

# INF5120 and INF9120

## ”Modelbased System development”

Lecture 1: 16.01.2017

Arne-Jørgen Berre

[arneb@ifi.uio.no](mailto:arneb@ifi.uio.no) and [Arne.J.Berre@sintef.no](mailto:Arne.J.Berre@sintef.no)

# Welcome to INF5120 and INF9120

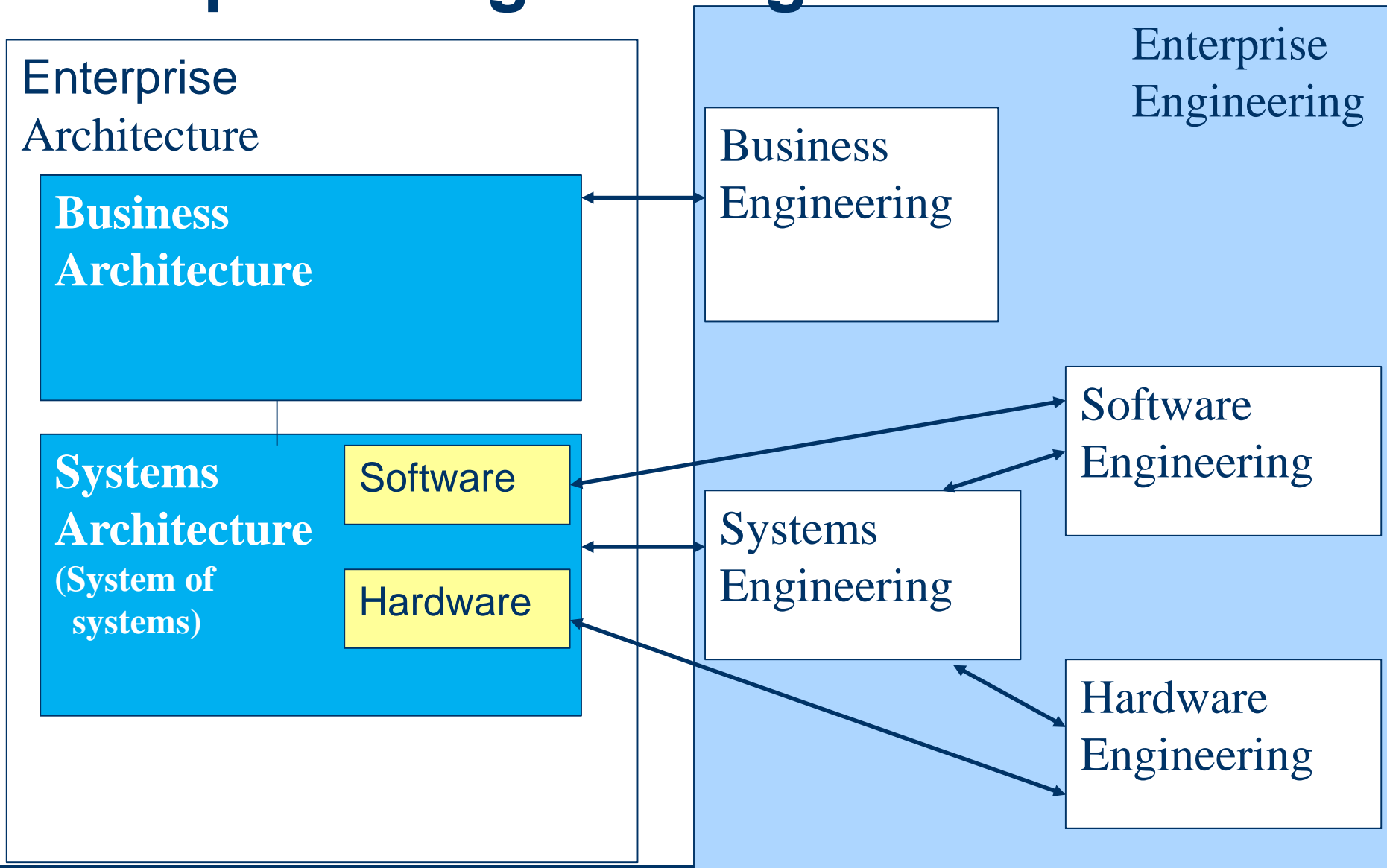
## “Model based System development”

- Model based System Development
  - <http://www.uio.no/studier/emner/matnat/ifi/INF5120/>
- Lecturers:
  - Arne-Jørgen Berre
  - Guest lecturers
  - Email: [inf5120-forelesere@ifi.uio.no](mailto:inf5120-forelesere@ifi.uio.no)
  - Collaboration with Professor Øystein Haugen, Østfold University College, Halden (previous at SINTEF and UiO) – on Cyber Physical Systems and IoT
- Teaching Assistant responsible for Obligatory exercises:
  - Imad Munir
  - Extra support: Erik Forsen
  - Email: [inf5120-oppgaver@ifi.uio.no](mailto:inf5120-oppgaver@ifi.uio.no)

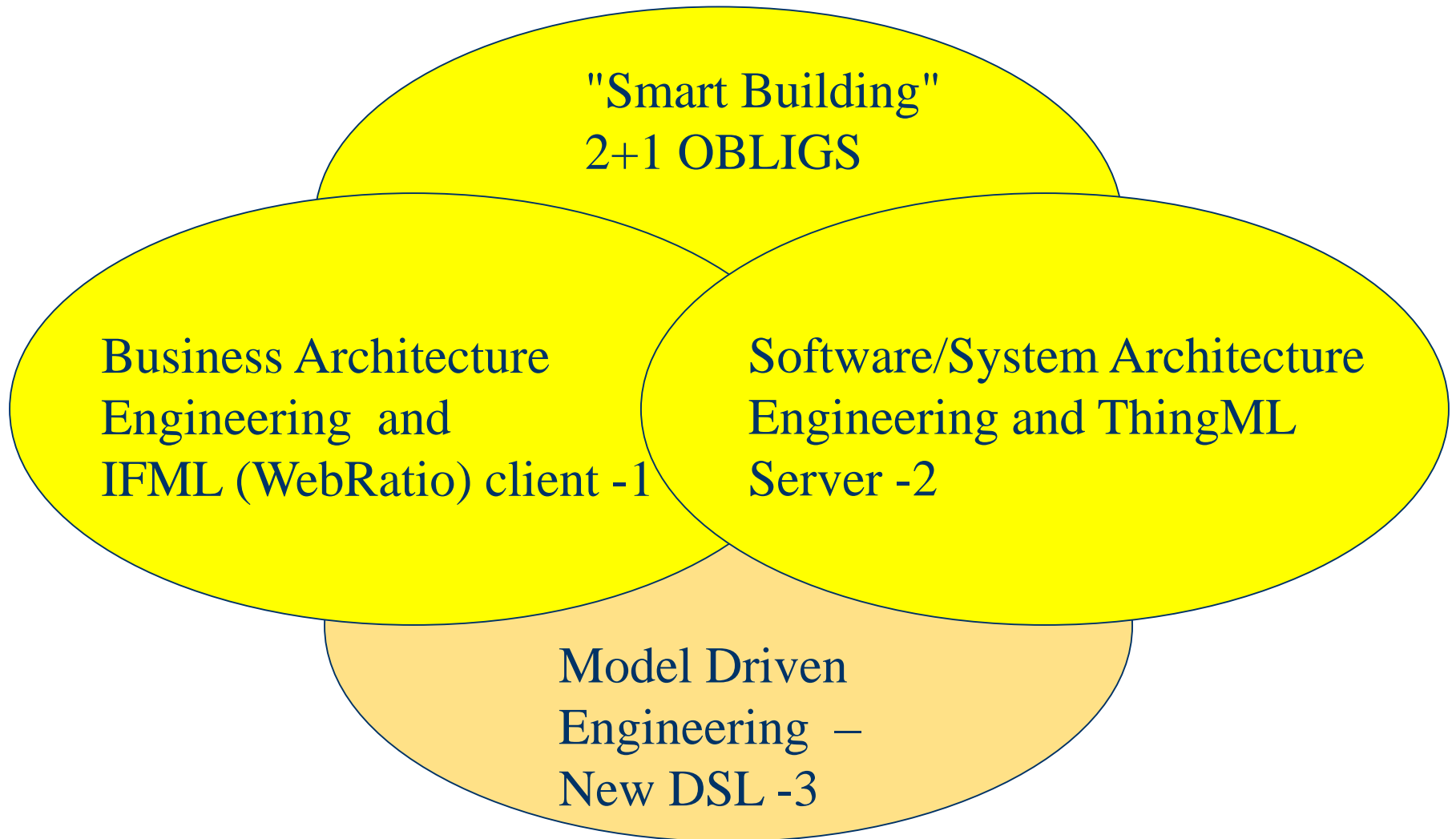
# 3 parts of the course

- MDE-Client: (Business architecture engineering and Requirements models, with service innovation and design)
- MDE-Server: (System and Software Architecture Engineering - Model Driven system architecture and realisation)
- MDE-DSL (Model Driven Engineering) – Modeling of Structure and Behaviour in "systems"- Design of domain specific languages and editors

# Enterprise Architecture and Enterprise Engineering



# Course components



# Course parts (16 lectures) - 2017

- January (1-3) (Introduction to Modeling, Business Architecture and the Smart Building project):
- 1-16/1: Introduction to INF5120
- 2-23/1: Modeling structure and behaviour (UML and UML 2.0 and metamodeling) - (establish Oblig groups)
- 3-30/1: WebRatio for Web Apps/Portals and Mobile Apps – and Entity/Class modeling – (Getting started with WebRatio)
  
- February (4-7) (Modeling of User Interfaces, Flows and Data model diagrams, Apps/Web Portals - IFML/Client-Side):
- 4-6/2: Business Model Canvas, Value Proposition, Lean Canvas and Essence (Smart Building project) - User stories and Use case
- 5-13/2: IFML – Interaction Flow Modeling Language, WebRatio advanced – for Web and Apps
- 6-20/2: BPMN process, UML Activ.Diagrams, Workflow and Orchestration modelling value networks
- 7-27/2: Modeling principles – Quality in Models
- 27/2: Oblig 1: Smart Building – Business Architecture and App/Portal with IFML WebRatio UI for Smart Building
  
- March (8-11) (Modeling of IoT/CPS/Cloud, Services and Big Data – UML SM/SD/Collab, ThingML Server-Side):
- 8-6/3: DSL and ThingML, UML State Machines and Sequence Diagrams
- 9-13/3: UML Composite structures, State Machines and Sequence Diagrams II
- 10-20/3: Architectural models, Role modeling and UML Collaboration diagrams
- 11-27/3: UML Service Modeling, ServiceML, SoaML, REST, UML 2.0 Composition, MagicDraw
- 27/3: Oblig 2: Smart Building – Internet of Things control with ThingML – Raspberry Pi, Wireless sensors (temperature, humidity), actuators (power control)
  
- April/May (12-14) (MDE – Creating Your own Domain Specific Language):
- 12-3/4: Model driven engineering – Metamodels, DSL, UML Profiles, EMF, Sirius Editors
- EASTER – 10/4 og 17/4
- 13-24/4: MDE transformations, Non Functional requirements
- 1. Mai – Official holiday
- 14-8/5: Enterprise Architecture, TOGAF, UPDM, SysML – DSLs etc.
- 8/5: Oblig 3 - Your own Domain Specific Language
  
- May (15-17): (Bringing it together)
- 15-15/5: Summary of the course – Final demonstrations
- 16-22/5: Previous exams – group collaborations (No lecture)
- 17-29/5: Conclusions, Preparations for the Exam by old exams
- June (Exam)
- 13/6: Exam (4 hours), (June 13<sup>th</sup>, 0900)-1300

# Update to the course in 2017

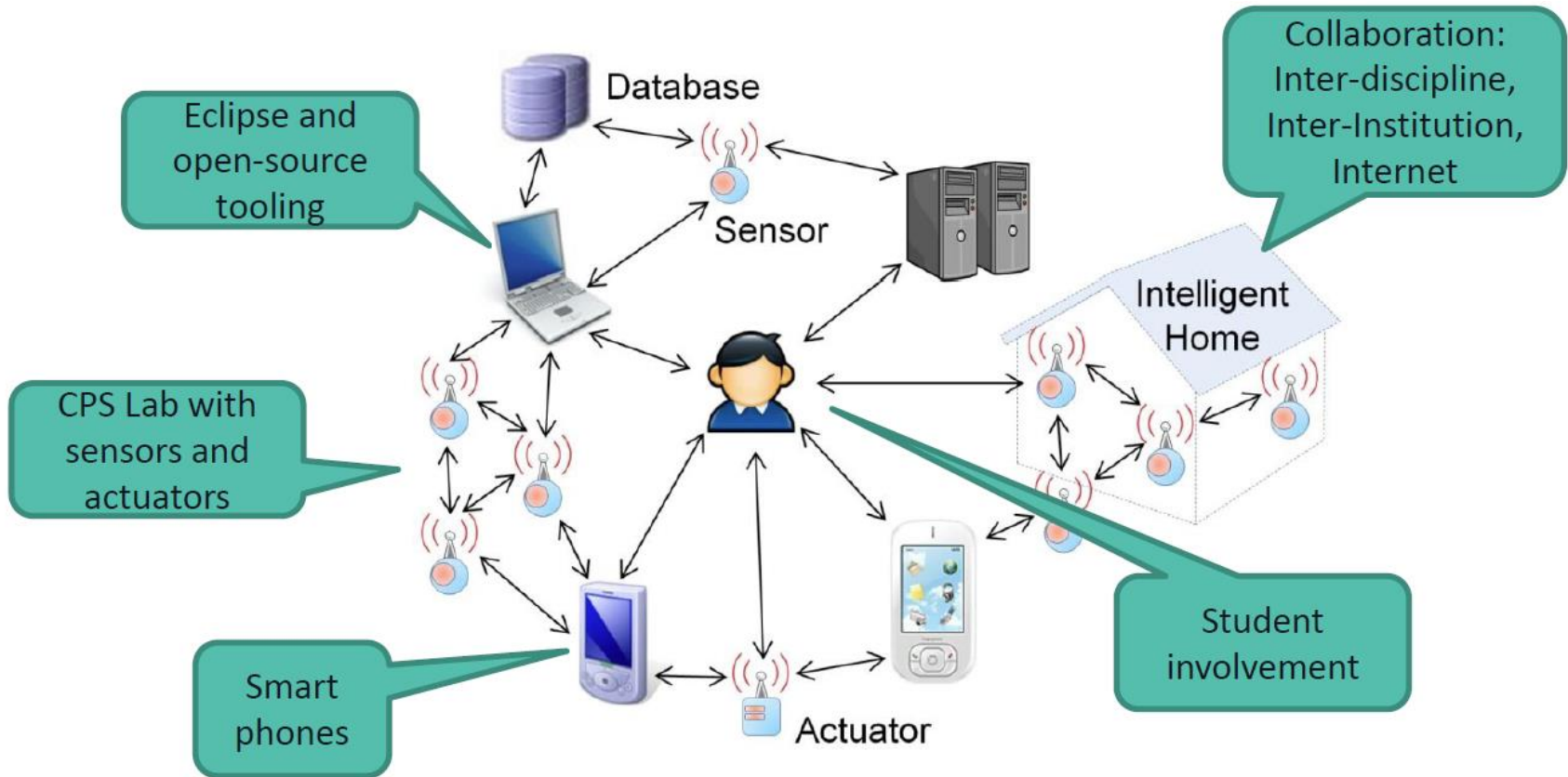
- We will have a project focus on "Smart Buildings" with Apps/Web portals and IoT/Sensors
- We will reduce the focus on Business Modeling and Service Design from previous years
- We will increase the focus on App/Web Portal development with the IFML domain specific language and supporting UML/Metamodels
- We will introduce a new focus on IoT/Cyber Physical System/Big Data support through the ThingML domain specific language – and supporting UML/Metamodels
- The core MDE part will focus on the Modeling of Structure and Behaviour in "systems" including the creation of Model Driven Engineering tools – and the creation of tool support for new domain specific languages.

