

Design Defects and Restructuring

Engr. Abdul-Rahman Mahmood

 abdulrahman@nu.edu.pk

 alphapeeler.sf.net/pubkeys/pkey.htm

 pk.linkedin.com/in/armahmood

 www.twitter.com/alphapeeler

 www.facebook.com/alphapeeler

 abdulmahmoodsss  alphasecure

 armahmood786

 http://alphapeeler.sf.net/me

 alphapeeler#9321

 reddit.com/user/alphapeeler

 www.flickr.com/alphapeeler

 http://alphapeeler.tumblr.com

 armahmood786@jabber.org

 alphapeeler@aim.com

 mahmood_cubix  48660186

 alphapeeler@icloud.com

 pinterest.com/alphapeeler

 www.youtube.com/user/AlphaPeeler

Architectural design issues

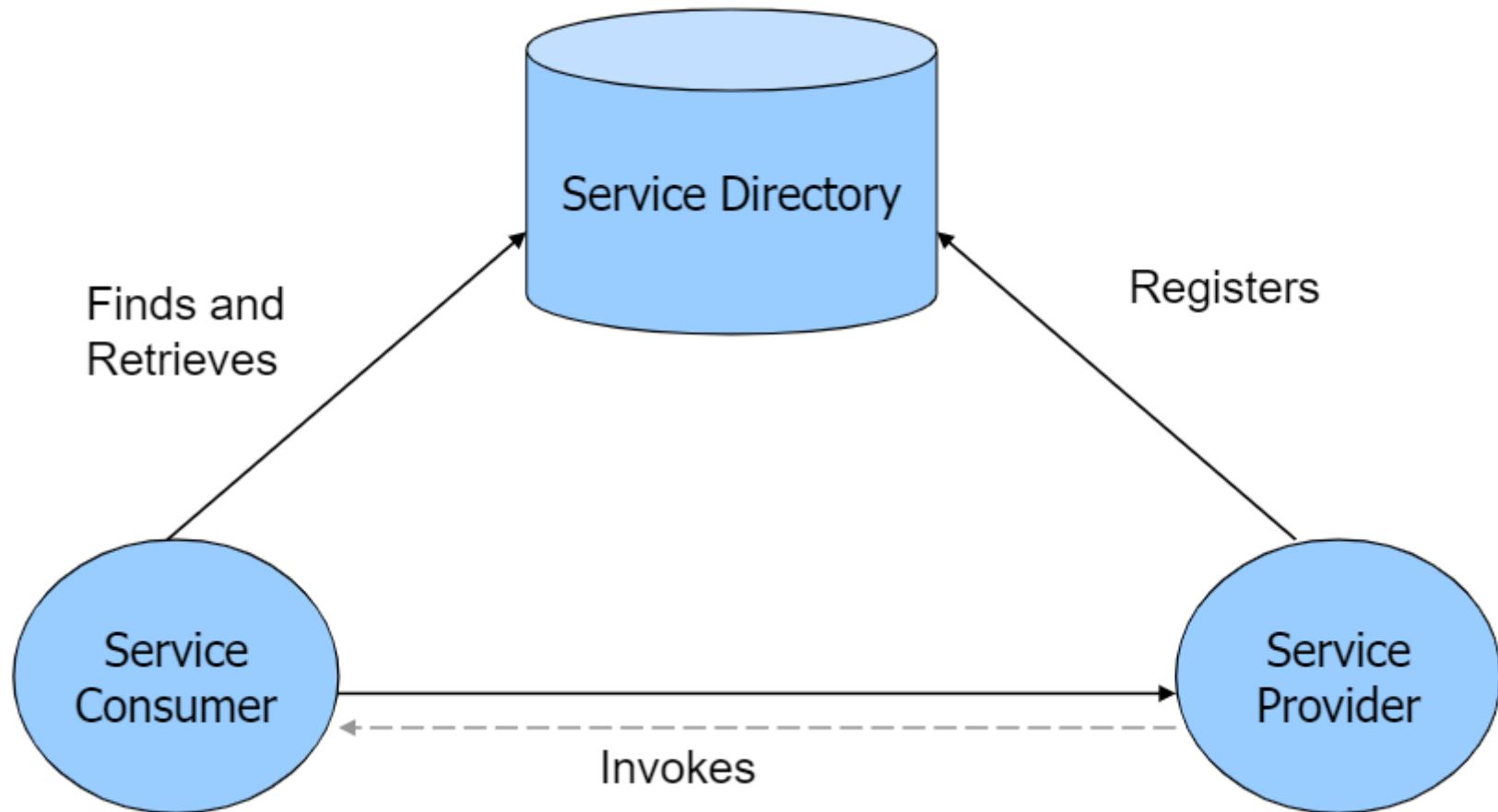
Issues of service oriented architecture

SOA

