

# Model-Driven Software Engineering (MDSE)

Bahman Zamani, Ph.D.

[bahmanzamani.com](http://bahmanzamani.com)

Computer Engineering Dept.  
University of Isfahan

Presented at  
Sheikhbahaee University  
Baharestan - Isfahan

Feb. 4<sup>th</sup>, 2010 (1388/11/15)

# Model

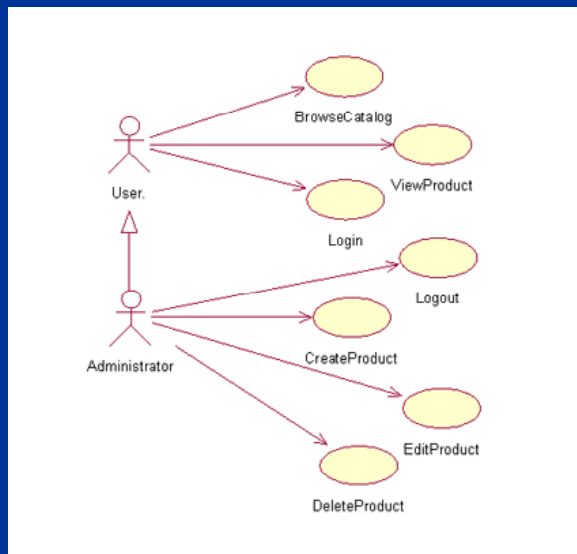
- What is a model?
  - A model is a representation of a system
- Engineering model?
  - A representation of a system that hides some of the properties and highlights the ones that are of interest for the user [Sel 06].

# Model Forms

- Mathematical, eg Linear Programming
- Physical, eg for aircraft or bridge
- Diagrammatic, eg Use case modeling

$$\begin{aligned} &\text{minimize} && \sum_{j=1}^n c_j x_j \\ &\text{subject to} && \sum a_{ij} x_j \leq b_i \quad (i = 1, 2, \dots, m) \\ &&& x_j \geq 0 \quad (j = 1, 2, \dots, n) \end{aligned}$$

<http://www.cise.ufl.edu/~davis/Morgan/Image13.gif>



<http://www.javaworld.com/javaworld/jw-07-2004/images/jw-0719-jsf1.gif>



<http://yourmilitaryaircraft.com/pics/c130-20.jpg>



<http://texnrails.com/images/track/tms/1200701.jpg>

# Why Model?

- In most of the engineering disciplines, it is de rigueur to use models when designing a complex system [Se1 03].
- Since today's software systems are becoming more and more complex, benefiting from using models is inevitable [Se1 03].

# Model-Driven Approach of Software Development

- Despite the processes that are code-centric, in these approaches, models are the main artifacts which **drive** the development → **Model-Driven**
- The ultimate goal is to automatically generate programs from the corresponding models [Se1 03].