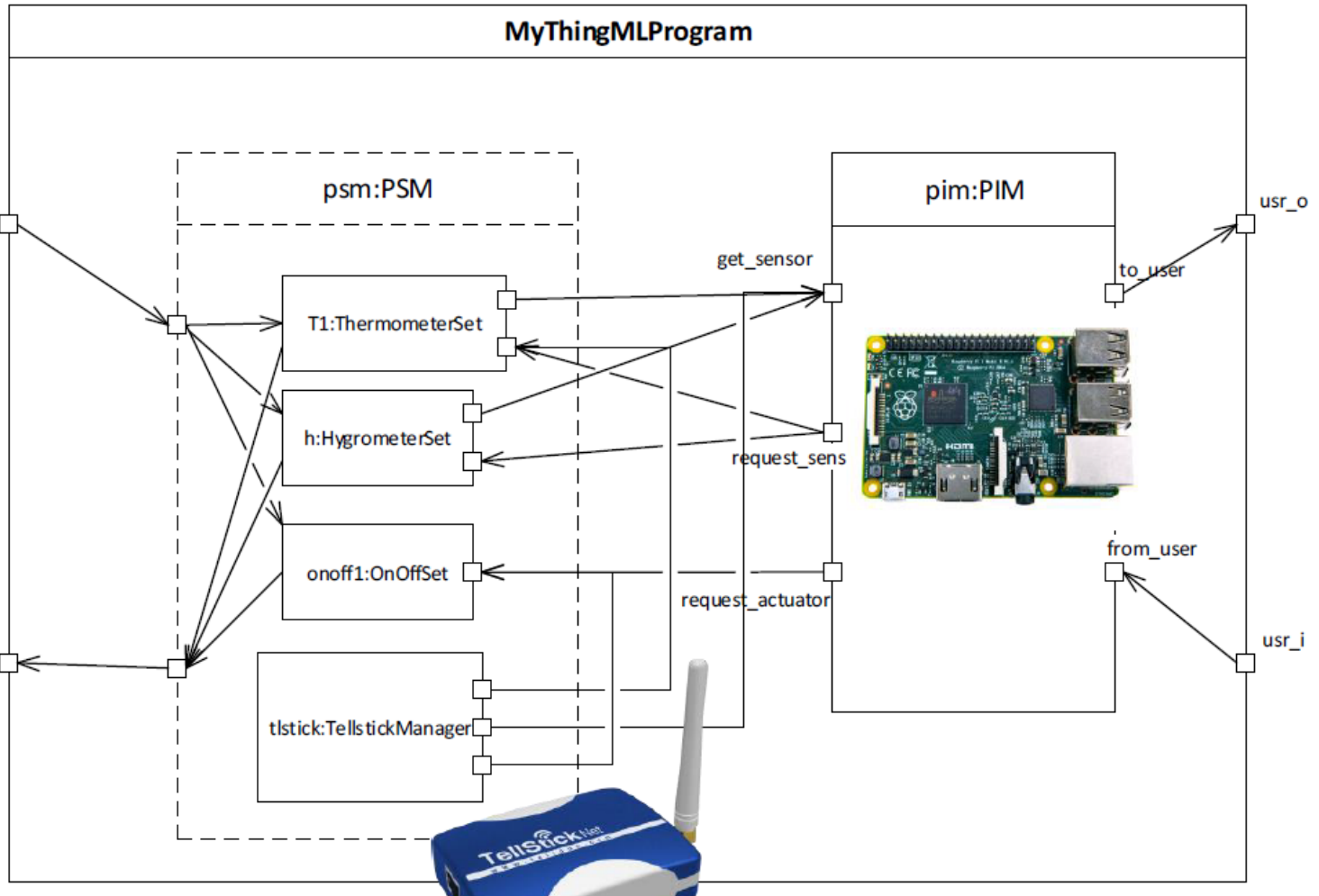
# Obligs

- Partially individual, partially group - in multiple, incremental parts
- Oblig 1, 2 and 3 "Smart Building and DSLs" – your "own" company to develop a mobile app service/Web portal and sensor system control for Smart Buildings – will be presented in more detail on January 30<sup>th</sup>
- Oblig 1, 2 and 3 – Evaluation will count as part of your final grade (10+10+10 = 30%)



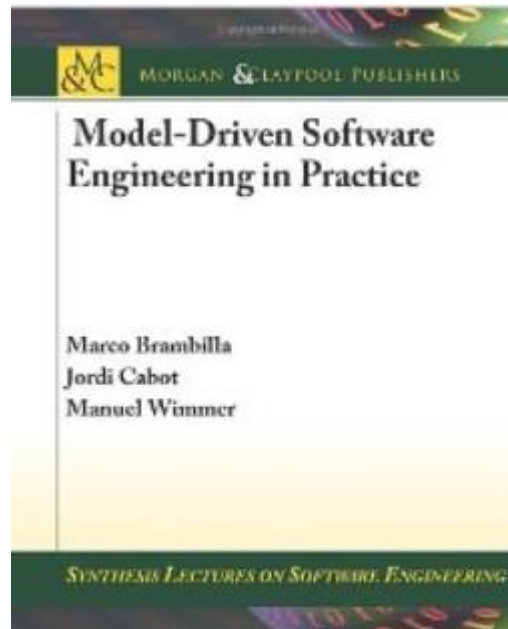
# Architecture for "Smart Building" project Spring 2017





# Book on Model-Based system development

- Model-Driven Software Engineering in Practice
- ISBN 978-1-60845-882-0
- Morgan&Claypool Publishers, Synthesis lectures on Software Engineering
- 2012, 166 pages
- Marco Brambilla, Jordi Cabot and Manuel Wimmer



# UML 2.0

- UML 2.0 and SysML Background and Reference material
- See [www.uml-forum.com/specs.htm](http://www.uml-forum.com/specs.htm)
  
- Also at OMG:
- <http://www.omg.org/uml/> (UML)
- <http://www.omg.org/mda/> (MDA)
- <http://www.omg.org/cwm/> (MOF, XMI, CWM)

## UML 2.0 recommended books:

**UML 2.0 in a Nutshell**

**by Dan Pilone (Author), Neil Pitman (Author)**

**The Unified Modeling Language User Guide  
Second edition (ISBN 0-321-26797-4)**

**(G, Booch, J. Rumbaugh, Jacobsson)**

# Requirements for the course

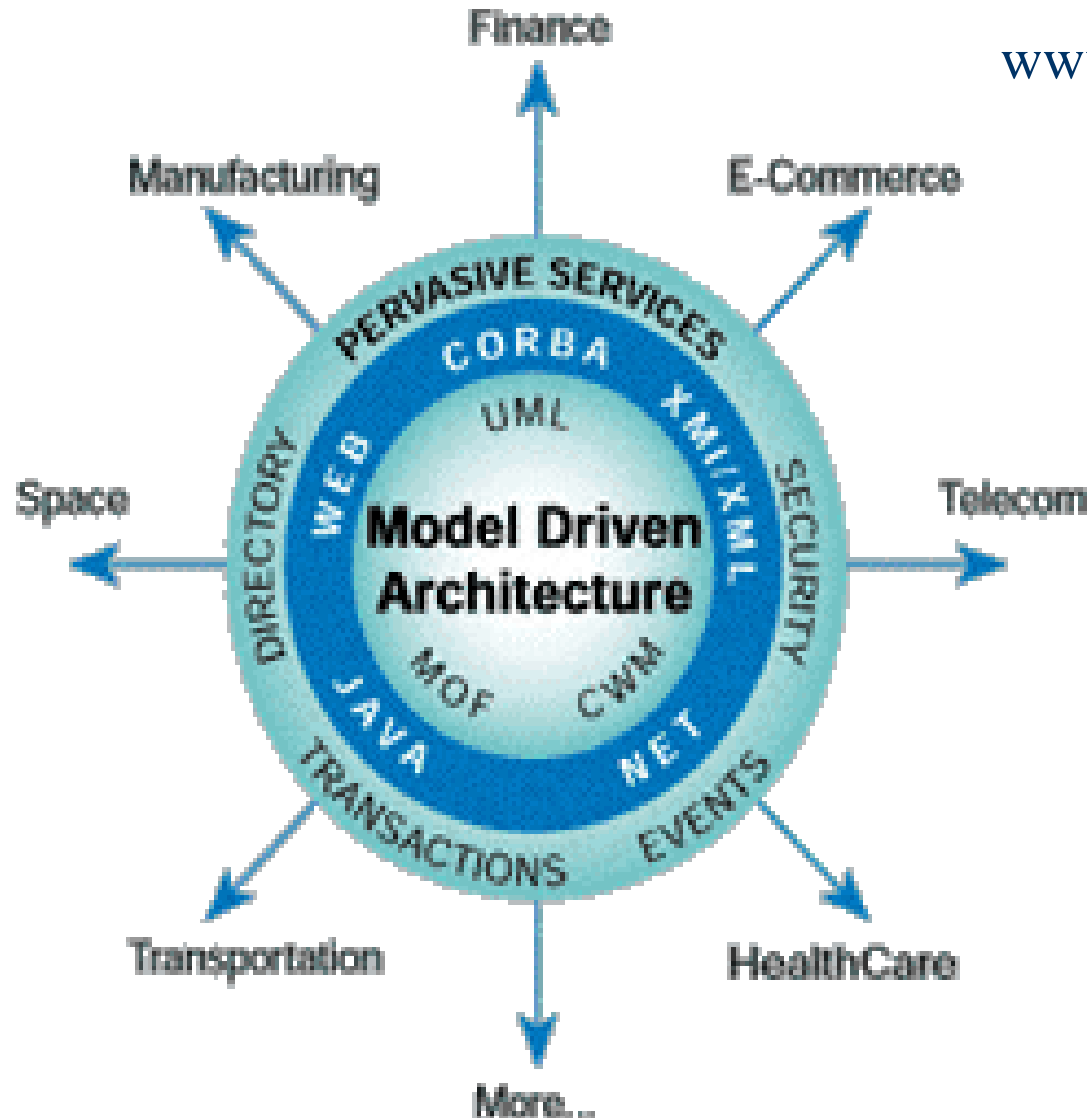
- Student at UIO
- Only assumption is basic knowledge of UML and Java (but not necessarily UML 2.0)
- Links to other courses on software engineering, entrepreneurship, user interaction etc.

# Exam

- Case-based (ref. earlier exams)
- All written material can be used
- 4 hours
- Tentative: Monday June 13<sup>th</sup>, 2017, 0900-1300 (4 hours)
- The grades from the OBLIGS count 30% as part of the final grade of the course

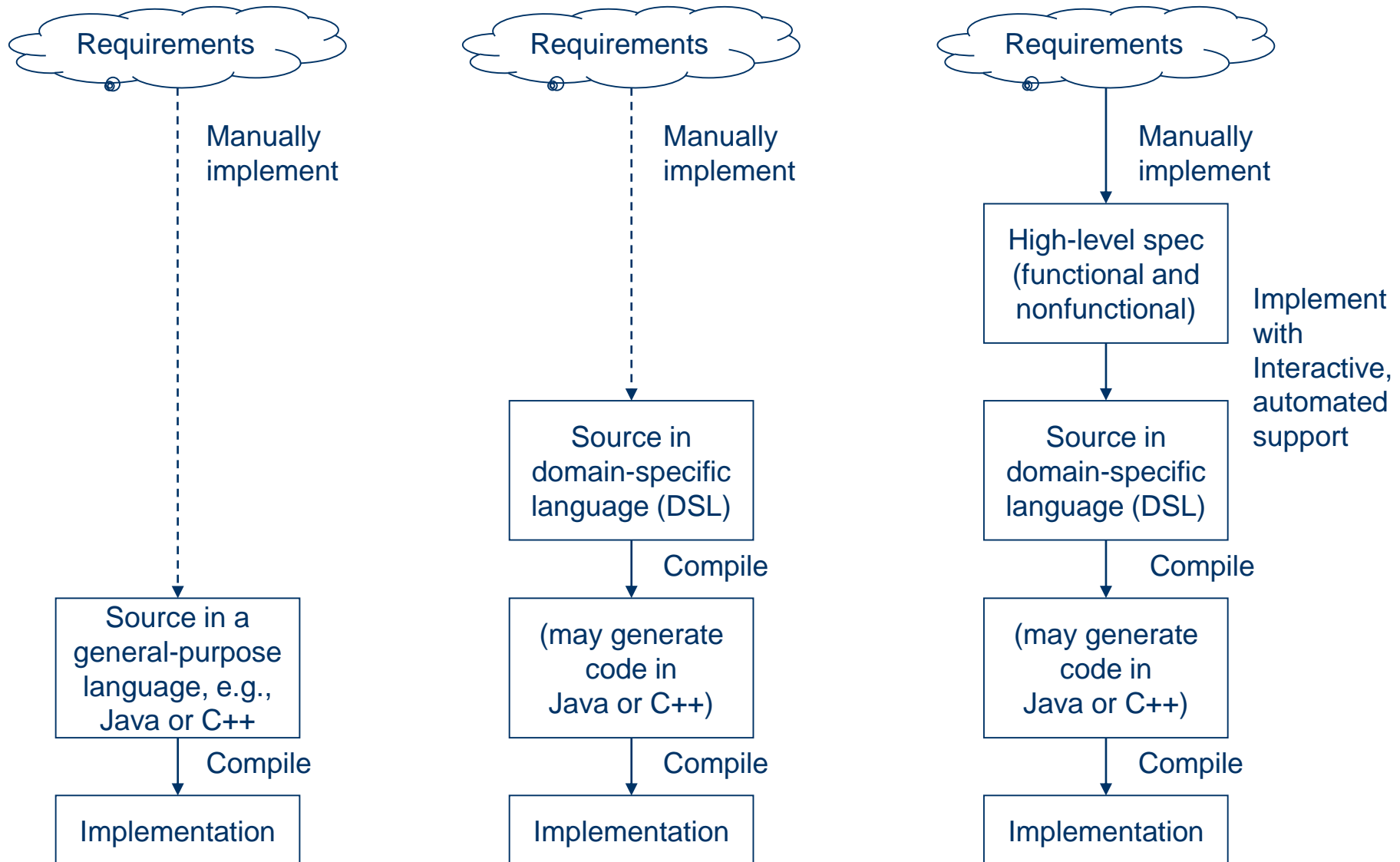
# OMG Model-Driven Architecture (MDA)

[www.omg.org/mda](http://www.omg.org/mda)

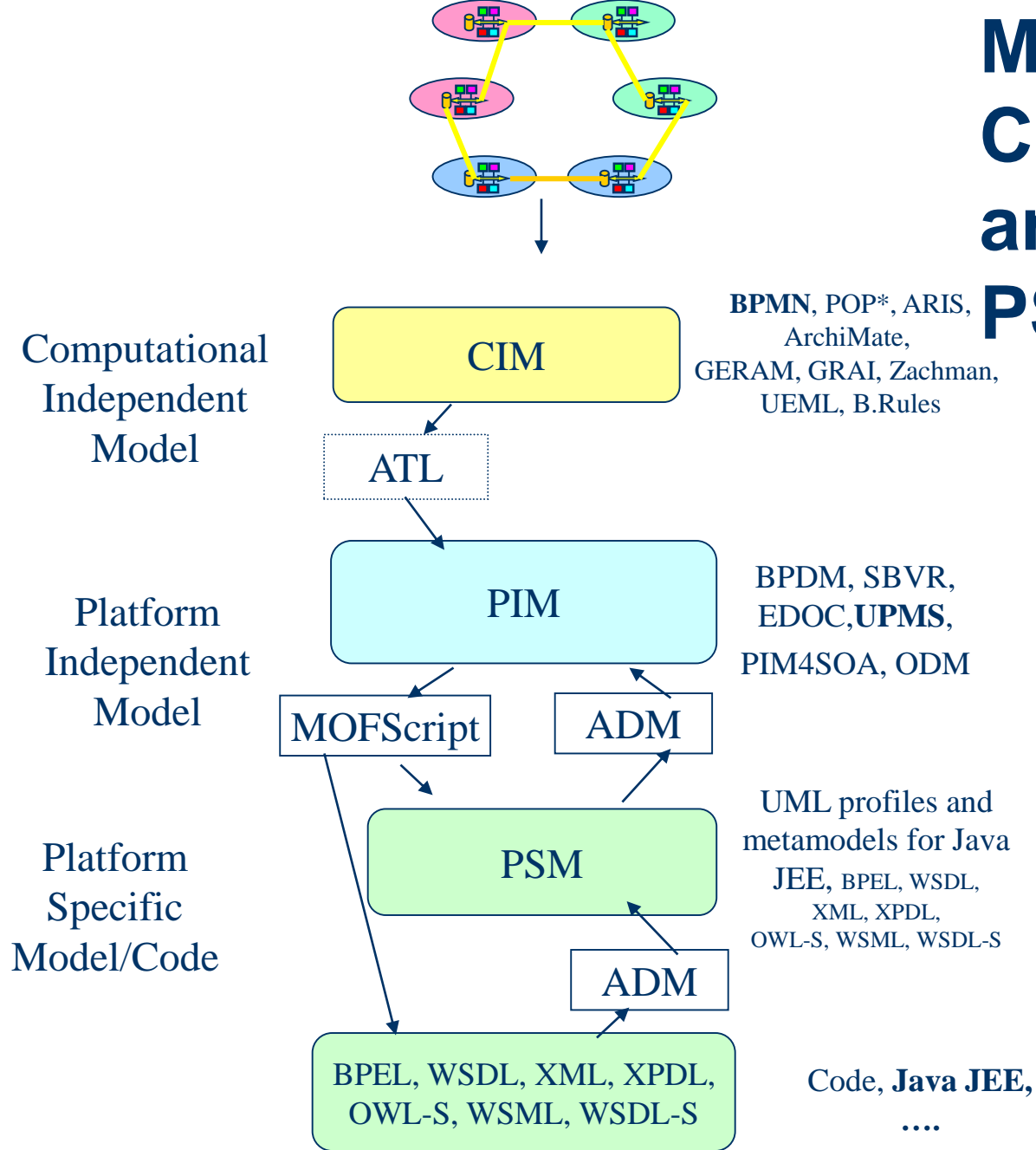




# Automation in Software Development



# MDA CIM, PIM and PSM/Code



# Which OMG modeling standards will you learn ?

- UML 2.0 – what is new in version 2
- VDML – Value Delivery Modeling Language – with VNA
- SoaML – SOA Modeling Language
- MDA – Model Driven Architecture
- BPMN 2.0 – Business Process Modeling Notation
- BMM \_ Business Motivation Model
- SysML – Systems Engineering Modeling Language
- Essence – Software Engineering Framework
- SPEM – Software Process Engineering Metamodel
- QVT, MOF2Text – Query, View, Transformation

# Which tools/environments will you learn ?

- WebRatio - IFML for Web and Mobile Apps
- Agile team support – Symphonical/Someone, Scrumwise
- BMI – Business Model Innovation/Generation - Strategyzer
- Balsamiq – UI Mockups – for further UI modeling
- UML and BPMN modeling tools - MagicDraw
- AT ONE – Service Design – use of Smaply
- MagicDraw with UML and BPMN
- Eclipse EMF and XMI, Principles of GMF
- Sirius for Eclipse
- EPF/SPEM Software Process Modeler
- Overview of ATL, MOFScript, KerMeta, OpenArchitectureWare-OAW,  
...

