

Two Thirds of “SDR” is “SD”

MDE



ASD



DRE



SPL



SDR

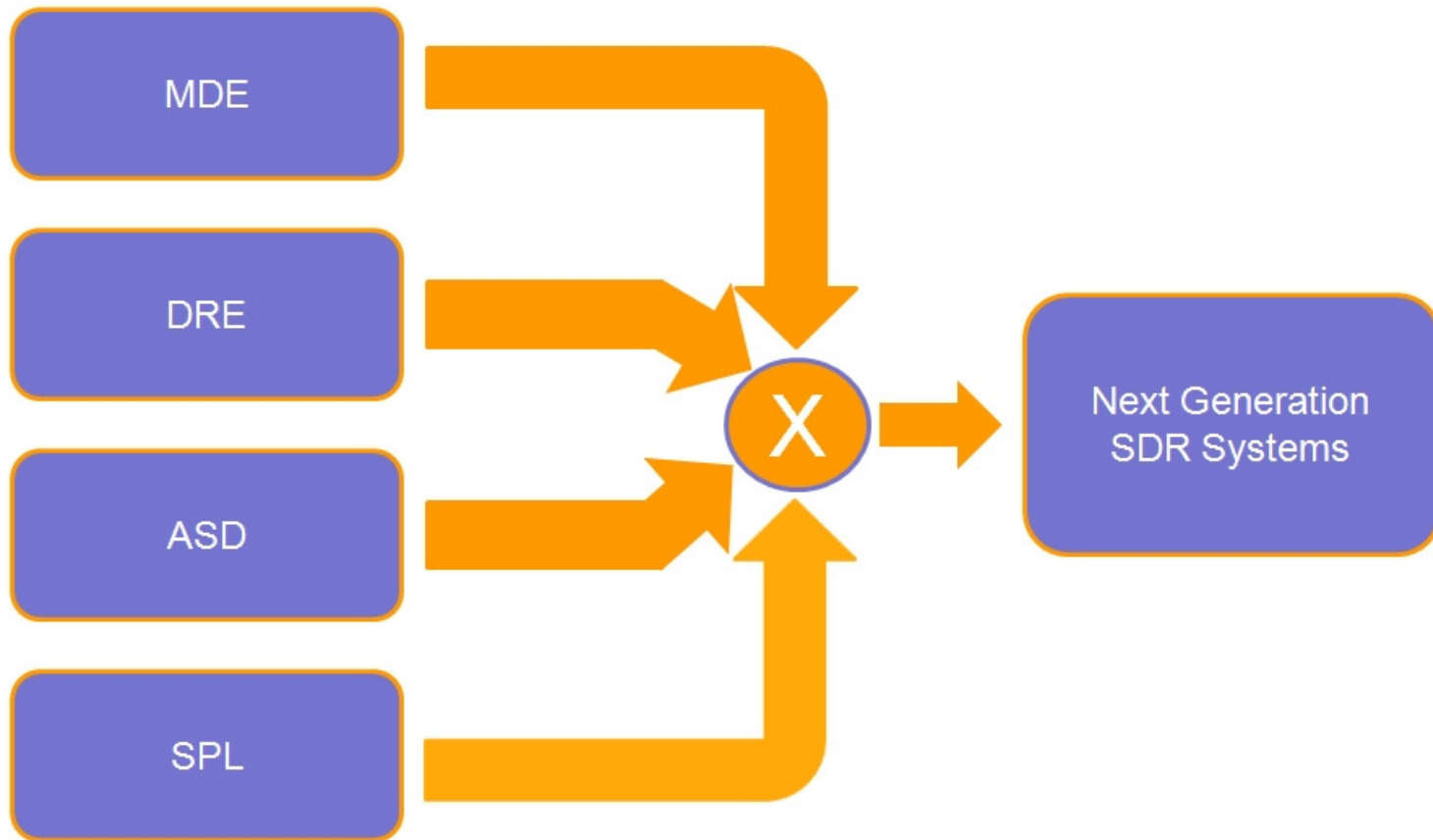
Bruce Trask
Angel Roman
MDE Systems

Agenda

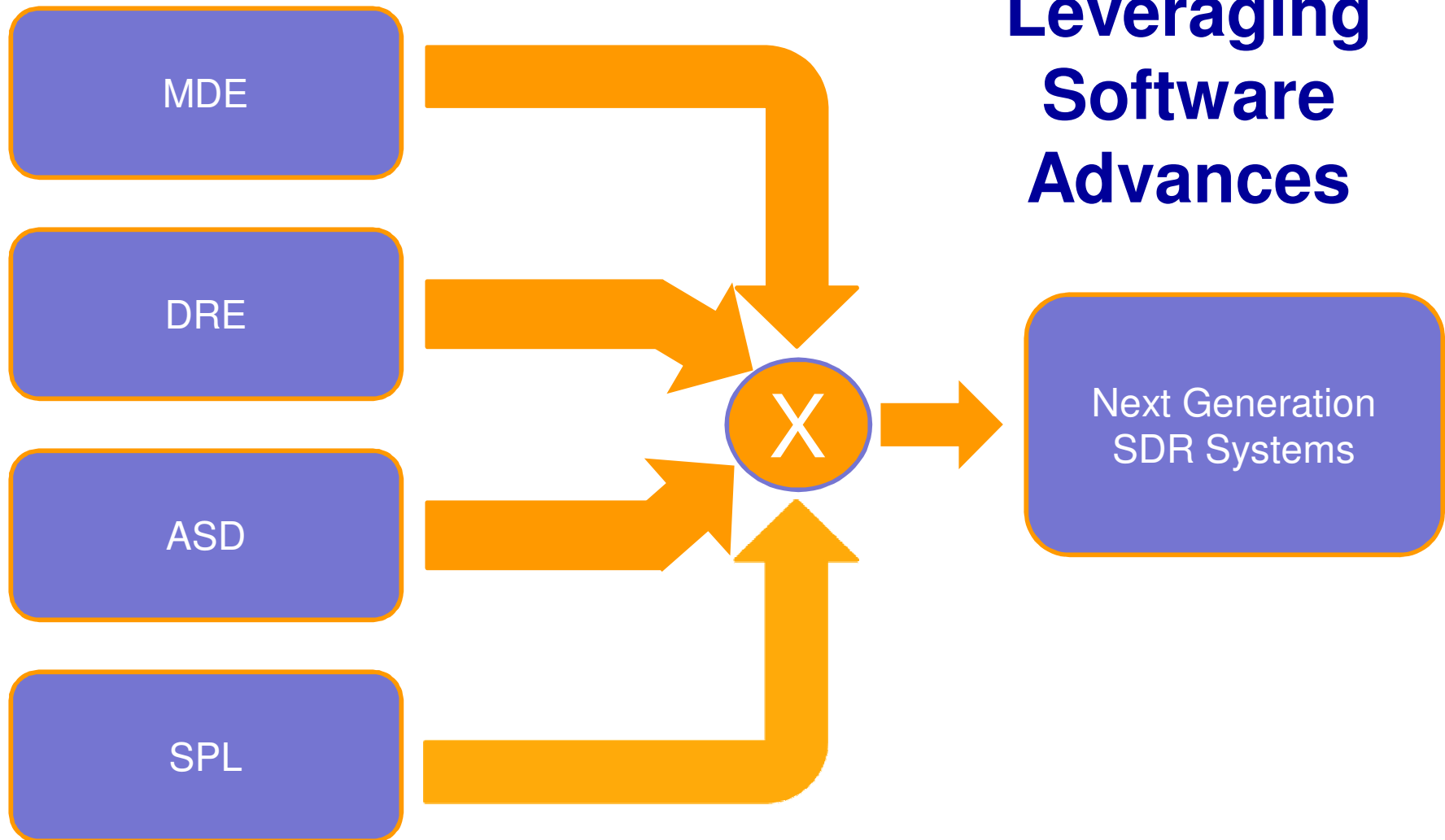
- **Comments on Software and the Software Industry**
- **Comments on SDR**
- **MDE, ASD, SPL, CI, CD, DRE**
- **Definitions**
- **Advances in Software – Enabling Technologies**
- **Making it all work**



