Interaction Flow Modeling Language

Model-Driven Development of Software Front Ends

Marco Brambilla

Politecnico di Milano and WebRatio

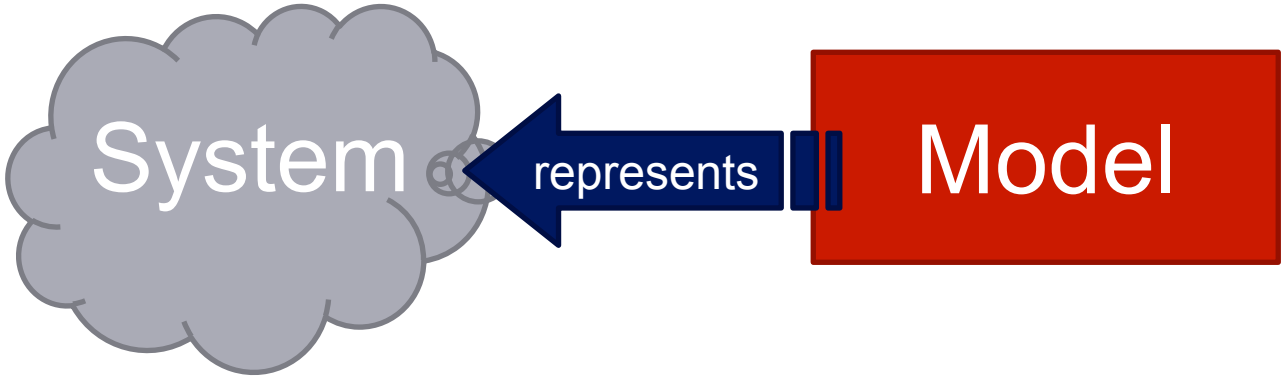
@marcobrambi

marco.brambilla@polimi.it

The modeling approach

Models

What is a model?



Mapping Feature	A model is based on an original (=system)
Reduction Feature	A model only reflects a (relevant) selection of the original‘s properties
Pragmatic Feature	A model needs to be usable in place of an original with respect to some purpose

Purposes:

- descriptive purposes
- prescriptive purposes

MDSE aim at large

MDSE considers models as first-class citizens in software engineering

The way in which models are defined and managed is based on the actual needs that they will address.

MDSE defines sound engineering approaches to the definition of

- models
- transformations
- development process.

WEB RATIO

POLITECNICO MILANO

Concepts

Abstraction from specific technologies

- model once, build everywhere

Automated code generation from abstract models

- Increased **productivity** and **efficiency** (models stay up-to-date)

Separate development of application and infrastructure

- Separation of application-code and infrastructure-code (e.g. Application Framework) increases **reuse**

Modeling Languages

Domain-Specific Languages (DSLs):

languages that are designed specifically for a certain domain or context

DSLs have been largely used in computer science. Examples: HTML, Logo, VHDL, Mathematica, SQL

General Purpose Modeling Languages (GPMLs, GMLs, or GPLs):

languages that can be applied to any sector or domain for (software) modeling purposes

The typical examples are: UML, Petri-nets, or state machines

Model Transformations

Purpose: Transforming items

- defining a mapping between elements of a model to elements to another one (**model mapping or model weaving**)
- Code is just another model

Transformations themselves can be seen as models

Types of models

Static models:

Focus on the static aspects of the system in terms of managed data and of structural shape and architecture of the system.

Dynamic models:

Emphasize the dynamic behavior of the system by showing the execution

Modeling Levels

CIM, PIM, PSM

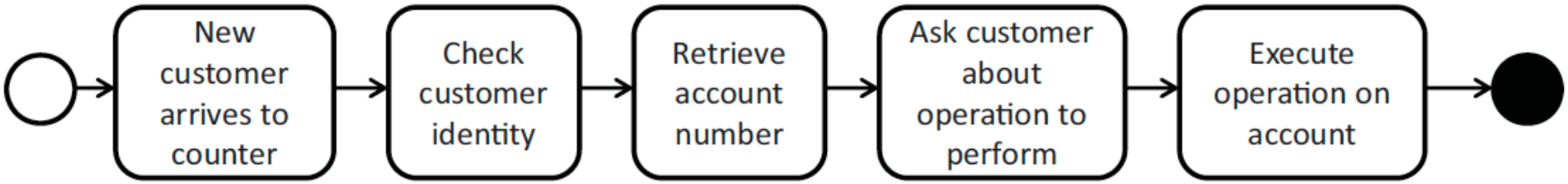
Computation independent (CIM): describe requirements and needs at a very abstract level, without any reference to implementation aspects