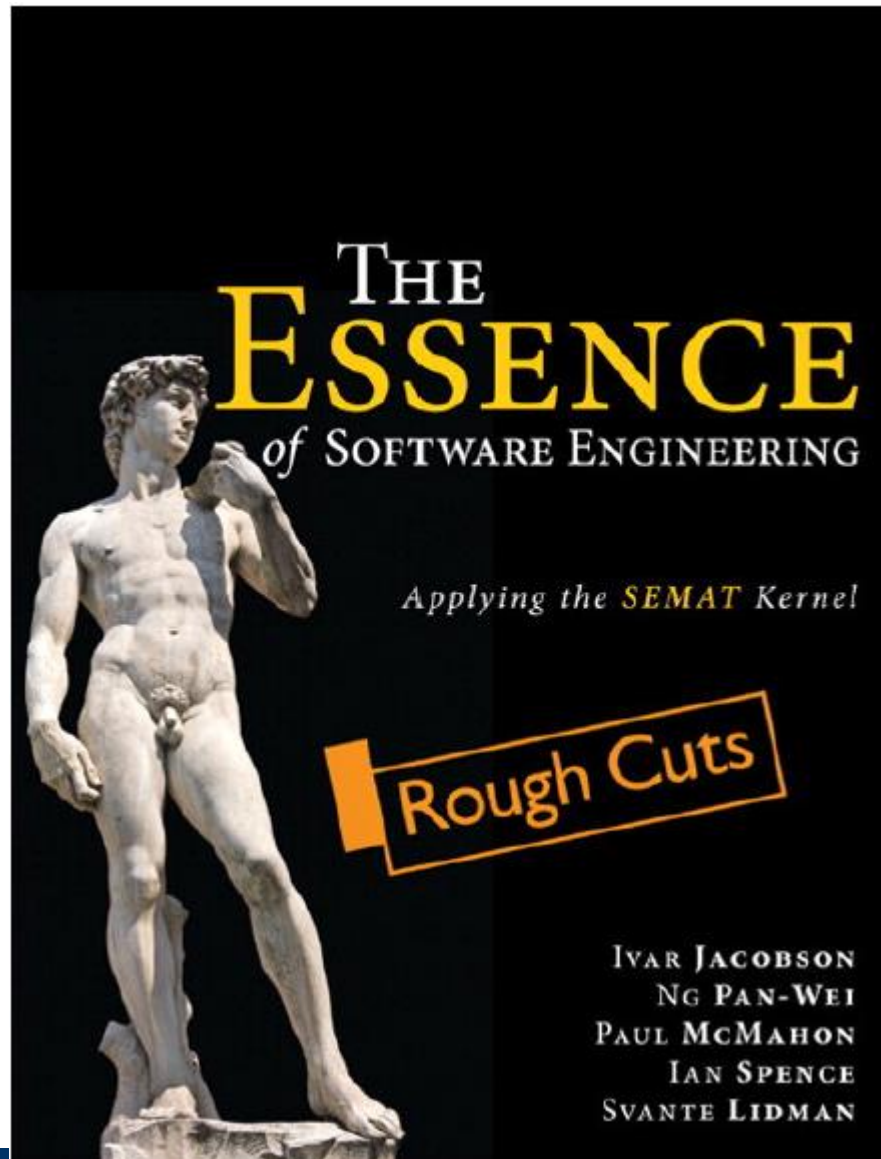
# Model Driven Engineering techniques

- Structure modeling: UML Class (Information) and Components – MagicDraw/Cameo, <http://www.nomagic.com/>
- Behaviour modeling: UML Behaviour modeling, BPMN
- WebRatio with IFML – for UI and App/Server development, - <http://www.webratio.com/>
- Non functional modeling: OCL and Planguage
- Metamodeling and DSLs: EMF and Sirius
- Business Model Canvas and Value Proposition Canvas – Strategyzer [www.strategyzer.com](http://www.strategyzer.com) (license will be provided)
- Scrum/Agile Project Management – Symphonical, <https://www.someone.io/> (see also Scrumwise.com )
- Service Design – Smaply, [www.smaply.com](http://www.smaply.com) (license will be provided)
- Business Process Model – MagicDraw/Cameo, <http://www.nomagic.com/> (license will be provided)
- User stories and Use cases – Someone.io and MagicDraw/Cameo
- UI Mockup – Balsamiq, <http://balsamiq.com/> (license will be provided)

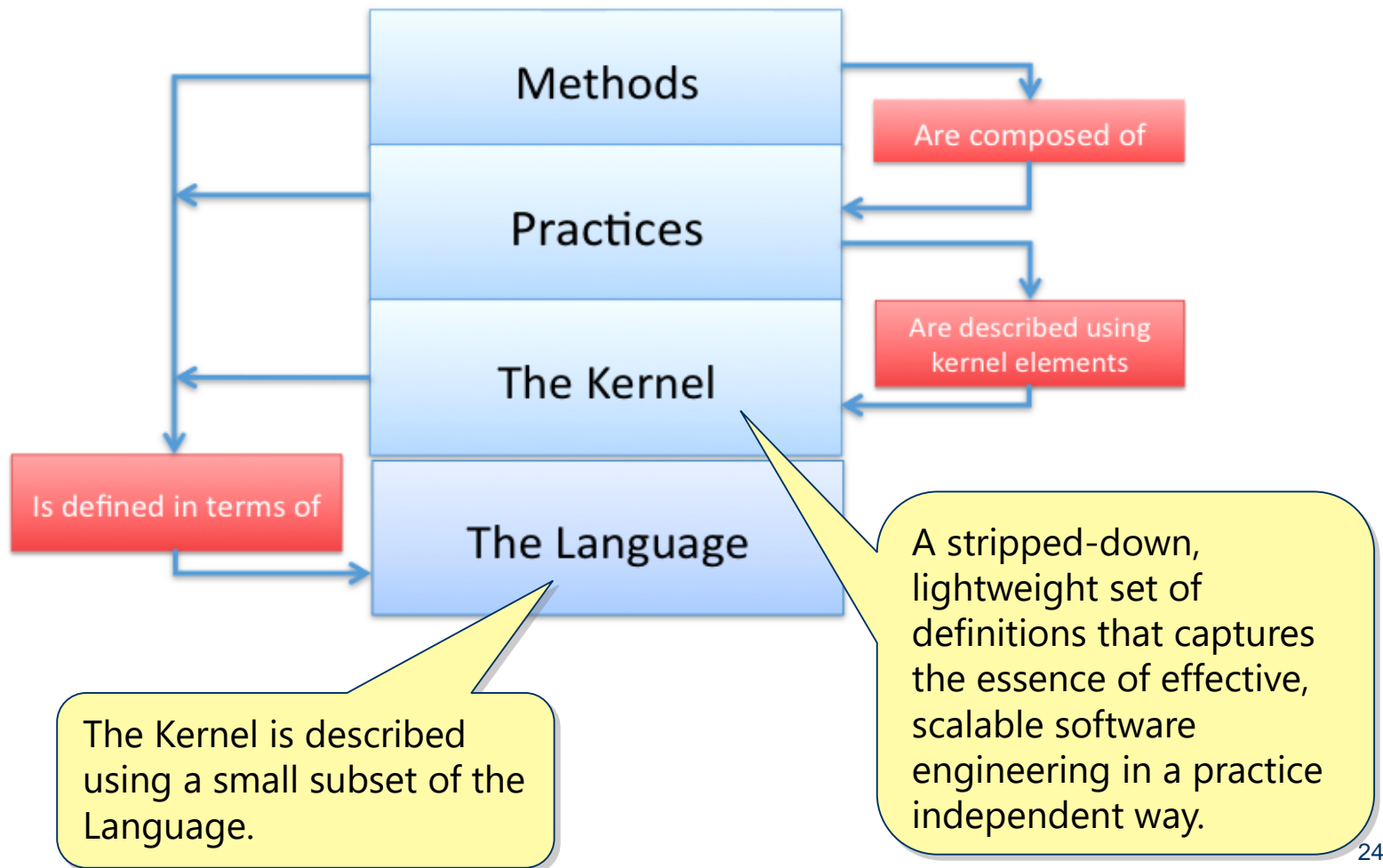
# Software engineering practices and methods

- modelbased.net
- practices.modelbased.net
  
- A practices framework, SEMAT, [www.semat.org](http://www.semat.org)

# Book – Safaribooksonline/Addison Wesley

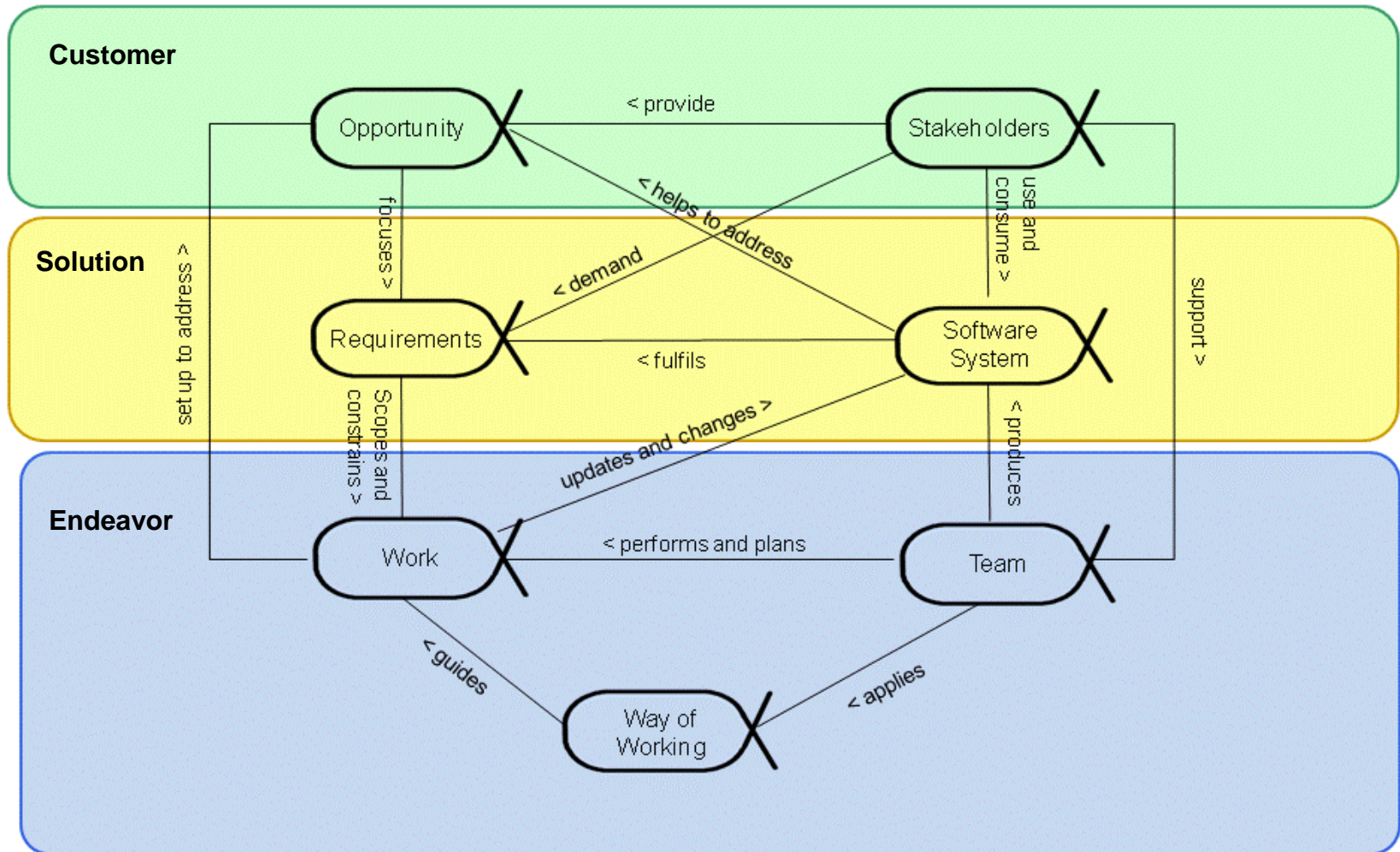


# The Kernel





# Alphas: The Essential Things to Work With



# Alphas: Example

## Requirements

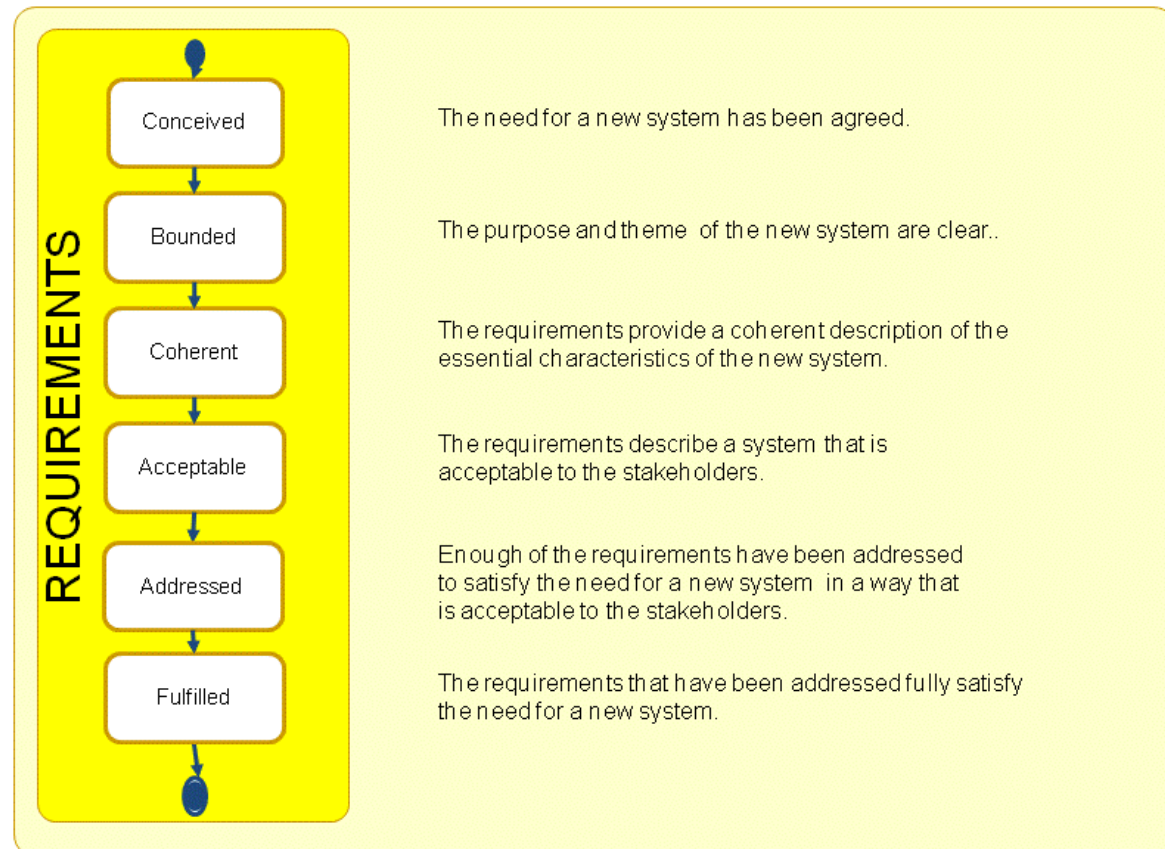
### Description

What the software system must do to address the opportunity and satisfy the stakeholders.