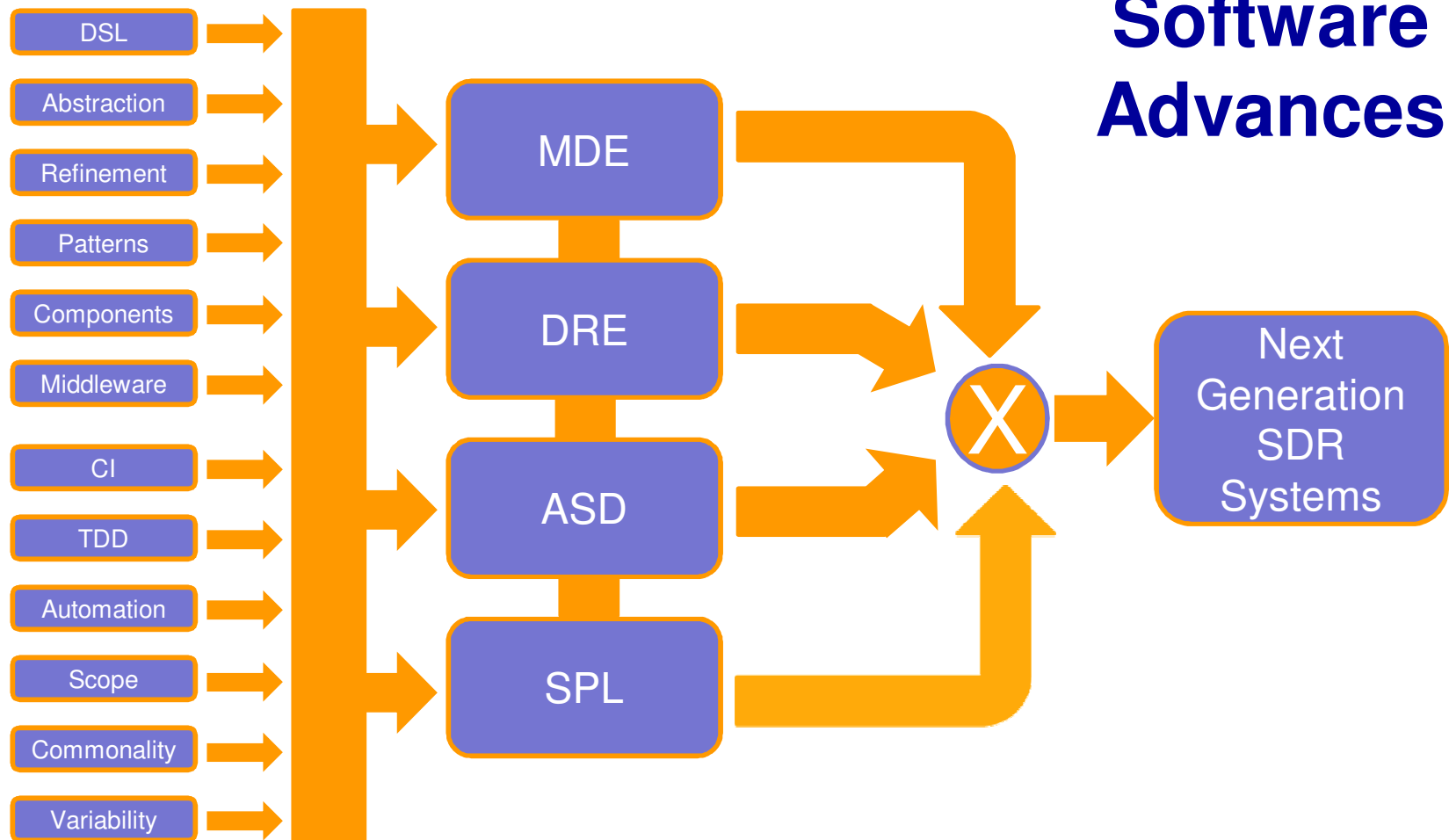
Summary



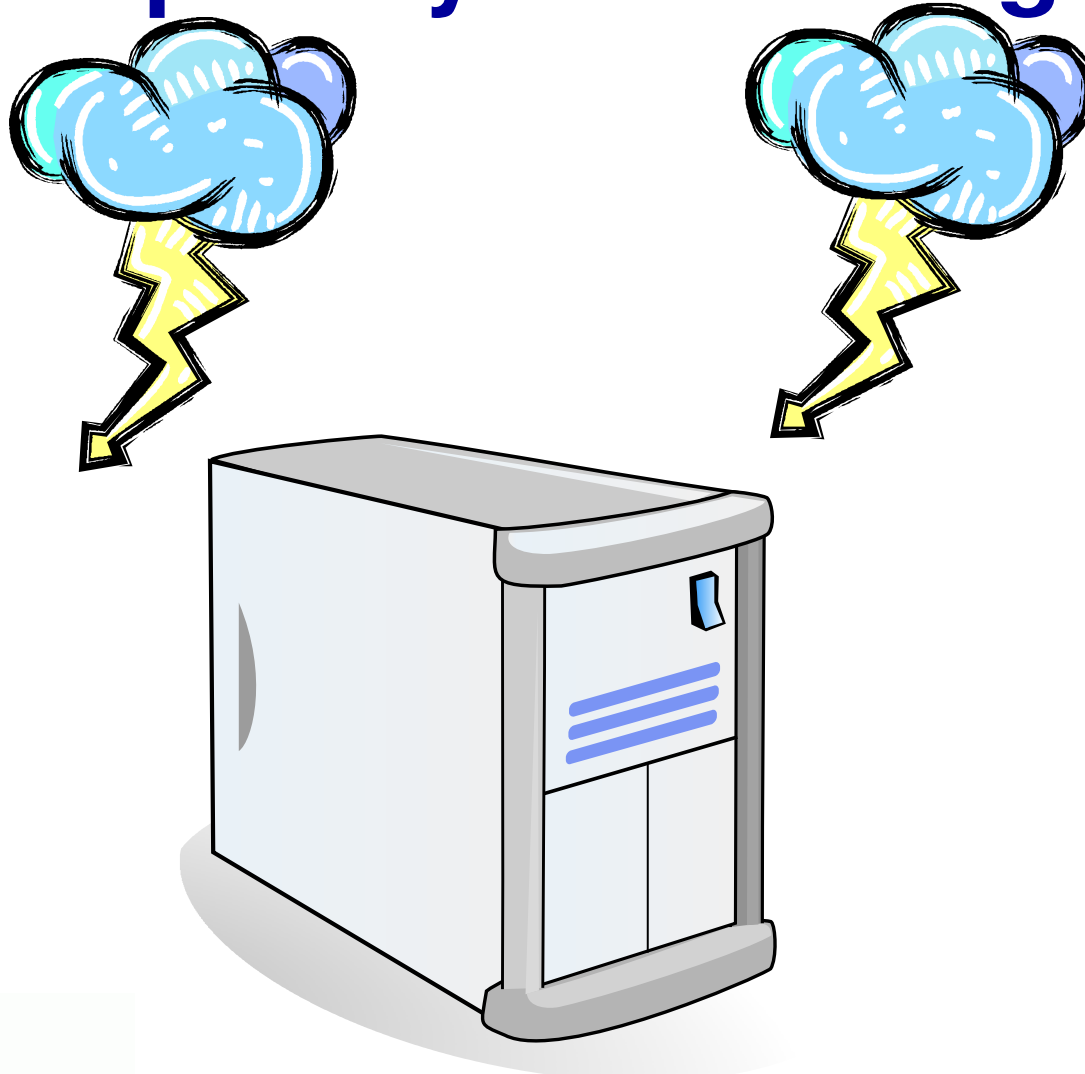
Leveraging Software Advances



Leveraging Software Advances



Complexity and Change



Abstraction Gap

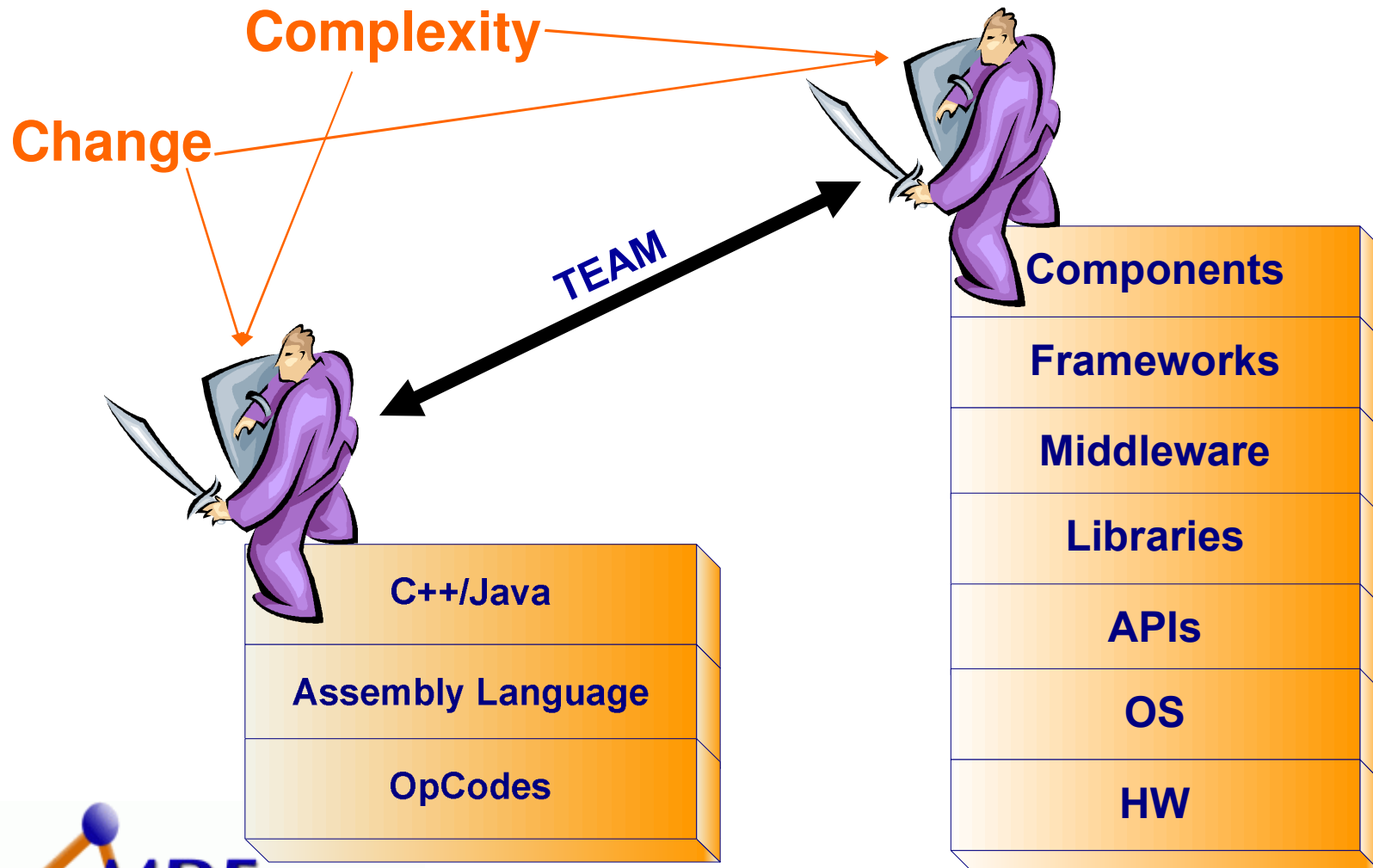


Software Engineering/Computer Science

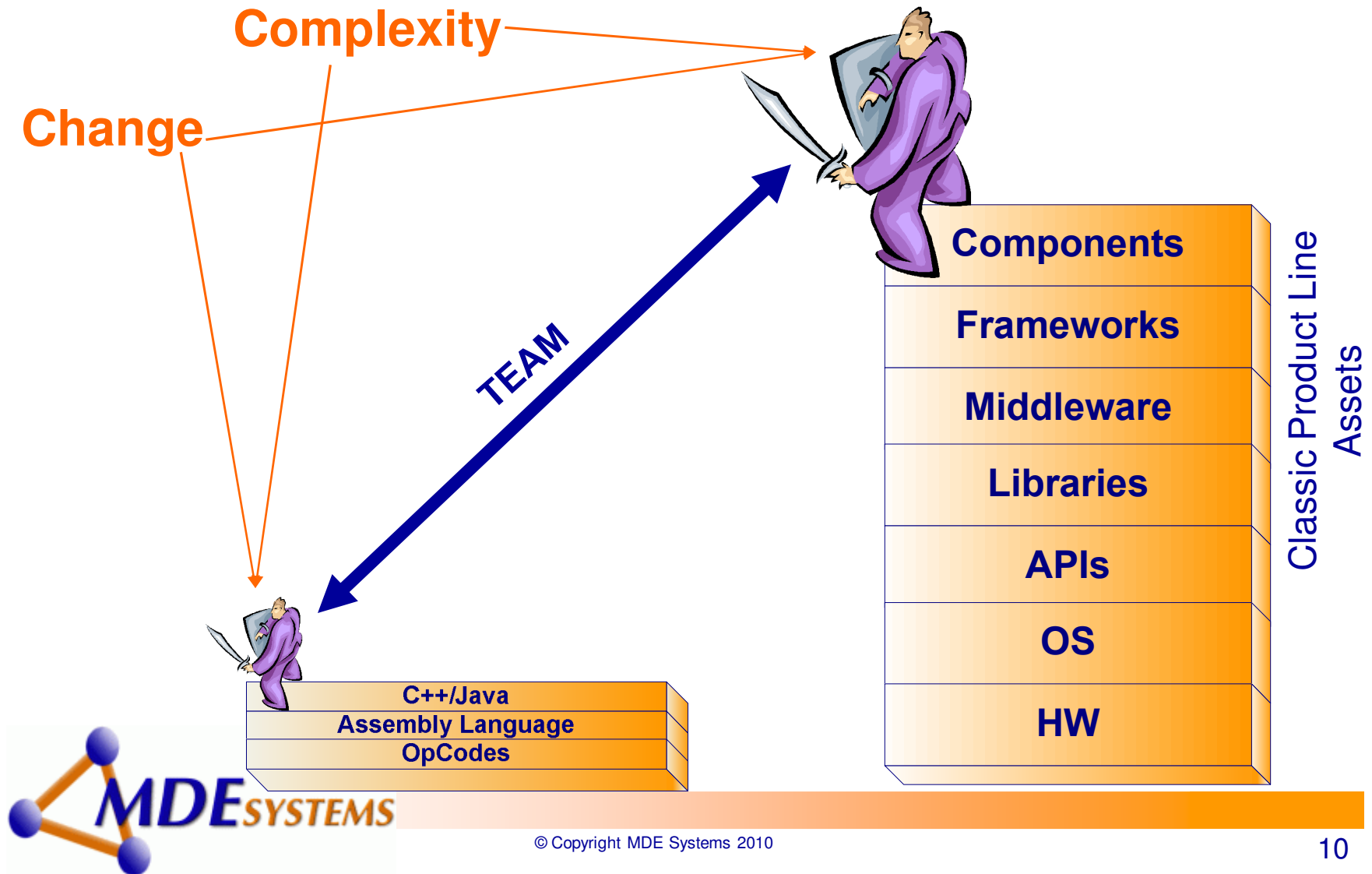
- Raising the level of Abstraction
- Introducing layers of indirection to solve problems¹



Language and Platform



Language and Platform – where we are now



What's underneath the problems

- Language technology has not kept pace with platform technology¹
- Insufficient *linguistic power* to tackle platform and domain complexity
- Lack of *tools* to deal with increased complexity



¹ Douglas C. Schmidt IEEE Computer Magazine February 2006

Solution

- Leverage recent critical innovations to provide a quantum leap of language technology and tools to overcome the complexity gap



Orders of Magnitude

- 1000x more processing power
- 1000x more dynamic memory
- 1000x more disk space
- 1000x more power efficiency
- 1000x smaller
- 15 orders of magnitude



Forces Upon SW Architecture

- Conway's Law
- Education – how far
- Fear of change
- The “real world” – square/rectangle

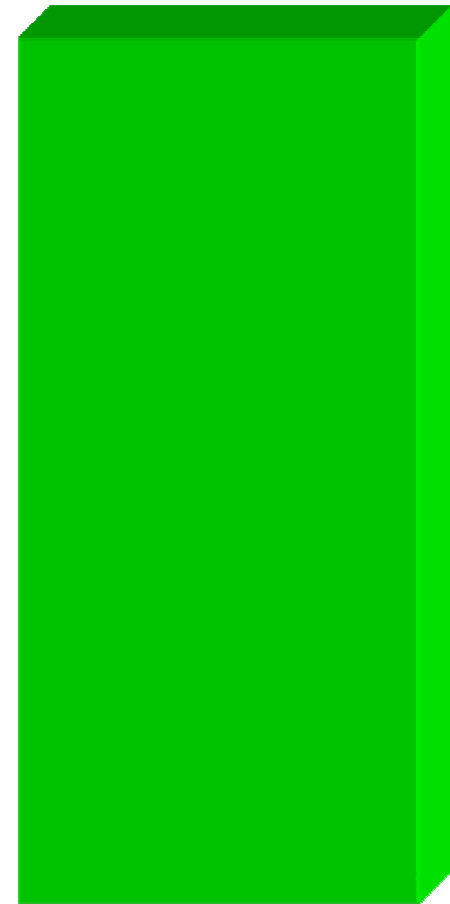
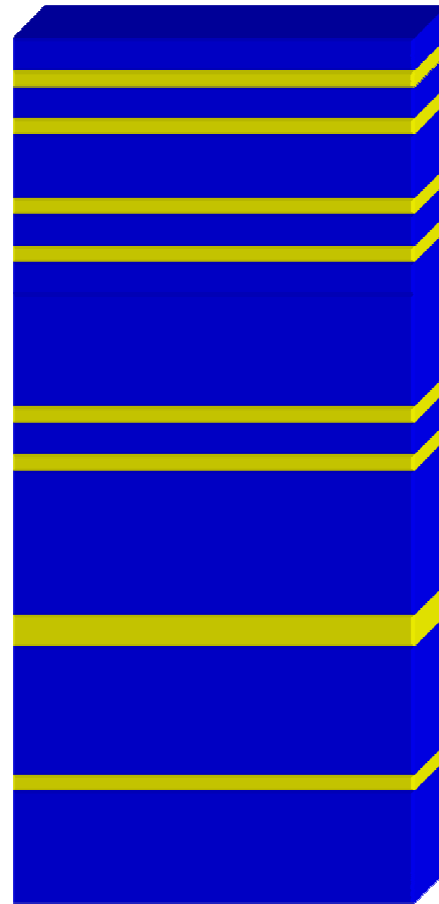
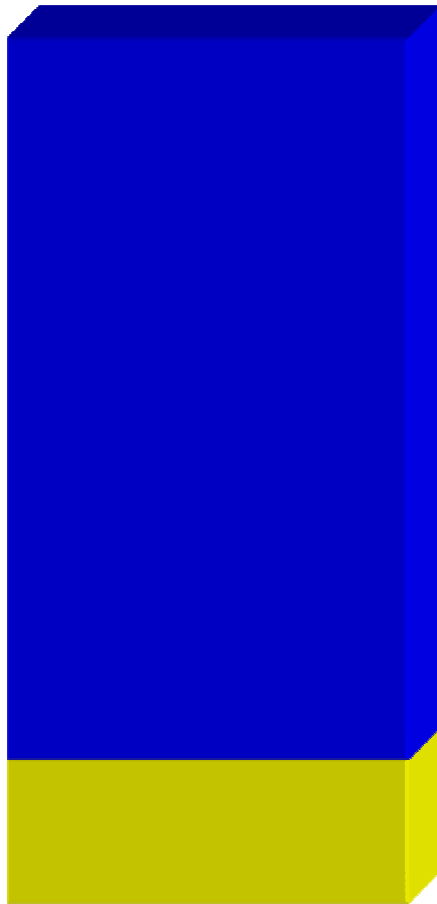


Software

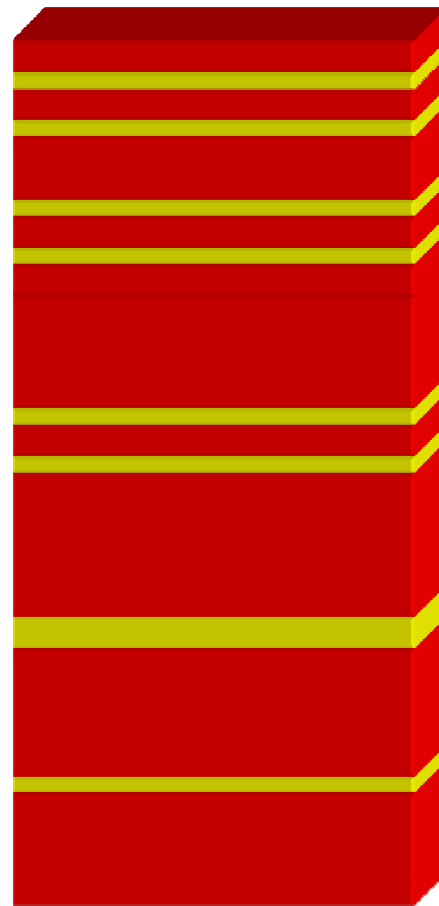
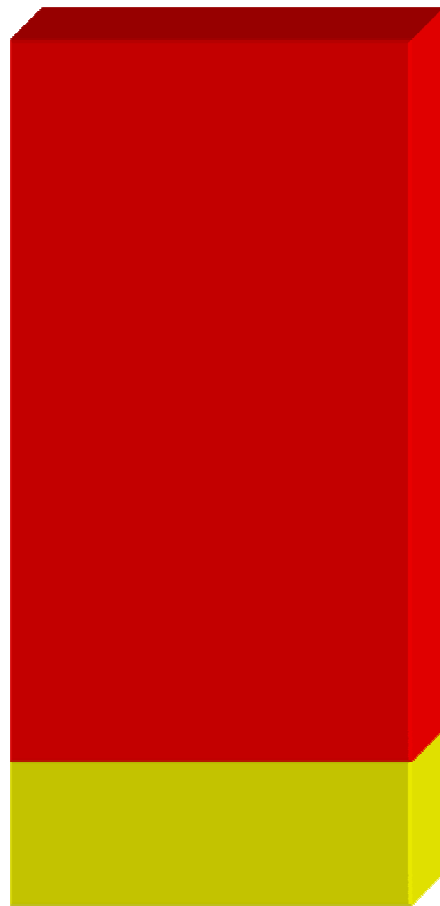
- Invisible
- Natural Laws
- Multiplicity of Software Design
- Malleable



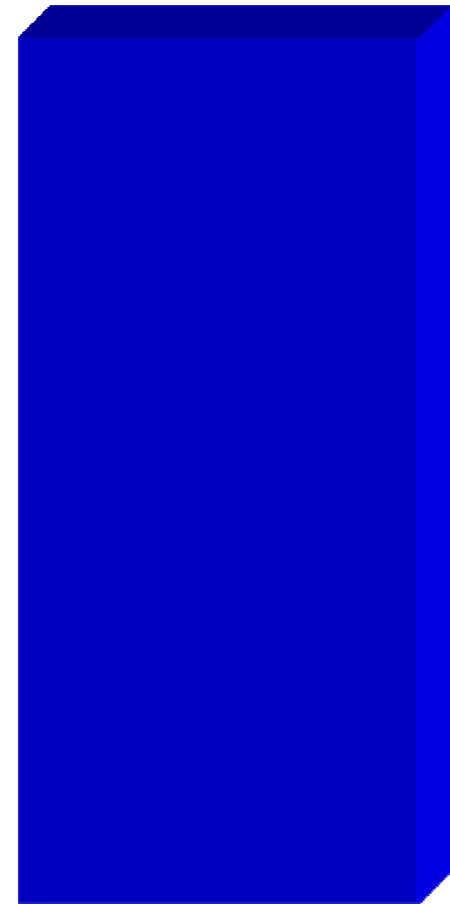
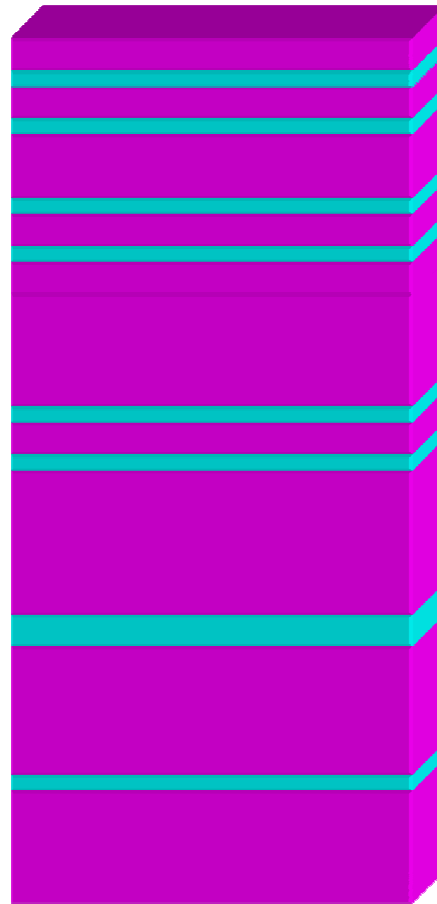
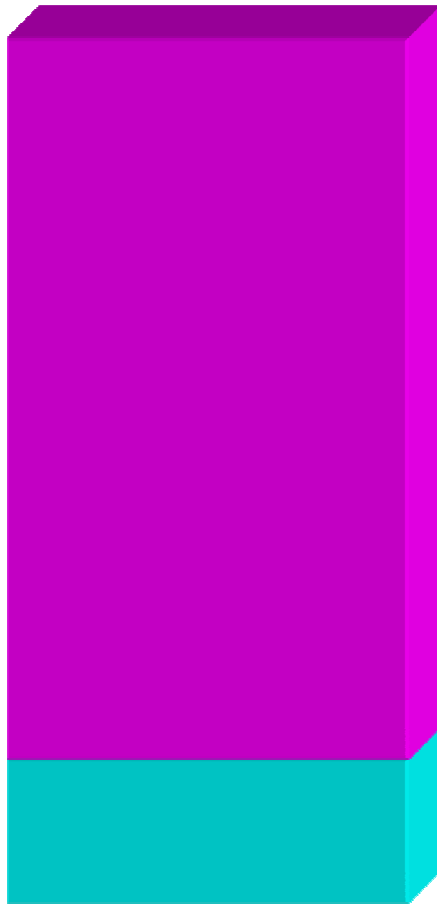
Rote vs. Creative Code