- need to assume business processes as a smart services that is loosely coupled and form the basis of target architecture.
- SOA uses the Web services standards and technologies
- resources are made available to other participants within the network as independent services that are accessible across the network in a standardized way.

Service Oriented Architecture

- Each Service Oriented Architecture plays one or more of three roles:



SOA Development Issues

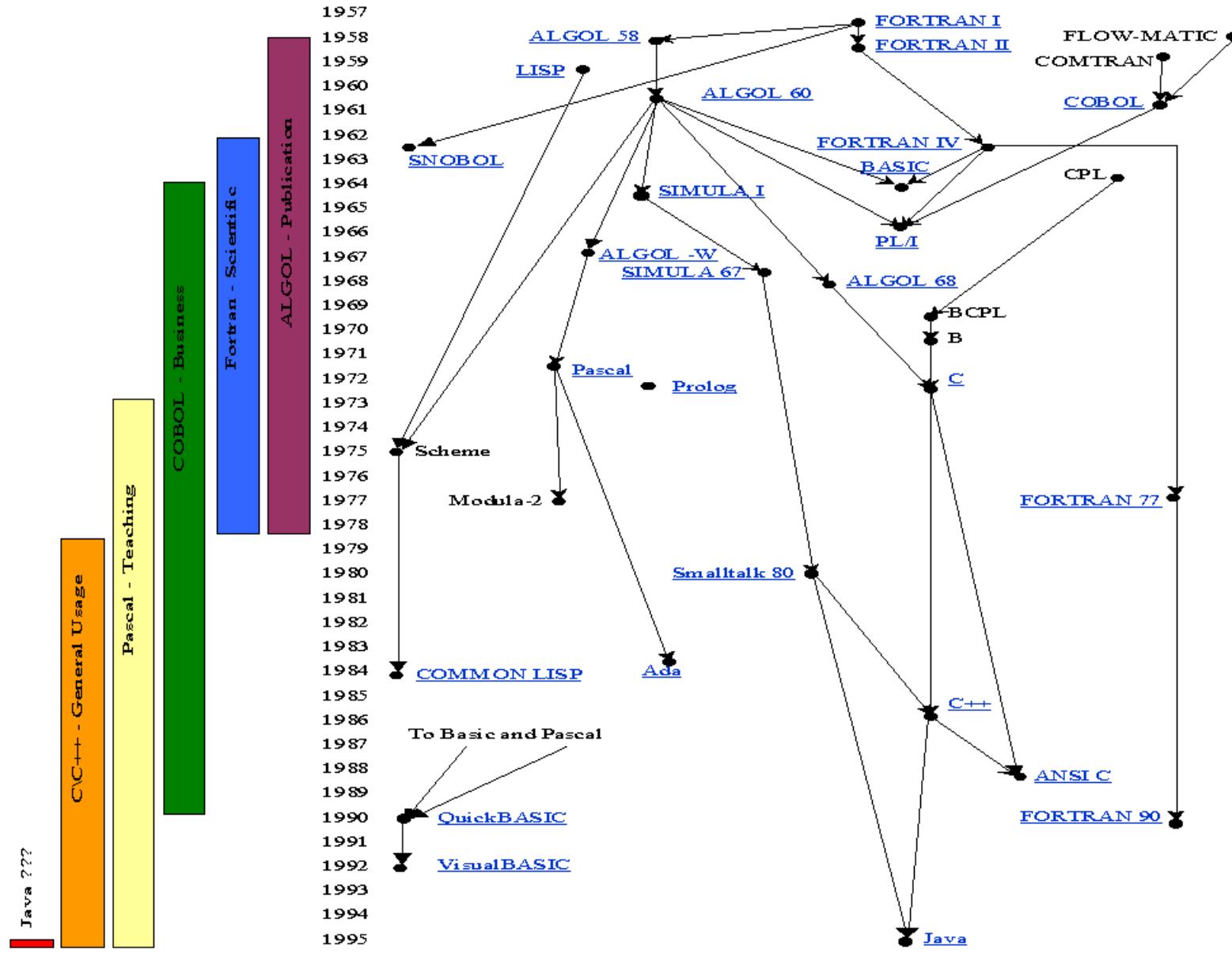
- service-oriented information systems analysis and design (SOAD) have roots in 3 major existing disciplines: (OOAD), (BPM), and (EA).
- Hybrid approach: includes elements from OOAD, EA, & BPM to come up with a 3 layers SOAD approach to include component, software service & business service layers.
- **Component layer** -> business objects (OOAD)
- **Business service layer** -> BPM
- **Software service layer** -> granularity or details of services
- 3 Phase approach: includes service abstraction, service analysis & design activities.
- **abstraction**: service discovery & conceptualization (high-level abstractions of business logic and re-usable processes)
- **analysis phase**: service descriptions, business integration, EA and meta-data specifications.
- **design phase**: component and architecture logical and physical
- designs will be outlined.

