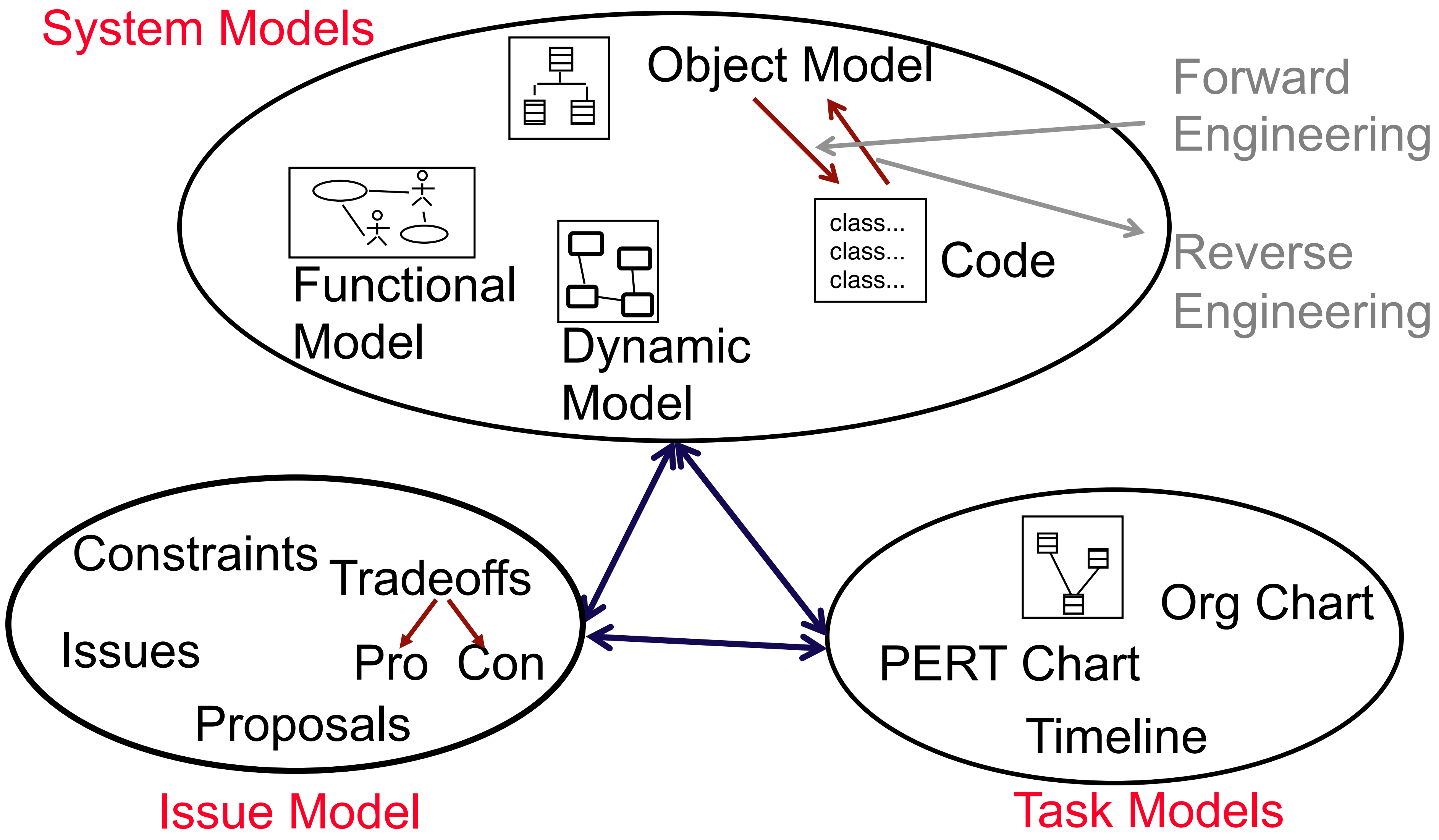
Architectural IDEs (e.g. ArcStyler)


PIM
Model (UML, ...) &
Modeling Style (J2EE, .NET, COBOL, ...)
Generator
Projection
Translative Generator Engine
Models to Code
Models to Models
PSM
Code for specific platforms,
Refinement models, ...