Domain Specific vs. Domain Independent Code



Duplicate vs. Unique Code

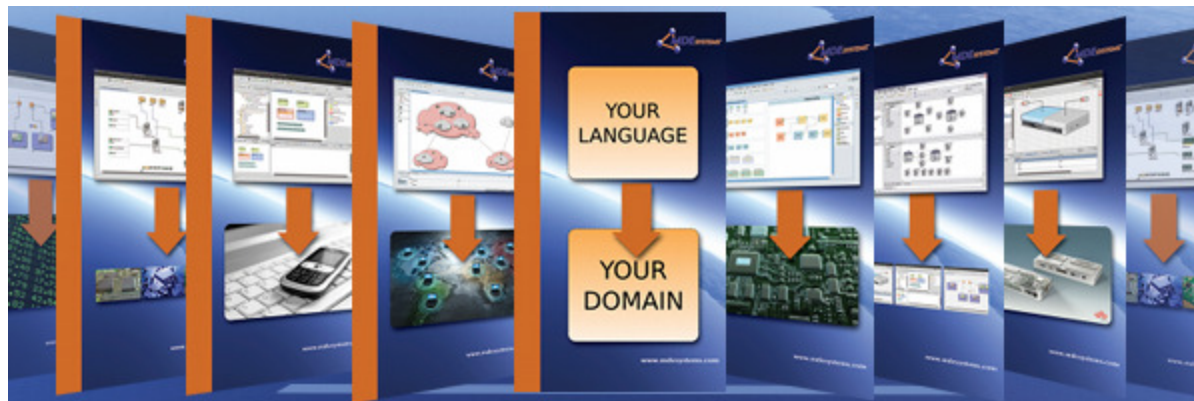


Comments on SDR Software

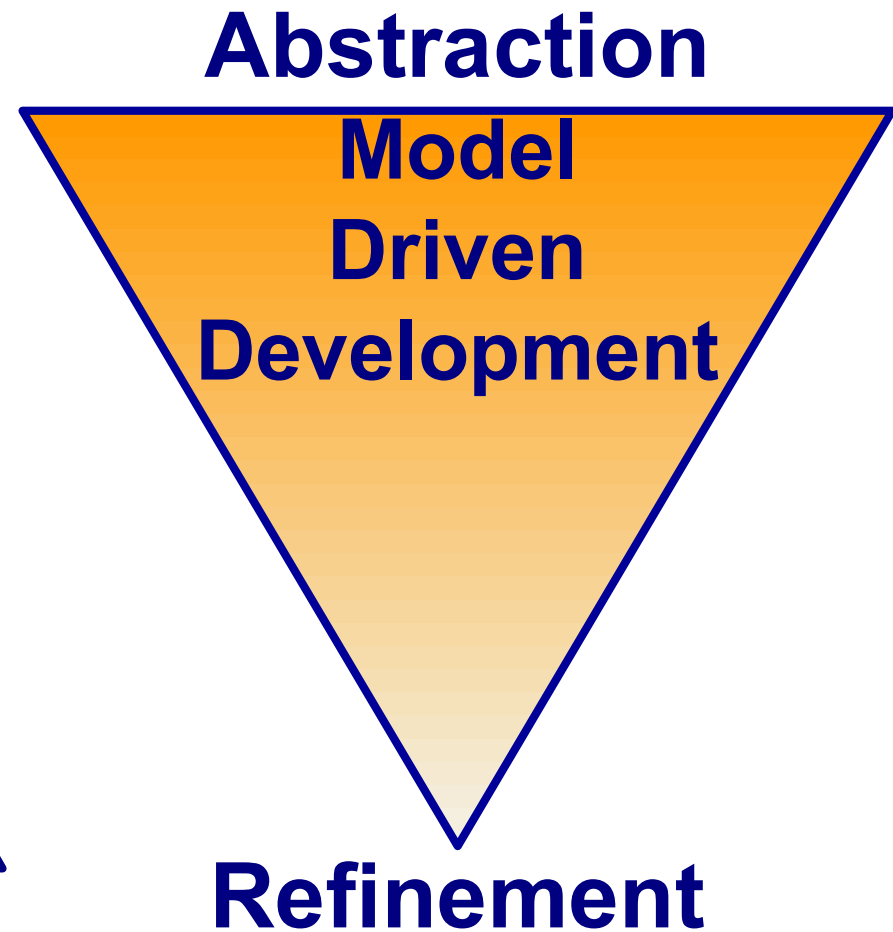
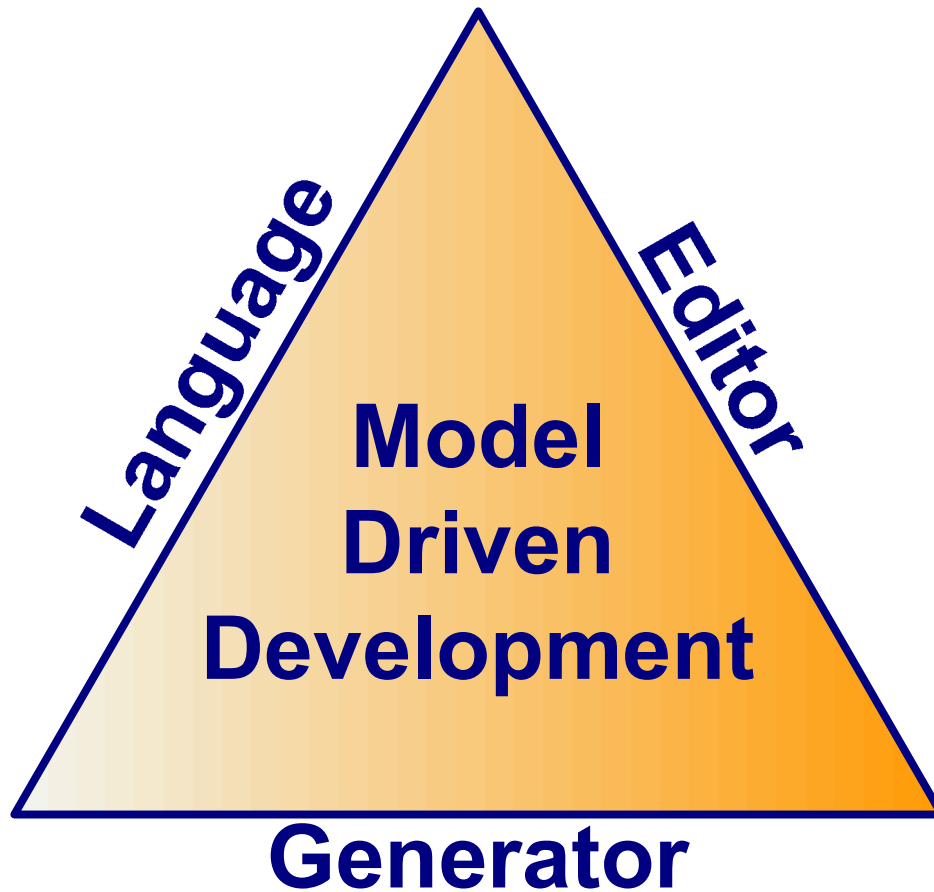
- Modifiers
- Degrees of Freedom



Model Driven Engineering (MDE)



Summary



Definitions - Basics



Model Driven Development

- “Model”
- “Driven”
- “Development”



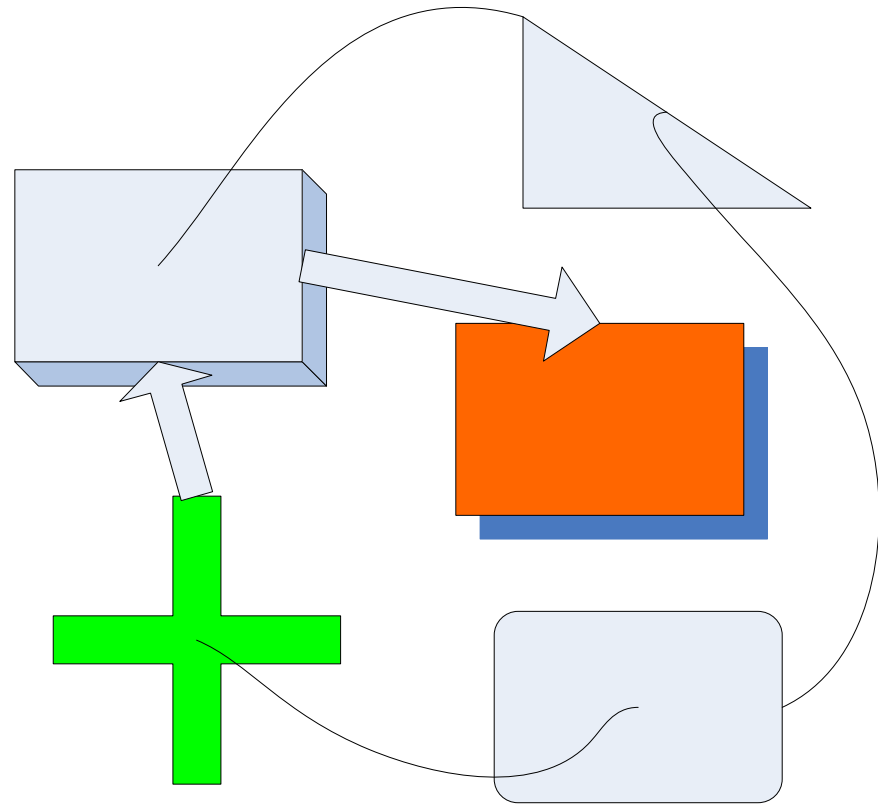
Model – what it is

- “a *simplified* representation of a system”¹
- Derivation – Latin *modus*, measure amount, limit, manner, kind, tone¹
- Primarily a model is *information* at a particular level of abstraction
- An *abstraction* of reality
[www.dictionary.com]

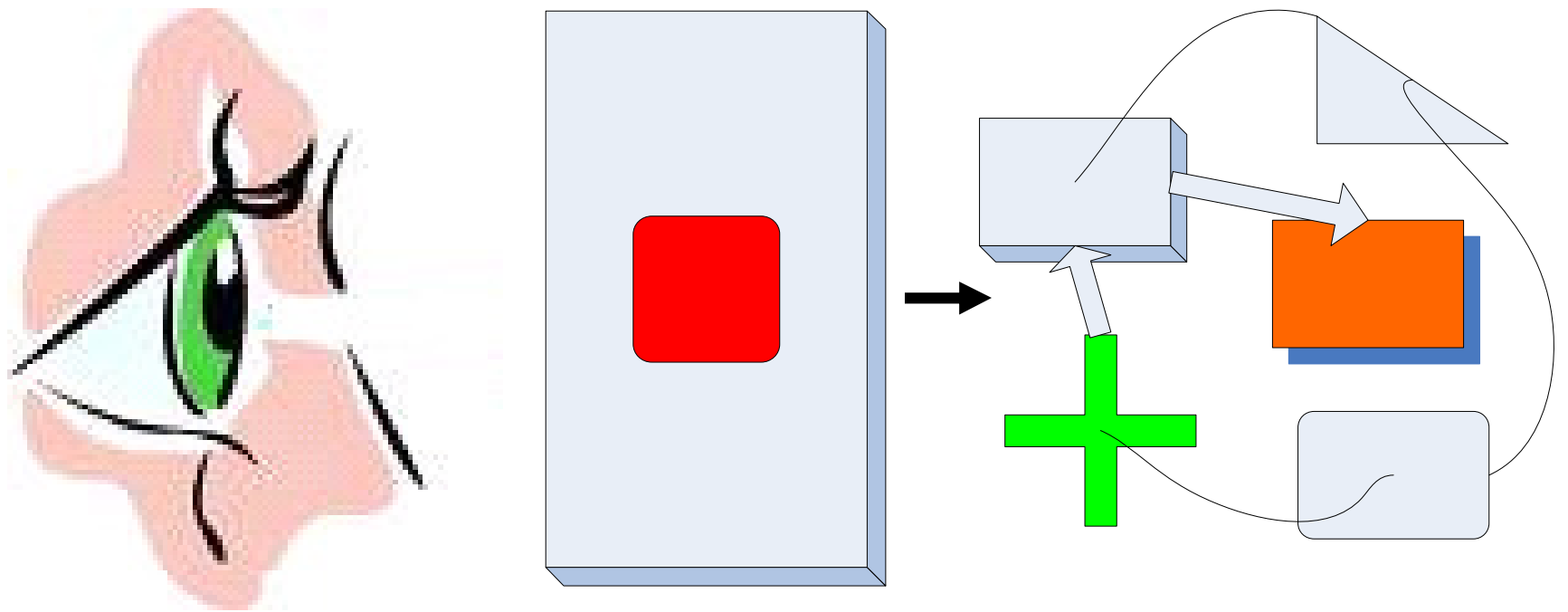


¹Random House Unabridged Dictionary 2nd Edition, Random House 1993

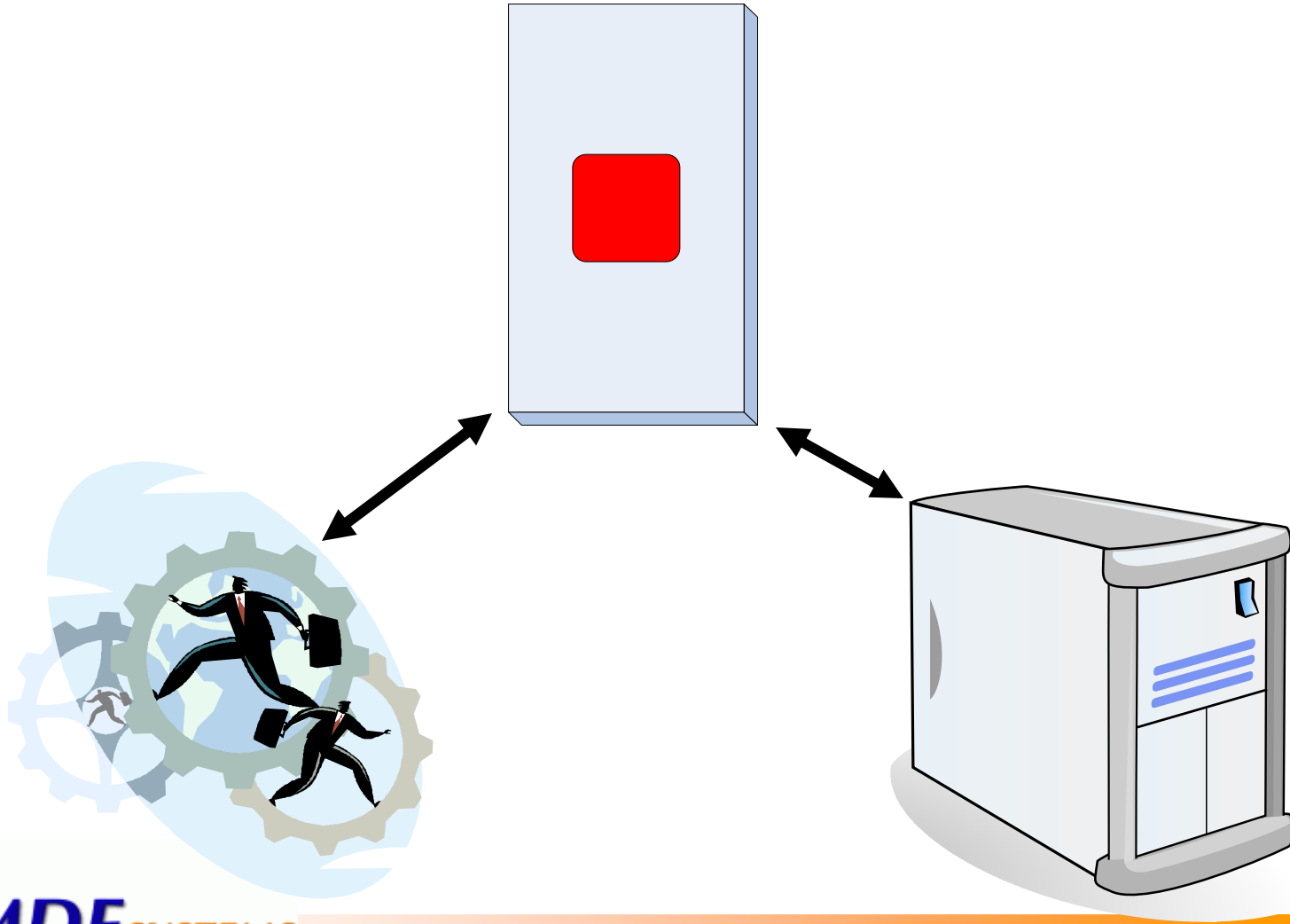
Model – Dealing with Complexity



Model – Dealing with Complexity



Models - participants



Abstract(ion)