# Model-Driven Approaches

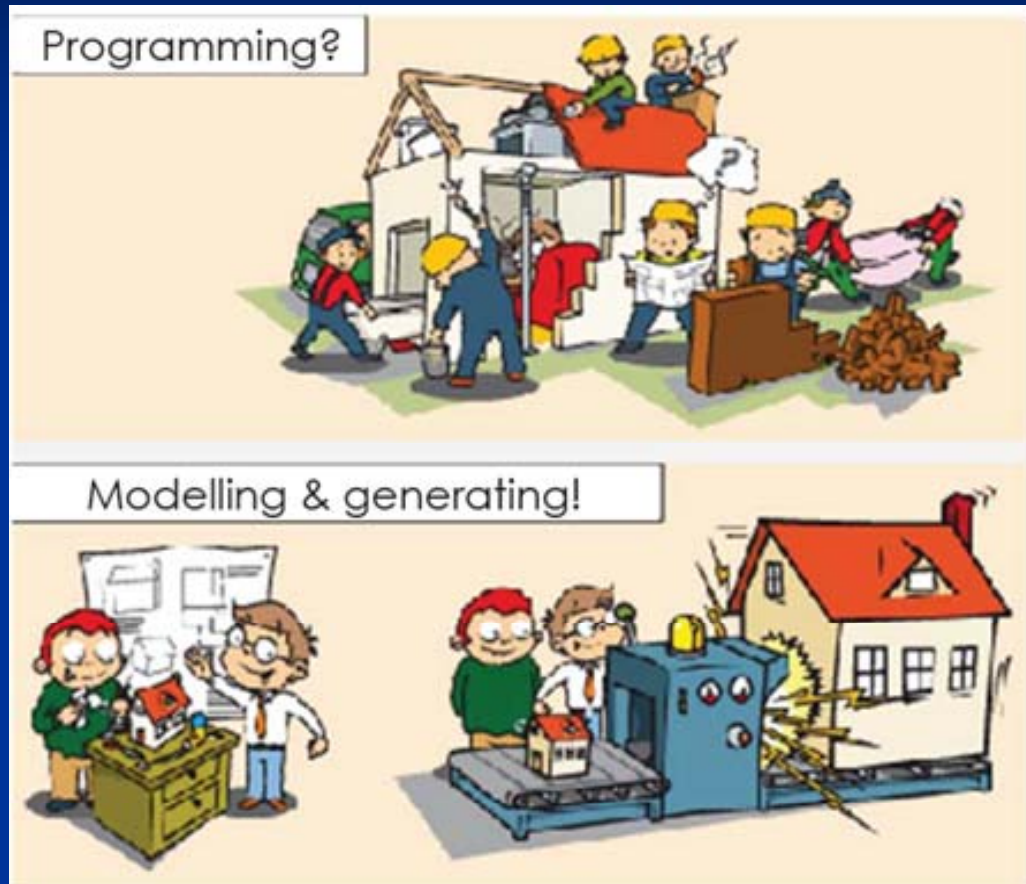
- MDA: Model-Driven Architecture
- MDD: Model-Driven Development
- MDE: Model-Driven Engineering
- MDSD: Model-Driven Software Development
- MDSE: Model-Driven Software Engineering