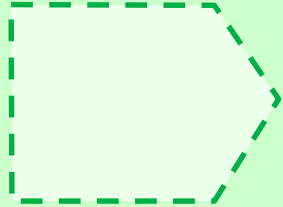
It is important to discover what is needed from the software system, share this understanding among the stakeholders and the team members, and use it to drive the development and testing of the new system.

### Associations

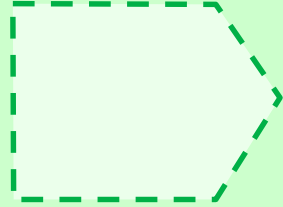
scopes and constrains : Work



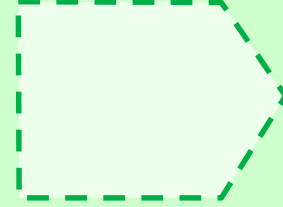
# Activity Spaces: The Essential Things to Do



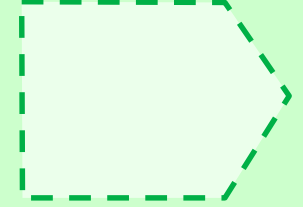
*Explore  
Possibilities*



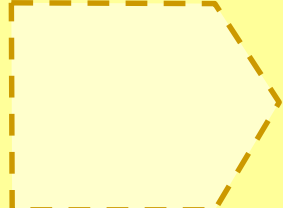
*Understand  
Stakeholder Needs*



*Ensure Stakeholder  
Satisfaction*



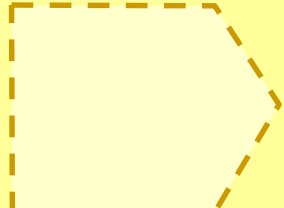
*Use the System*



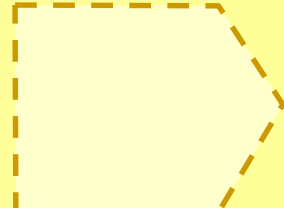
*Understand the  
Requirements*



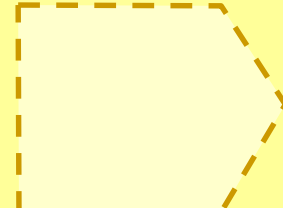
*Shape  
the System*



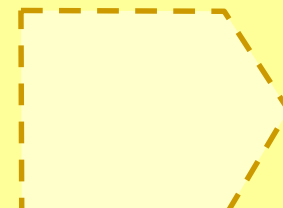
*Implement the  
System*



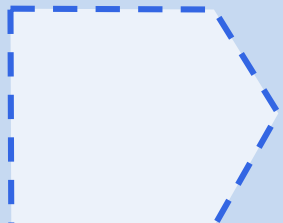
*Test  
the System*



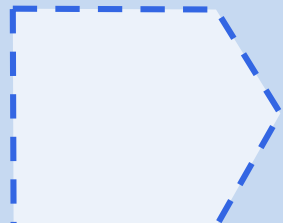
*Deploy  
the System*



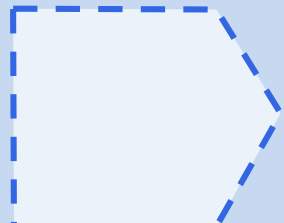
*Operate  
the System*



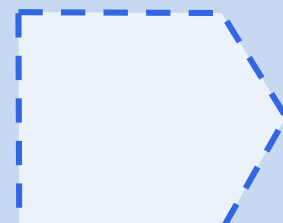
*Prepare to do  
the Work*



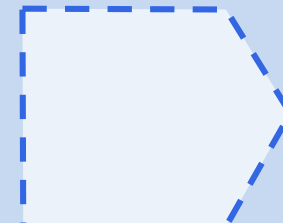
*Coordinate  
Activity*



*Support the Team*



*Track Progress*



*Stop the Work*

# Requirements and Agile development

The Agile Manifesto: <http://agilemanifesto.org/>



# Upwave.io – for Scrum

The image shows a promotional banner for Upwave.io, a task management tool. The banner has a red background and features the Upwave.io logo (a stylized 'u' with a face) and the text 'someone.io' in the top left. The main headline reads 'Task management for teams made easy'. Below this, a sub-headline states: 'Boost your team's productivity, strengthen communication, improve efficiency and make worklife simpler!'. A registration form is centered, with fields for 'Email address' and 'Company name', and a 'Get started for FREE' button. In the top right corner of the banner, there is a 'Sign in' link.

Below the banner is a screenshot of the Upwave.io task management interface. The interface shows a project titled 'Let's make a team T-Shirt!' under the user 'Someone'. It features a Kanban board with three columns: 'To do', 'In Progress', and 'Completed'. The 'To do' column has a task 'How many T-Shirts do we need to order? And what colours?' with a due date of 'in 2 days'. The 'In Progress' column has a task 'Find a printing place, get a good price!' with a due date of 'in a day'. The 'Completed' column has a task 'Finalize the design' which is marked as 'Completed, 27 minutes ago'. There is also an 'Add card' button in the 'In Progress' column and a task 'Make the order!' in the 'To do' column.

# Scrum – Scrumwise.com



[Home](#) [Features](#) [Pricing](#) [FAQ](#) [Support](#) [About](#)

[Try now](#)

[Log in](#)

## The most intuitive Scrum tool you've ever tried

And it's lightning fast!

[Try the instant demo](#)  
Just one click. No signup.



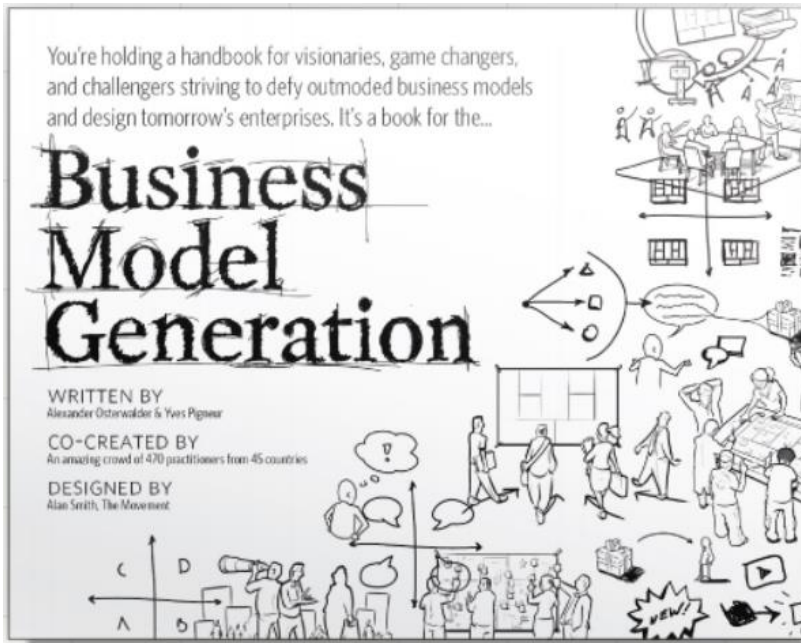
Loved by thousands of teams all over the world

## Built for Scrum

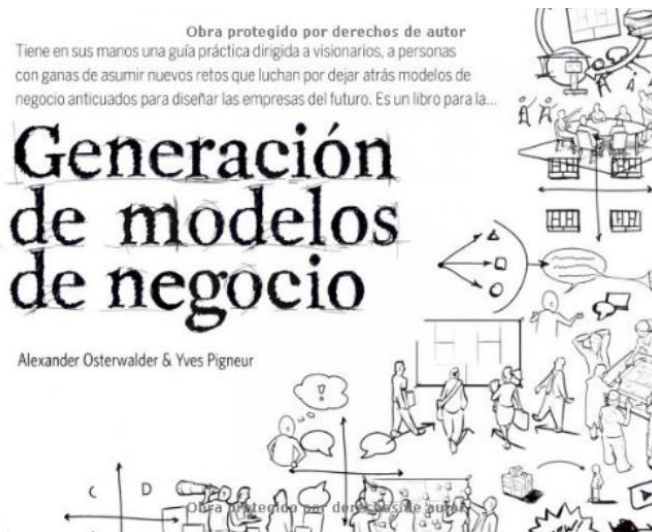
- Teams and roles
- Backlog management
- Release planning
- Sprint planning

- Task boards
- Burndown charts
- Kanban
- Time tracking





> 1 million copies sold

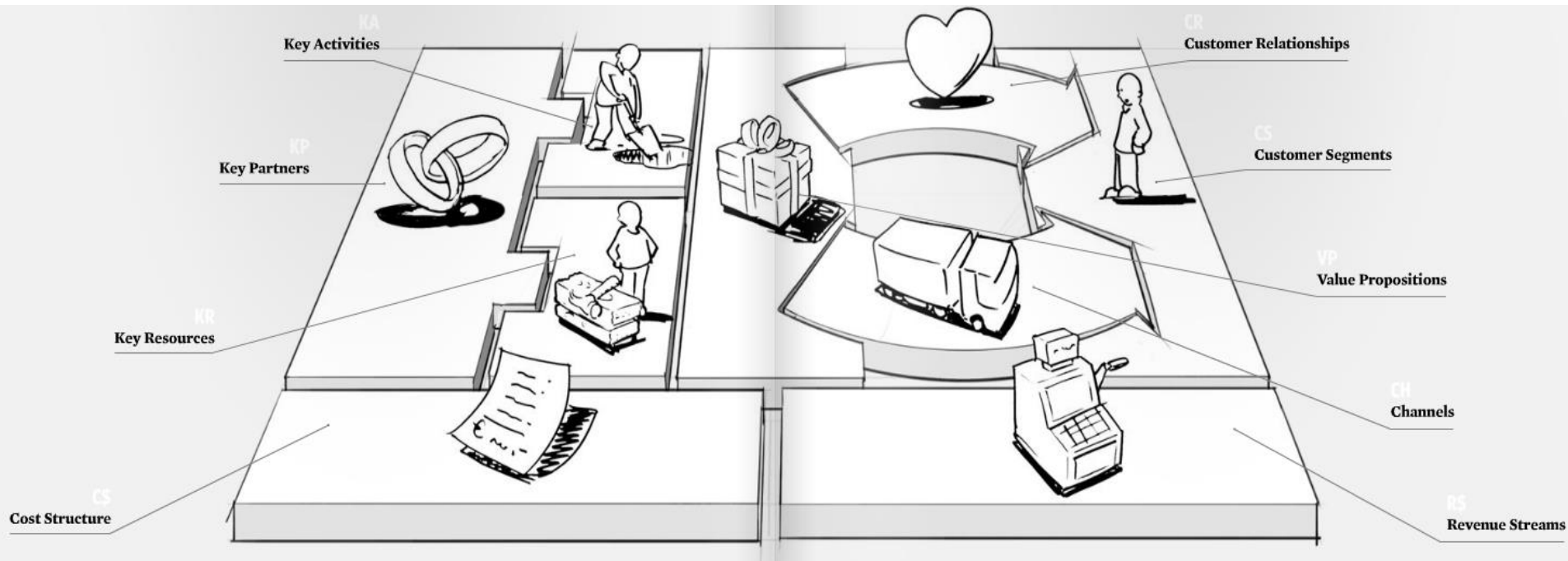


30 languages



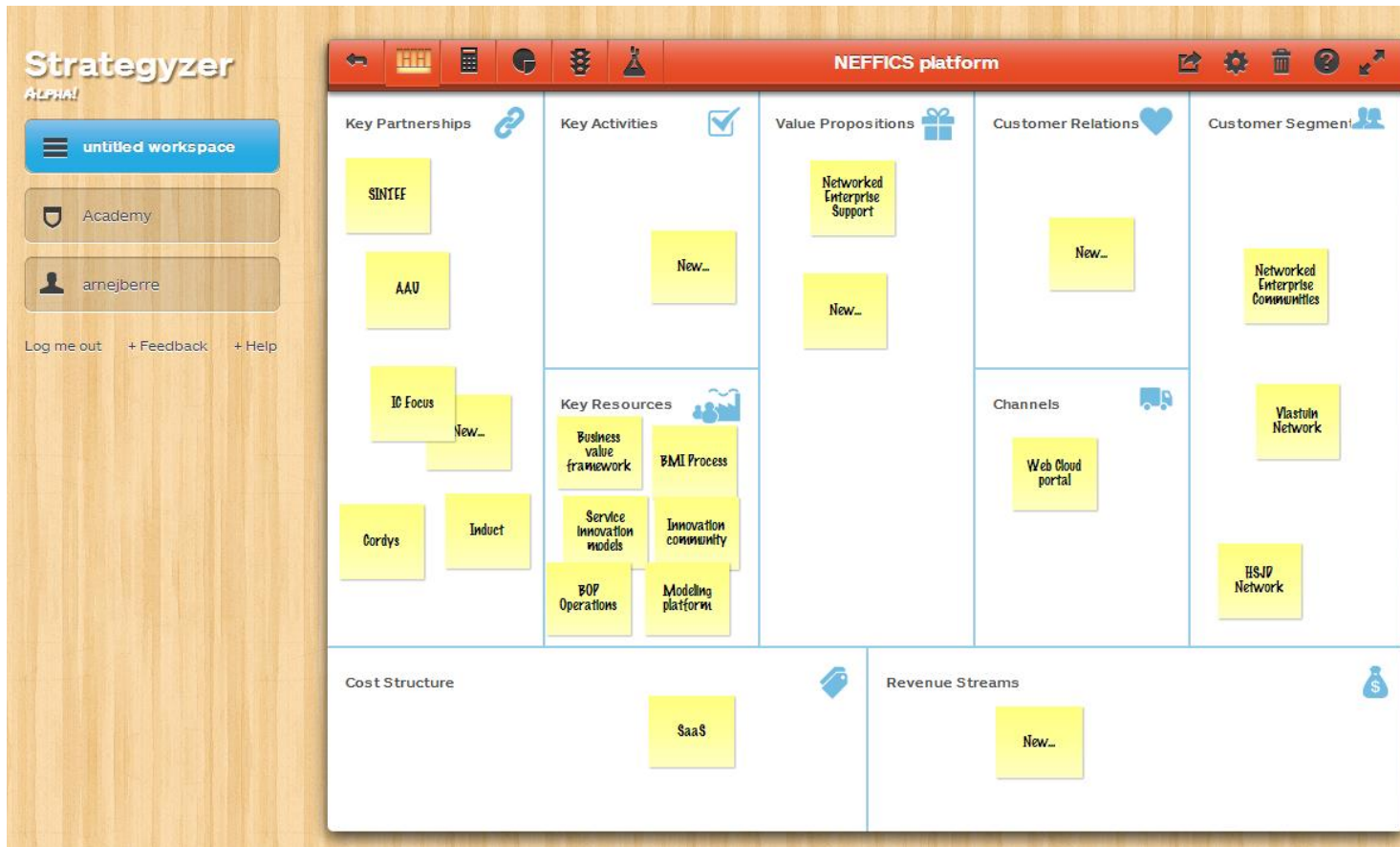
# Business Model Innovation

## The Business Model Canvas





# Strategyzer (Osterwalder)



# Business Model Canvas and Value Proposition Canvas Resources

- [www.strategyzer.com](http://www.strategyzer.com)
- <http://www.alexandercowan.com/business-model-canvas-templates/>
- BizCanvas App for the iPad
- ...