Platform independent (PIM): define the behavior of the systems in terms of stored data and performed algorithms, without any technical or technological details

Platform-specific (PSM): define all the technological aspects in detail

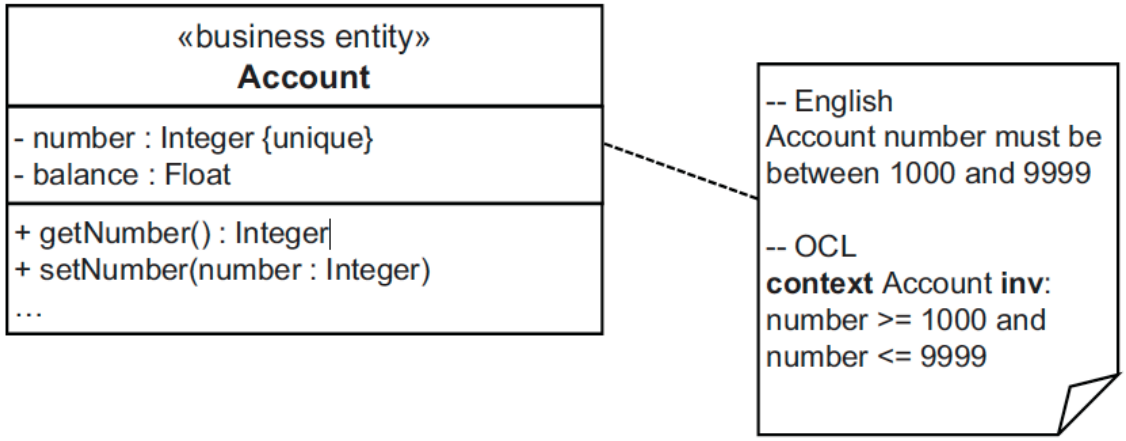
Modeling levels - CIM

Eg., business process

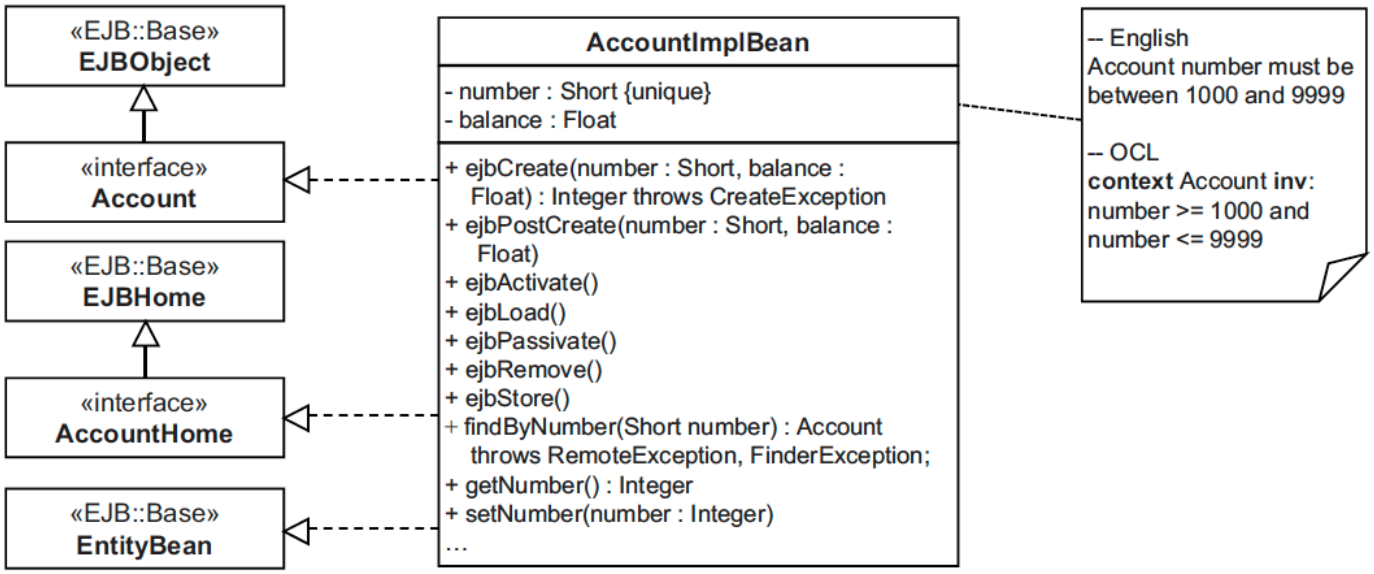


Modeling levels - PIM

Eg., business object description and constraints



Modeling levels - PSM



How the functionality in the PIM is realized on a certain platform

Using a UML-Profile for the selected platform, e.g., EJB

The UI Modeling Problem

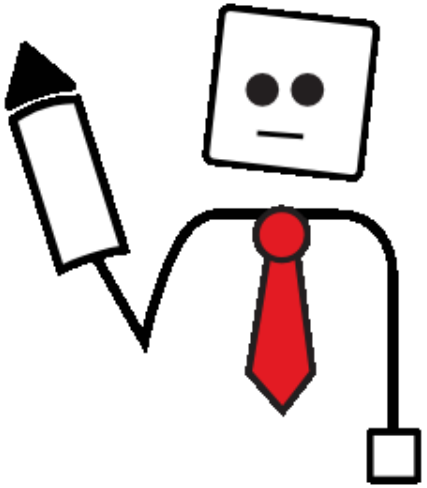
UI Modeling Problem