Don't worry, just go **Model-Driven!**

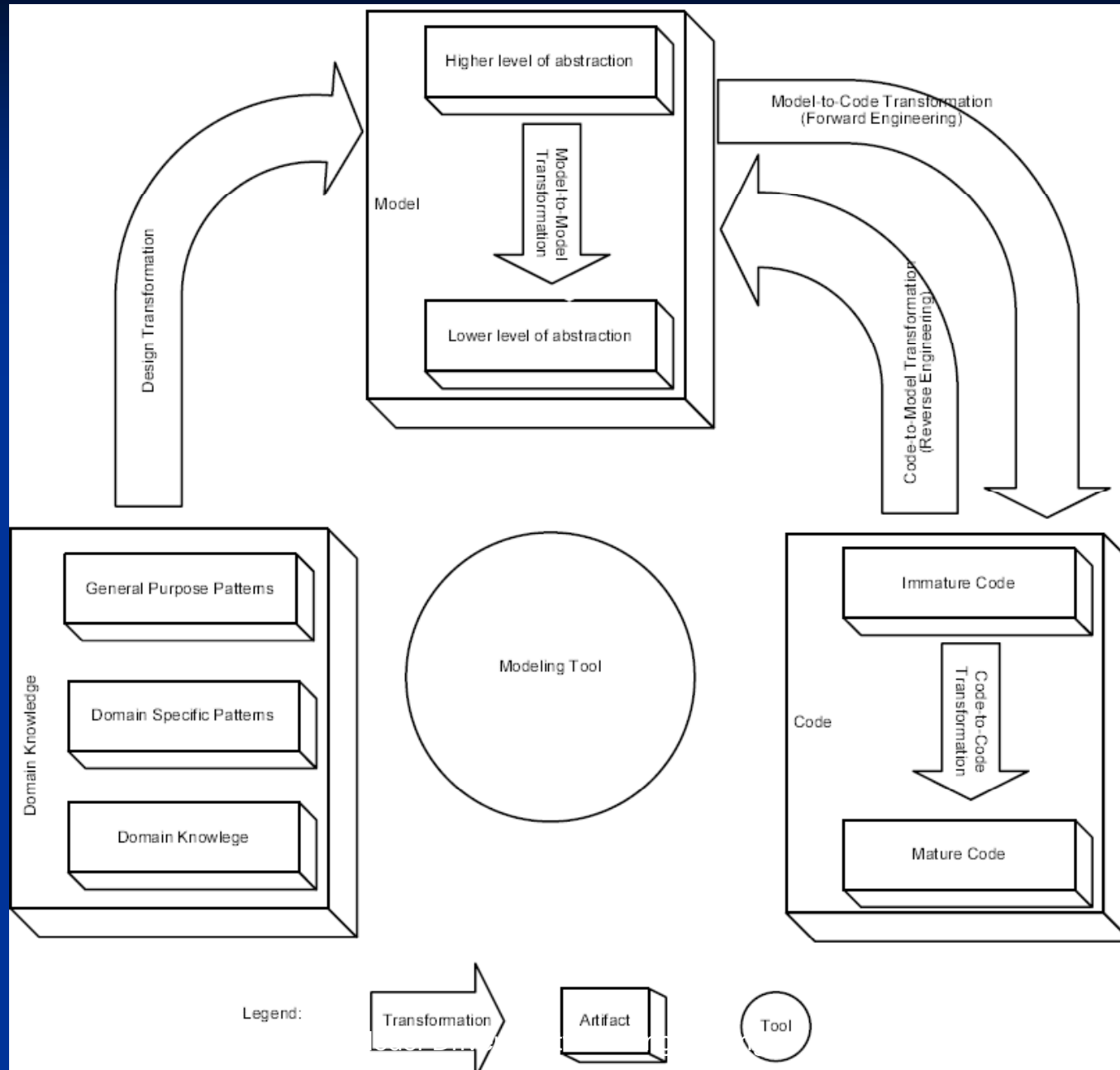
[http://www.t-one.net/~om/NCTblog/ist2\\_2771207\\_dizzy.jpg](http://www.t-one.net/~om/NCTblog/ist2_2771207_dizzy.jpg)

# A metaphor for MDE



<http://www.theenterprisearchitect.eu/archive/2009/08/05/a-metaphor-for-model-driven-engineering>

# An MDE Roadmap





# The Unified Modeling Language (UML)

- UML is a graphical language for
  - specifying
  - visualizing
  - constructing
  - documentingthe software artifacts.



<http://www.uml.org/>

- In the graphical modeling of OO software systems, Unified Modeling Language (UML) is the dominant approach.
- UML 1.1 adopted by OMG in November 1997
- Current release is UML 2.3 (released 9/9/09).

# UML 2.0 Diagrams

[<http://www.agilemodeling.com/essays/umlDiagrams.htm> ]