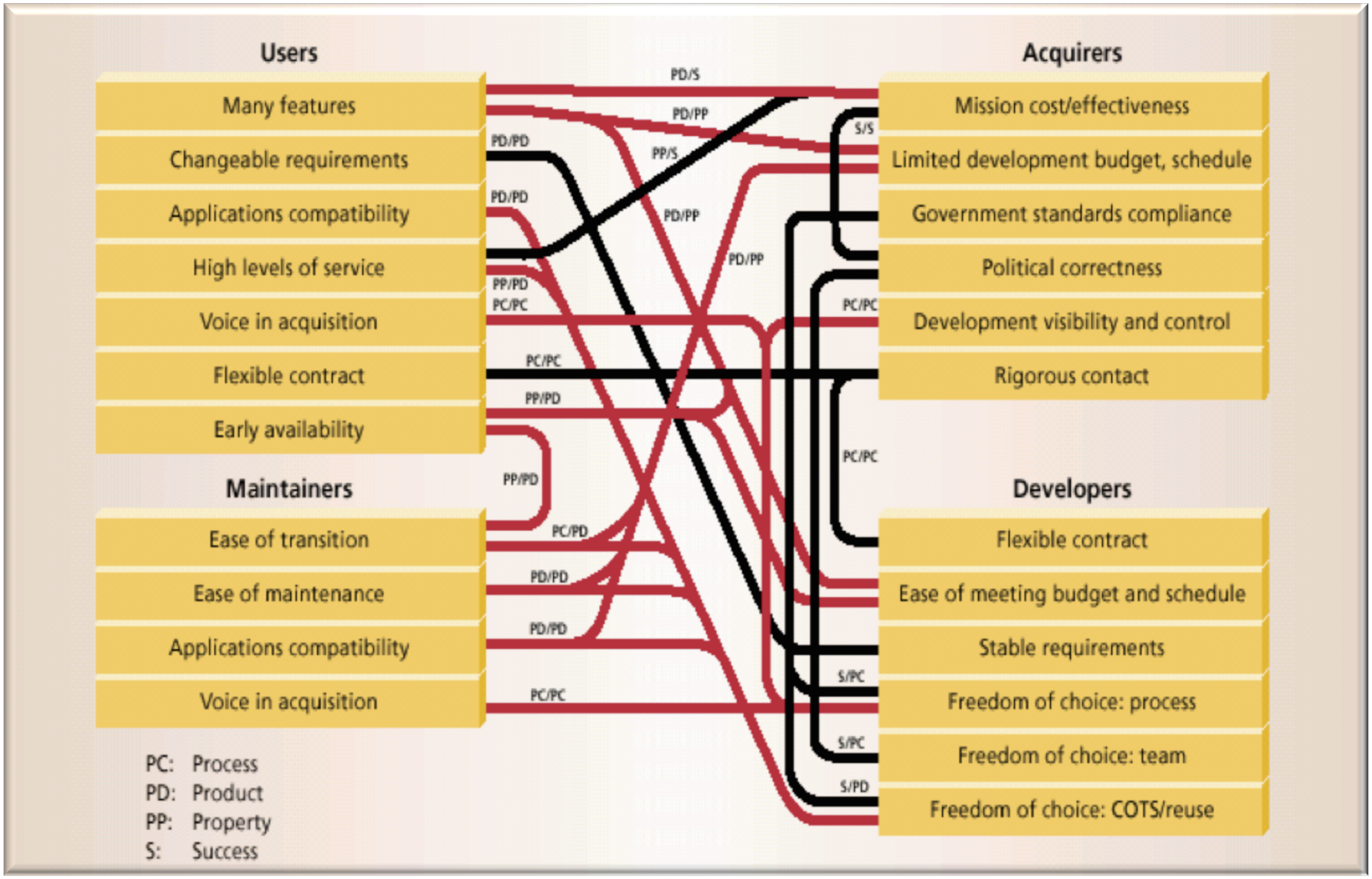
- Higher level of expression
- Easier to understand
- Portable
- Standardized

- Dependable
- Flexible
- Configurable
- Debuggable
- Optimizing
- Complete

Key Information Resides in Models

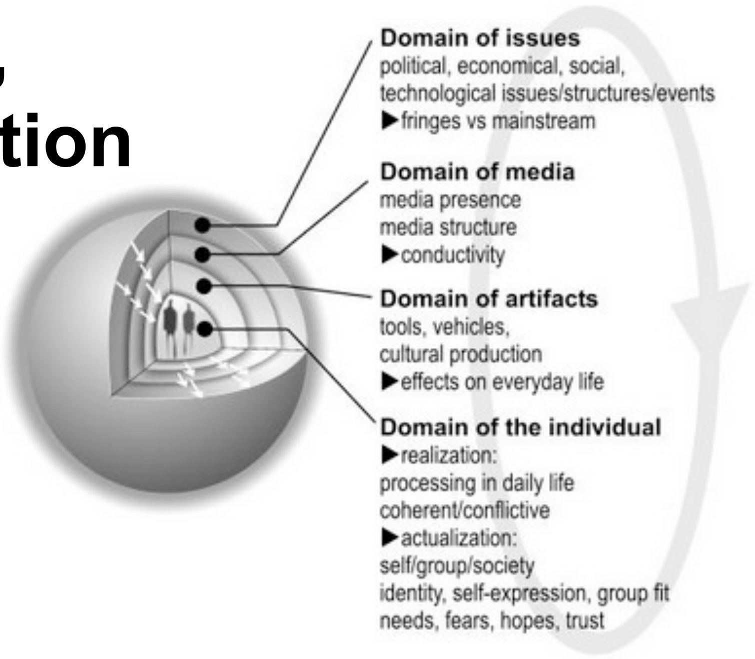


Model Clashes



MBSE Starting Point: Domain

- Domain engineering is the starting point in most cases
- Establish the key concepts, actors, objects, and interaction in the domain
  - Determine what parts done by a computer, people, roles, mechanisms, …
  - Understand forces of change
- Establish basic vocabulary (later use in DSL)



Domain of issues
political, economical, social, technological issues/structures/events
▶fringes vs mainstream
Domain of media
media presence
media structure
▶conductivity
Domain of artifacts
tools, vehicles, cultural production
▶effects on everyday life
Domain of the individual
▶realization: processing in daily life coherent/conflictive
▶actualization: self/group/society identity, self-expression, group fit needs, fears, hopes, trust