- “Thought of as apart from concrete realities, specific objects, or actual instances
- “Expressing a quality or characteristic apart from any specific object or instance”¹
- Derivation: abs— away or away from, tract — to pull or draw¹ — relate to domain/product
- “Doing more with less” [Unknown]



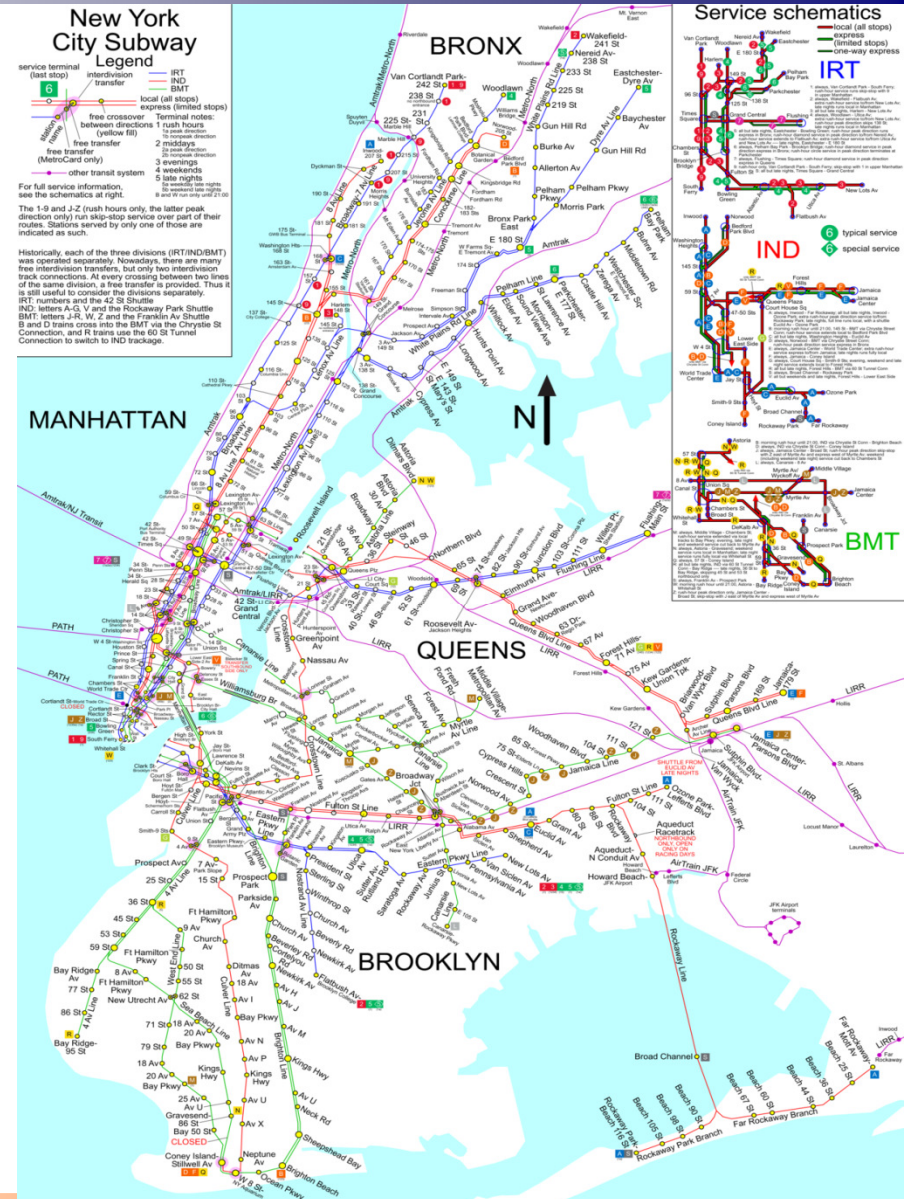
¹Random House Unabridged Dictionary 2nd Edition, Random House 1993

Abstraction – Five steps

- Selectively ignore detail not pertinent to the viewpoint
- Create a new element that encapsulates and relates the details pertinent to the viewpoint
- Give this new element a *name*
- Provide a connection between this new element and the details – High Fidelity
- Package the abstraction to be used by others



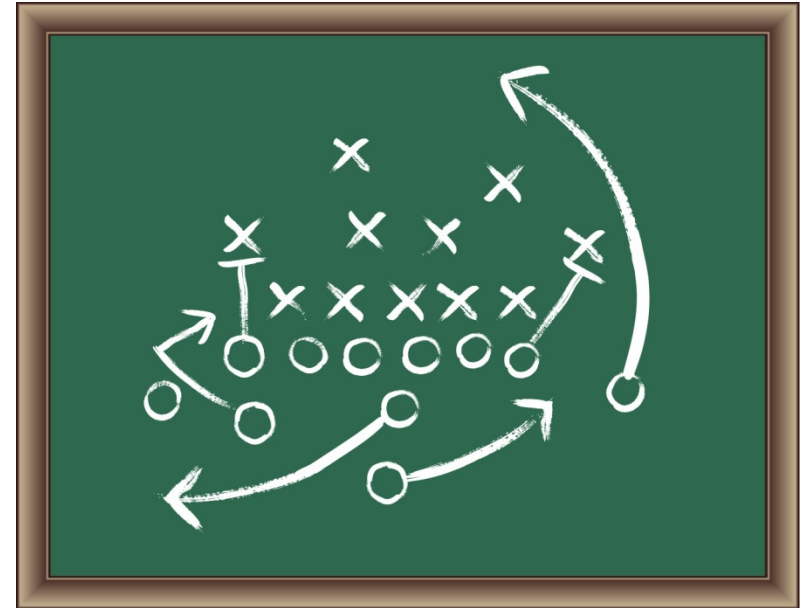
Abstraction



<http://www.nycsubway.org/maps/spui.html>

© Copyright MDE Systems 2010

Abstraction - Language



Groups of Abstractions

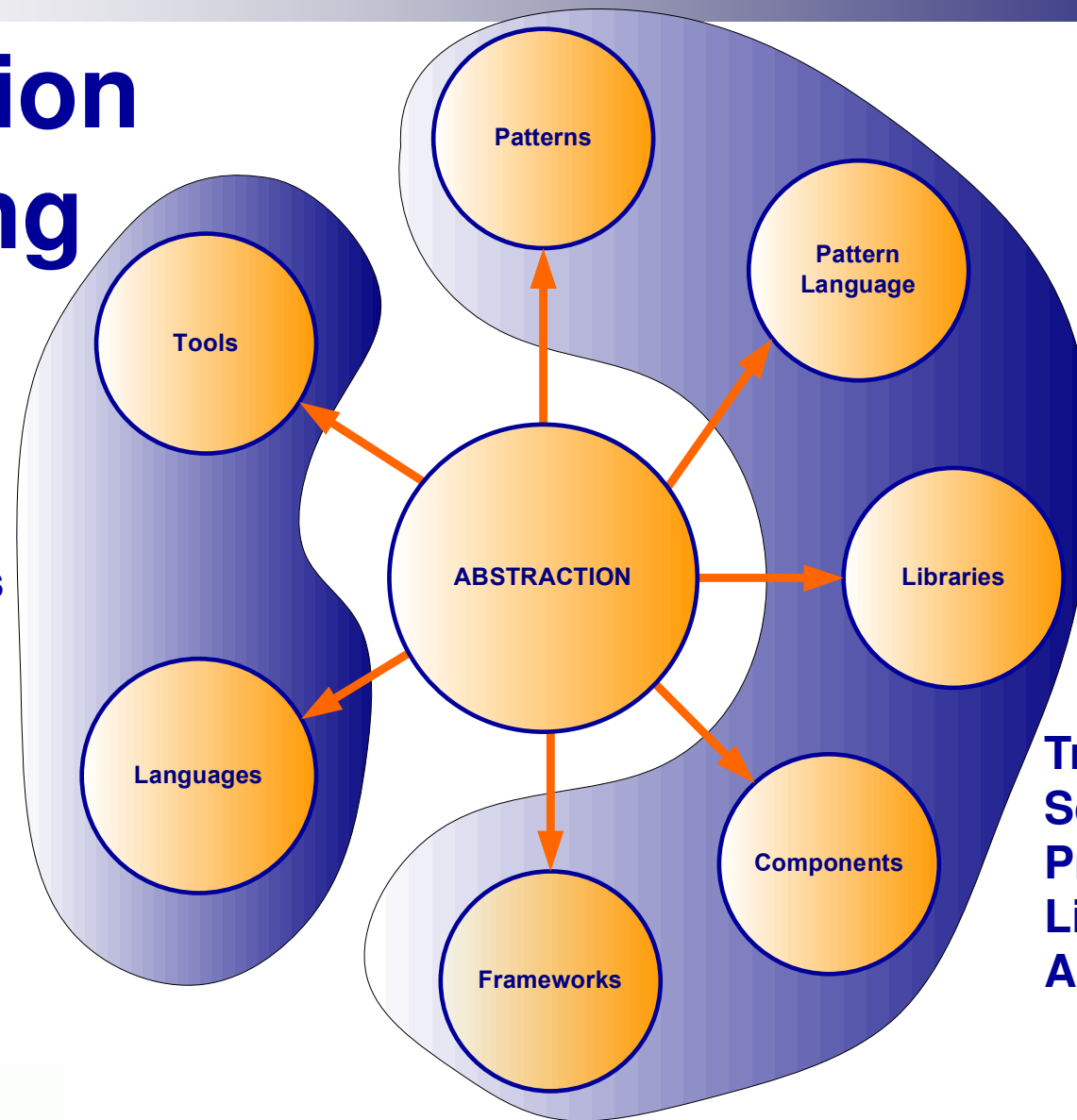
- Groups of abstractions are used to tackle complex problems
- Abstractions have particular relationships and can be combined in meaningful ways¹
- New names, new relationships
- Creation of a *vocabulary* and grammar = language¹



¹Software Factories, Greenfield et. al. Wiley 2004

Abstraction Packaging

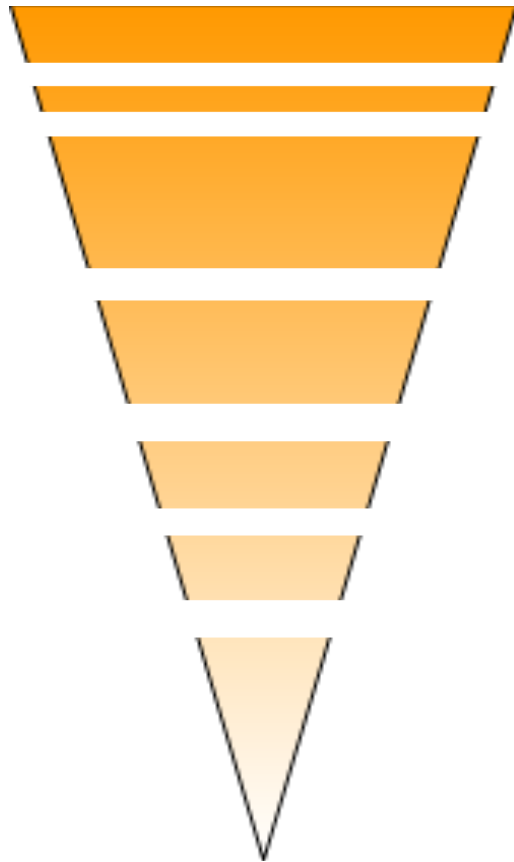
**MDD
Artifacts**



**Traditional
Software
Product
Line
Assets**

Abstraction

Productivity/Power



Refinement



Control



Misconceptions about Models

- Models must be visual or graphical
- Models must be declarative
- Modeling and programming are completely separate activities
- Modeling should be separated from programming



Driven

- Caused, sustained, or stimulated: *an export-driven economic recovery.*
- Powered, operated, or controlled: *a piston-driven airplane; a menu-driven software program.* [both from dictionary.com]
- To cause and guide the movement of¹
- To impel; constrain; urge; compel¹
- Contrast with oriented, centric, integrated



¹Random House Unabridged Dictionary 2nd Edition, Random House 1993

Develop(ment)

- In contrast with architecture, design, documentation
- To bring into being or activity, generate, evolve¹

Engineering

- the art or science of making practical application of the knowledge of pure sciences, as physics or chemistry, as in the construction of engines, bridges, buildings, mines, ships, and chemical plants.



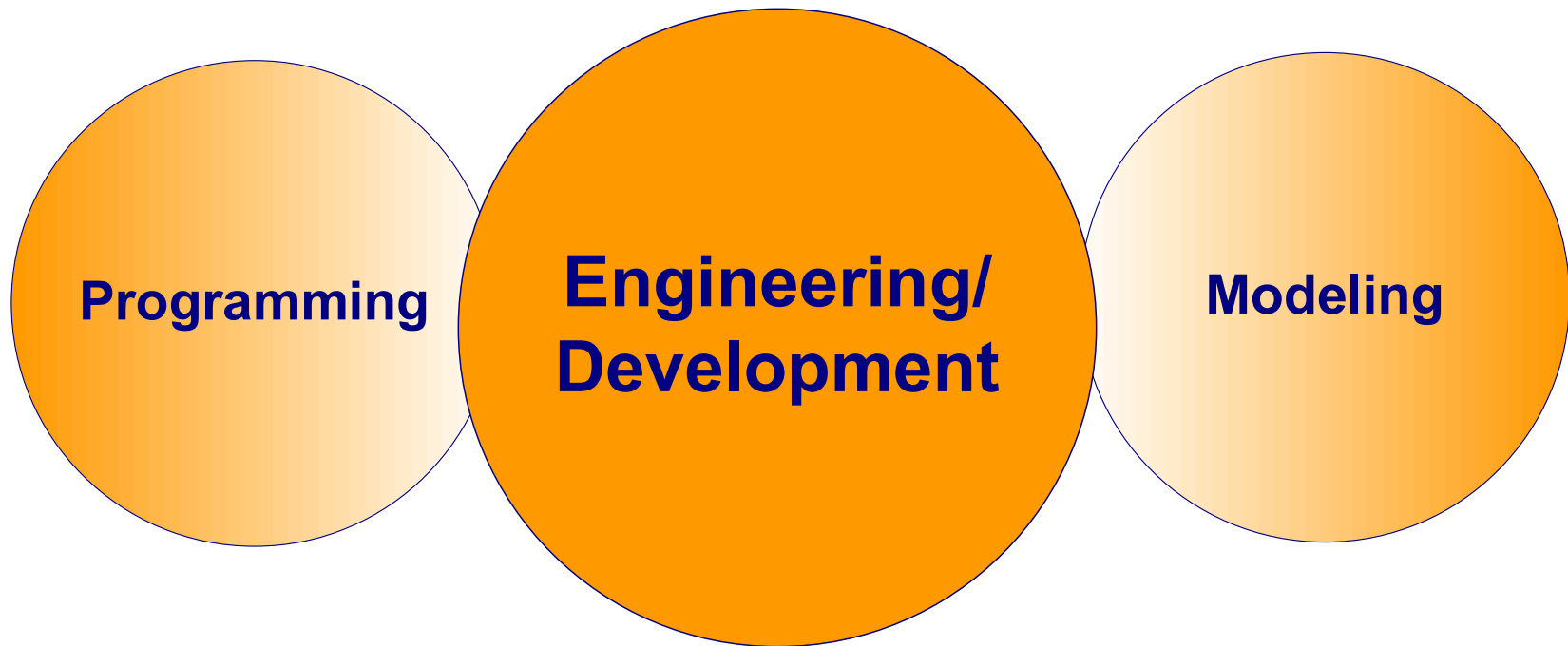
¹Random House Unabridged Dictionary 2nd Edition, Random House 1993