SOA Security issues

- **Security is business enabler:** e.g. car brake. Because brakes we drive faster.
- Objects & components use binary runtimes.
- **Web services,** don't rely on binary controls for security.
- Web services security (**WS-Security**): XML Digital Signature & XML Encryption for message integrity and confidentiality.
- **SOA architecture** contains components service requestor, service provider and service repository
- the communication exists between service requestor to provider in the form of message passing and is the scope for intruder to enter, therefore it is within the message exchange where the authentication, authorization, audit, and assurance services add true value.

Model Driven Development

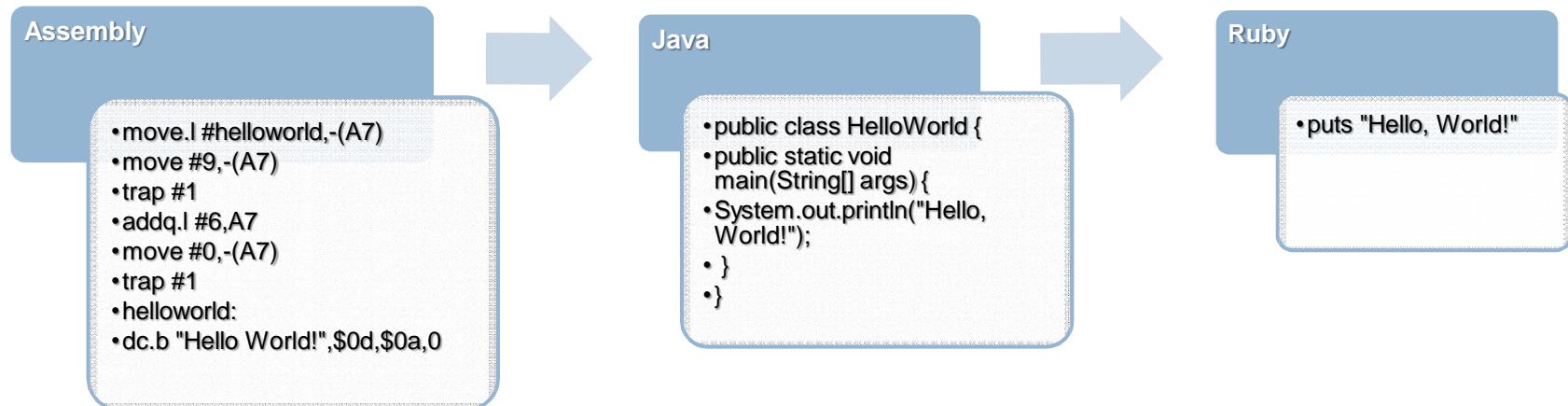
Evolution of Languages



Source: <http://www.eecs.ucf.edu/~leavens/ComS541Fall97/hw-pages/history/>

Evolution of Languages

- Evolution of programming languages
 - Machine language to Assembly language to higher level languages such as C++, Java, C#, Ruby, etc.
 - More time was spent on understanding “how” to solve the problem in early languages (understand the language)
- Evolution of tools, frameworks and application servers
 - Abstraction and reuse of common services
- Each language and framework raised the level of **Abstraction** by hiding low level details



What is Abstraction?

- What is Abstraction?
 - Abstraction is concentration on relevant aspects of the problem and ignoring those that are not important
 - Focus on solution to the problem by working with concepts and terms that are familiar to the problem space and ignoring the low level details
- Abstraction is the key to building modern complex software with multiple moving parts
- Model based development is the natural next step in the evolution of Abstraction

How can Models help us?

- Models provide a simplified abstraction of the complexity in the target domain
 - Models provide an abstraction layer that focuses on the higher level concepts of the domain and decouples “what” from “how”
 - Models can be Visual or non-visual (DS Model)
 - Different models provide different views of the problem domain
- Used in Daily Life, Science & Engineering
 - Ex: Maps, Engineering (CAD/CAM), Architecture (Structural Modeling)
- Used in software engineering primarily for white boarding, communication and analysis & design



Model Driven Development (MDD)

□ What is MDD?

- A software development approach that uses models to capture application logic during the development of end-to-end enterprise applications
- **Forrester's Definition:**
"An iterative approach to software development where models are the source of program execution with or without code generation."