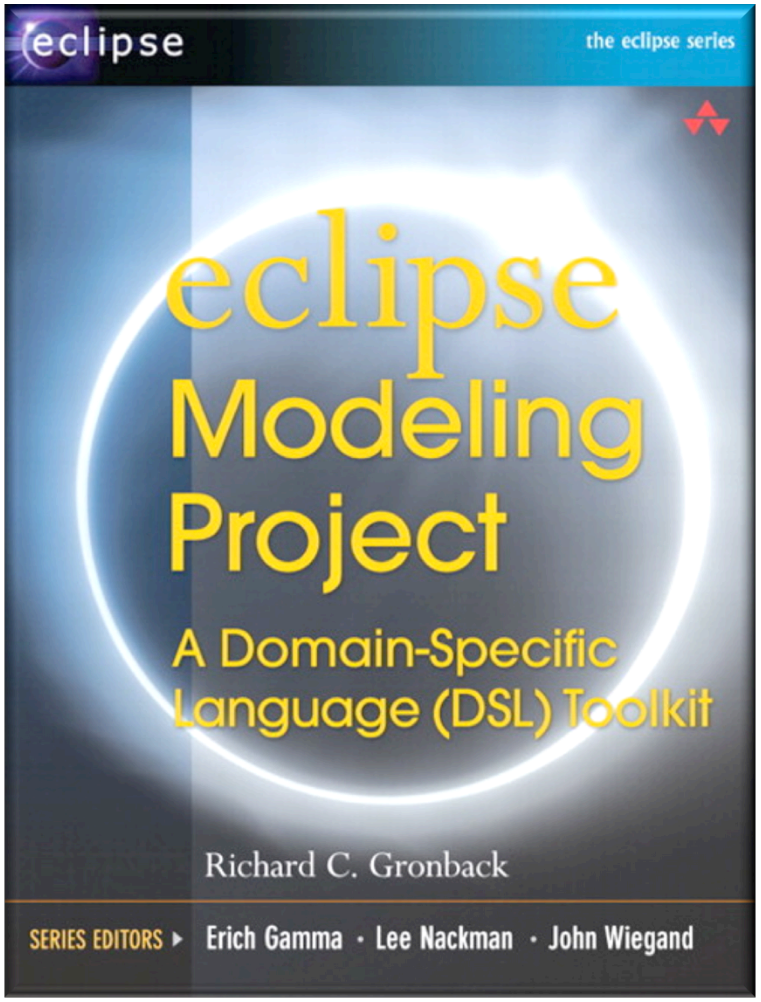
MBSE: Metamodel

- Starting with a clear structure of the domain (an ontology), establish the structure for introducing automation

- Metamodel captures the abstract syntax and the static semantics of a language
  - Abstract syntax (unlike concrete syntax) merely states what the language structure looks like
  - Static semantics determine the well-formedness (e.g., a rule that variables must be declared)

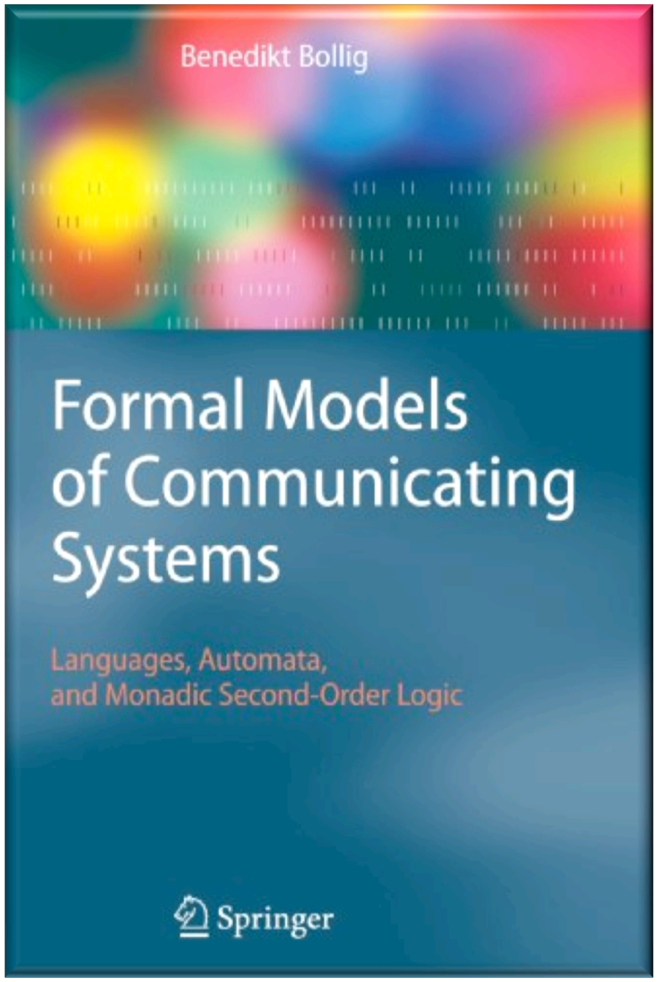
Domain Specific Language

- Specialized to a domain
  - e.g., telecommunications or telephony billing
- Makes a domain's aspects formally expressible and model'able
- Model will sometimes be used synonymously with DSL


eclipse
the eclipse series
eclipse Modeling Project
A Domain-Specific Language (DSL) Toolkit
Richard C. Gronback
SERIES EDITORS ▸ Erich Gamma · Lee Nackman · John Wiegand