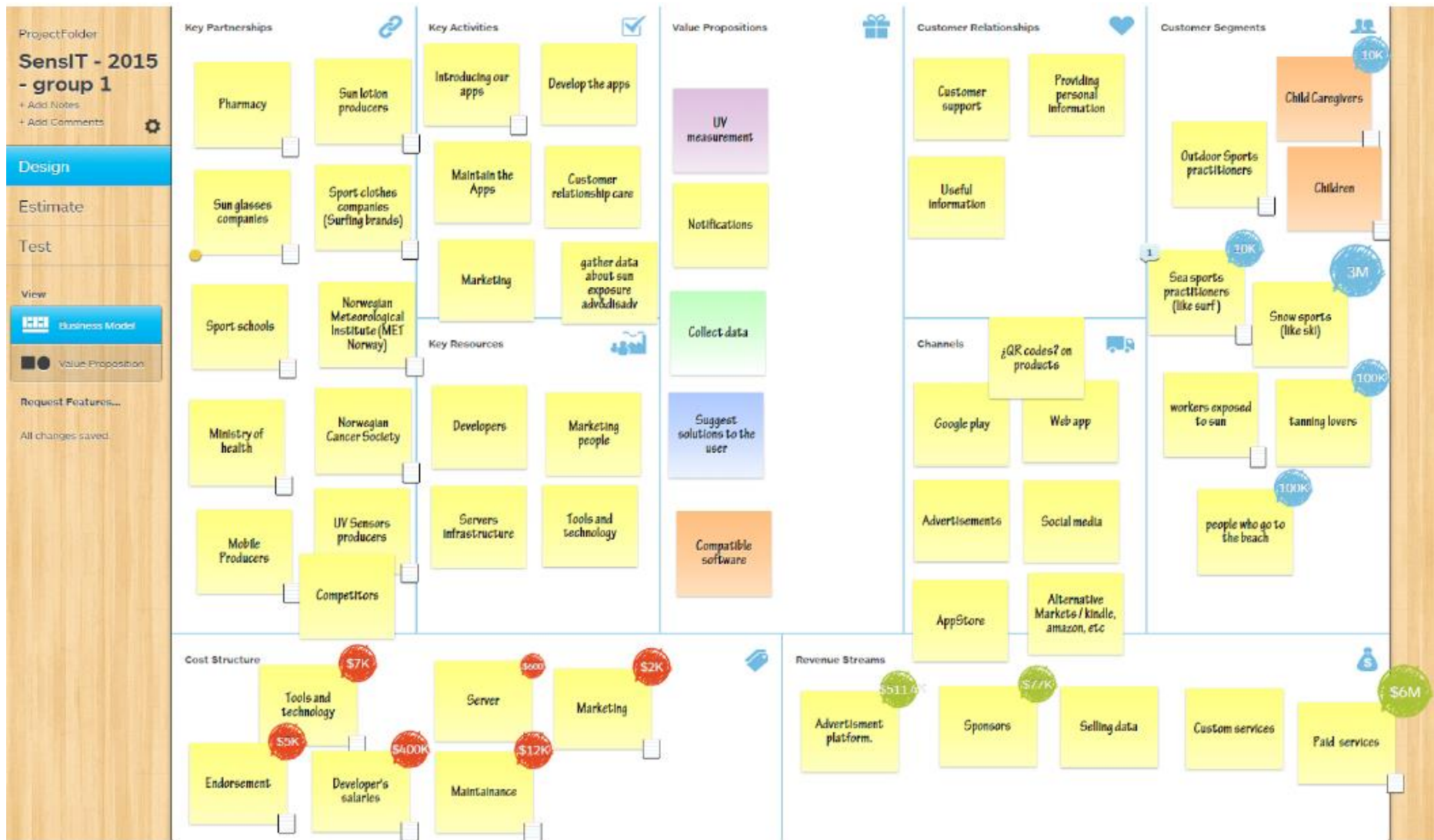
# Business Model (Definition)

- A Business Model describes the rationale of how an organization creates, delivers and captures value.

# Reference examples in the course

- Concierge: A company with a system/service that offers advice and recommendations to people with respect to current and upcoming events, concerts, exhibitions etc.
- TravelAdvisor: A company with a system/service that offers advice and booking possibilities to travelers
- Senselt: UV sensor measurements (from the course of 2015)
- BioCaching: Citizen Science for Biodiversity (2016)
- Project 2017: Smart Building

# Senselt



You're holding more than a book,  
it's the first step to design, test and deliver  
what really matters for your customers.

# Value Proposition Design

By Alex Osterwalder, Yves Pigneur,  
Greg Bernarda, & Alan Smith  
Designed by Trish Papadakos

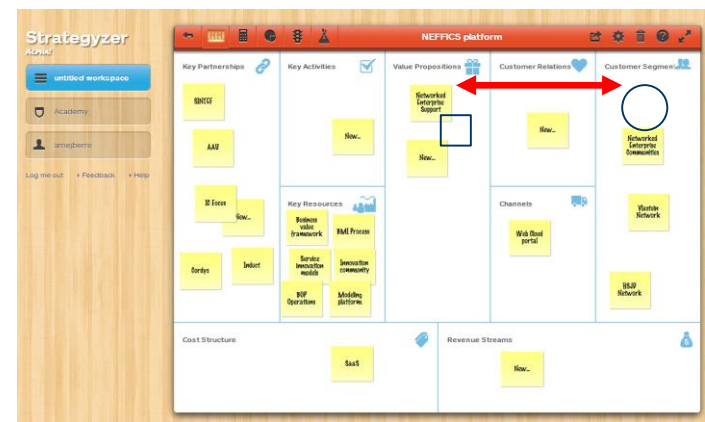


From the team behind  
*Business Model Generation*,  
the global bestseller of over  
1 million copies in 30+ languages.

 **Strategyzer**

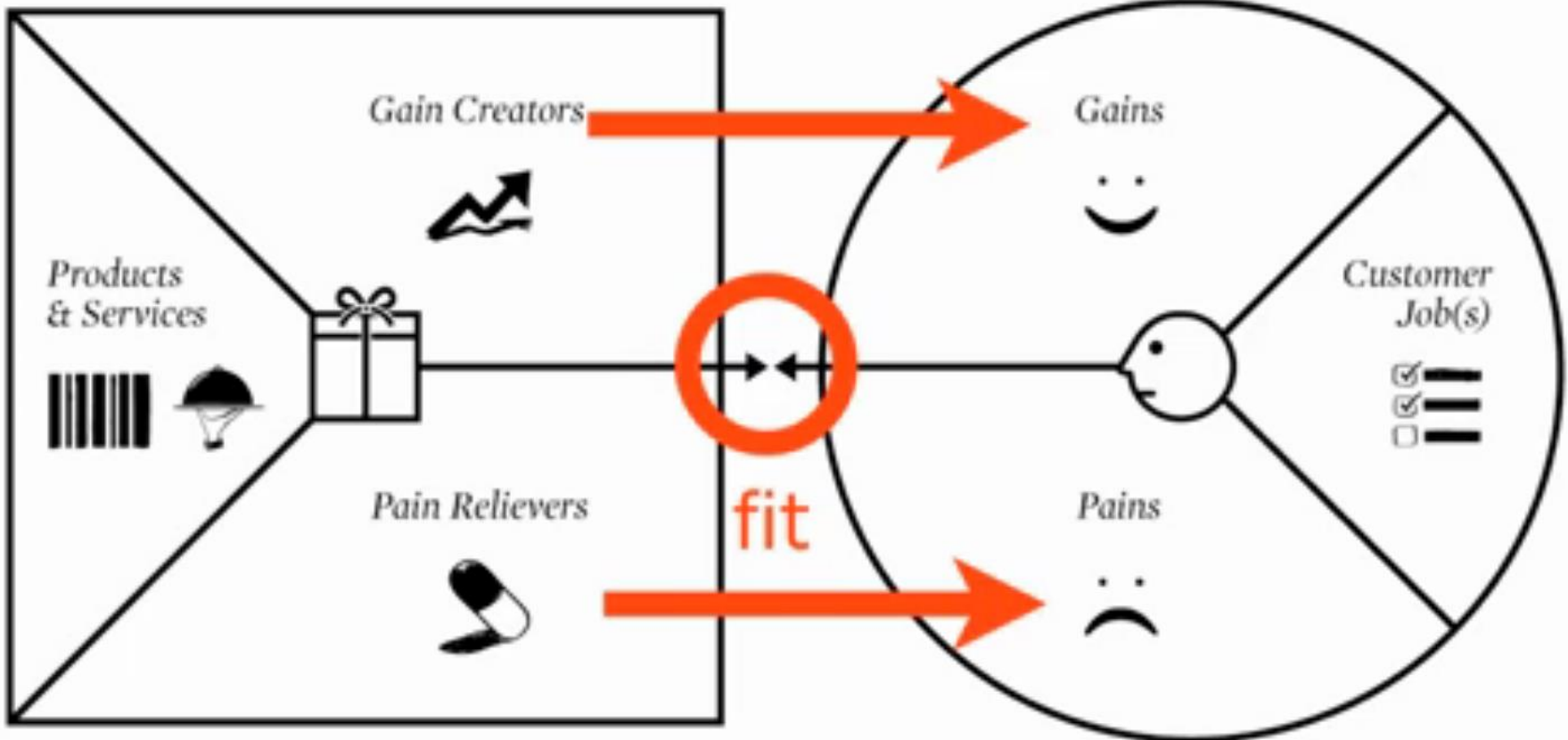
October, 2014

# Value Proposition Canvas

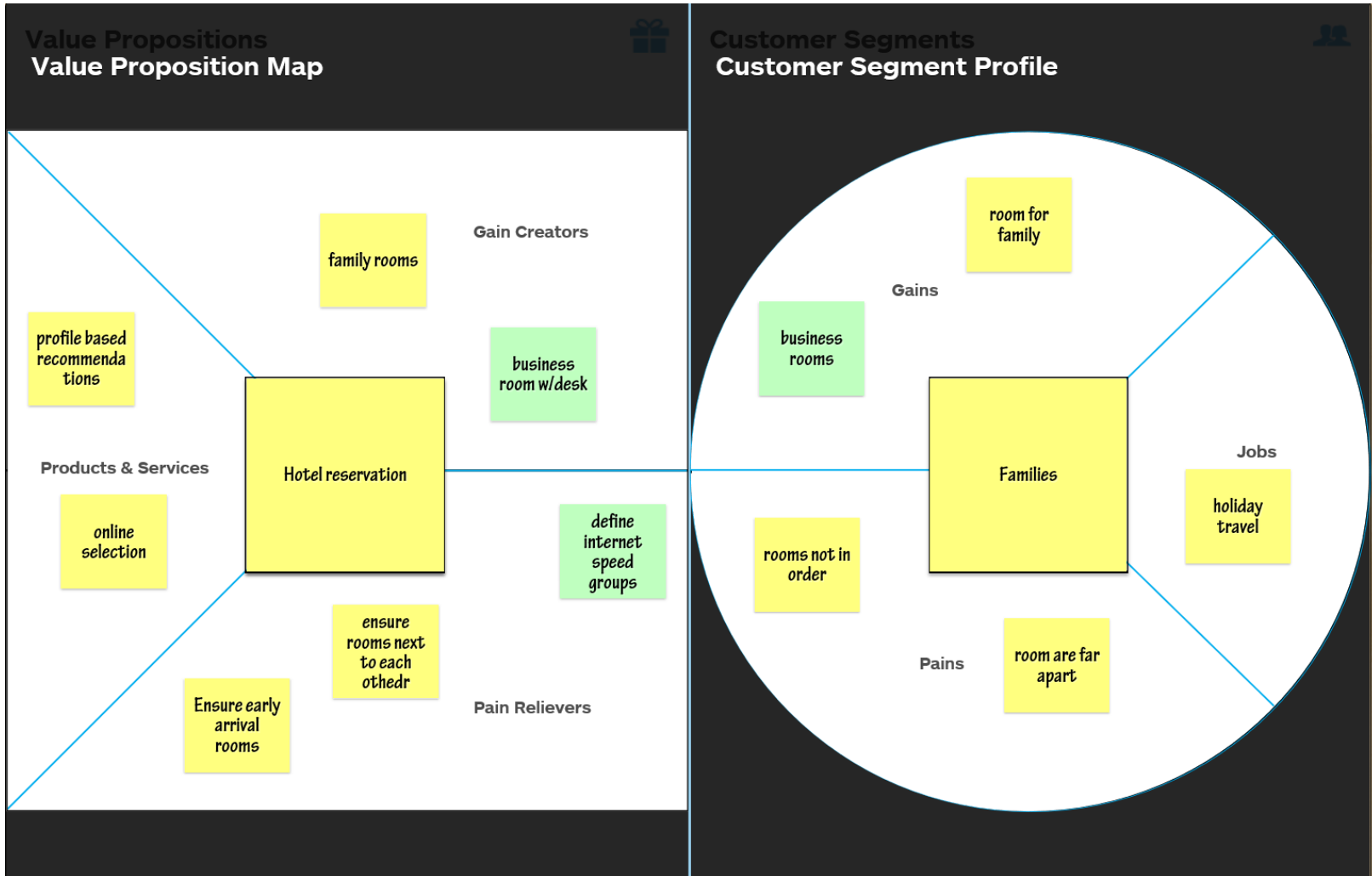


Value offer  
(Opportunity)

Value needs (Requirements)



# Value Proposition Canvas





# BPMN and UML

- BPMN
- UML 2.0

# MagicDraw - [www.nomagic.com](http://www.nomagic.com)

No Magic

Phone: +1-214-291-9100

Login

No Magic **WORLD SYMPOSIUM** **EARLY BIRD SPECIAL**  
Register Now and SAVE \$200  
MAY 22-25, 2016 | Allen TX USA

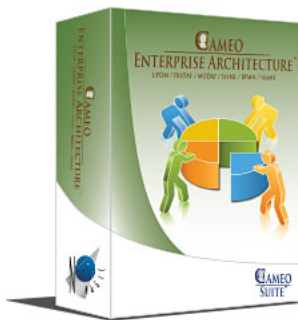
search...

Getting Started Products Services Support News About Us Events Contact

Products > Cameo Enterprise Architecture

DoDAF, MODAF, and NAF with UPDM Compliance

## Cameo Enterprise Architecture



INTRO FEATURES FRAMEWORKS EDITIONS REQUIREMENTS DEMOS RELATED

No Magic has deep experience with DoDAF 2.0, MODAF, NAF 3 and the Defense Industry. Our Cameo Enterprise Architecture product, based on our core product

[MagicDraw](#), offers the most robust standards compliant DoDAF 2.0, MODAF and NAF 3 via a UPDM standardized solution. And what's more, No Magic fully supports all architectural framework products ensuring you achieve mission results. No Magic also leads the industry in its integration of DIEA requirements, ensuring that you achieve net-centric success. Meet your interoperability challenges with proven, tested No Magic solutions.



### No Magic Specifically Meets DoDAF 2.0, MODAF, NAF 3 and UPDM Needs

Improved Mission Results - Your team will do a better job of mining available data, measuring and visualizing architecture and overall success factors resulting in improved mission results.

- Convey the knowledge faster and easier
- Easily represent and communicate complex architecture
- Reduce assumptions, misconceptions and risk

Buy Cameo Enterprise Architecture

Download Trial Version

What's New

### Testimonials

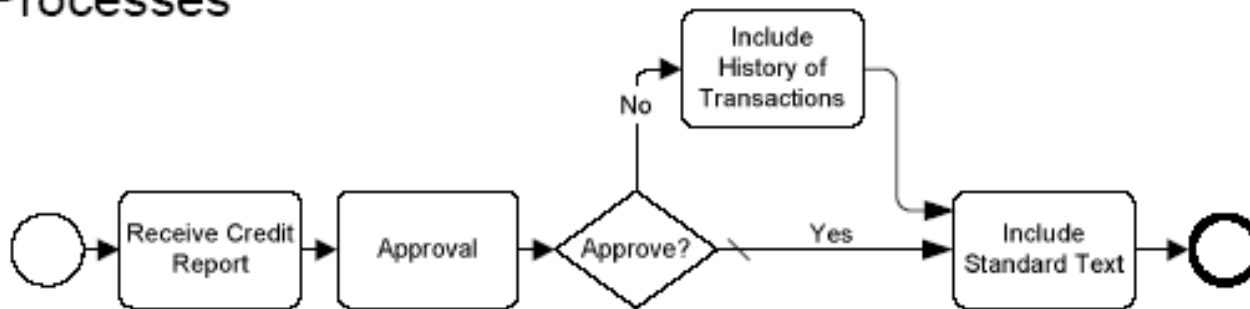
“ Please give a big thumbs up to whomever had the idea to throw in Cameo Requirements Management into MagicDraw 18. I'm using the plugin a ton now.