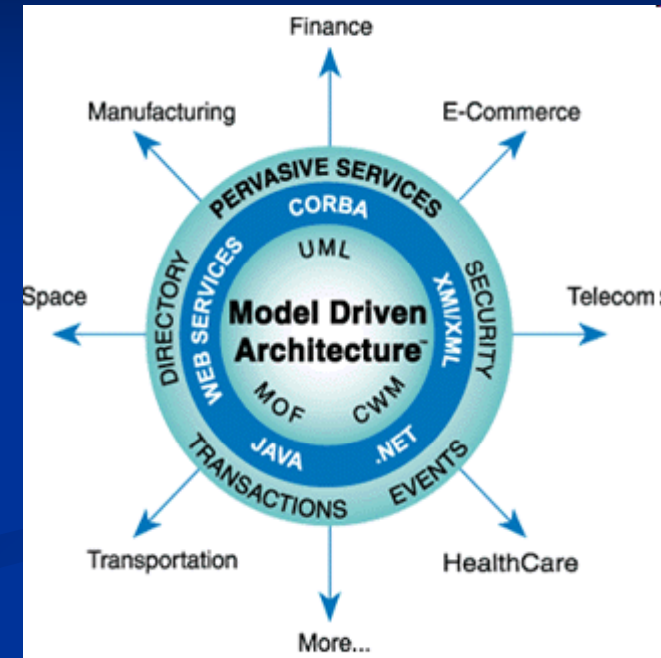
1. Activity Diagram: Depicts high-level business processes
2. Class Diagram: Shows a collection of static model elements such as classes
3. Communication Diagram: Shows instances of classes, their interrelationships
4. Component Diagram: Depicts the components that compose an application
5. Composite Structure Diagram: Depicts the internal structure of a classifier
6. Deployment Diagram: Shows the execution architecture of systems
7. Interaction Overview Diagram: Overviews the control flow within a system
8. Object Diagram: Depicts objects and their relationships at a point in time
9. Package Diagram: Shows how model elements are organized into packages
10. Sequence Diagram: Models the sequential logic
11. State Machine Diagram: Describes the states an object or interaction may be in
12. Timing Diagram: Depicts the change in state or condition of a classifier instance
13. Use Case Diagram: Shows use cases, actors, and their interrelationships

# Does Model-Driven work?

- Working paradigms:
  - MDA
  - xUML

# Model-Driven Architecture (MDA)

- Separates
  - the specification of functionality (business and application logic) from
  - the specification of the implementation (underlying platform technology)
- There are two types of models
  - Platform-Independent Model (PIM)
  - Platform-Specific Model (PSM)
- Design once, build it on any platform (.NET, J2EE, CORBA,...)