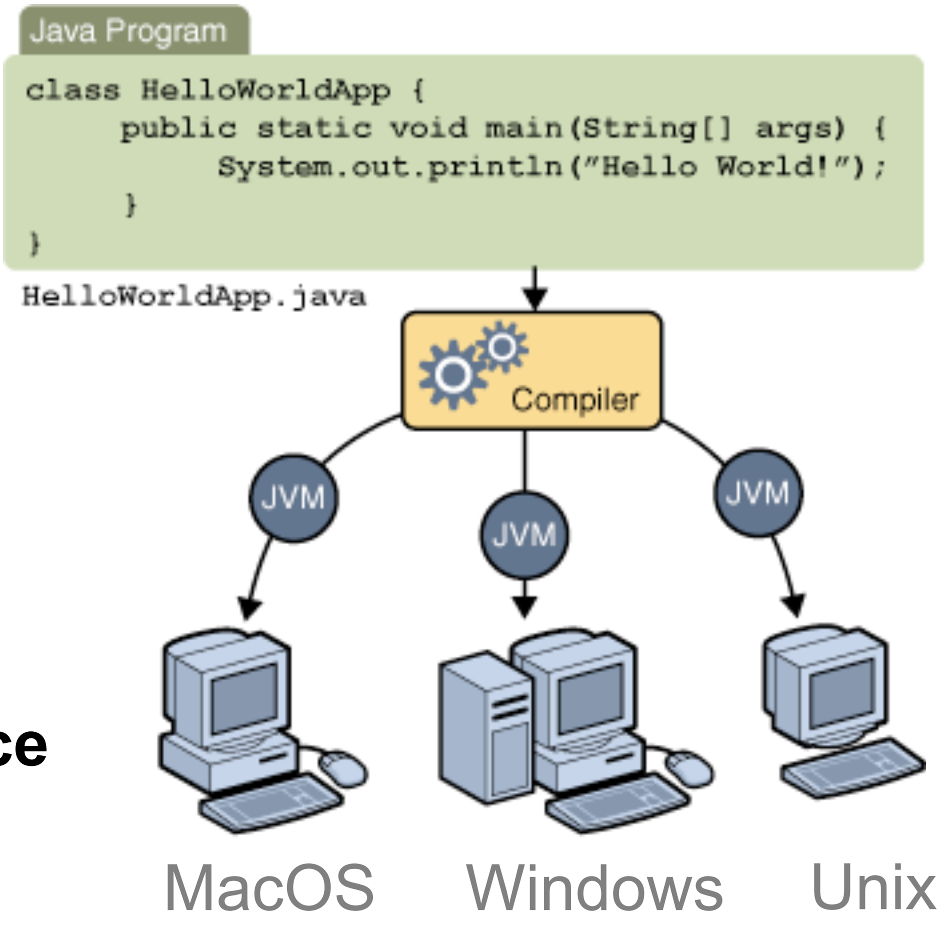
Formal Models

- Starting point for automated transformations
- Formal model needs a DSL
- Mappings
- Transformations
  - Always based on metamodel
  - Model to model
  - Model to platform


Benedikt Bollig
Formal Models of Communicating Systems
Languages, Automata, and Monadic Second-Order Logic
Springer

MBSE: Platform

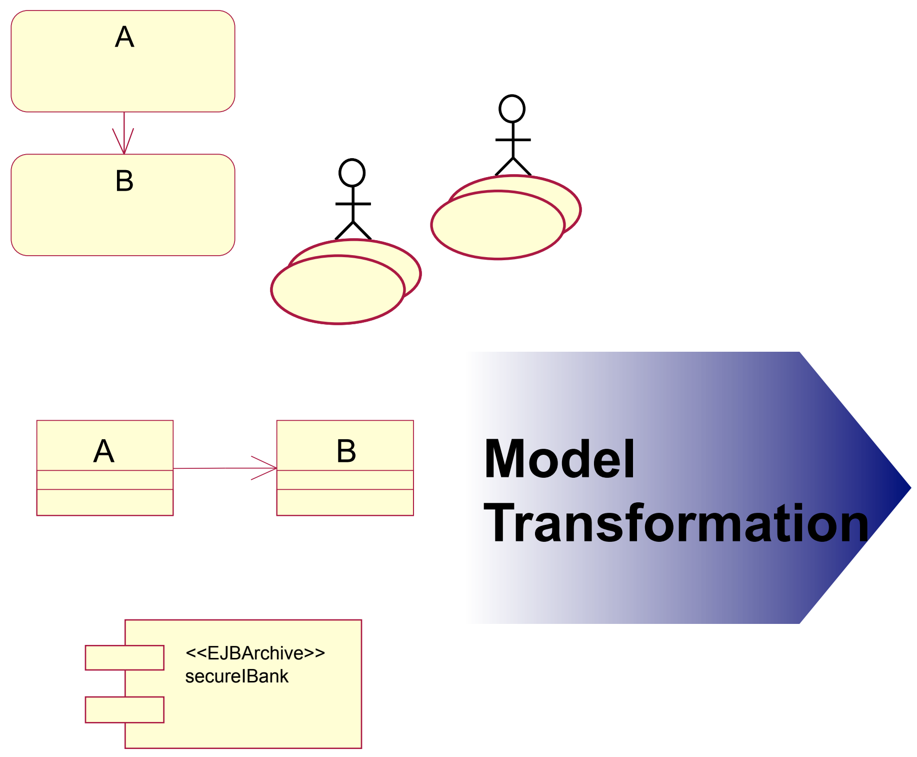
- Target of the system
  - System
  - Hardware
  - Communications
  - Operating System
  - Language
  - Database
  - Graphical User Interface
  - etc.