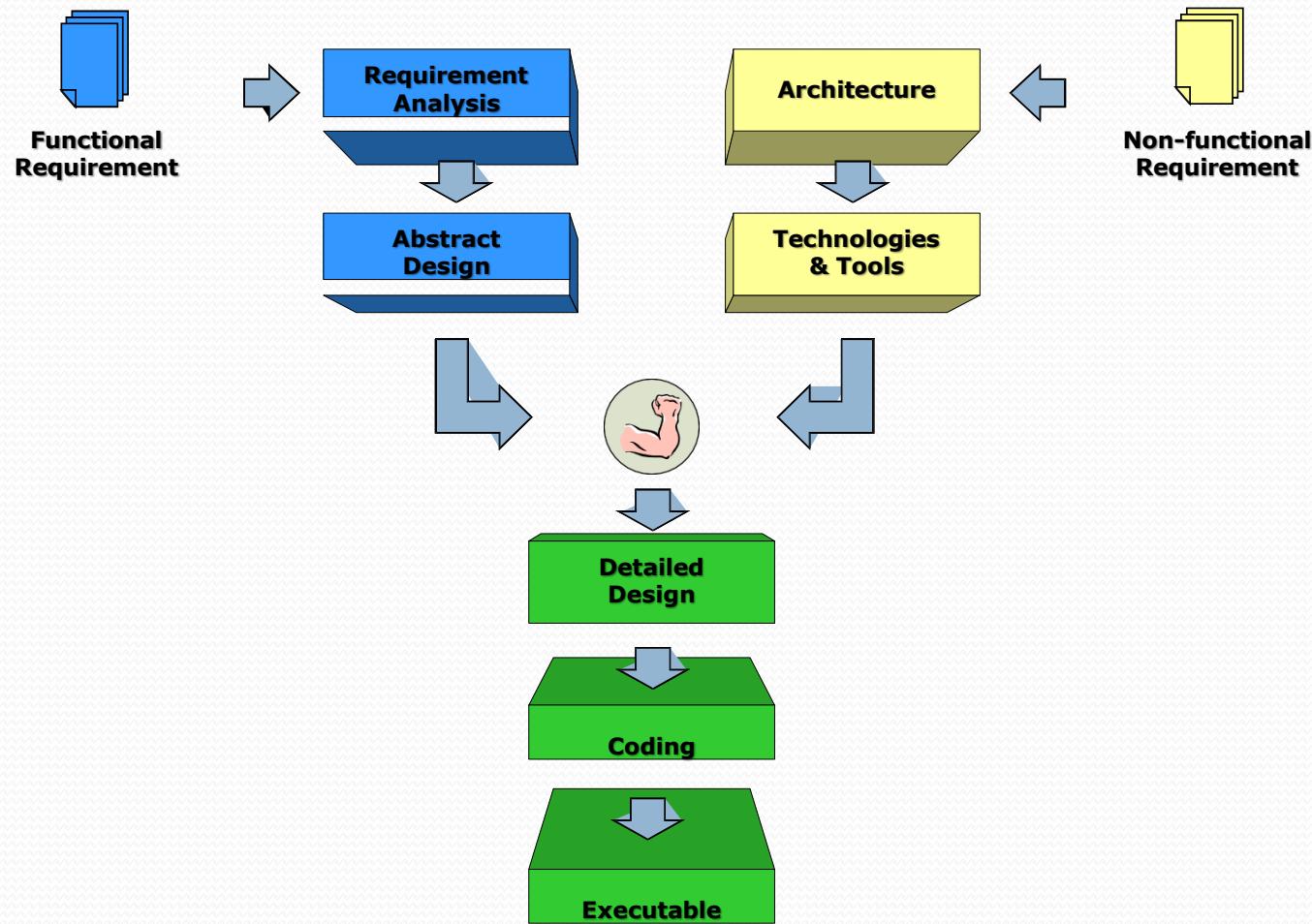
□ MDD Objectives

- Raise the level of abstraction for application development
- Reduce development time and improve application quality
- Reduce maintenance cost and Total Cost of Ownership of enterprise applications