<http://www.omg.org/mda/>

# MDA: Platform-Independent Model (PIM)

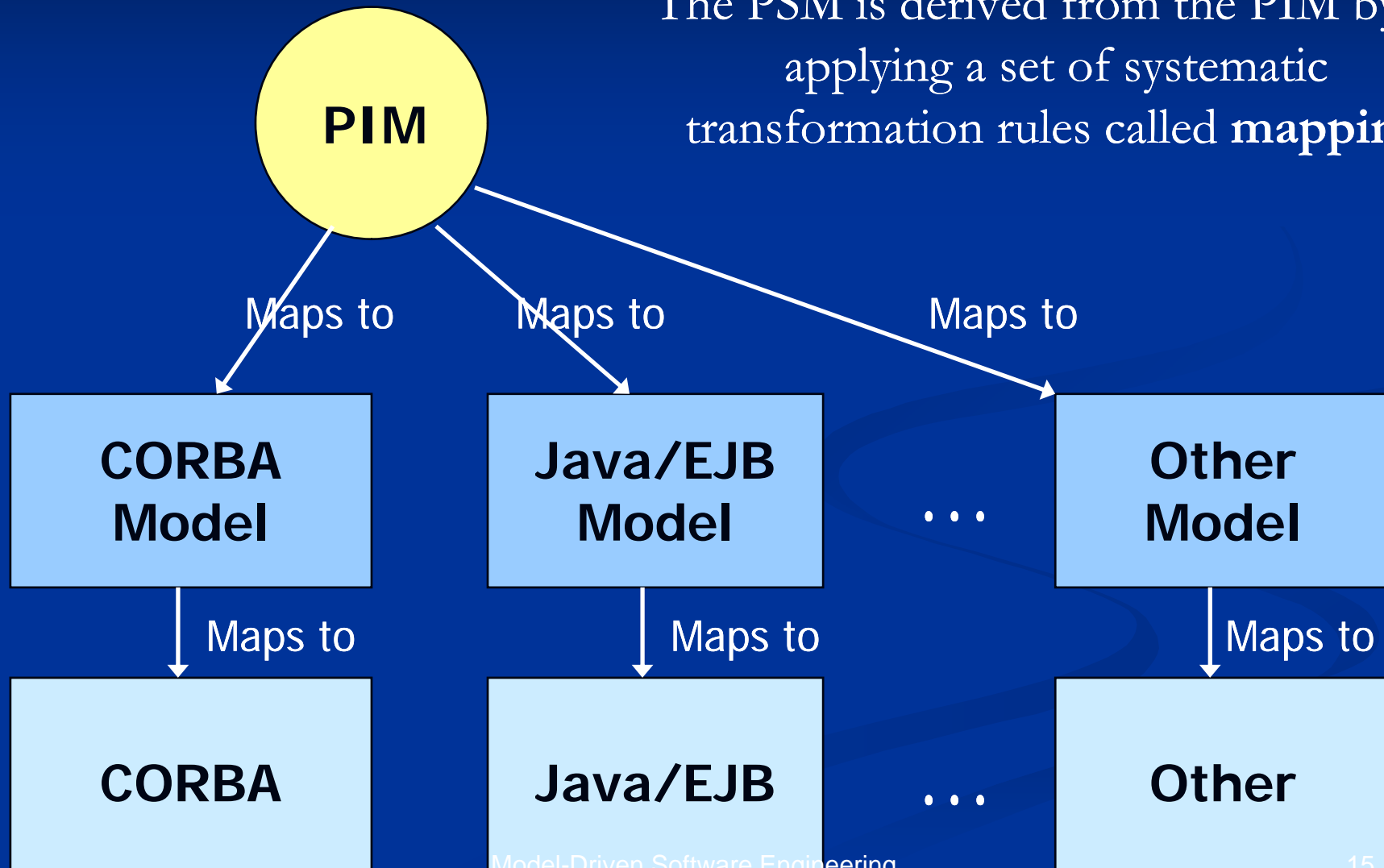
- An abstract “formal” specification of the structure and function of a system, leaving out the technical details
  - Expressed using UML
- Each PIM captures an aspect (or viewpoint) of the system
  - Viewpoints are called **domains**
  - Domains are combined to produce the system we want to construct

## MDA: Platform-Specific Model (PSM)

- The PSM comprises all the functionality expressed in the PIM with the added design concerns
- PSM is expressed using UML extended by adding platform specific details
  - Using UML extension mechanisms (UML Profiles)

# MDA: Mapping from PIM to PSM

The PSM is derived from the PIM by applying a set of systematic transformation rules called **mapping**



# Executable UML (xUML)

- An executable version of UML
- Automated methodology utilizing a highly specialized subset of the UML notation