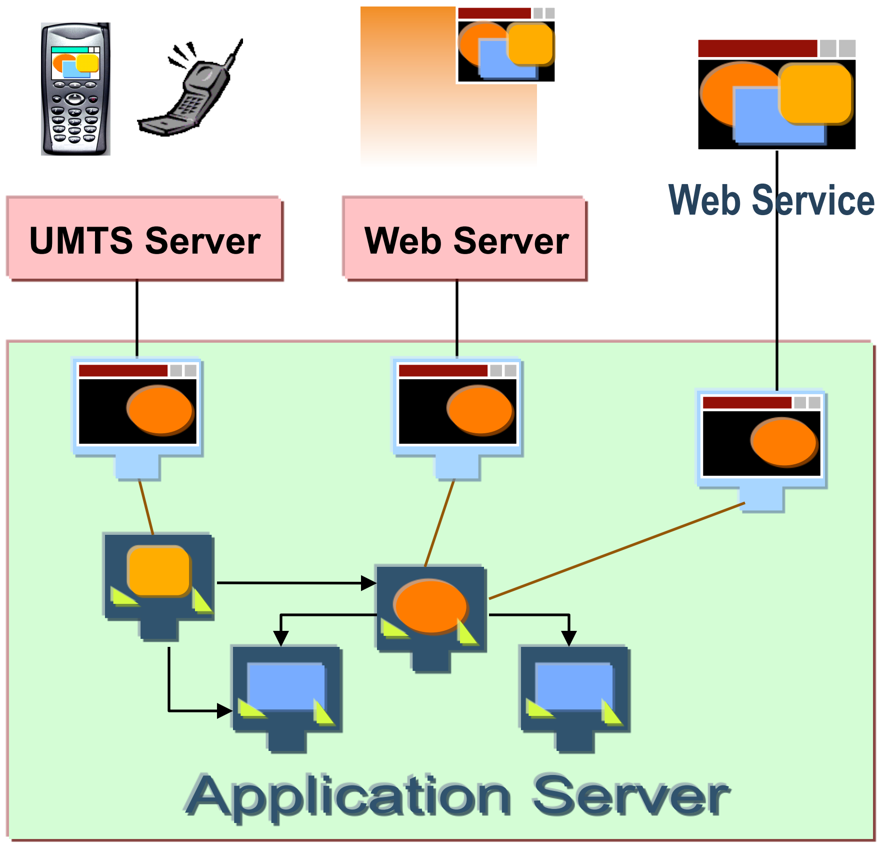


```
class HelloWorldApp {
      public static void main(String[] args) {
            System.out.println("Hello World!");
      }
}
```

MBSE: Car Sharing Example

Domain Modeling



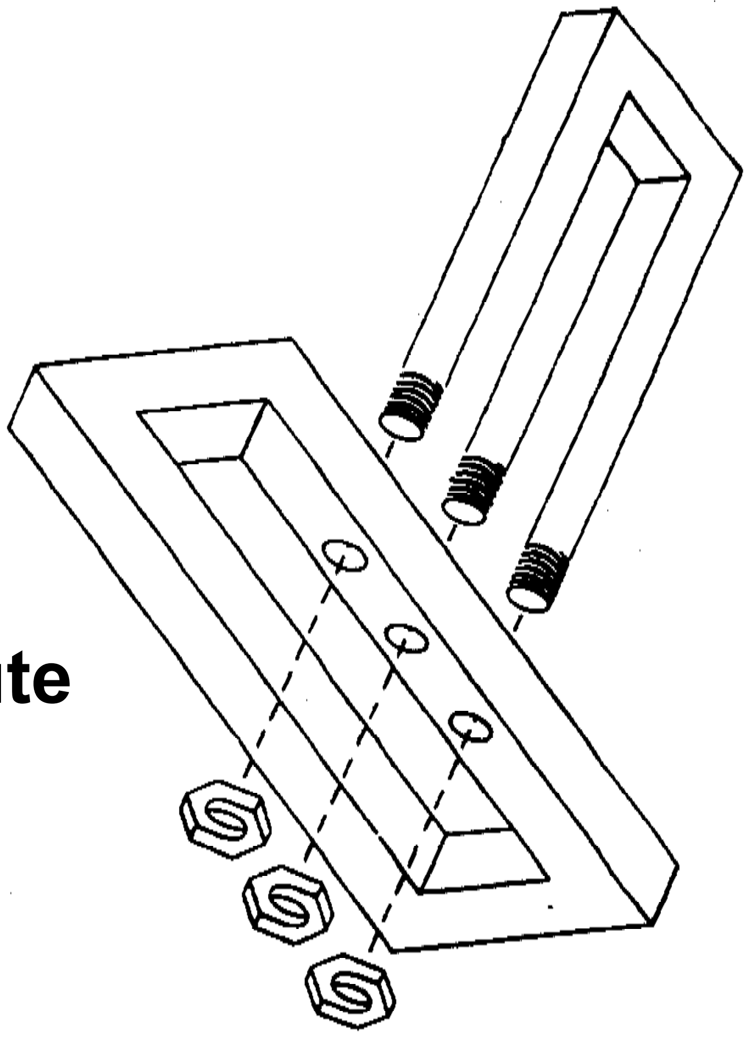


- Domain Object Models
- Use Cases
- ...