User interface and interaction development is a painful phase of software process

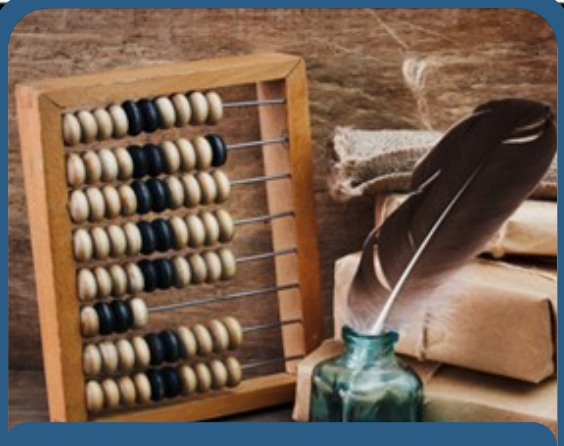
… for everybody!

The UI Design Problem

Costly and Inefficient process



Complexity of user interfaces (UIs)

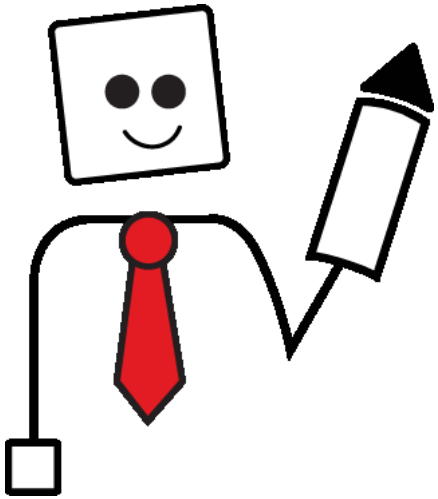


Ineffective tools

Manual development

No MDE technology

The UI Design solution: IFML



Platform independent description of UIs

Focused on user interactions

No definition of graphics and styles

Reference external models

Standardization gap

- User interaction has been overlooked in software engineering standards
- Hence **the Interaction Flow Modeling Language (IFML)**

- In less than 2 years (a record in OMG!), we obtained approval of the IFML standard