*Cheers!  
Chris Mellroth  
Northrop Grumman*

“ No Magic's team were exceptional at delivering software that was on time

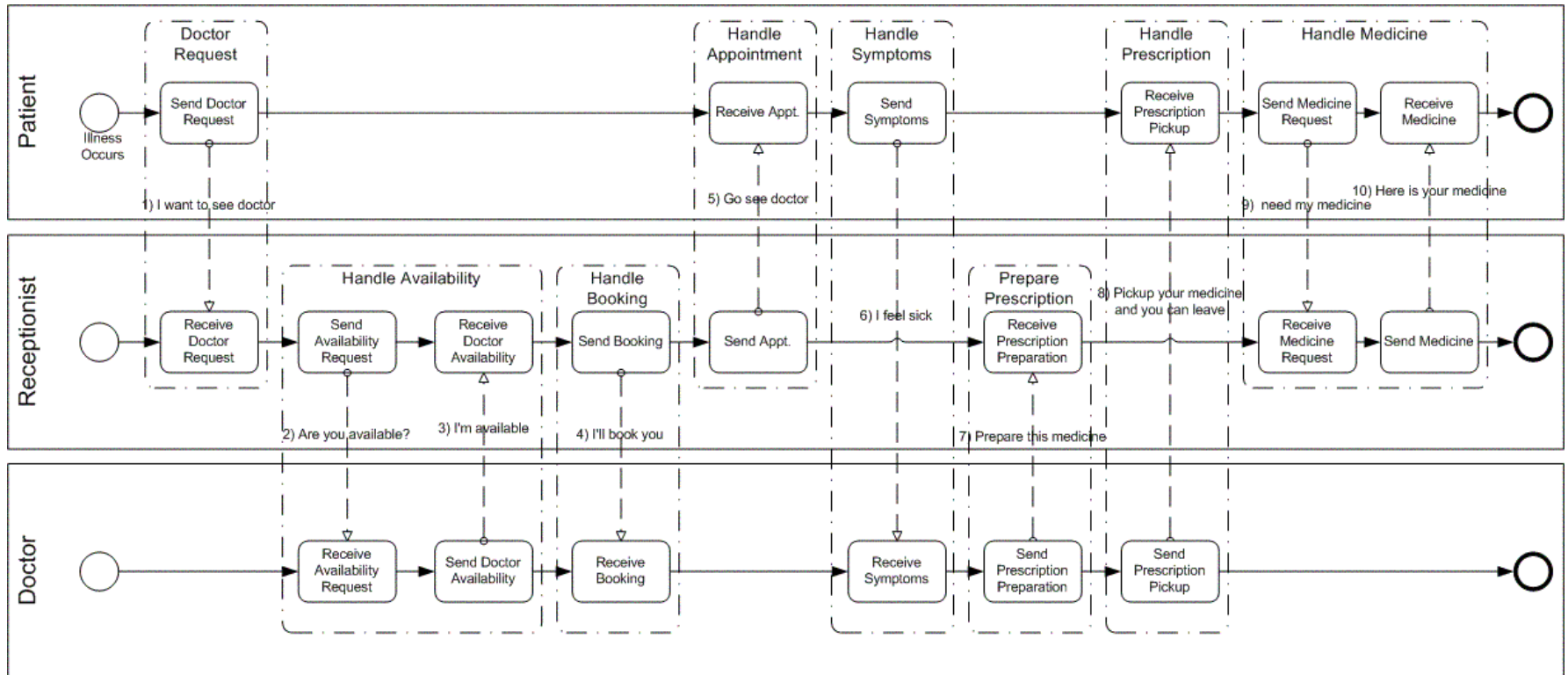
# What is BPMN (Business Process Modeling Notation) ?

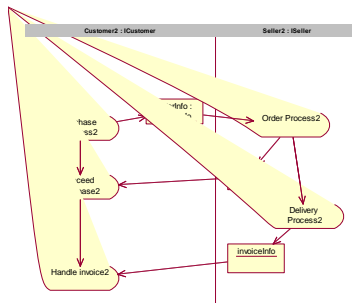
- BPMN is flow-chart based notation for defining Business Processes



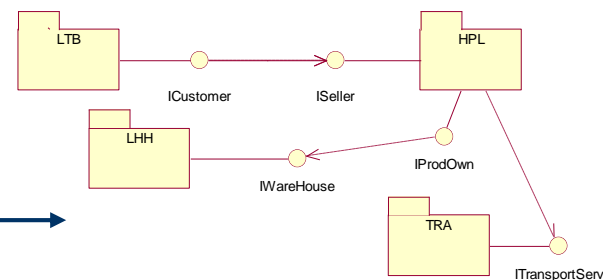
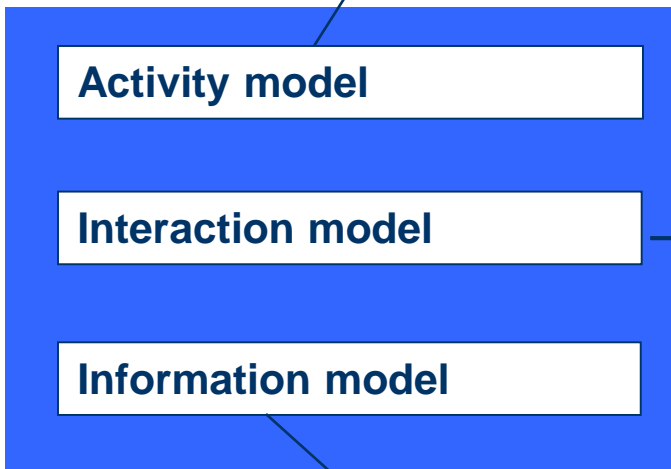
- BPMN is an agreement between multiple modeling tools vendors, who had their own notations, to use a single notation for the benefit of end-user understand and training
- BPMN provides a mechanism to generate an executable Business Process (BPEL) from the business level notation
  - ▶ A Business Process developed by a business analyst can be directly applied to a BPM engine instead of going through *human* interpretations and translations into other languages

# BPMN example





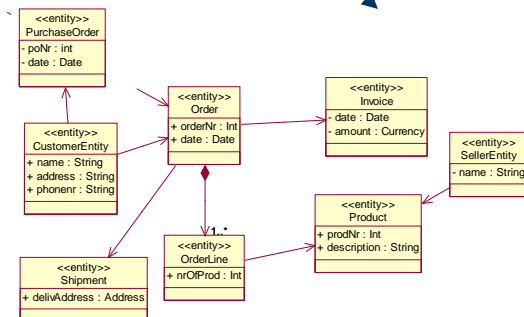
UML Activity model (or BPMN)



UML component diagram (enhanced in UML 2.0), SoaML

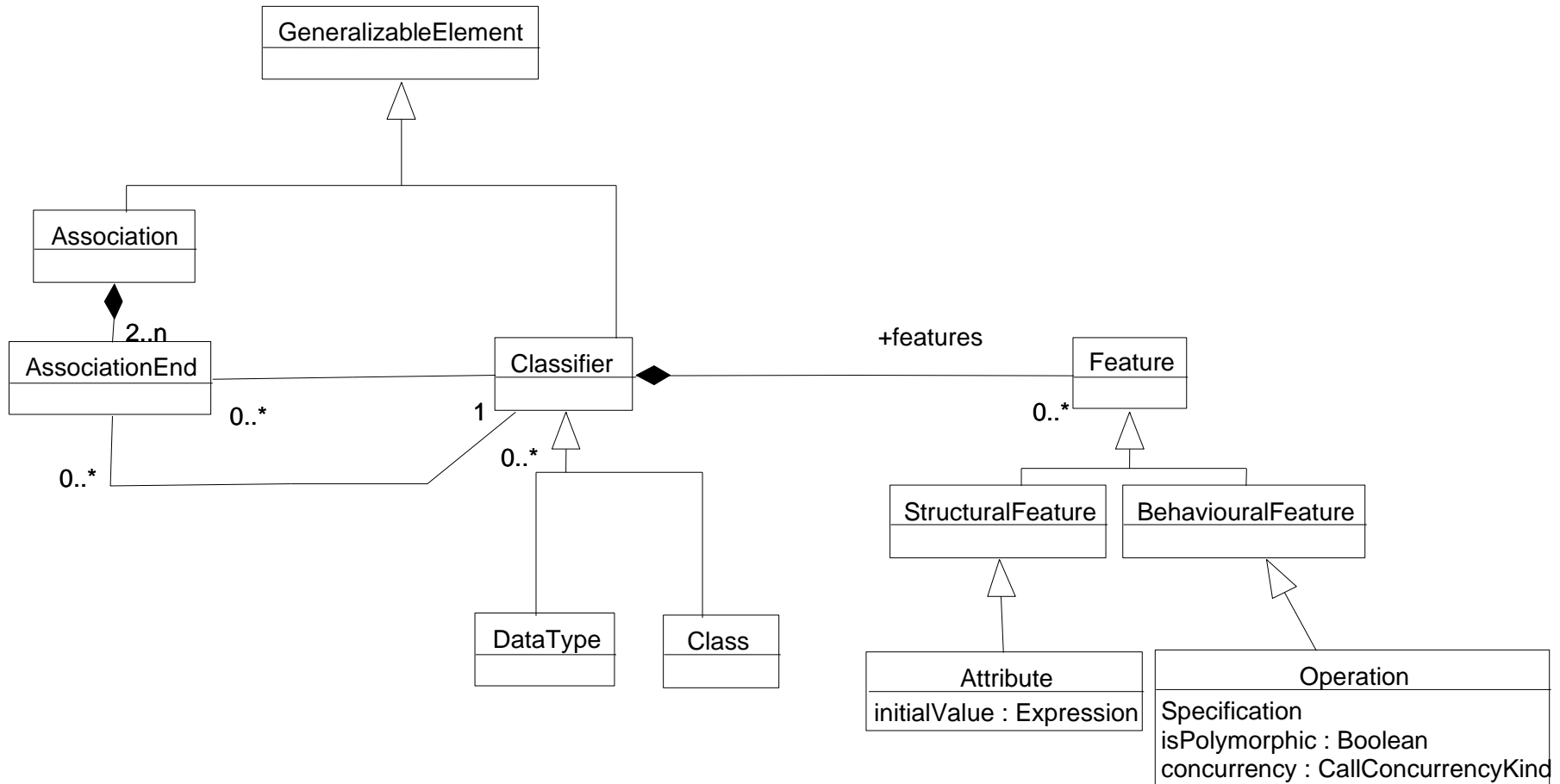
Information modeling

Semantic Models



UML Class diagram

# Parts of UML Metamodel



# UML Information Modeling

- Ref also ISO 19103 Standard for Conceptual Modeling
- The following material is for reference .....

# Objects

Can represents

- One instance

Ola :Person

- One type, interface

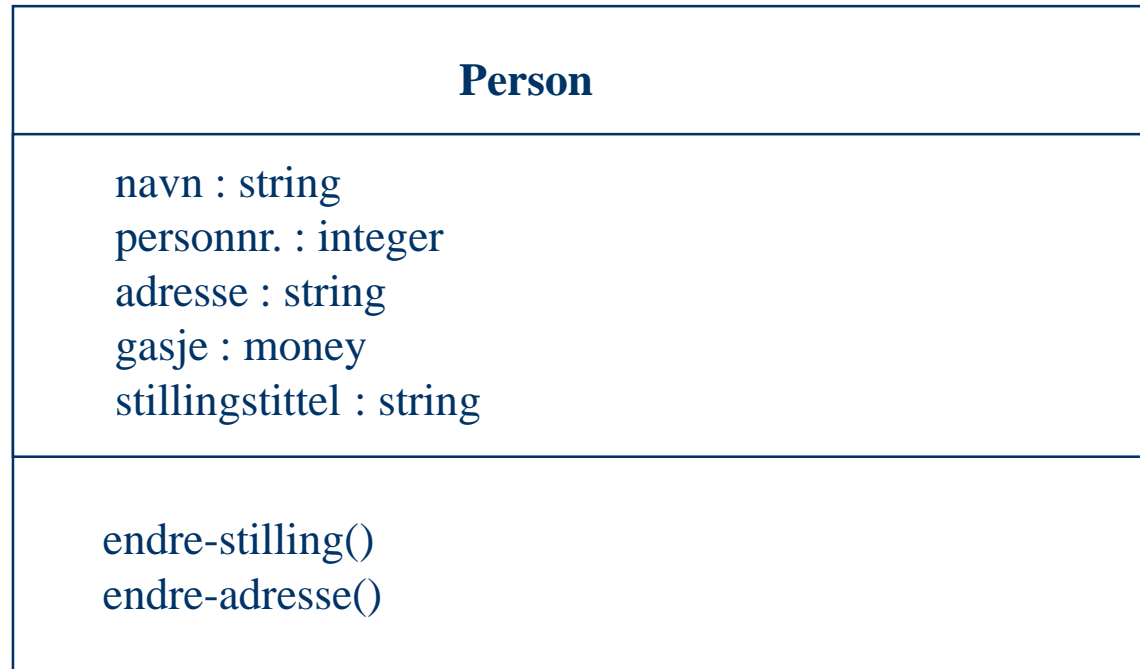
<<Interface>>  
**Person**

- One class

**Person**



# Object and classes - notation



*Example - object class*

# Agile Software Requirements

Lean Requirements Practices for Teams, Programs, and the Enterprise

Dean Leffingwell

Foreword by Don Reinertsen

Agile Software Development Series

Alistair Cockburn and Jim Highsmith,  
Series Editors

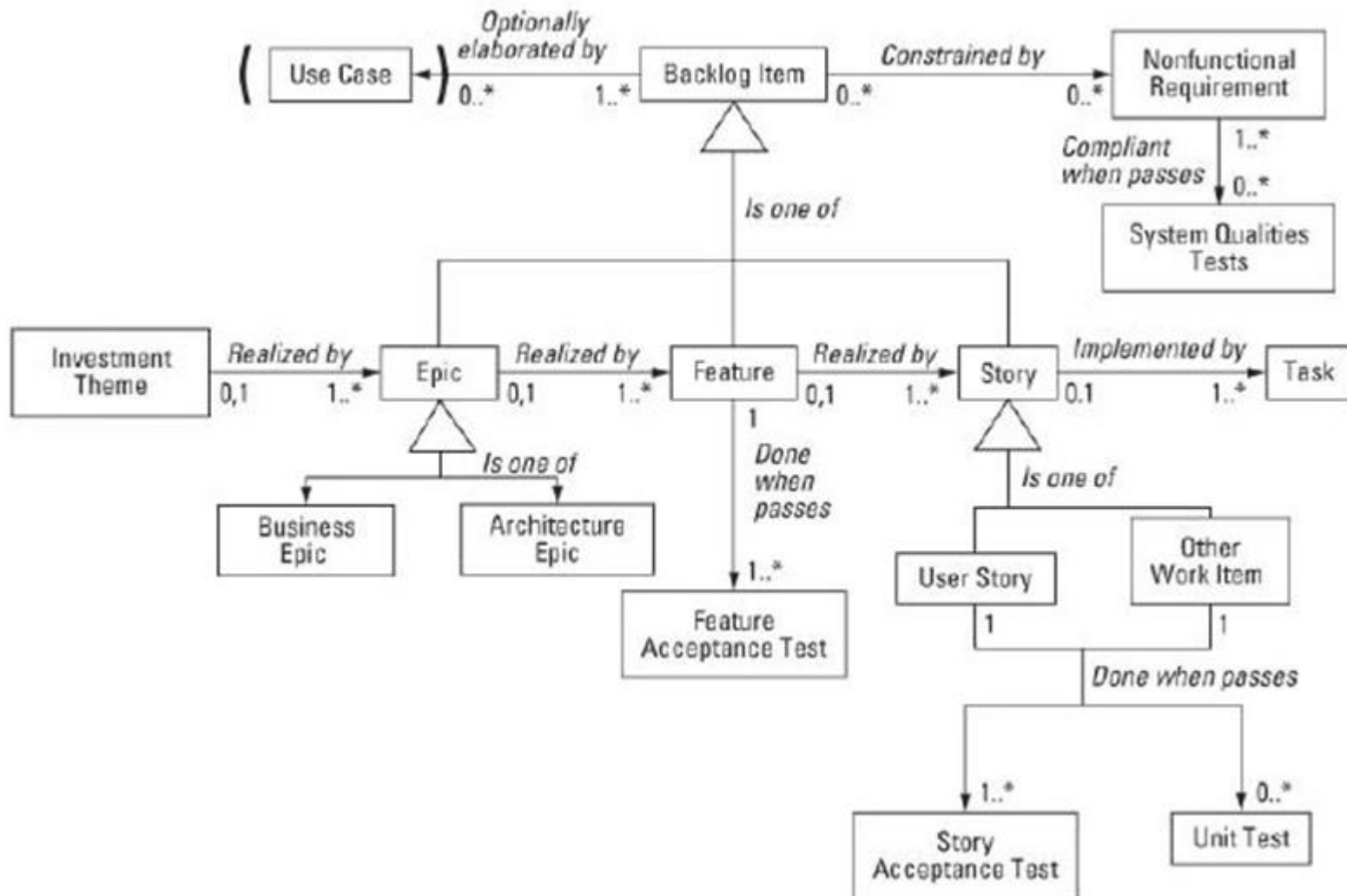
## Latest Books By Dean Leffingwell

Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise  
Scaling Software Agility: Best Practices for Large Enterprises

# User Story template

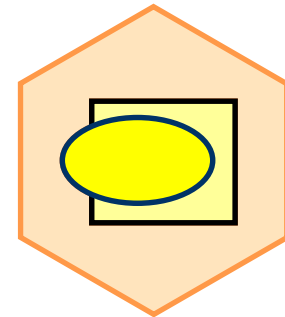
- I <in the role of XX> needs functionality <zzz> to achieve the goal of <YYY>