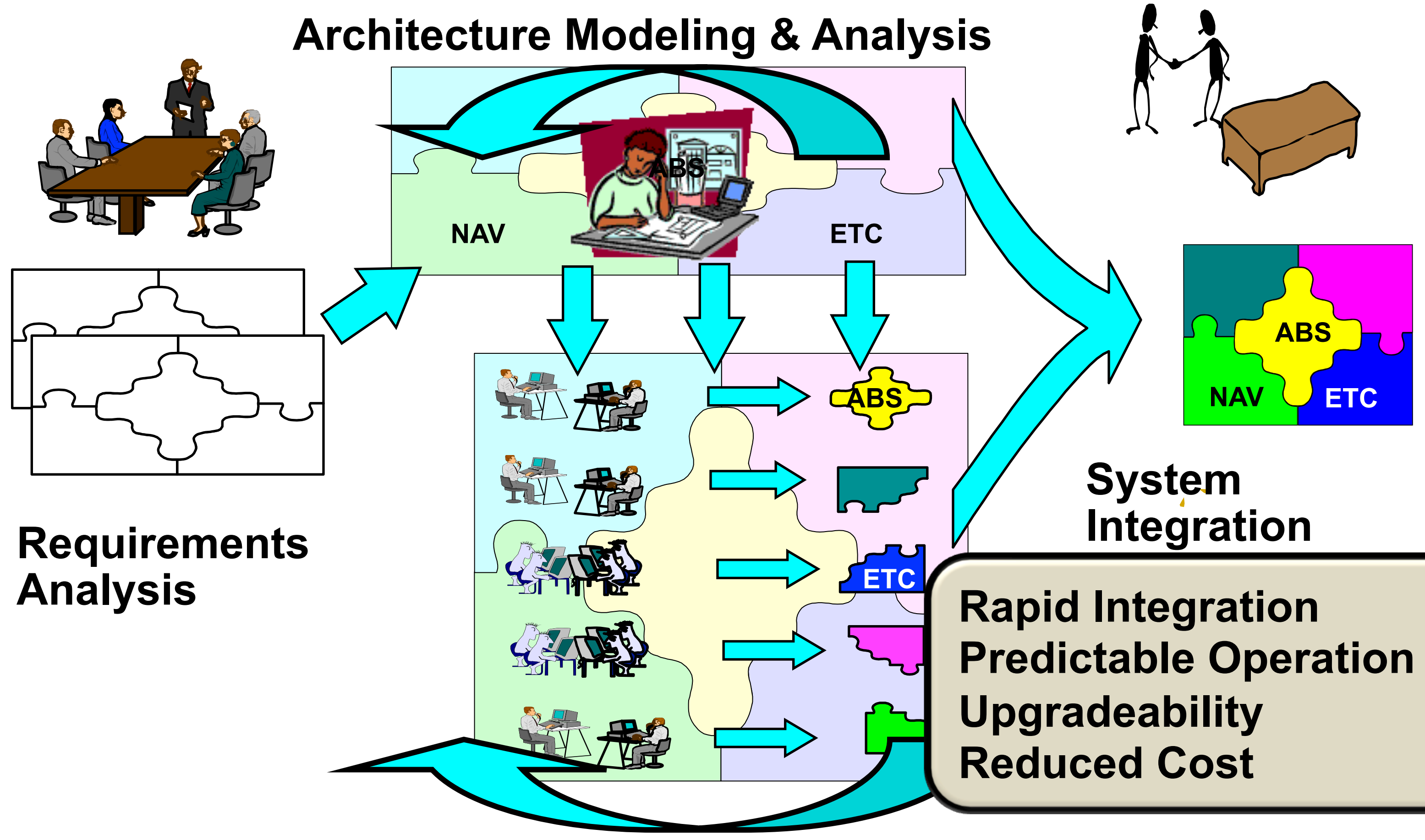
- Deployable Components
- Build & test environment
- ...

This is a model... Can you develop it?