Model Driven Development

- Program with Models
- Create Models that can be processed by a machine to produce lower level abstraction artifacts
- Language - Editor - Generator



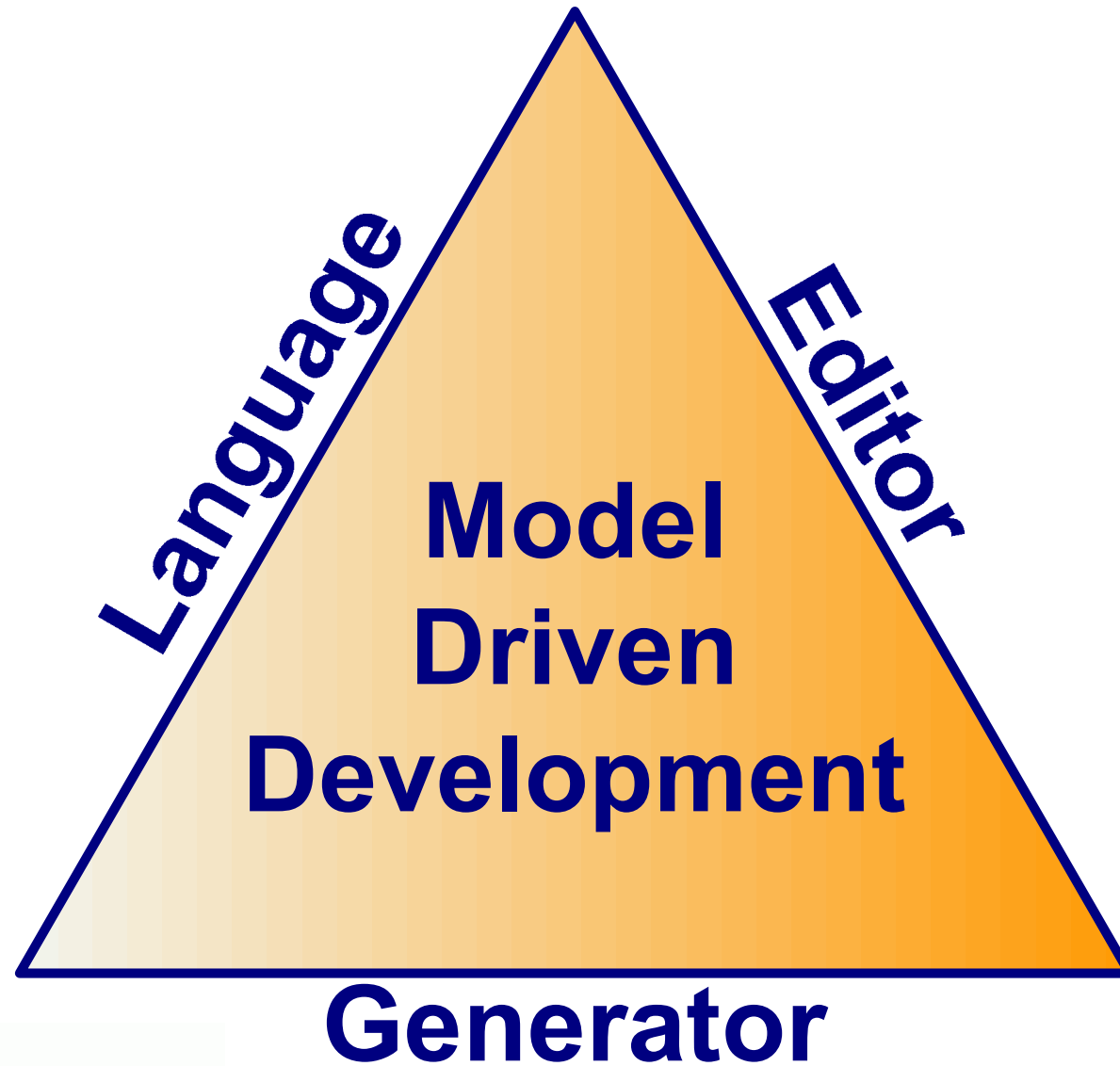
Programming/Modeling



Model Driven Development

The Anatomy and Pattern

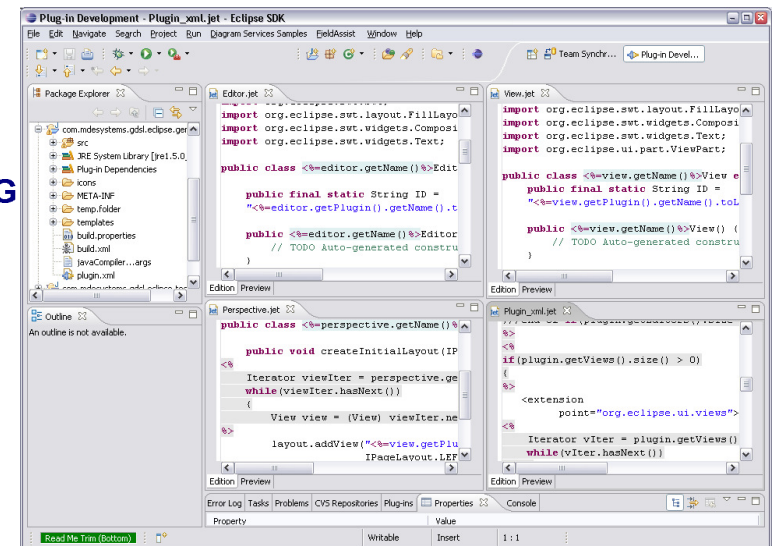
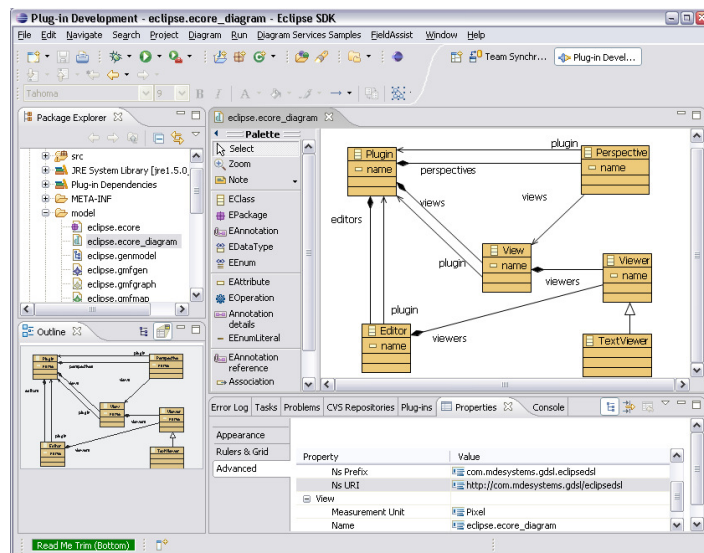
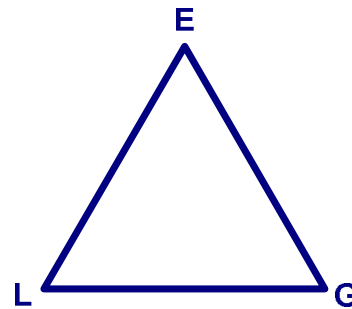
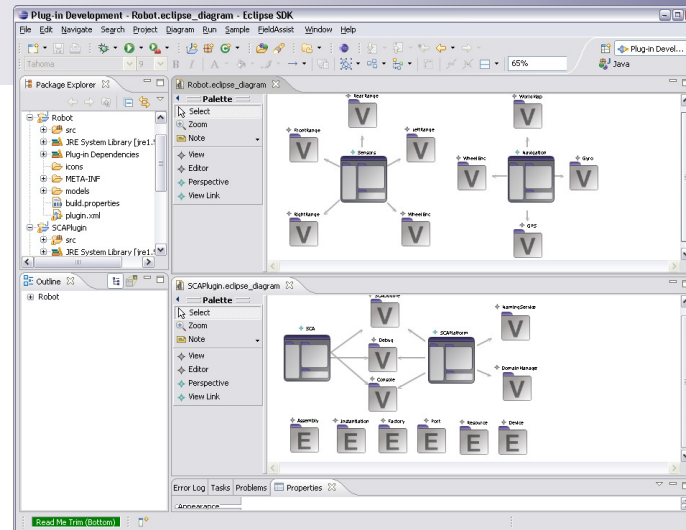


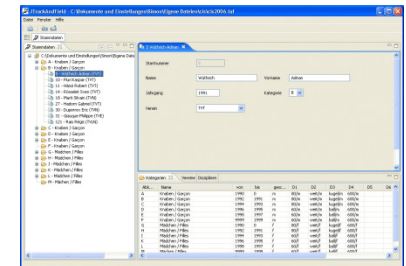
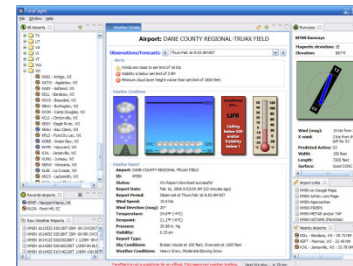
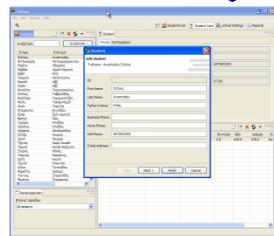
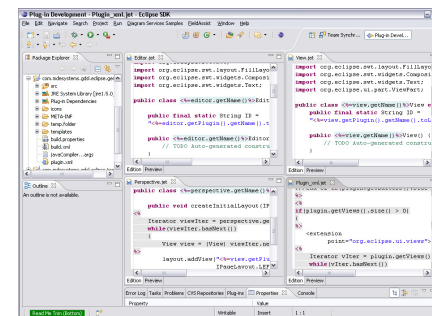
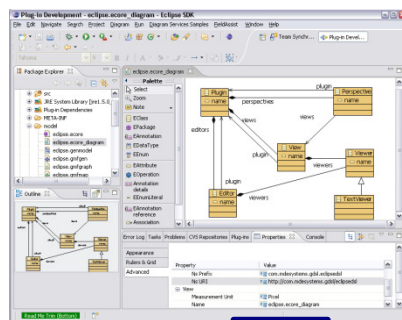
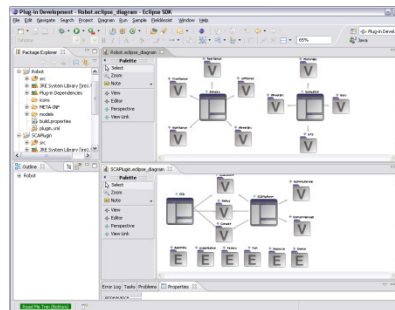


Purpose

- **Language**
 - define the abstract syntax, or grammar, upon which the editor, serialization concrete syntax, and in-memory concrete syntax are based.
- **Editor**
 - The purpose of the editor is to render, capture and express *design intent*
- **Generator**
 - Provides automatic ways to get to lower levels of more executable forms of code.
 - Provides the actual meaning or semantics to the higher levels of abstractions.





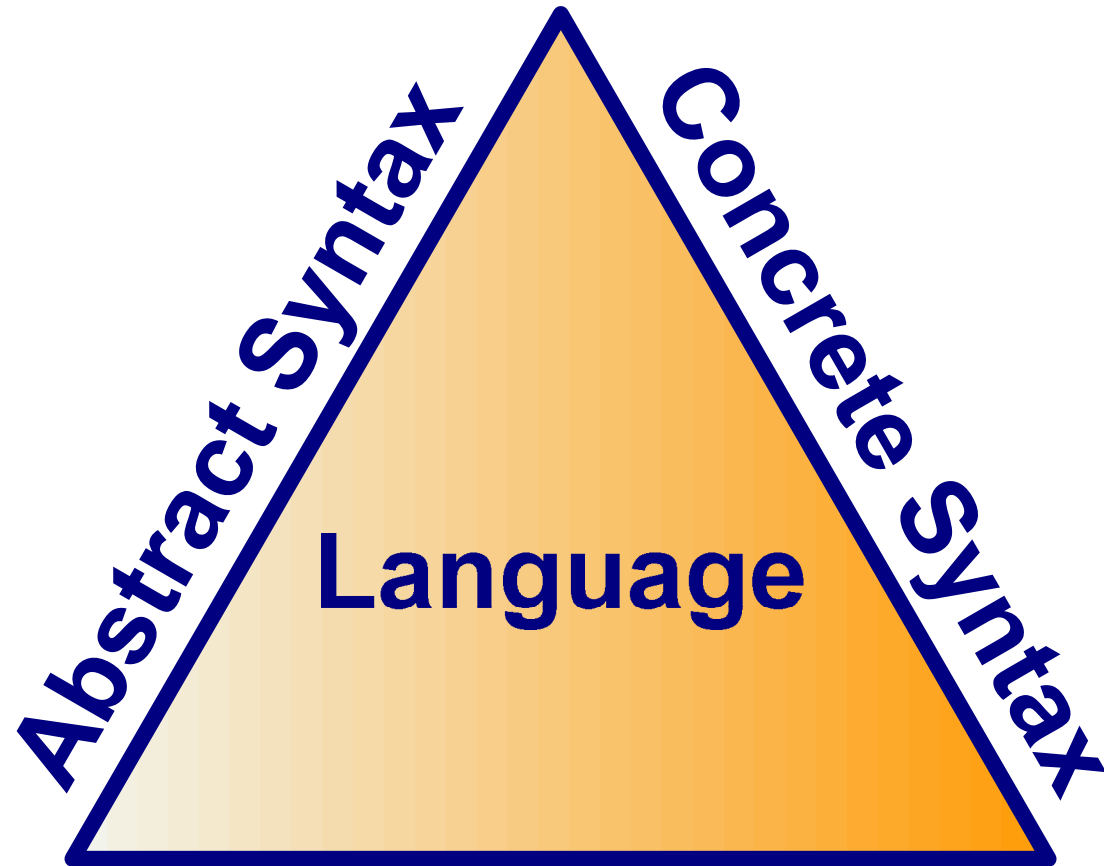


Synonyms

- Model Driven Development
- Domain Specific Language



Language



Semantics

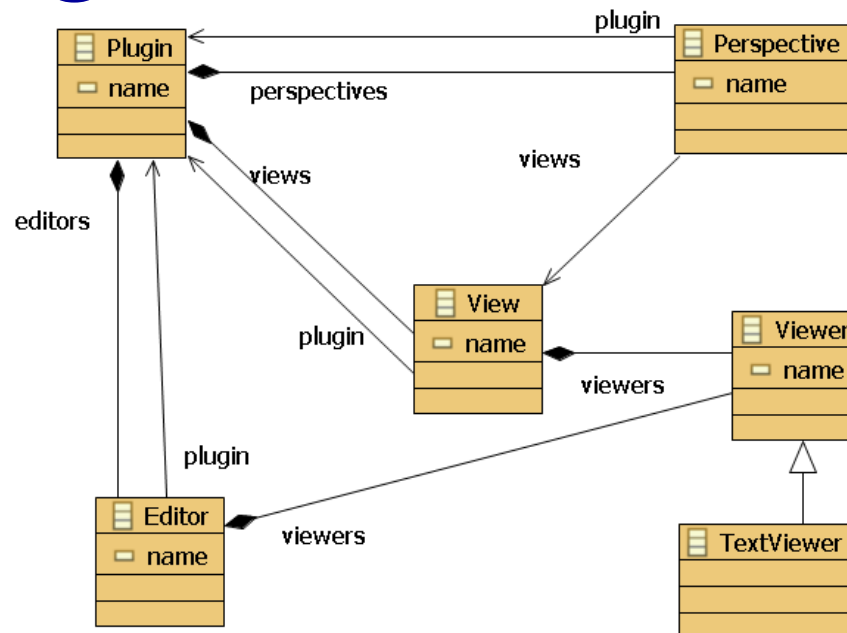
Language

Vocabulary
+ Grammar
= Language



Metamodel

Object oriented representation of the grammar of a language



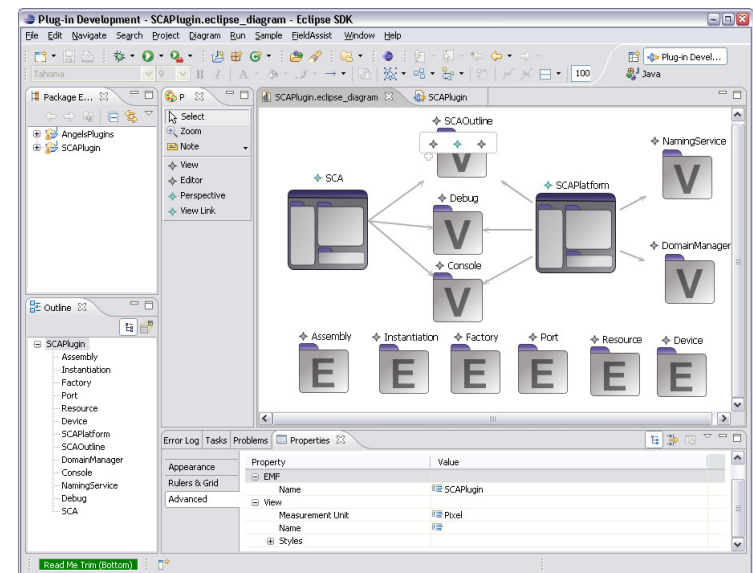
Domain Specific Language

- Model concepts found in a specific domain
- Contrast to General Purpose Language – encodes generic abstractions