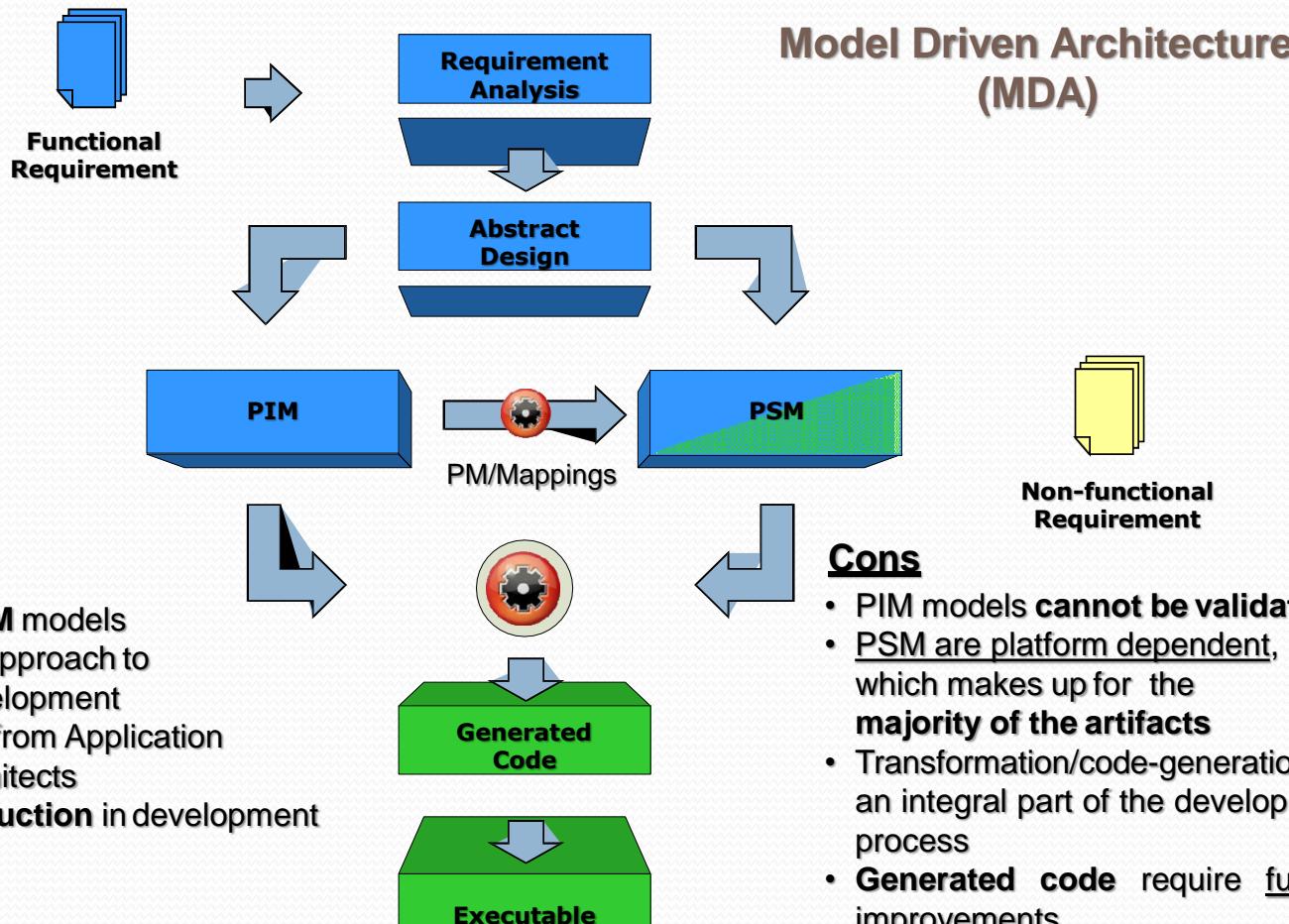
Model Driven Development

- How
 - Use models to implement application logic
 - The domain model (in Abstract Design) is the implementation model
 - Use automation to generate executables from the implementation model in runtime or build time
- Approaches
 - OMG Model-Driven Architecture (MDA)
 - Executable Models

Software Development



Model Driven Architecture (MDA)