The Interaction Flow Modeling Language

IFML Objectives

Content

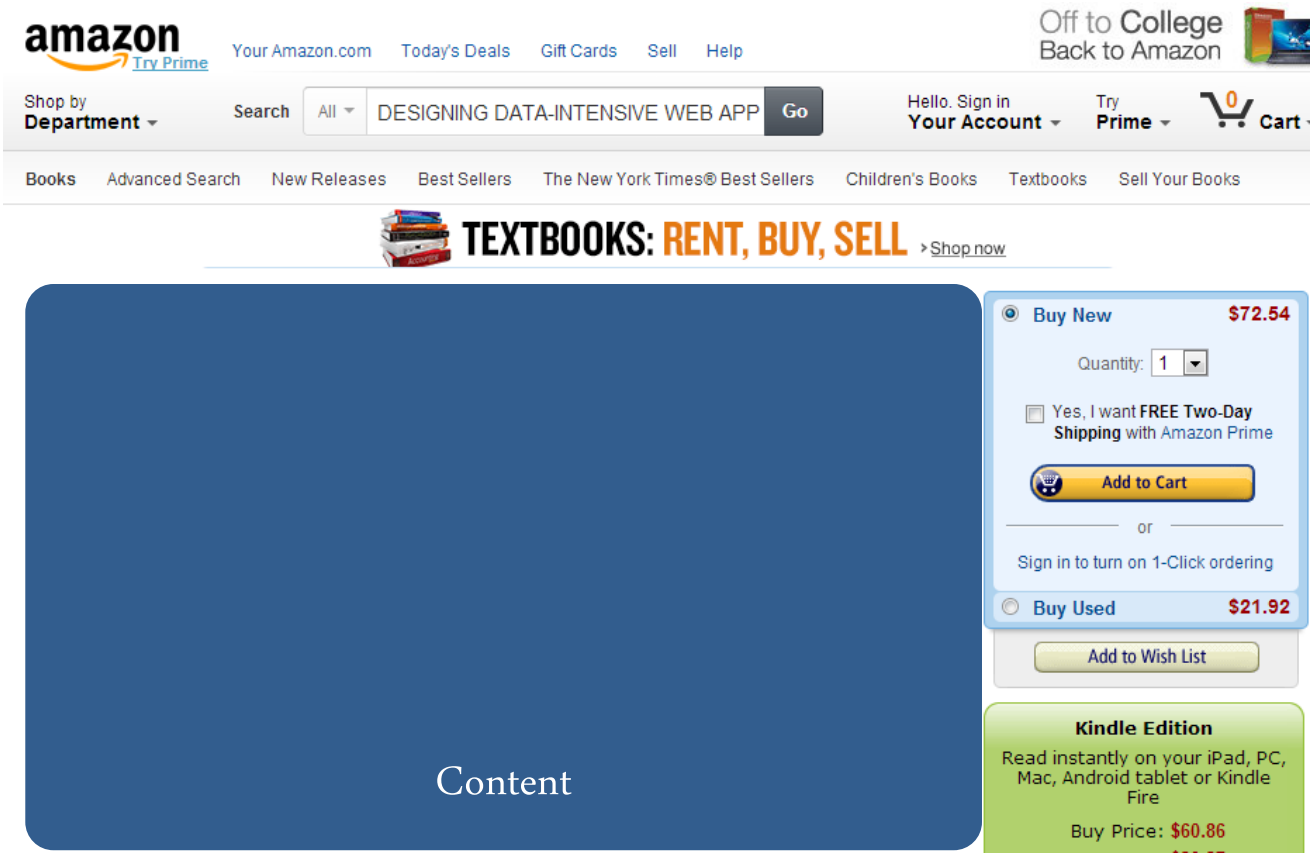
Navigation Path

Event

Binding to
Business Logic

Binding to Persistence Layer

IFML Objectives: Content



WEB RATIO

POLITECNICO MILANO

IFML Objectives: Navigation Path