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



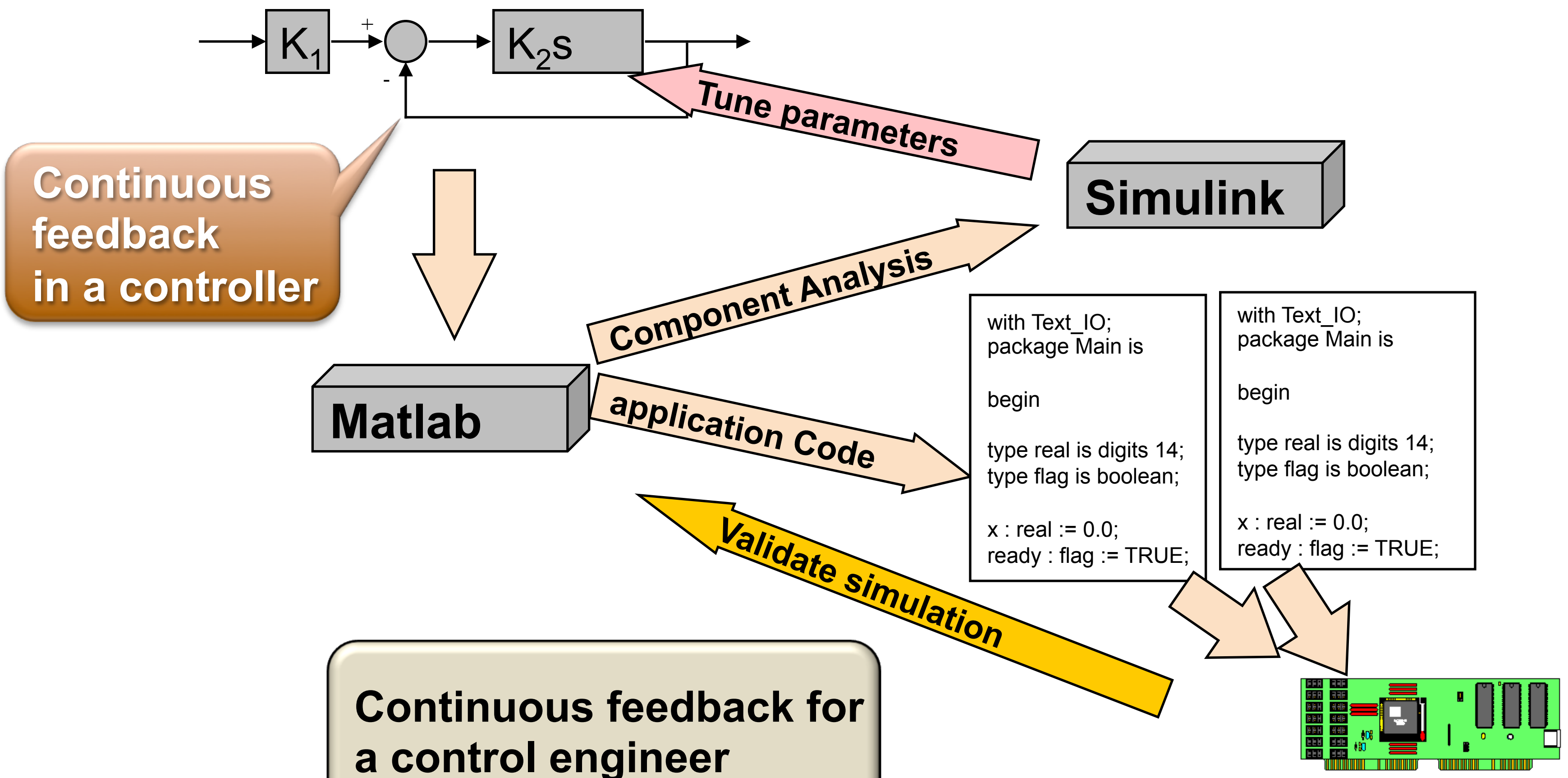
Model-Based System Engineering
(according to Software Engineering Institute)



Rapid Integration
Predictable Operation
Upgradeability
Reduced Cost

Predictive Analysis Early In & Throughout Life Cycle

A Control Engineer Perspective



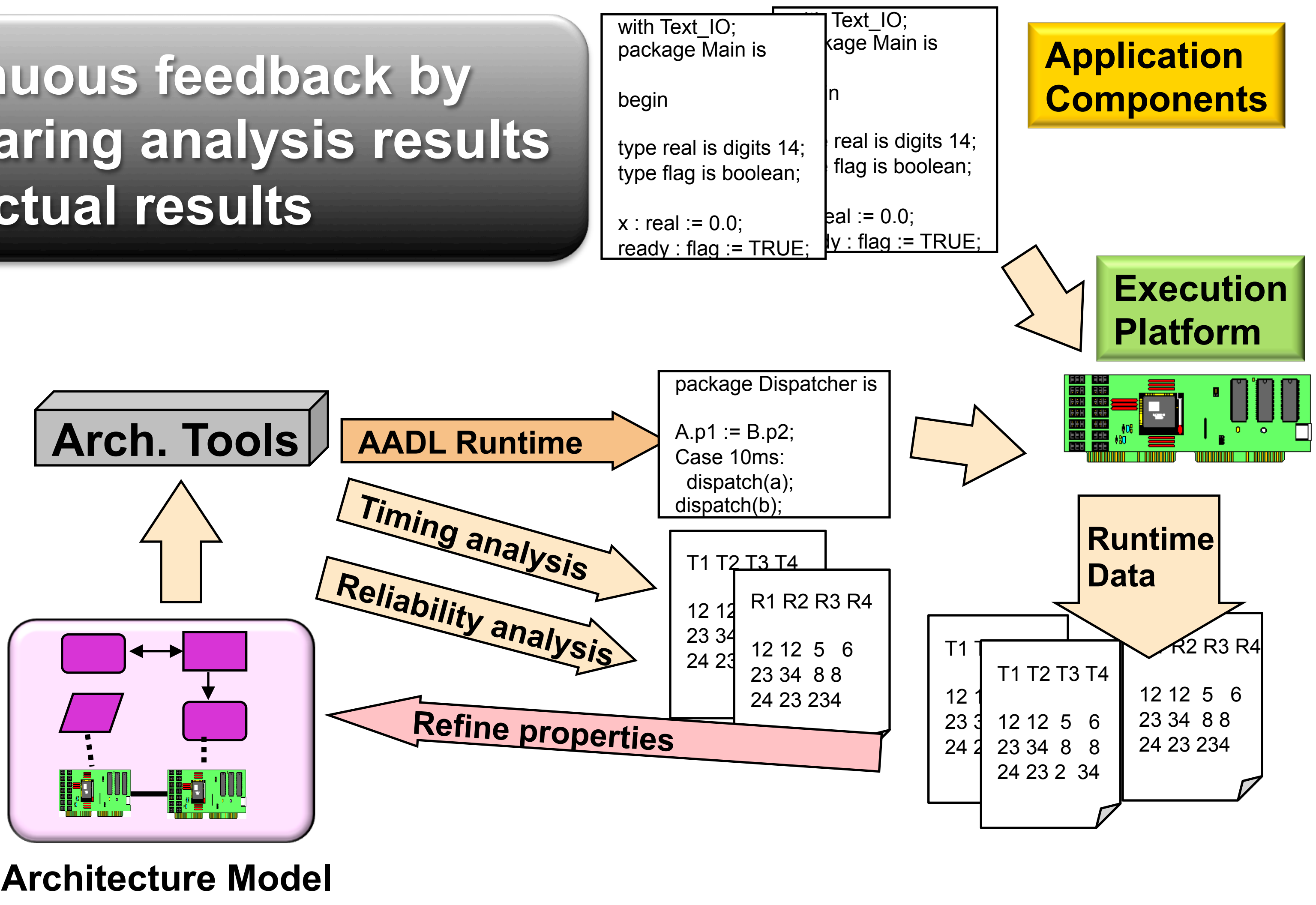
```
with Text_IO;
package Main is

begin

type real is digits 14;
type flag is boolean;

x : real := 0.0;
ready : flag := TRUE;
```

```
with Text_IO;
package Main is

begin

type real is digits 14;
type flag is boolean;

x : real := 0.0;
ready : flag := TRUE;
```