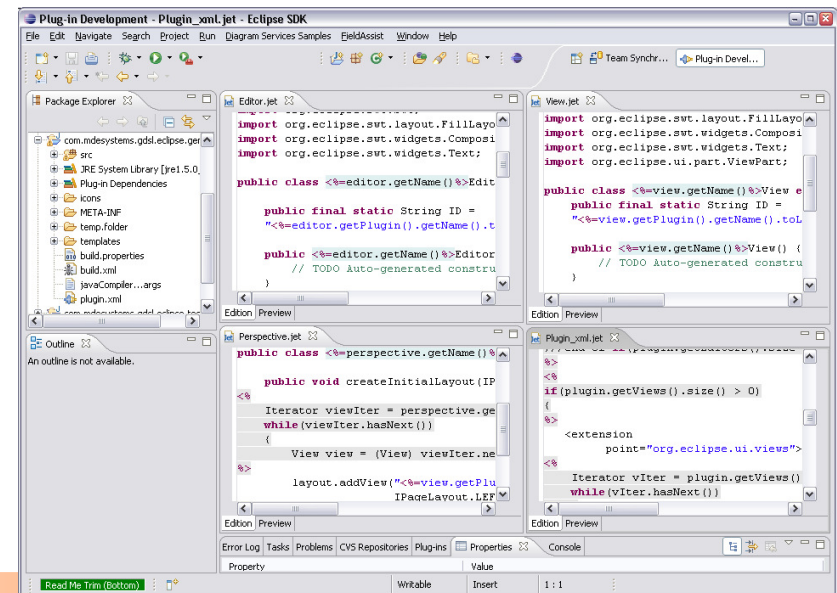
Editor

- Textual – Syntax and highlighting, intellisense
- Graphical – Domain Specific Graphics
- Meant for the human
- Nominated to a first class MDD participant
- Constraints at modeling time
- See the design
- Program in terms of domain elements and relationships
- Declarative



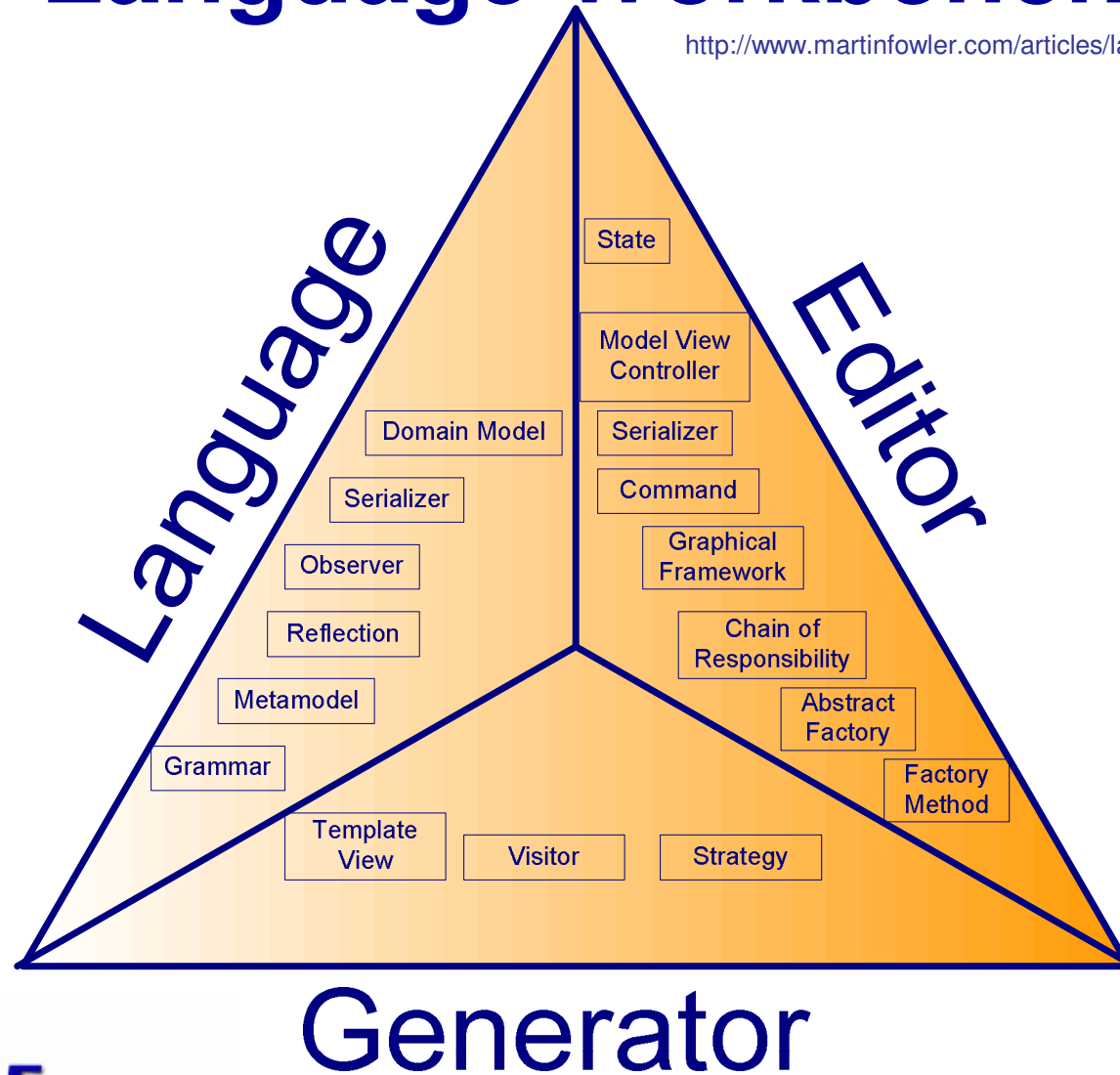
Generator

- Where the rubber meets the road in MDD
- Correct by construction
- Best Practice and Design Pattern replication
- Programming Power
- Maintainability – two sides of the same coin when bug occurs
- Portability of design
- Opens the world up to new programmers



Language Workbench¹

<http://www.martinfowler.com/articles/languageWorkbench.html>



Language Workbench



Artifacts needed

- Editors
- Views
- Actions
- Generators
- Constraints
- Debuggers
- Tests



Eclipse¹

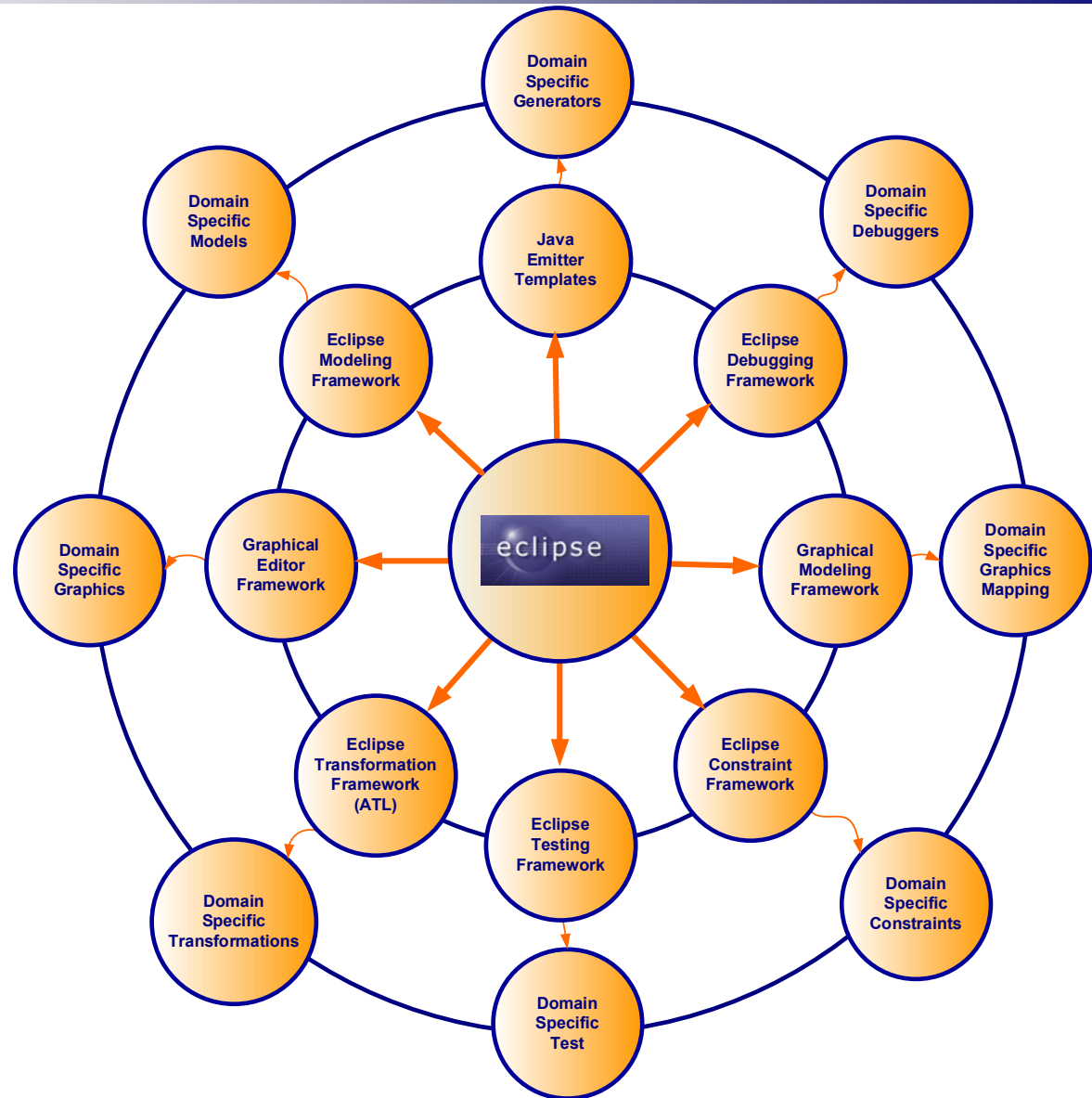


<http://www.martinfowler.com/articles/languageWorkbench.html>

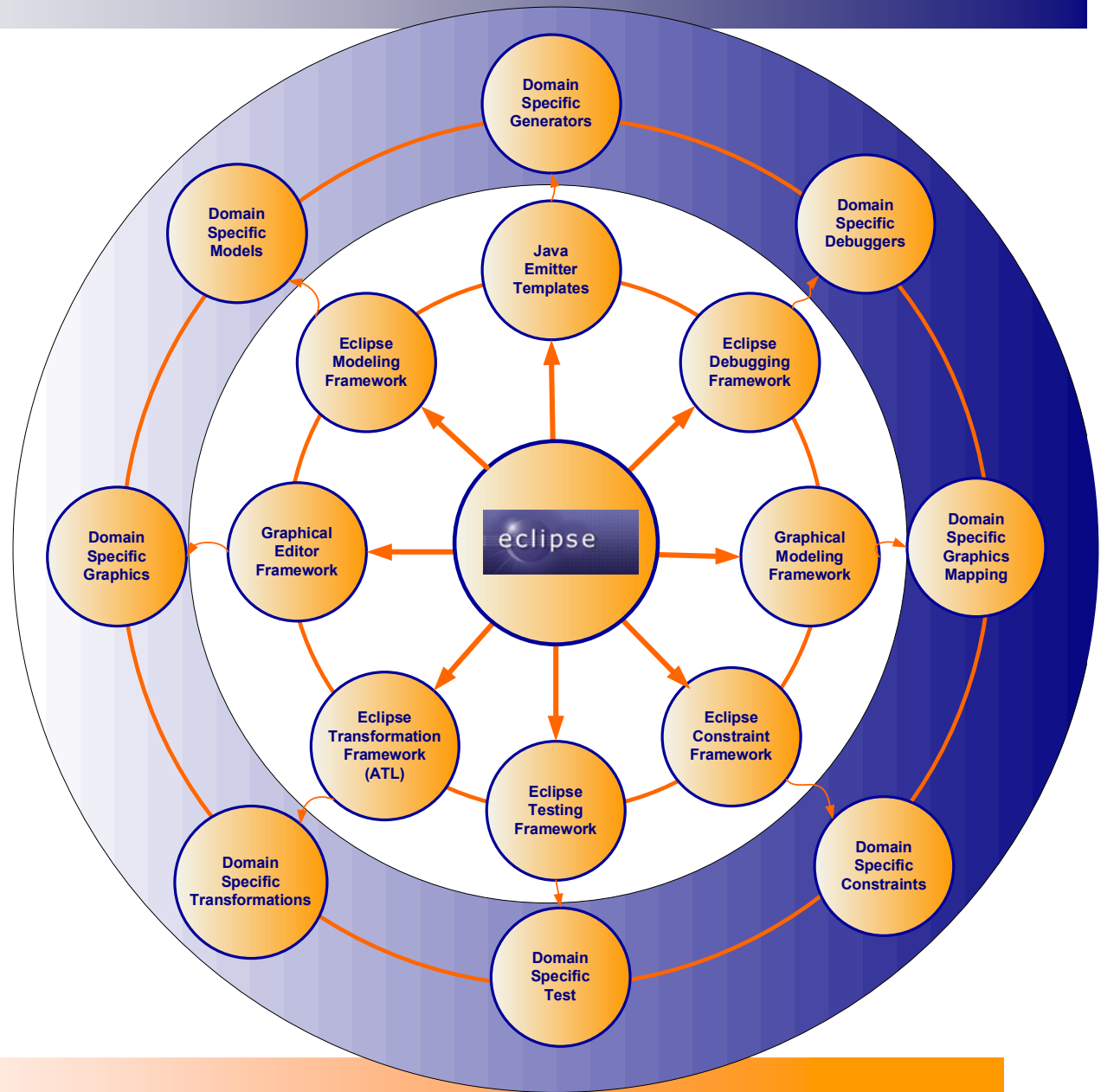
Eclipse – Language Workbench

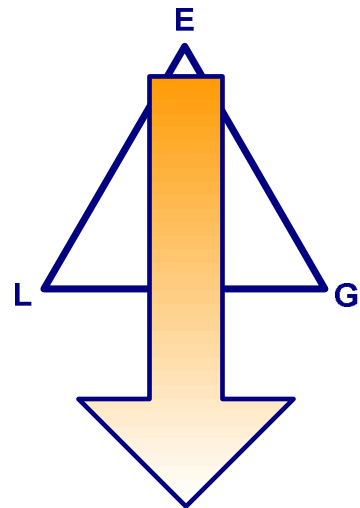
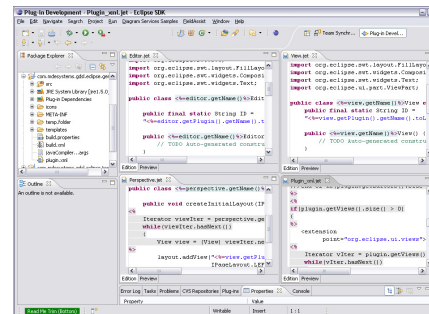
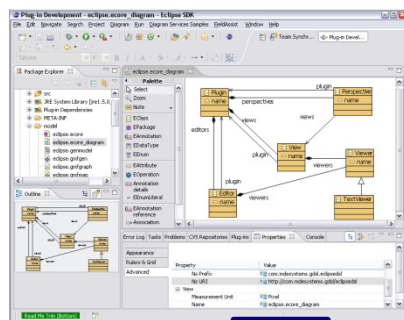
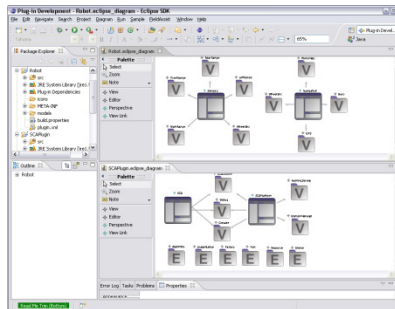
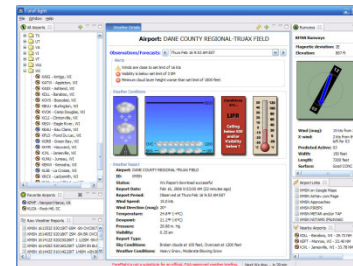
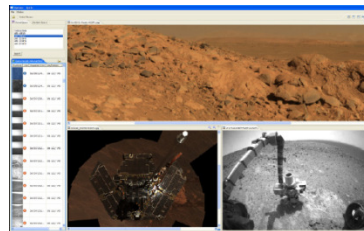