IFML Objectives: Navigation Path

IFML Objectives: Events

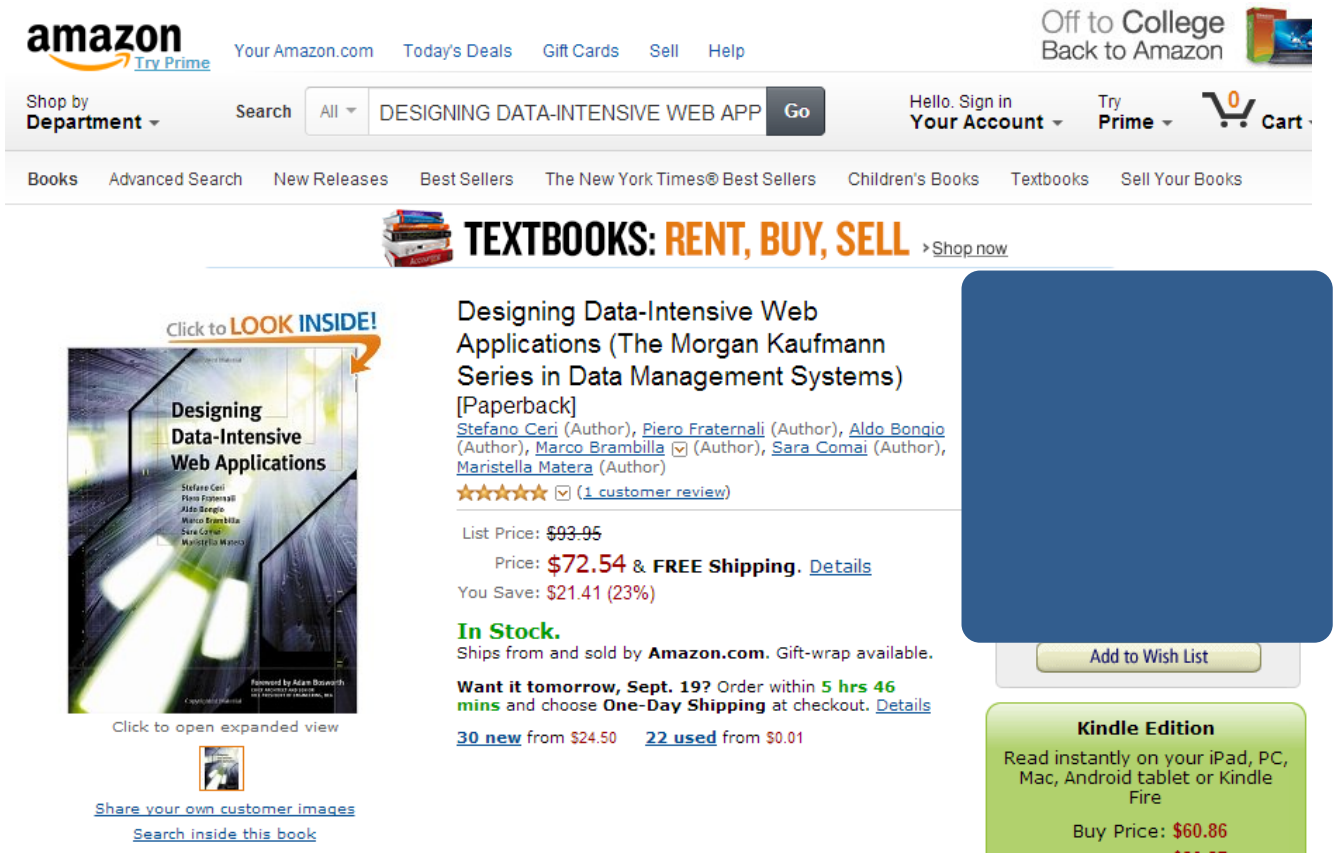
WEB RATIO

IFML Objectives: Events

WEB RATIO

POLITECNICO MILANO

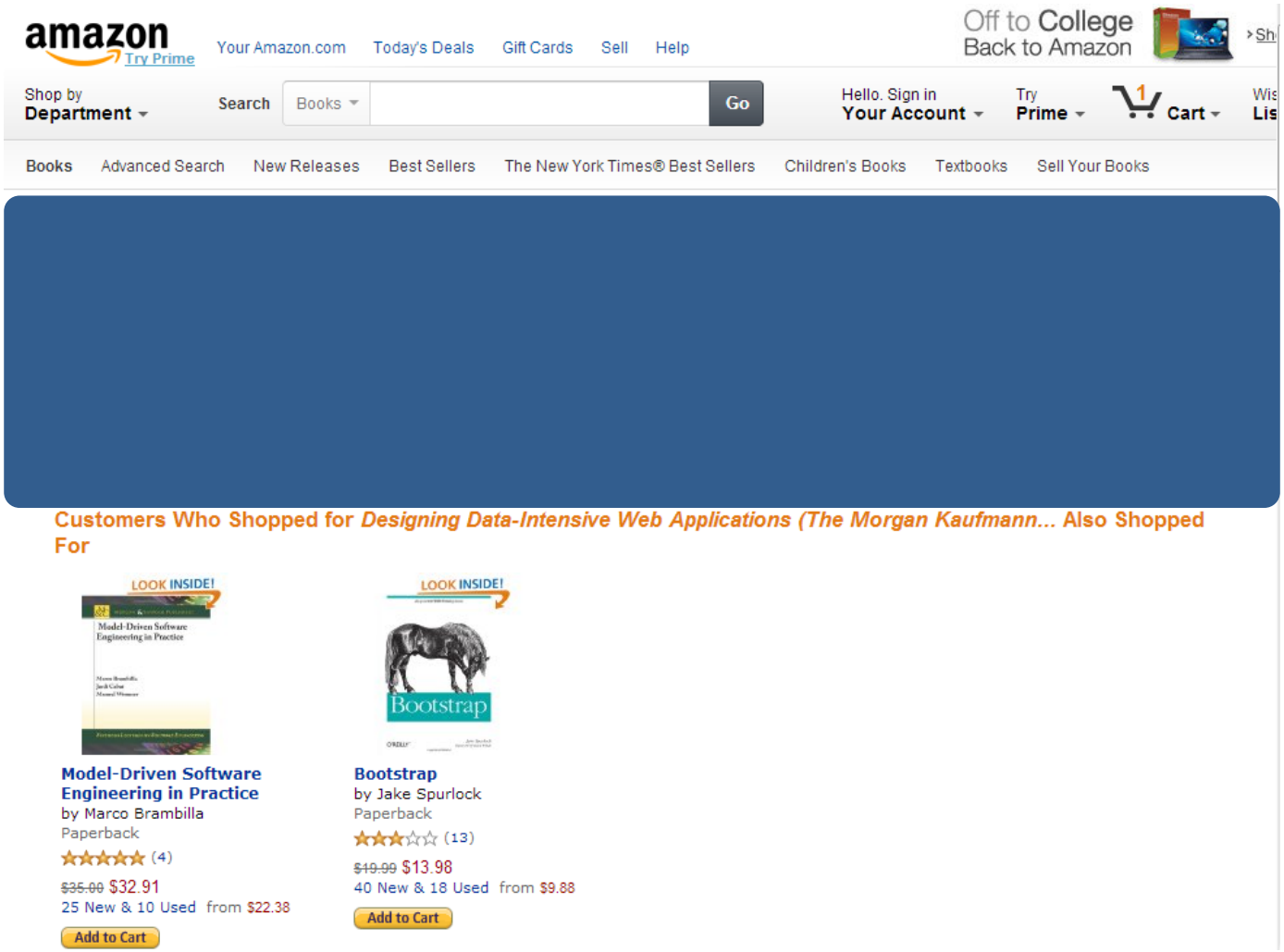
IFML Objectives: Binding to business logic