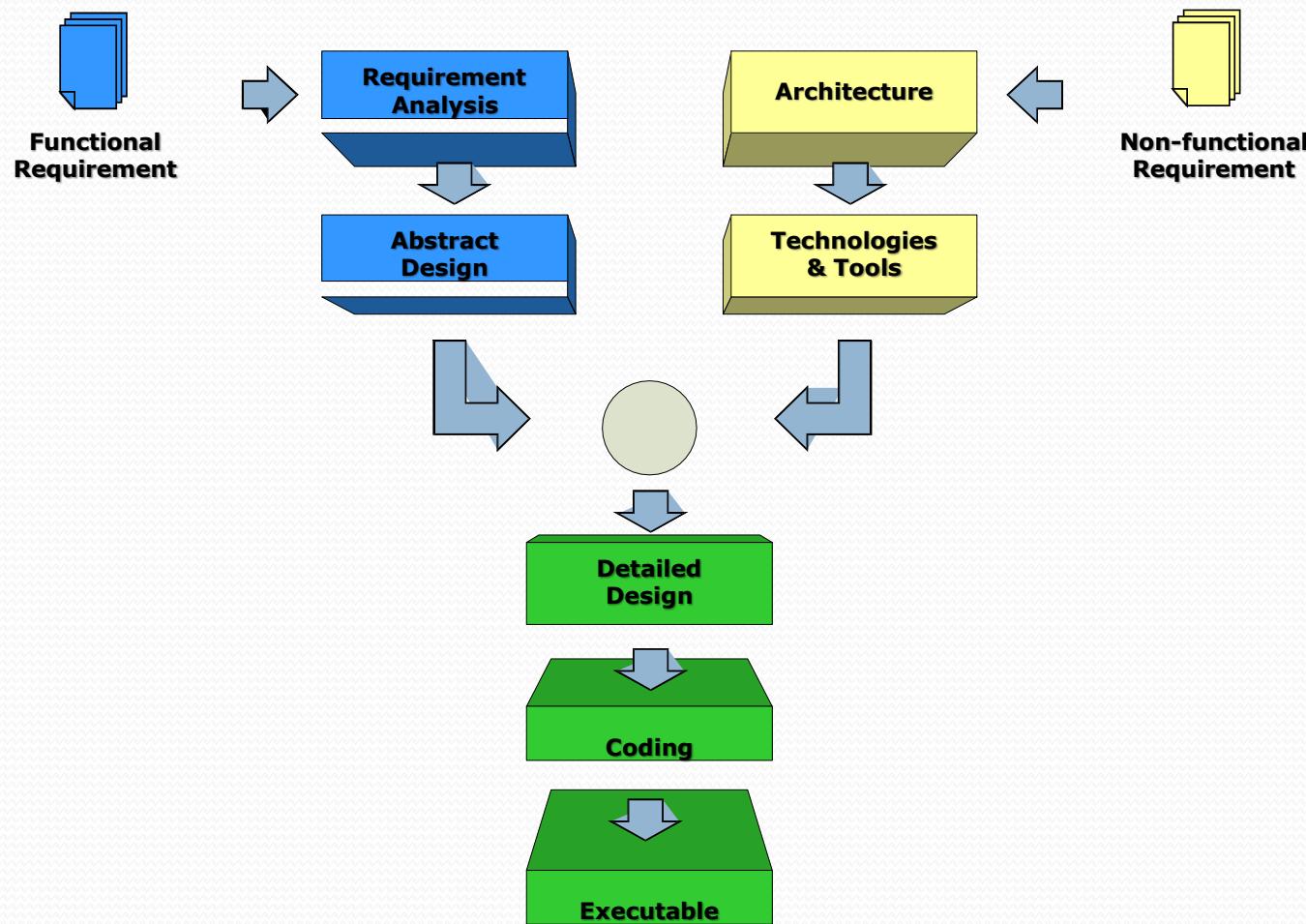
Pros

- **Reusable PIM models**
- **Methodical approach to software development**
- **Direct input** from Application Analysts/Architects
- **Marginal reduction** in development time