Eclipse as a Product Line



Eclipse as a Model Driven Development Engine



Index	Value	Unit	Label
1	1000	kg	Weight
2	2000	kg	Weight
3	3000	kg	Weight
4	4000	kg	Weight
5	5000	kg	Weight
6	6000	kg	Weight
7	7000	kg	Weight
8	8000	kg	Weight
9	9000	kg	Weight
10	10000	kg	Weight

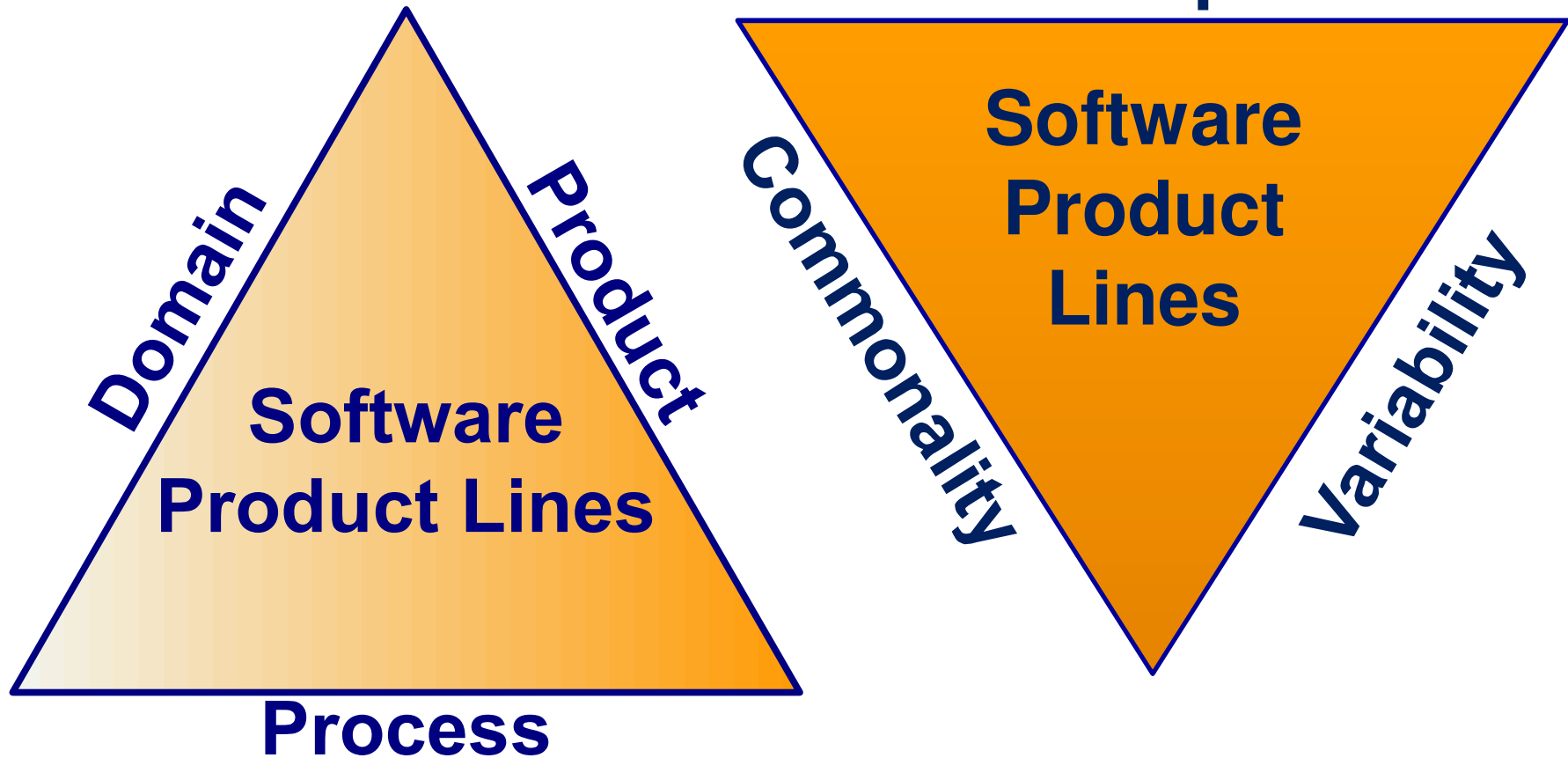


Software Product Lines (SPL)



Software Product Lines

Scope



Software Product Line

- “Software”
- “Product”
- “Line”



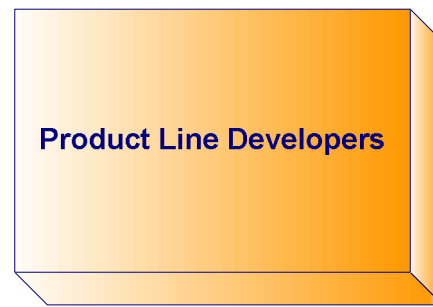
Software Product Line

- Contrast with one-off system
- “The Split” – Domain Engineering/Product Engineering
- Definition a *family* of software products
 - members vary
 - sharing common features”
- Aren’t all software products product lines?



The Central Dichotomy

- Frameworks and MDD tools
- Different skills, experience and expertise

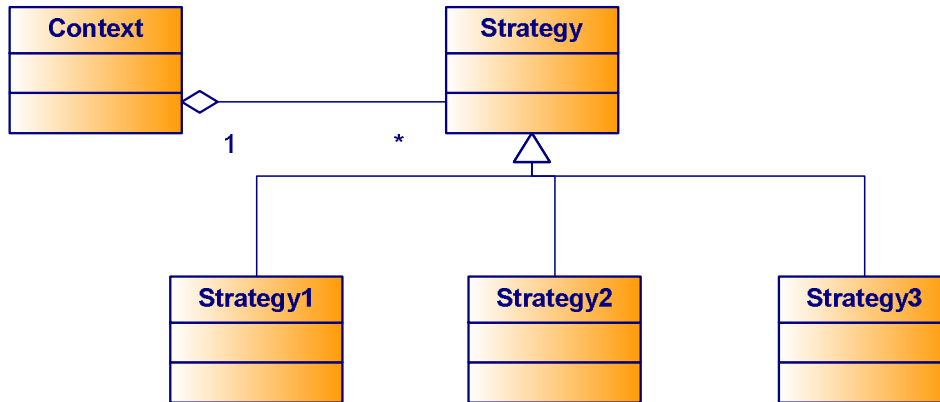


The Strategy Pattern

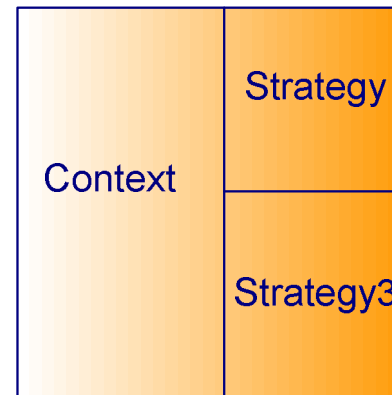
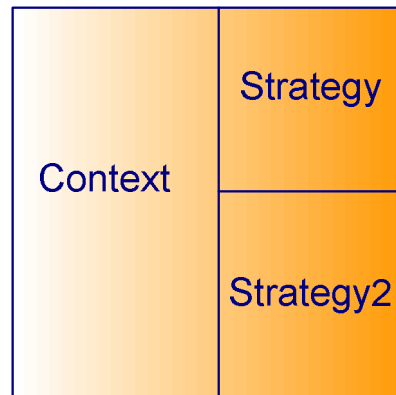
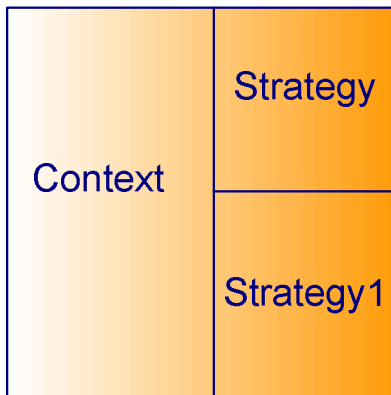
- Abstraction
- Commonality Variability
- Family of systems
- Complexities introduced
- Making it a language



Strategy Pattern – mini product line

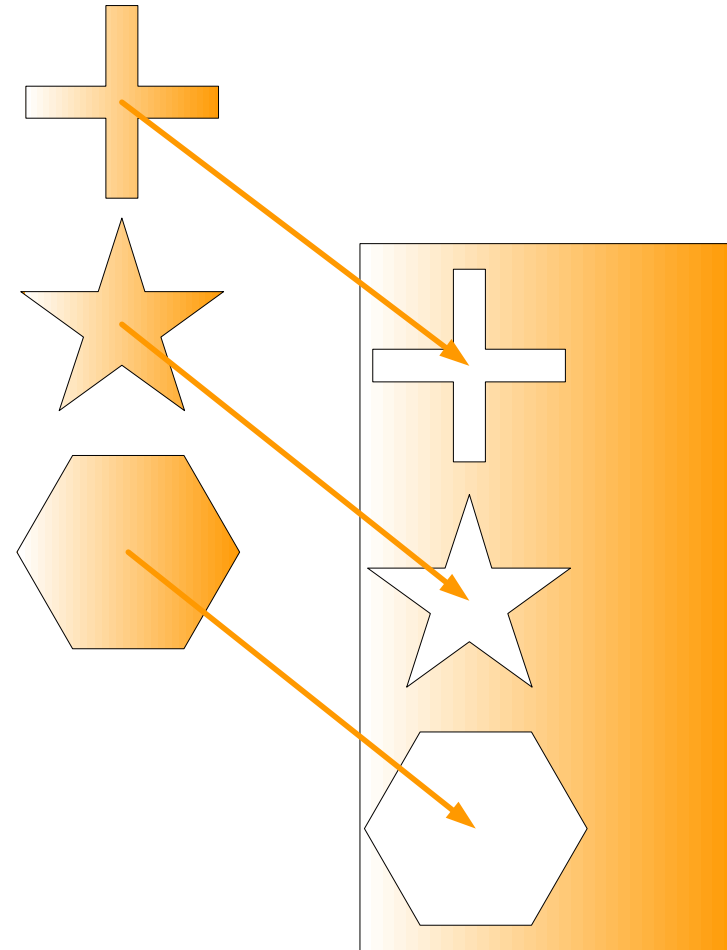


- ☐ Strategy1
- ☒ Strategy2
- ☐ Strategy3

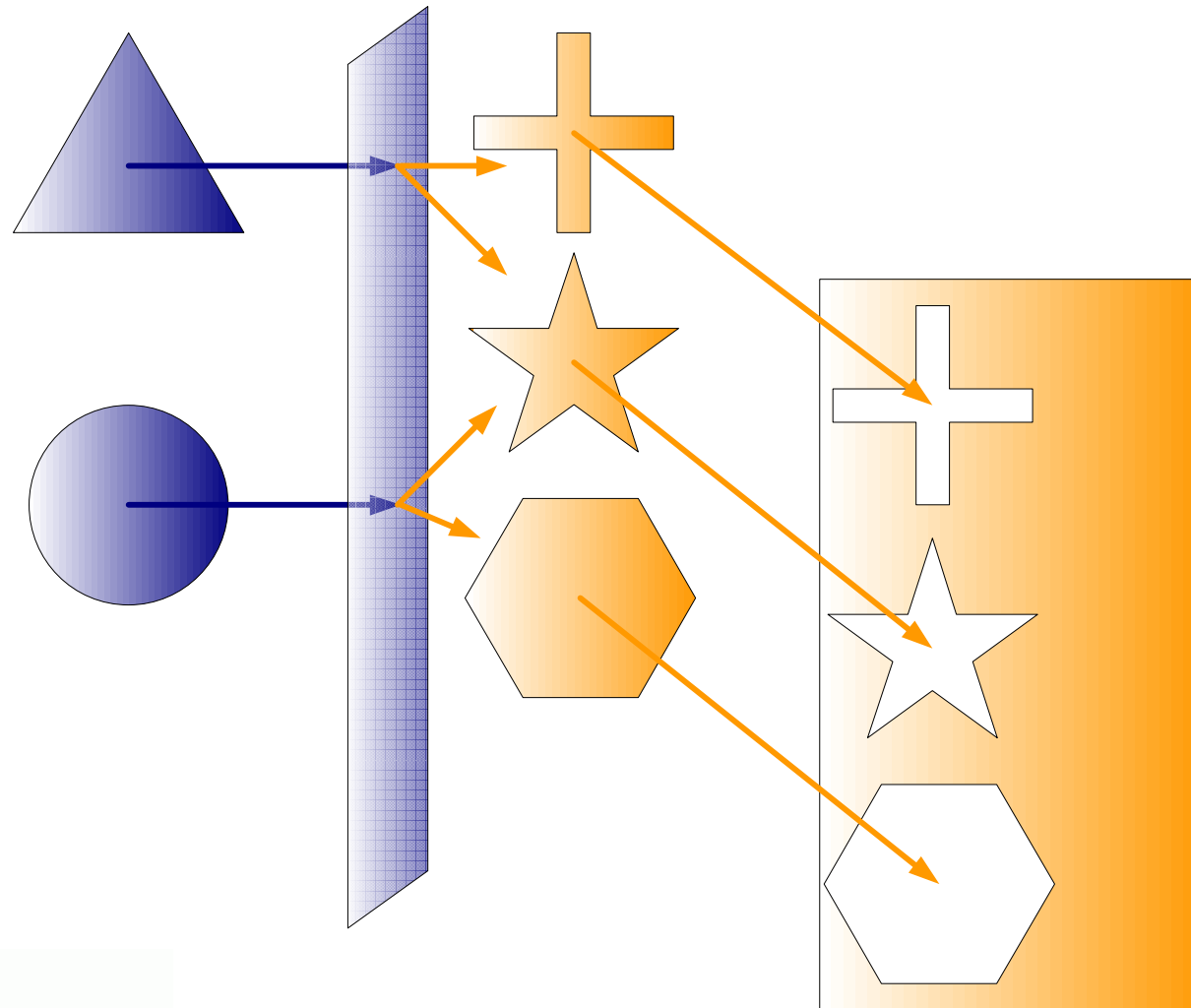


Framework

- Abstract architectures and classes that provide a partial solution to a family of problems.



DSLs, Models and Frameworks



MDD and SPL

- Synergy
- $MDD + SPL = SPL$
- MDD brings SPL into the hands of the Domain Expert
- Uniting the various aspect of SPL and Software Development

NOTE



MDD and SPL

- Product Line Engineering
 - isolation of the scope, commonalities and variabilities of a particular domain[1].
- Model Driven Engineering
 - *capturing* of these commonalities and variabilities in formal models so as to provide effective domain specific *tooling*.
- Synergy between MDE and Product Line Engineering.
- Tools then become the perfect medium with which developers can express, manipulate, analyze, simulate and generate the commonalities and variabilities of a product line.
- MDE tools are frequently the overarching central and aligning production asset provided to the product engineering group from the product line engineering group.