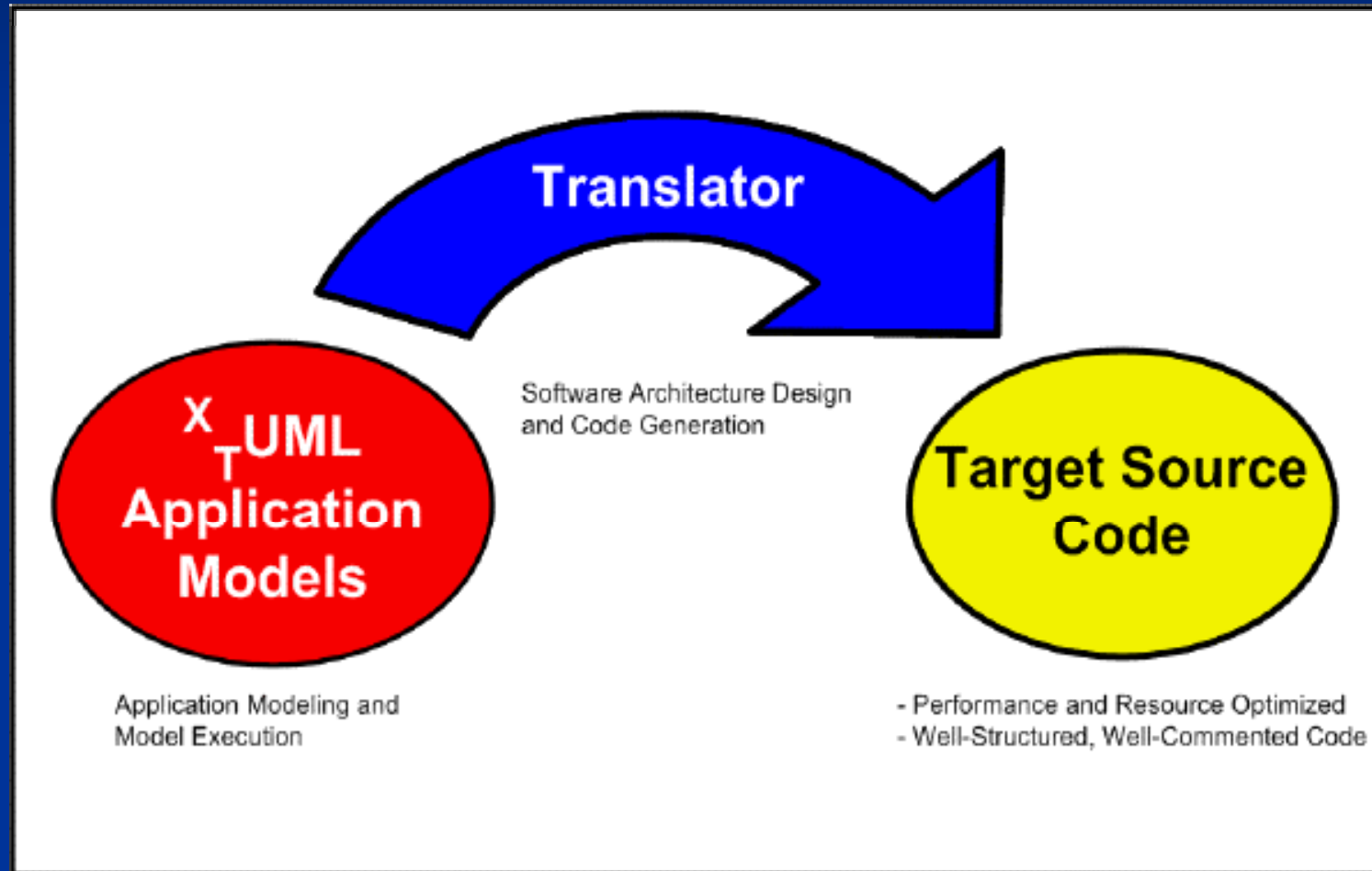
- xUML → 

UML
Semantically weak elements
Action Semantic

- UML Class Diagrams, State Charts, Action Language

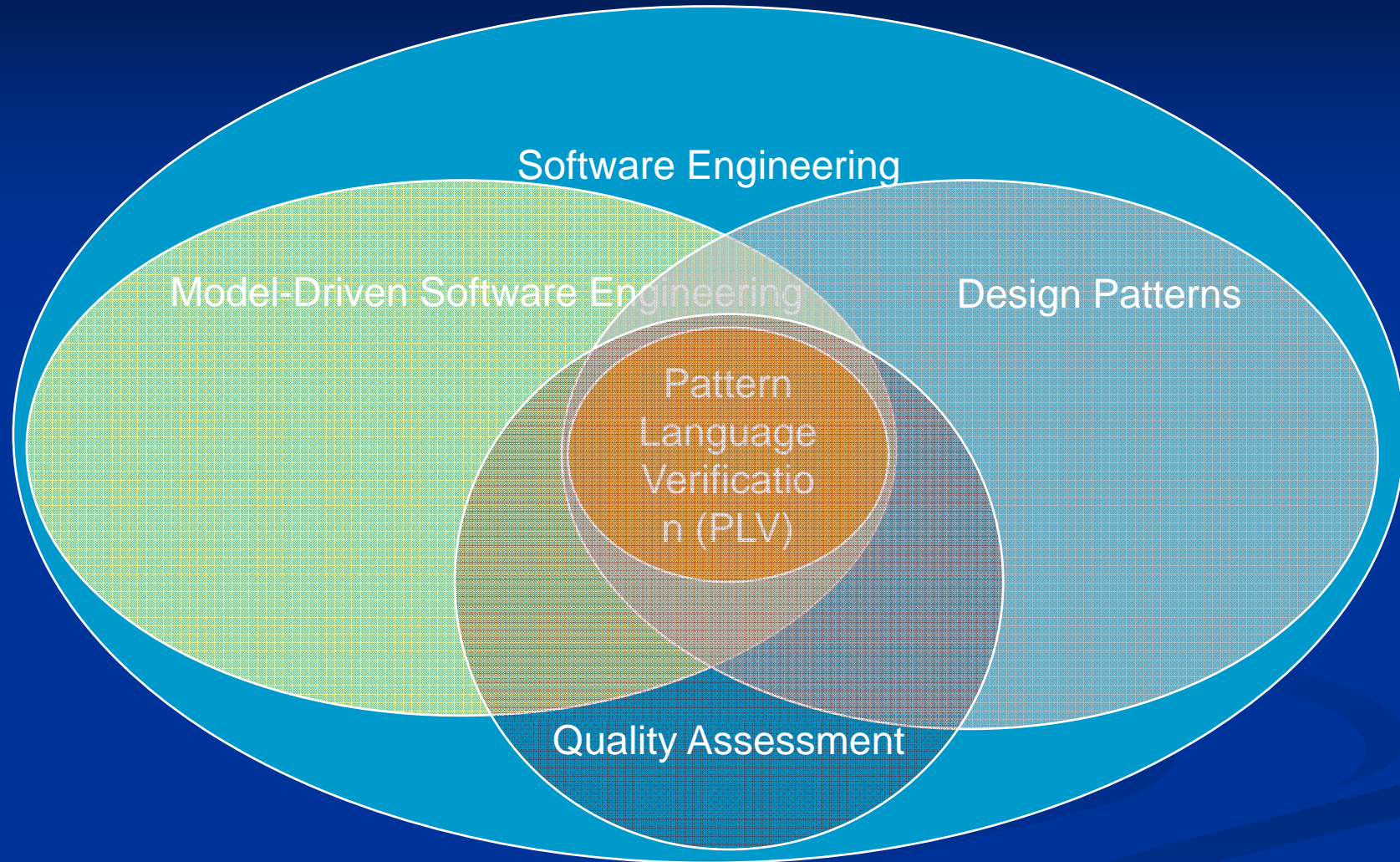


# xUML tool: BridgePoint Development Suite



[www.projtech.com](http://www.projtech.com)

# My Research



- For more information:
  - Visit my page at [bahmazamani.com](http://bahmazamani.com)

Thank you!  
Questions



<http://www.greateracadianaregion.net/edu/Portals/0/images/cct/Questions.jpg>

# References

- [Sel 03] Bran Selic. The pragmatics of model-driven development. *IEEE Software*, 20(5):19–25, Sep. 2003.
- [Sel 06] Bran Selic. Model-driven development: Its essence and opportunities. In *Proc. ISORC'06*, pages 313–319, Los Alamitos, CA, USA, 2006. IEEE Computer Society Press.
- [Obj 05] Object Management Group (OMG). Unified Modeling Language (UML): Infrastructure, v2.0. OMG document: formal/05-07-05, 2005.