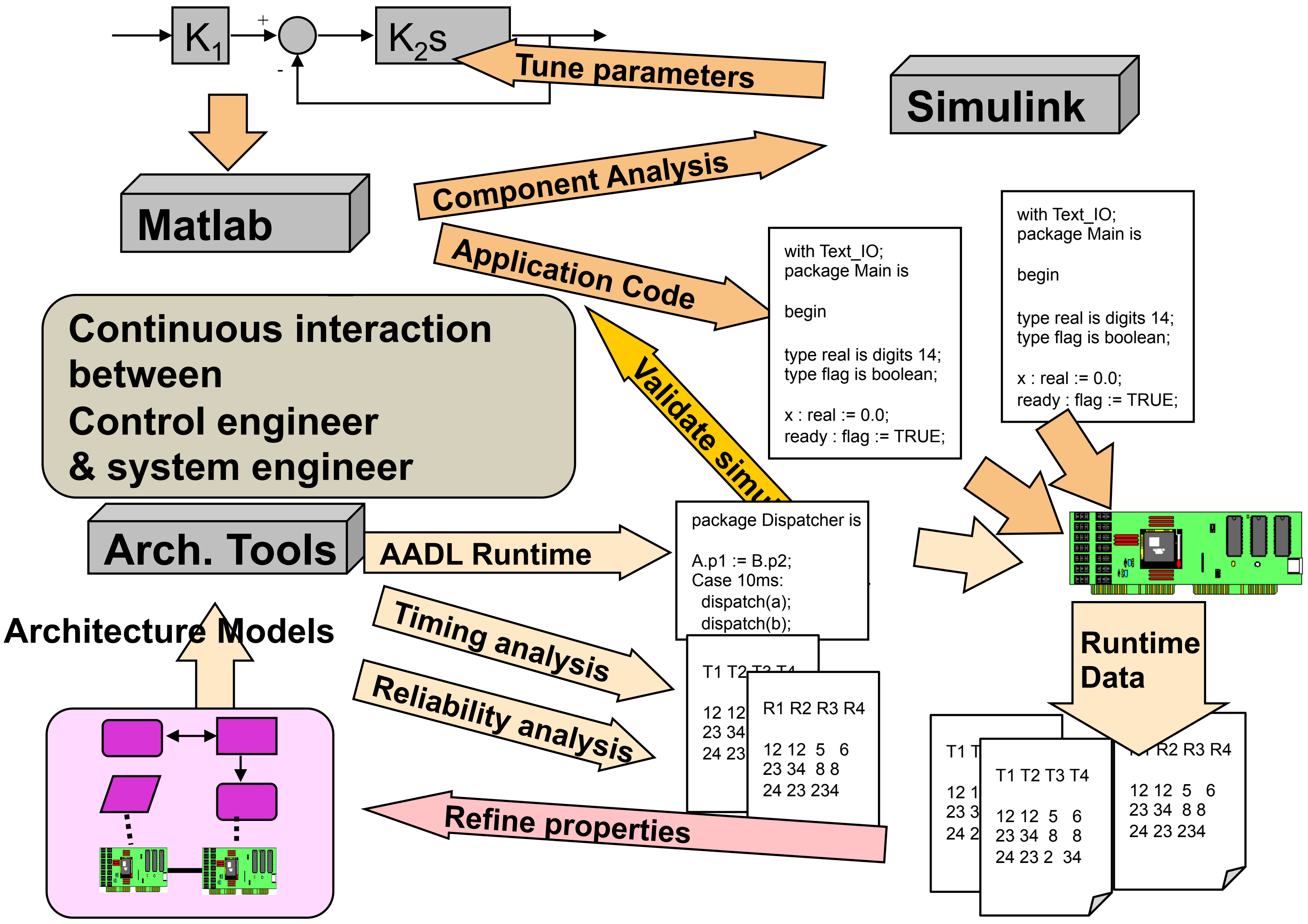
Continuous feedback for a control engineer

Software System Engineer Perspective

Continuous feedback by Comparing analysis results with actual results



A Combined Perspective



```
with Text_IO;
package Main is

begin

type real is digits 14;
type flag is boolean;

x : real := 0.0;
ready : flag := TRUE;
```

```
package Dispatcher is

A.p1 := B.p2;
Case 10ms:
  dispatch(a);
  dispatch(b);
```

Homework and Milestone Reminders

- Continue to Read Chapter 4 of MBSD Text – Concept Formulation
- Let’s talk tomorrow about representation forms