¹ Commonality and Variability in Software Engineering, Coplien IEEE Software 1998

Agile Software Development (ASD)



Enabling Technologies and Approaches

- Test Driven Development
- Continuous Integration
- Tracer Bullets



Continuous Integration

Tree is open (mac dbg->ignore, needs to be restarted)

continuous | Builds: [symbols](#) | [perf](#) | [status](#)

Development: [source](#) | [reviews](#) | [bugs](#)

Chromium: [news](#) | [blog](#) | [dev](#) | [support](#)

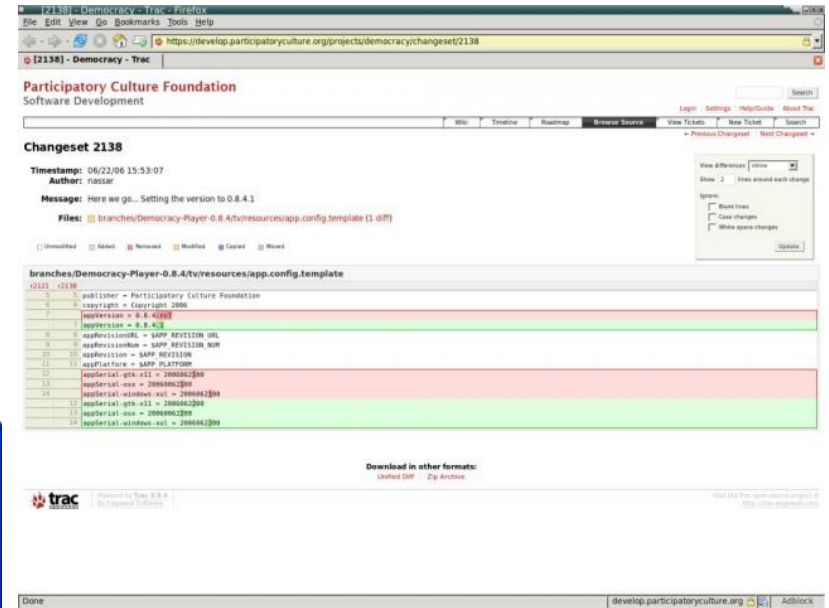
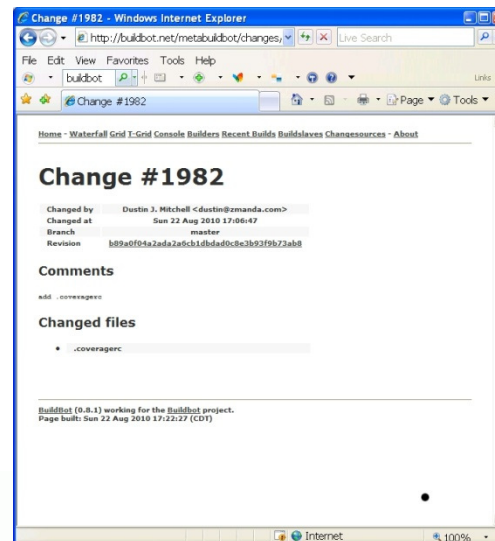
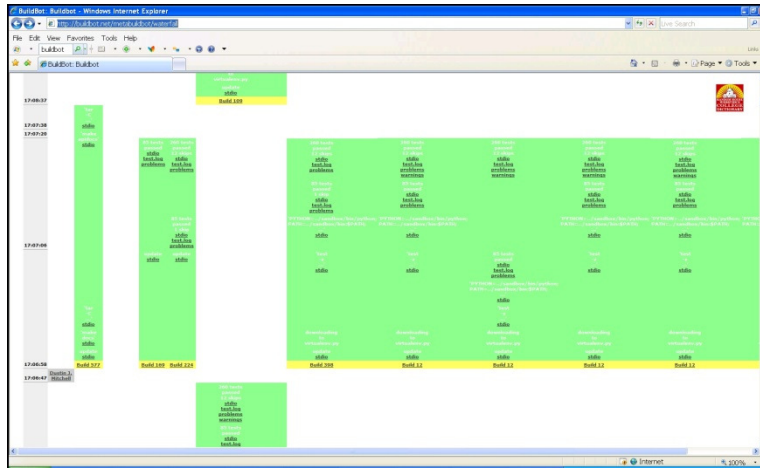
Sheriffs: [paulj](#), [nsylvain](#), [ojan](#)

Chromium last build	failed compile	build successful	build successful	build successful	failed ui_tests ui_tests_singleproc	failed compile
current activity	building ETA in 1 mins, 7 secs at 17:04:41 4 pending	building ETA in 8 mins, 30 secs at 17:12:04	building ETA in 1 mins, 28 secs at 17:05:01 1 pending	building ETA in 11 mins, 49 secs at 17:15:22	building ETA in 10 mins, 26 secs at 17:14:00	building ETA in 35 mins, 8 secs at 17:38:42 3 pending

time (PST)	changes	Chromium XP	Chromium Builder	Chromium Builder (dbg)	XP Tests	XP Tests (dbg)	XP Tests (purify)
		page_cycler_int2 stdio	compile stdio	update stdio	ui_tests stdio	ui_tests_singleproc (--single-process) stdio	purify test; unit stdio
16:14:33			check deps stdio				
16:14:47			tackkill stdio				
16:14:28			update stdio				
			update scripts stdio	update scripts stdio			
			svckill stdio	svckill stdio			
16:14:20			Chromium Builder Build 1428	Chromium Builder (dbg) Build 1344			



Continuous Integration



Test Driven Development, Automated Test, Continuous Integration

- When these happen and how often
- TDD – Refinement
- TDD – Abstraction
- TDD – Scope
- TDD – Commonality Variability
- CI – Refinement
- CI – Scope
- CI – Conway's Law
- Testing – forcing a family of two
- Mocking Technologies – increased quality



Distributed Real-time Embedded (DRE)



Elements

- Patterns
- Patterns Languages
 - Server Components
 - Remoting
 - Networked
 - Distribution
- Middleware
- Relation to MDE, SPL and ASD



Bringing it All Together



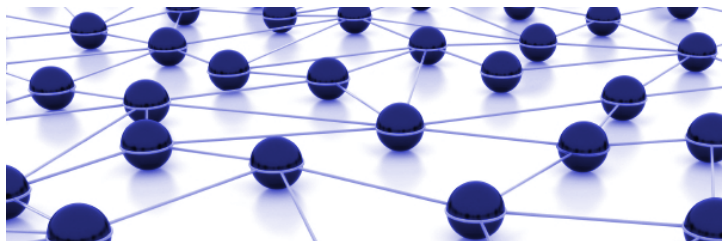
MDE



ASD



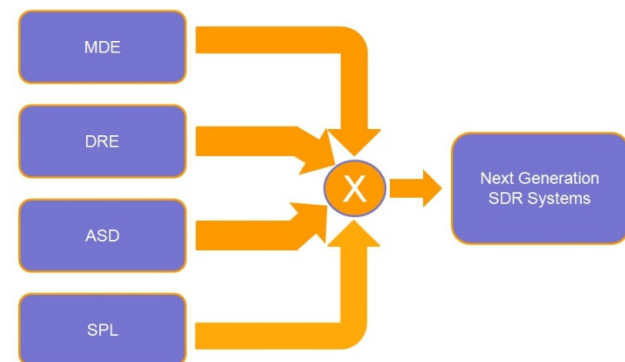
DRE



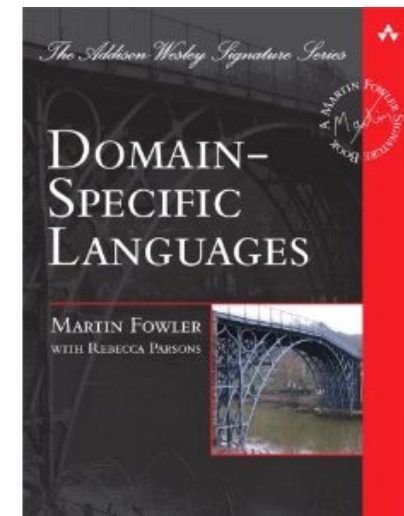
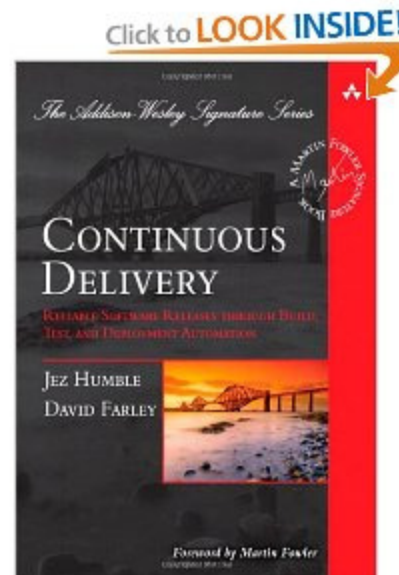
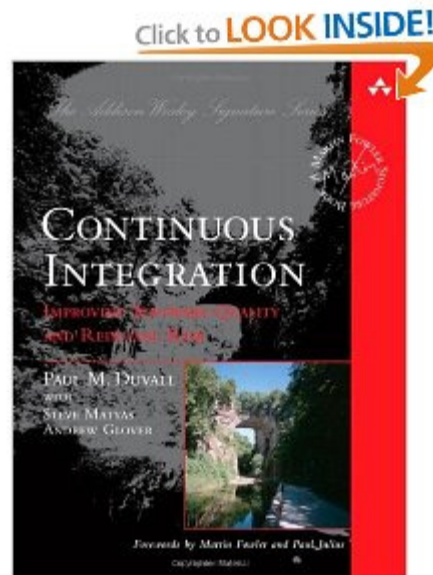
SPL



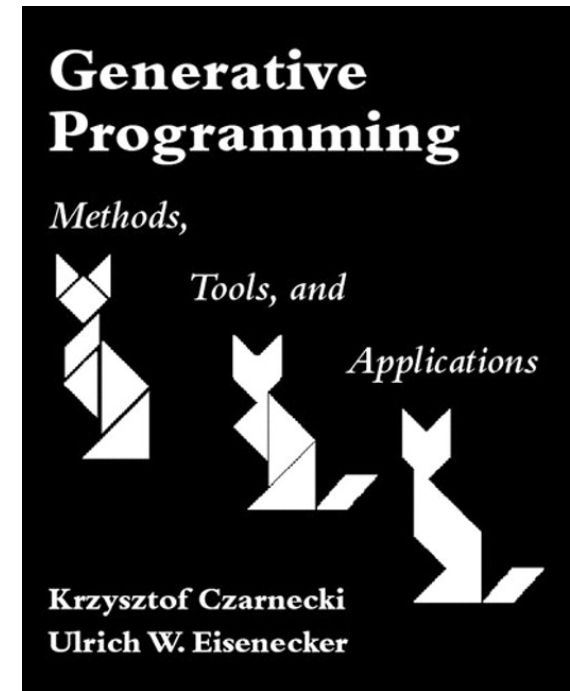
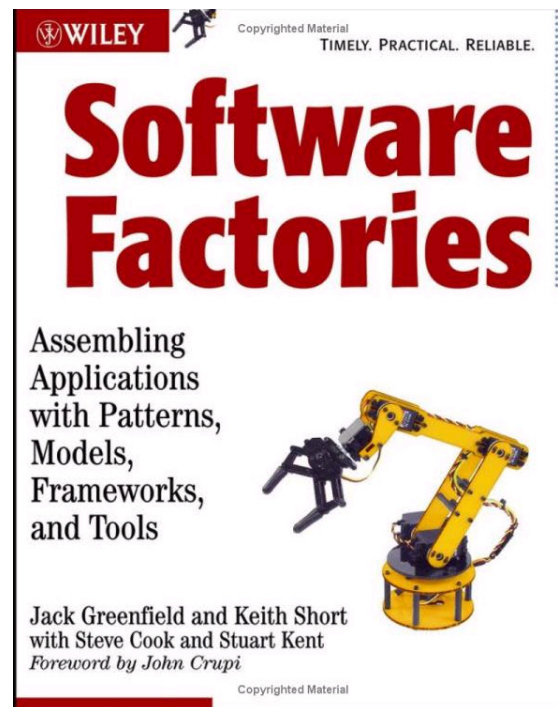
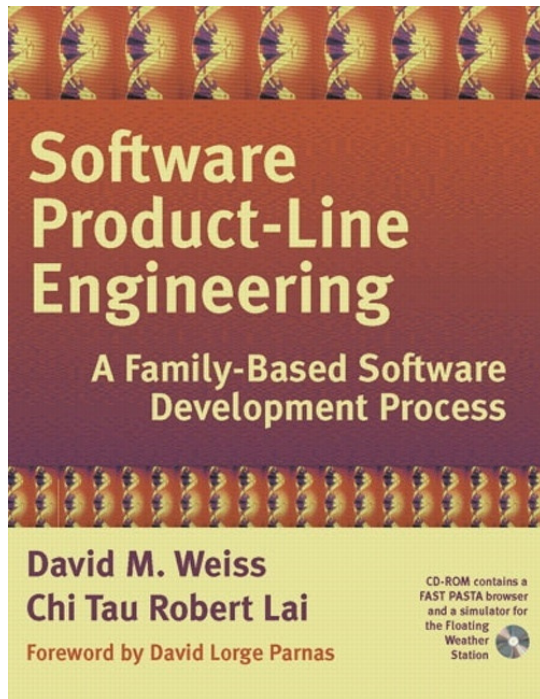
SDR



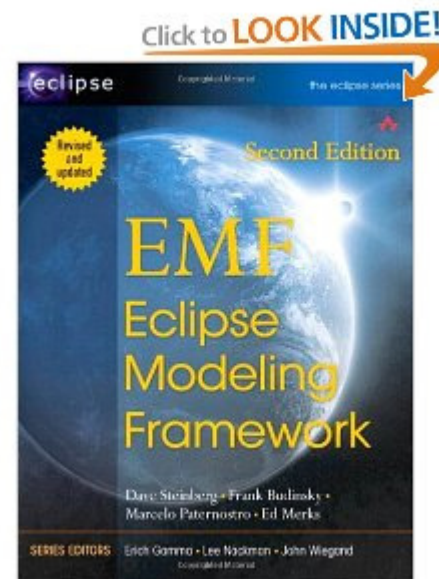
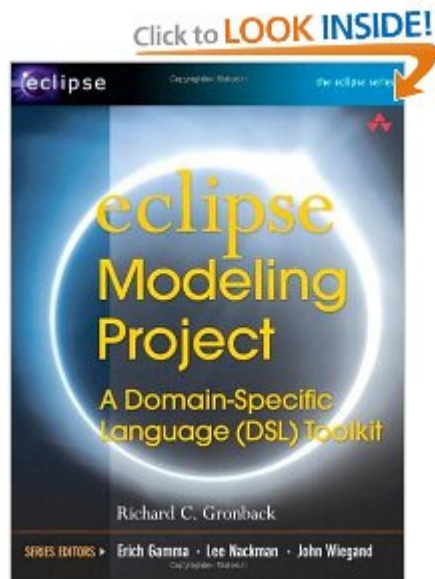
Books



Books



Books

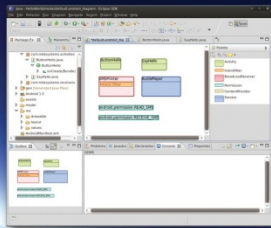




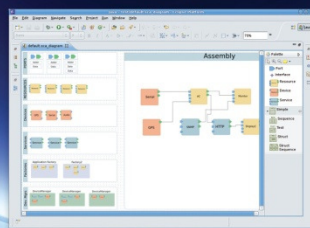
A New Dawn of Programming and Modeling

Tools to Express Yourself in Your Domain

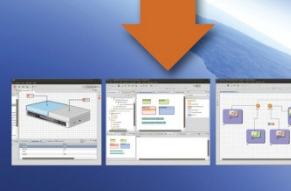
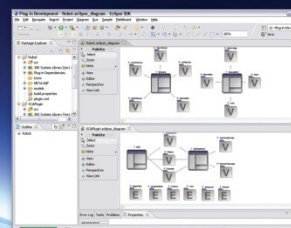
Infrastructure that Works with Tools



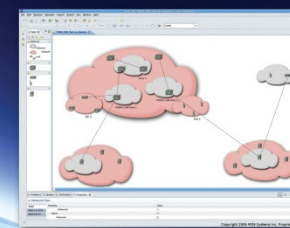
www.mdesystems.com



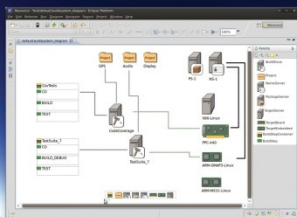
www.mdesystems.com



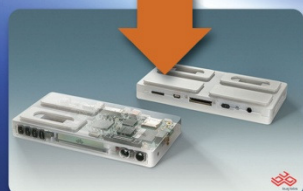
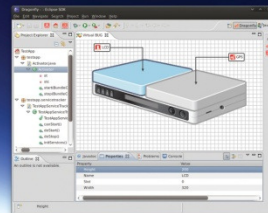
www.mdesystems.com



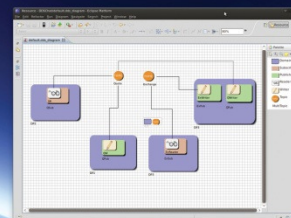
www.mdesystems.com



www.mdesystems.com



www.mdesystems.com



www.mdesystems.com



YOUR
DOMAIN
SPECIFIC
LANGUAGE
HERE

YOUR
DOMAIN

www.mdesystems.com

More Information

Website: www.mdesystems.com

News: www.mdesystems.com/news.html

Events: www.mdesystems.com/events.html

Phone: 718-451-6668

Email: bruce.trask@mdesystems.com
angel.roman@mdesystems.com