# Backlog metamodel



# Introduction to The Essentials – using Essence

Module 3 – Use-Case Essentials



# Use-Case Essentials

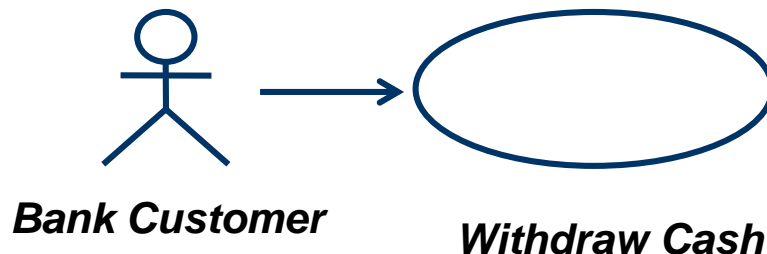
- A way to establish the requirements of the system
  - Use cases place requirements in context
- A way to establish the system boundary
  - The model identifies who or what interacts with the system and what the system should do
- A way to iteratively evolve the requirements
- A way to communicate the requirements to all the stakeholders
  - The use cases provide a common thread through all project activities
- A way to focus the development efforts on delivering customer value
- A way to verify that the requirements have been implemented

**A way to effectively gather requirements and ensure that the system delivers real value to the customers and users**

# What is a Use Case?

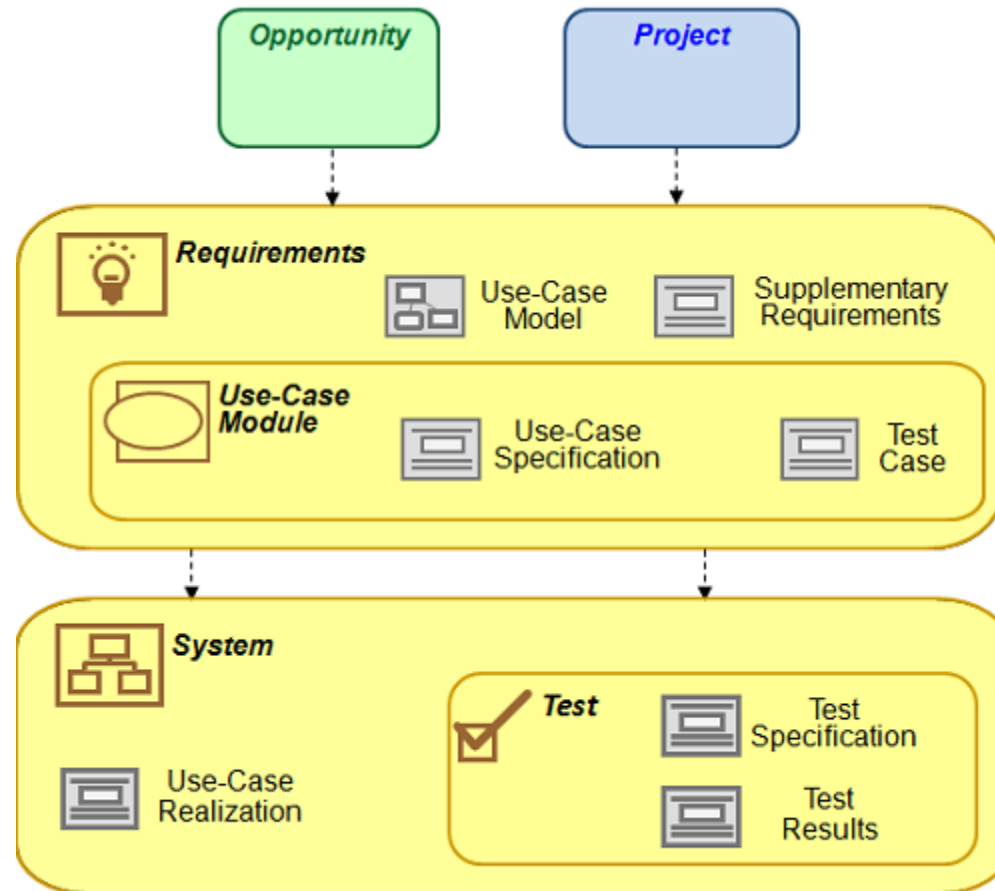
A use case describes a sequence of actions a system performs that yields an observable result of value to a particular actor

- Use cases are shown in UML diagrams



- Use cases are described in text
  - They tell the story of the interactions between actors and the system

# What do we need to produce?





# http://www.balsamiq.com/

File Edit View Favorites Tools Help

Secure Search

McAfee

Suggested Sites Web Slice Gallery Blackboard FAU Mail Technical Services Group

balsamiq®

PRODUCTS COMPANY SUPPORT BLOGS DOWNLOAD BUY LOG IN Search...

WATCH VIDEO TRY WEB DEMO

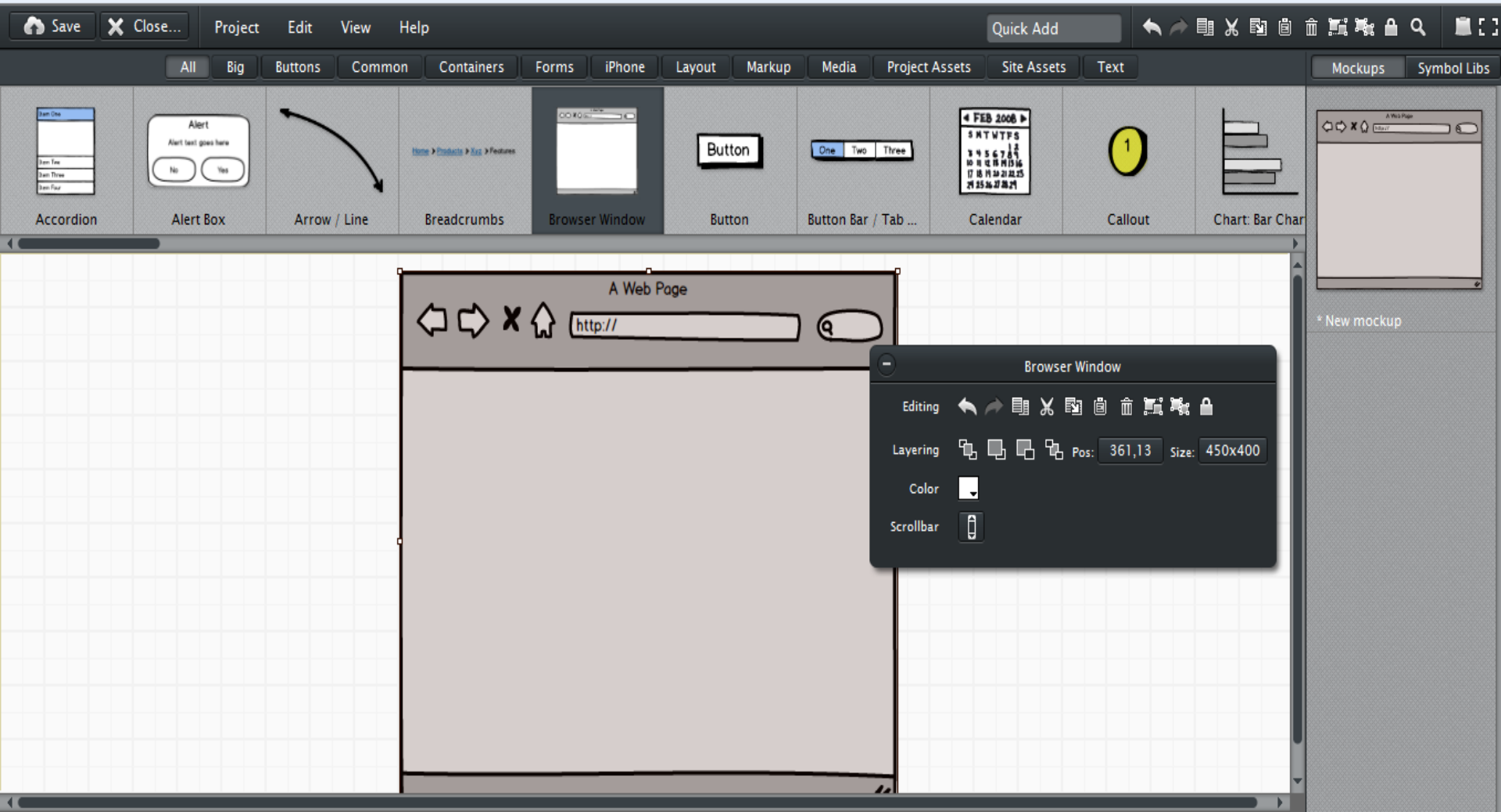
Download Mockups for Desktop

Feedback

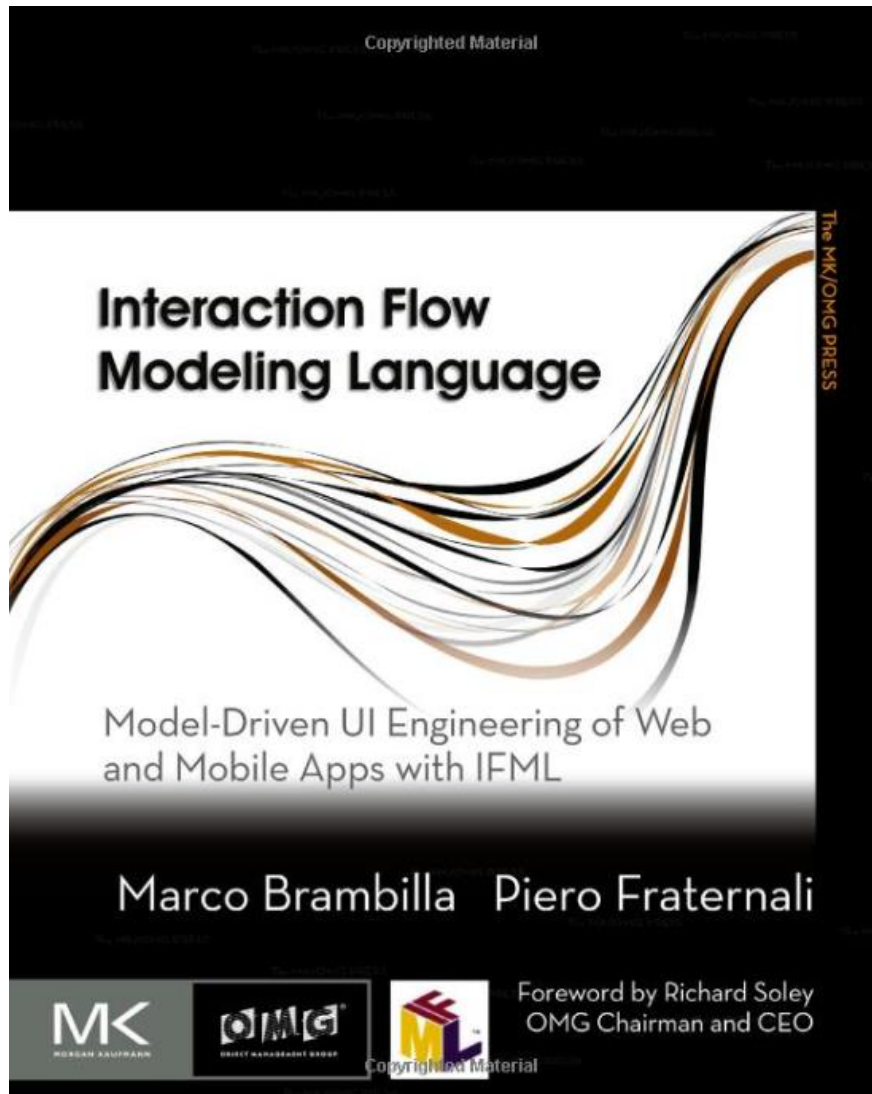
Balsamiq is a small group of passionate individuals who believe work should be fun and that life's too short for bad software.

Meet Balsamiq Mockups, our Rapid Wireframing Tool


# Creation of Mockups





# IFML – for Model Driven Mobile Apps



# WebRatio

WEB  RATIO

MENU  EN  CONTACT US LOGIN

## MOBILE PLATFORM

Build amazing mobile apps in days, not months, without writing a single line of code.

[LEARN MORE](#)

## BPM PLATFORM

Build tailored BPM applications with a fine-grained user experience definition.

[LEARN MORE](#)

## WEB PLATFORM

Focus on creative tasks and build composite web apps in a fraction of the time.







[LEARN MORE](#)

## ENTERPRISE PLATFORM

Build omni-channel applications with a consistent user experience.

[LEARN MORE](#)

... JUST AS THEY DID

<http://www.webratio.com>

# Underlying technologies



**HTML**



**CSS**

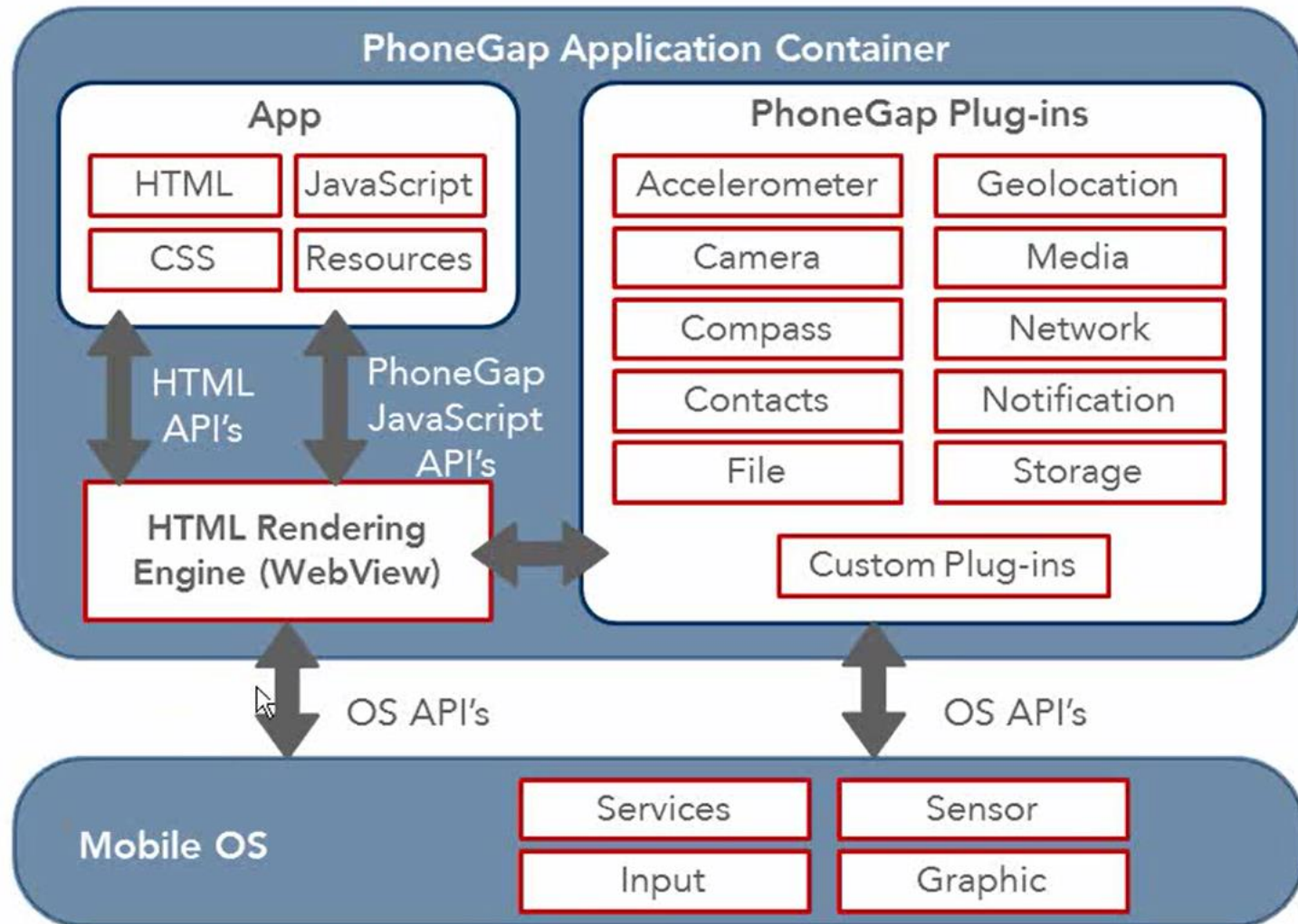


**JS**



**ANGULARJS**

# Mobile App Architecture

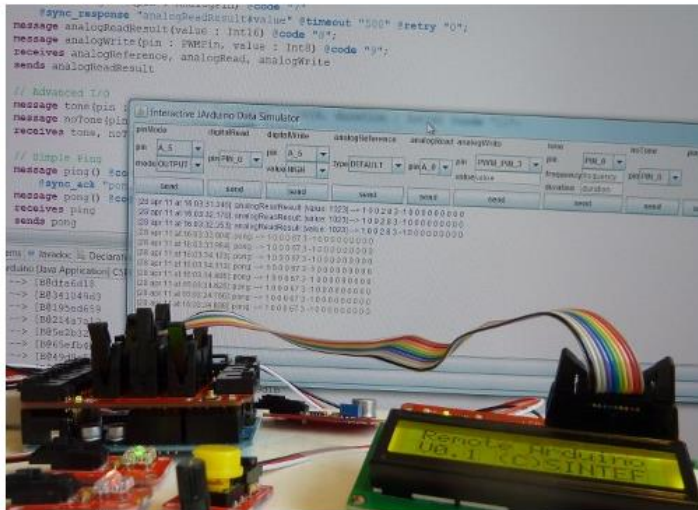


## What is ThingML?

**ThingML** is a modeling language for embedded and distributed systems. It is developed by the Networked Systems and Services department of SINTEF in Oslo, Norway.