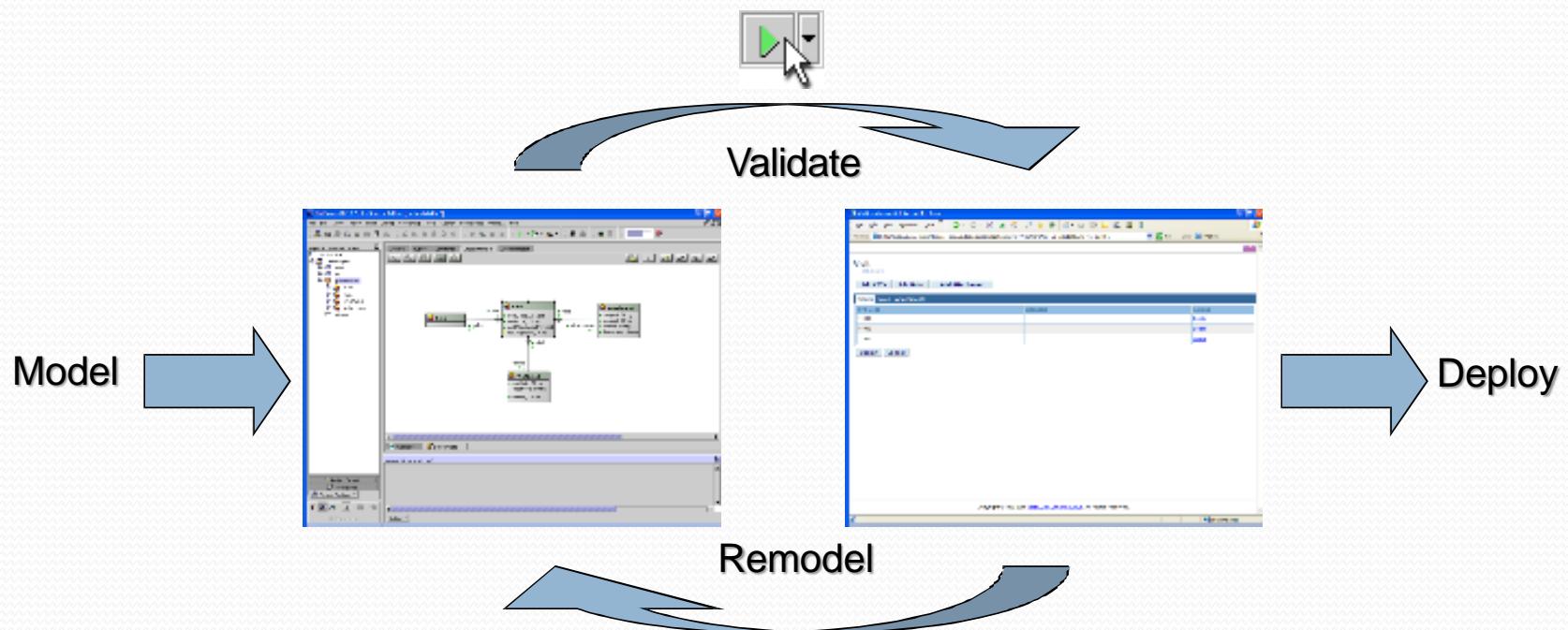
Cons

- PIM models **cannot be validated**
- **PSM are platform dependent**, which makes up for the **majority of the artifacts**
- Transformation/code-generation is an integral part of the development process
- **Generated code require further improvements**

Executable Models (xUML)



Executable Models (xUML)



The Model is the Executable

Advantages

- Captures application logic in platform independent UML models
- Simplifies application development by reducing the number of required skills in the underlying technologies, specifications and standards
- Provides immediate validation of business requirements
- Improves communication among stakeholders
- Protects business IP investments from evolving technologies
- Radically reduces the development, time, cost and effort of business applications and Web Services
- Increases application agility to better align with continuously changing business needs

MDD Vendors

- MDA
 - IBM's Rational Software Architect
 - Interactive Objects (ArcStyler)
- Executable UML
 - **Intelliun Corporation (The Virtual Enterprise)**
 - Kennedy Carter (iUML)
 - CARE Technologies (OLIVANOVA)
 - Mentor Graphics (BridgePoint UML Suite)
 - E2E Technologies, Ltd. (E2E Bridge)

Myths of MDD

- ✖ Modeling can only be used during the analysis and design phases
- ✖ Only objective of modeling is to generate code
- ✖ MDD is similar to CASE Tools
- ✖ Models are not sufficient to develop an enterprise application
- ✖ MDD tools are not mature and can only be used to for small apps and prototypes
- ✖ MDD tools are expensive

State of MDD

- Many active vendors and products
- Proven to work – Several MDD applications in production !
- MDD and SOA
 - Model Driven Service Development
- MDD and BPM
 - Model Driven Business Process Management
- MDD continues to build momentum
 - Increased interest and involvement of industry leaders
- Emerging standards for Executable UML (xUML)
 - OMG's RFP for Concrete Syntax for a UML Action Language

Resources

- OMG's MDA Resources
 - MDA: <http://www.omg.com/mda/>
 - MDA Vendors: <http://www.omg.com/mda/committed-products.htm>
- Intelliun' Virtual Enterprise
 - Corporate : <http://www.intelliun.com>
 - Technical Overview:
<http://www.intelliun.com/Products/TheVirtualEnterprise>
 - Free Download:
<http://www.intelliun.com/Developers/Downloads>