WEB RATIO

POLITECNICO MILANO

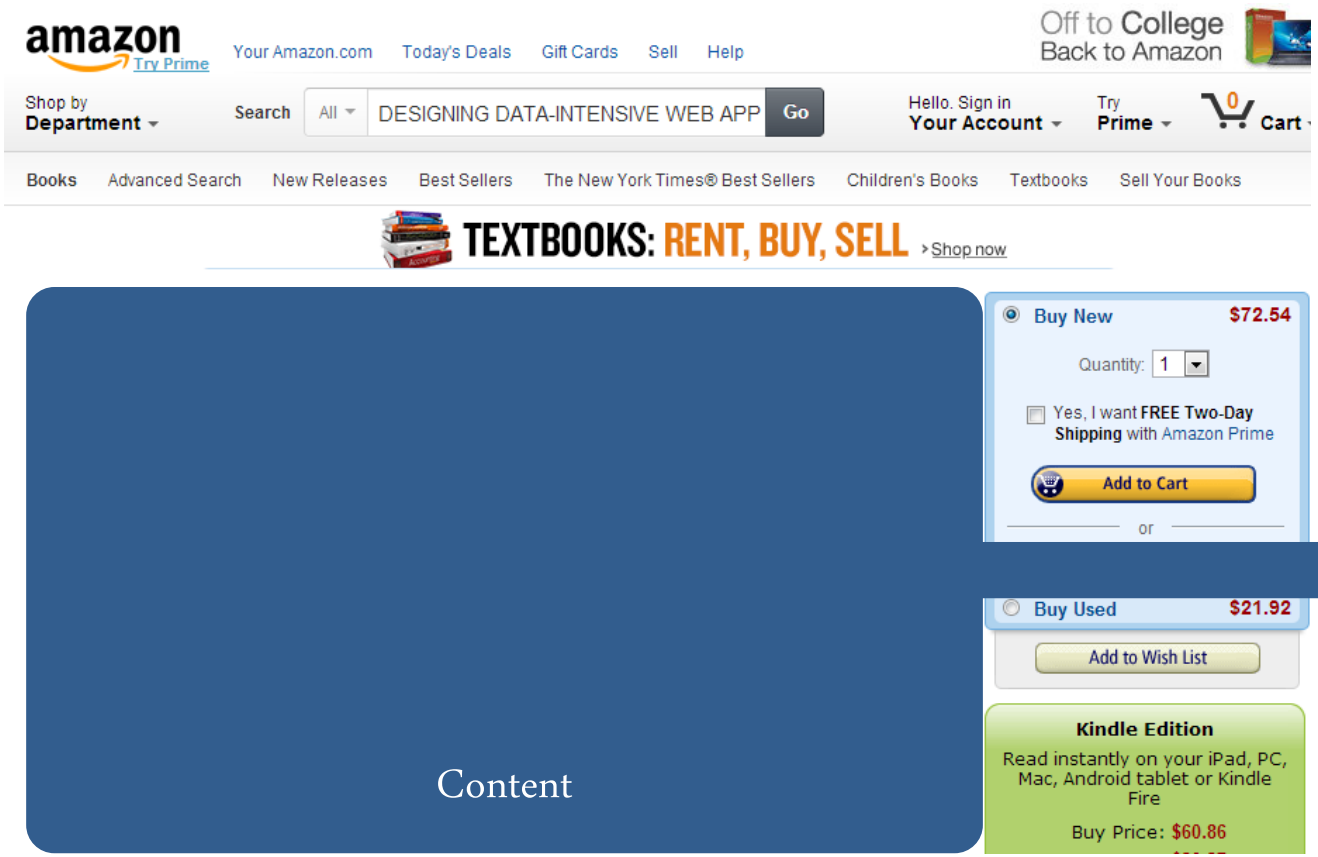
IFML Objectives: Binding to business logic