ThingML stands for "Thing" Modeling Language as a reference to the so called *Internet of Things*.

The idea of ThingML is to develop a practical model-driven software engineering tool-chain which targets resource constrained embedded systems such as low-power sensor and microcontroller based devices.



ThingML is developed as a domain-specific modeling language which includes concepts to describe both software components and communication protocols. The formalism used is a combination of architecture models, state machines and an imperative action language.

The ThingML toolset includes text editors to create and edit ThingML models, a set of transformations to create diagrams from ThingML models and a set of code generators to compile ThingML to C, Java and JavaScript.

## SEARCH

 Go

## THINGML

- ThingML Home
- Getting Started
- Research
- Contact

## DOCUMENTATION

- Conventions and Naming
- Data Types
- Things
- State Machines
- Actions
- Configurations
- Optimisation

## ARDUINO

- Arduino Compiler
- Core Library
- Devices Library
- Electronic Bricks
- Sample Applications

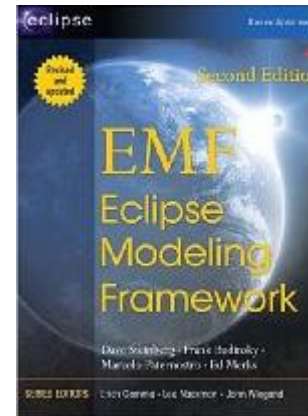
## LINUX C/C++

- Linux C/C++ Compiler
- Linux C-C++ Examples?
- ROS Node Compiler
- ROS Node Examples?

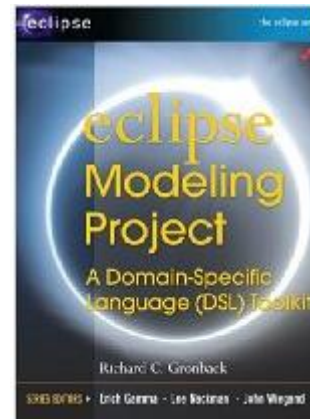
<http://thingml.org/>

# Supporting literature – EMF and GMF

- Book: Eclipse Modeling Framework (2nd Edition) (Paperback)
- [Dave Steinberg](#) (Author), [Frank Budinsky](#) (Author), [Marcelo Paternostro](#) (Author), [Ed Merks](#) (Author)



- Book: Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit (Paperback)
- [Richard C. Gronback](#)





# Sirius

Modeling - platform/resource/org.obeone.network.sample.robot/representations.aird/Topography diagram - Obeo Designer

File Edit Diagram Navigate Search Project Run Window Help

Model Explorer

type filter text

- flow.design
  - org.obeone.network.sample.robot
    - Project Dependencies
    - representations.aird
      - Robot.flow
        - System
          - Topography diagram
          - Flow matrix
          - Processors table
          - Composite Processor Robot Central Unit
            - Processor DSP
            - Processor Motion Engine
            - Fan active
            - Power Input
          - Composite Processor Captor Unit
            - Processor Camera Capture
            - Processor Laser Capture
            - Data Source Front Camera
            - Data Source Back Camera
            - Fan active
            - Data Source Laser
            - Data Source Wifi
            - Data Flow standard

\*Topography diagram

Palette

- Creation Tools
  - Composite Processor
  - Processor
  - State Processor
  - Data Source
  - Flow

\*Flow matrix

	DSP	Motion Engine	Camera Capture	Laser Capture
DSP		X		
Motion Engine			X	
Camera Capture				X
Laser Capture				
Front Camera			X	
Back Camera			X	
Laser				X
Wifi	X			

\*Processors table

	capacity	consumption	load	status	usage
DSP	4	0	4	inactive	low
Motion Engine	9	90	9	active	standard
Camera Capture	8	80	8	active	standard
Laser Capture	4	40	6	active	high

Properties

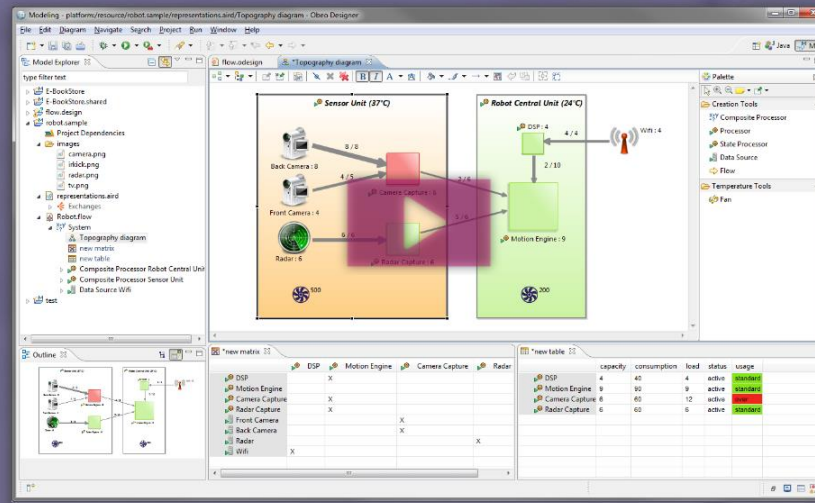
Composite Processor Captor Unit

Semantic	Property	Value
Style	Composite Processor Captor Unit	
	Capacity	10
Appearance	Consumption	120
	Incoming Flows	
	Load	0
	Name	Captor Unit
	Outgoing Flows	

Outline



The easiest way to get your own Modeling Tool



Visual

Declarative

Easy



Sirius 2.0 has been released!



WHAT'S NEW



TRY IT NOW

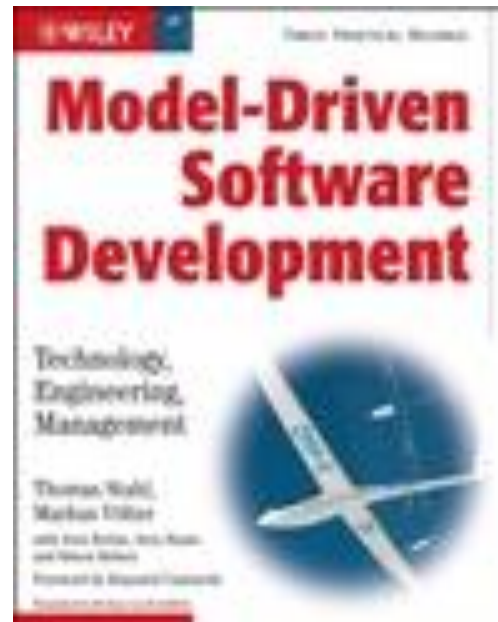


# MDSD

- Book: **Model-Driven Software Development: Technology, Engineering, Management (Paperback)**

by [Thomas Stahl](#), [Markus Voelter](#), [Krzysztof Czarnecki](#)

ISBN: 978-0-470-02570-3



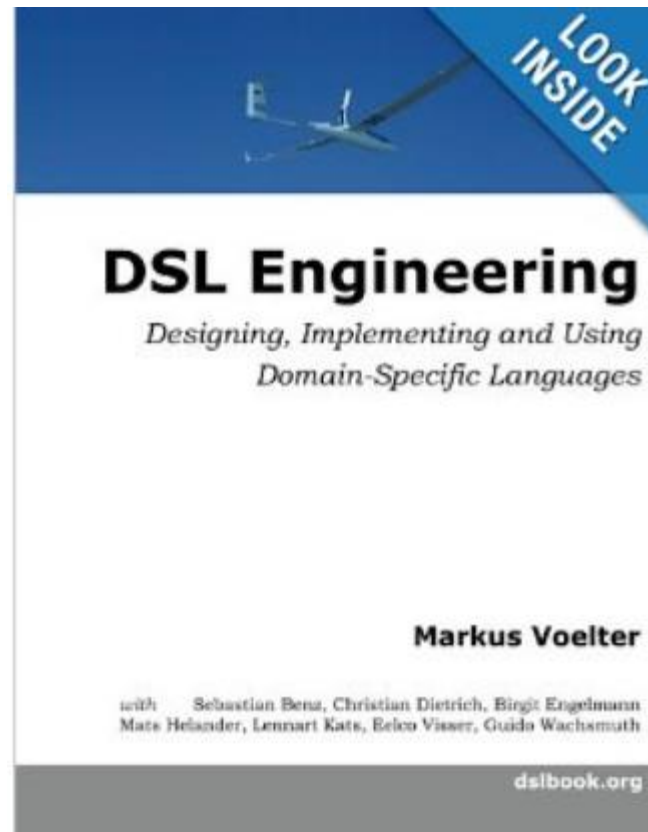
- **Engineering Service Oriented Systems: A Model Driven Approach**, Karakostas, Bill; Zorgios, Yannis

ISBN10: 1599049686 ISBN13: 9781599049687 Cover: Hardcover

April 2008

# DSL Engineering

- **DSL Engineering: Designing, Implementing and Using Domain-Specific Languages Paperback**– January 23, 2013 , 560 pages
- Markus Voelter



# What is Enterprise Modelling?



**Enterprise Modelling (EM)** is a capability for externalising, making and sharing enterprise knowledge.

EM tools can either be:

- used stand-alone to produce various kinds of model views,
- integrated as front-ends to other systems,
- part of an environment providing a contextual user-environment.

# Why Enterprise Architecture?

How can I  
involve my people  
in improving the  
performance of the  
business

































How can I use best  
practices to ensure  
the success of the  
business?

How can I  
ensure that the IS  
technology  
helps the work of  
my people?



# Zachman Framework – for Enterprise Architecture (IBM, 1987)

VA Enterprise Architecture	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why	Based on work by John A. Zachman
<b>SCOPE (CONTEXTUAL)</b>  <i>Planner</i>	Things Important to the Business  Entity = Class of Business Thing	Processes Performed  Function = Class of Business Process	Business locations  Node = Major Business Locations	Important Organizations  People = Major Organizations	Events Significant to the Business  Time = Major Business Event	Business Goals and Strategy  Ends/Mears = Major Business Goals	<b>SCOPE (CONTEXTUAL)</b>  <i>Planner</i>
<b>ENTERPRISE MODEL (CONCEPTUAL)</b>  <i>Owner</i>	Semantic Model  Ent = Business Entity Rel = Business Relationship	Business Process Model  Proc = Business Process I/O = Business Resources	Business Logistics System  Node = Business Location Link = Business Linkage	Work Flow Model  People = Organization Unit Work = Work Product	Master Schedule  Time = Business Event Cycle = Business Cycle	Business Plan  End = Business Objective Mears = Business Strategy	<b>ENTERPRISE MODEL (CONCEPTUAL)</b>  <i>Owner</i>
<b>SYSTEM MODEL (LOGICAL)</b>  <i>Designer</i>	Logical Data Model  Ent = Data Entity Rel = Data Relationship	Application Architecture  Proc = Application Function I/O = User Views	Distributed System Architecture  Node = IS Function Link = Line Characteristics	Human Interface Architecture  People = Role Work = Deliverable	Processing Structure  Time = System Event Cycle = Processing Cycle	Business Rule Model  End = Structural Assertion Mears = Action Assertion	<b>SYSTEM MODEL (LOGICAL)</b>  <i>Designer</i>
<b>TECHNOLOGY MODEL (PHYSICAL)</b>  <i>Builder</i>	Physical Data Model  Ent = Segment/Table Rel = Pointer/Key	System Design  Proc = Computer Function I/O = Data Elements/Sets	Technology Architecture  Node = Hardware/Software Link = Line Specifications	Presentation Architecture  People = User Work = Screen Format	Control Structure  Time = Execute Cycle = Component Cycle	Rule Design  End = Condition Mears = Action	<b>TECHNOLOGY MODEL (PHYSICAL)</b>  <i>Builder</i>
<b>DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)</b>  <i>Sub-Contractor</i>	Data Definition  Ent = Field Rel = Address	Program  Proc = Language Statement I/O = Control Block	Network Architecture  Node = Addresses Link = Protocols	Security Architecture  People = Identity Work = Job	Timing Definition  Time = Interrupt Cycle = Machine Cycle	Rule Design  End = Sub-Condition Mears = Step	<b>DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)</b>  <i>Sub-Contractor</i>
<b>FUNCTIONING ENTERPRISE</b>	Data  Ent = Rel =	Function  Proc = I/O =	Network  Node = Link =	Organization  People = Work =	Schedule  Time = Cycle =	Strategy  End = Mears =	<b>FUNCTIONING ENTERPRISE</b>
	<b>DATA What</b>	<b>FUNCTION How</b>	<b>NETWORK Where</b>	<b>PEOPLE Who</b>	<b>TIME When</b>	<b>MOTIVATION Why</b>	

# Zachman with OMG standards

	<b>Data</b> (What)	<b>Function</b> (How)	<b>Network</b> (Where)	<b>People</b> (Who)	<b>Time</b> (When)	<b>Motivation</b> (Why)
<b>Scope</b> (Contexts)	List of things important to business <b>SBVR</b>	List of processes that the business performs <b>VDM</b>	List of locations which the business operates <b>VDM</b>	List of organizations important to the business <b>OSM</b>	List of events/cycles important to the business <b>DTFV</b>	List of business goals/strategies <b>BMM</b>
<b>Business</b> (Concepts)	Semantic Model <b>ODM, IMM (CWM)</b>	Business Process Model <b>BPMN, CMPM</b>	Business Logistics System <b>BPMN, CMPM</b>	Workflow Model <b>OSM, BPMN, CMPM</b>	Master Schedule <b>BPMN, CMPM, DTFV</b>	Business Plan <b>SBVR</b>
<b>System</b> (Logic)	Logical Data Model <b>ODM, IFML IMM (CWM), UML</b>	<b>IFML</b> Application Architecture <b>SoaML, UML</b>	Distributed System Architecture <b>SoaML, UML</b>	Human Interface Architecture <b>BPMN, CMPM</b>	Process Structure <b>BPMN, CMPM, DTFV</b>	Business Rule Model <b>SBVR</b>
<b>Technology</b> (Physics)	Physical Data Model <b>IMM (CWM), UML</b>	System Design <b>SoaML, UML</b>	Technology Architecture <b>SoaML, UML</b>	Presentation Architecture	<b>(ThingML)</b> Control Structure <b>BPMN, CMPM, DTFV</b>	Rule Design <b>SBVR</b>
<b>Component</b> (Assemblies)	Data Definition <b>IMM (CWM), UML</b>	Program <b>UML</b>	Network Architecture <b>UML</b>	Security Architecture	Timing Definition <b>DTFV</b>	Rule Definition <b>SBVR</b>
<b>Operation</b> (Instances)	Data	Function	Network	Organization	Schedule	Strategy

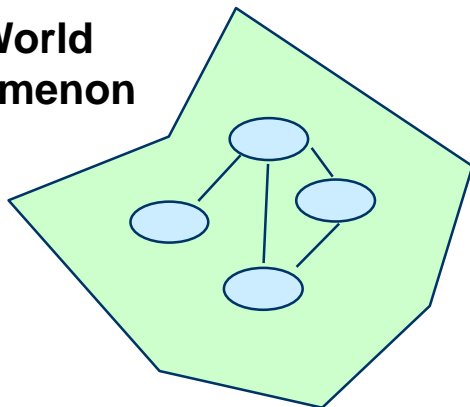


# System and objects

A system is a part of the real world which we choose to regard as a whole, separated from the rest of the world during some period of consideration.

A whole that we choose to consider as a collection of objects, each object being characterized by attributes and by actions which may involve itself and other objects.

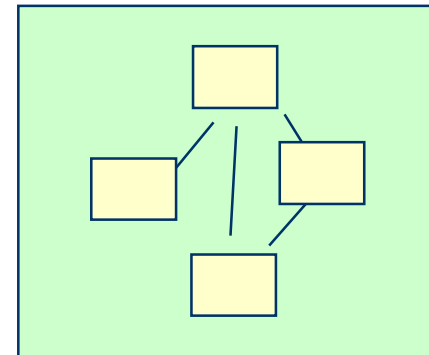
Real-World phenomenon



o o Mental modell



Manifest Model



**Next Lecture, January 23, 2017**

**Modeling structure and behaviour (UML and UML 2.0)**