WEB RATIO

POLITECNICO MILANO

IFML Objectives: Binding to persistence



Book

Title: string
Cover: file
List Price: currency
Price: currency
Save: currency
Rating: integer
……

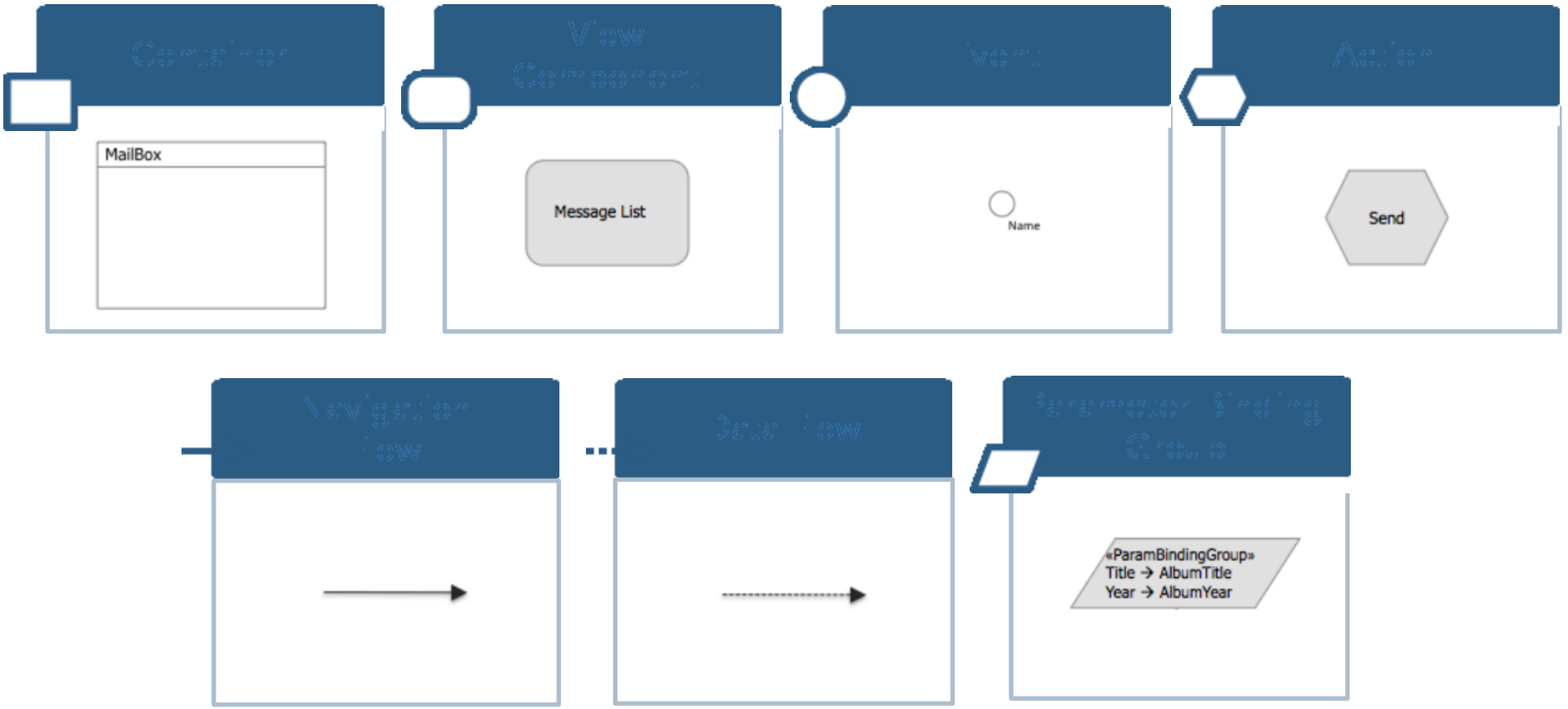
WEB RATIO

IFML Overview

WEB RATIO

POLITECNICO MILANO

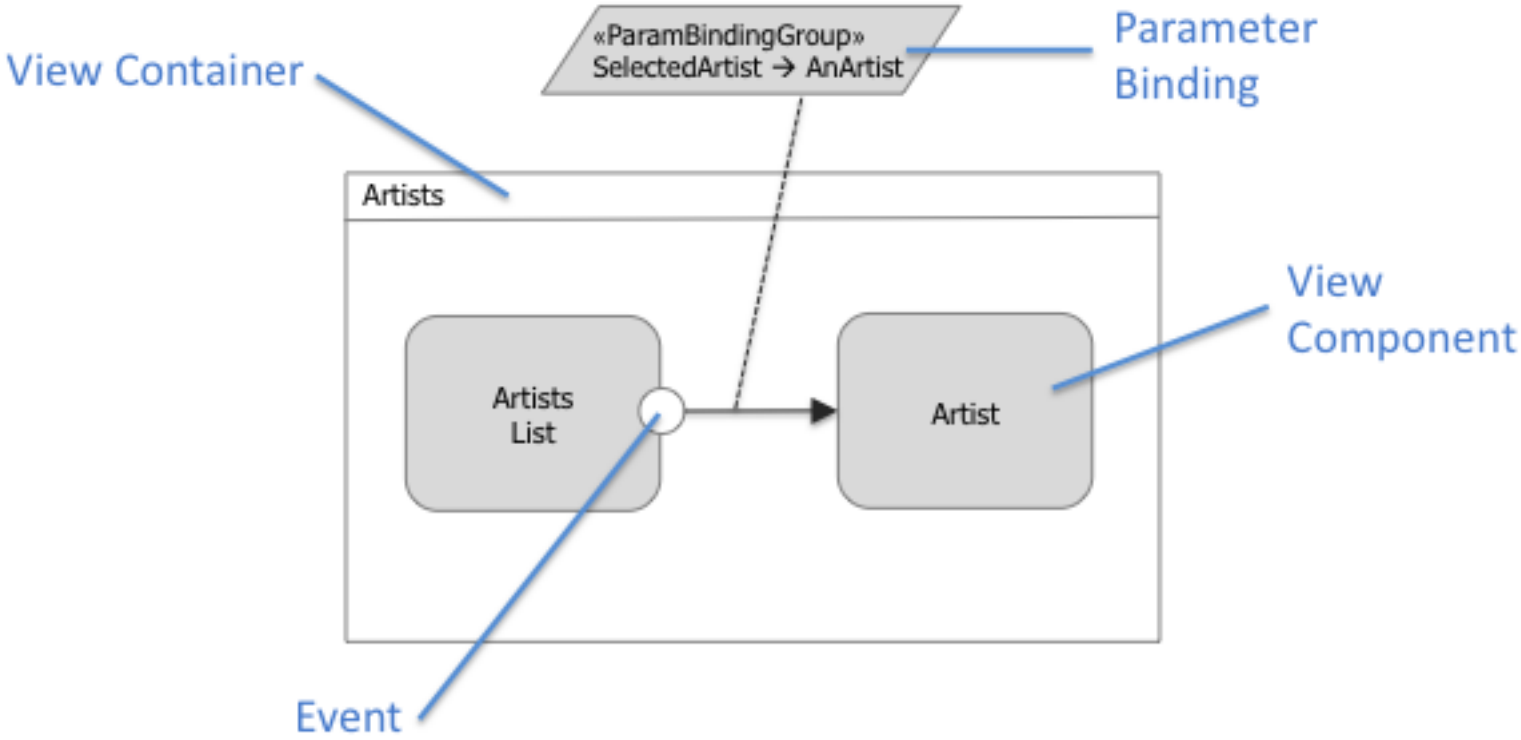
IFML Essentials



Covered aspects

- **Multiple views** for the same application
- **Mobile and multi-device** applications
- Visualization and input of data, and production of **events**
- **Components** independent of concrete widgets and presentation
- **Interaction flow**, initiated by the user or by external events
- **User context**: the user status in the current instant of the interaction (position, history, machine, platform,…)
- **Modularization** of the model (design-time containers for reuse purpose)
- User input **validation**, according to OCL or other existing constraint languages

IFML by example



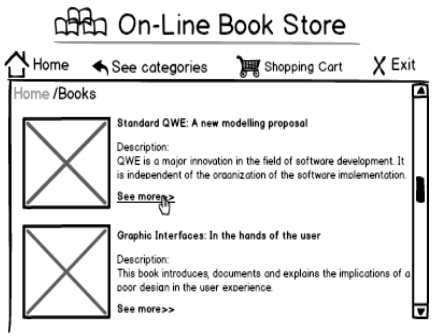
Basic navigation flow between ViewComponents

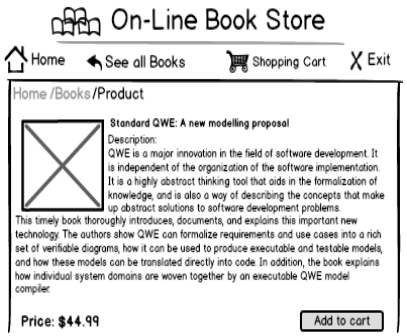
IFML ViewComponents

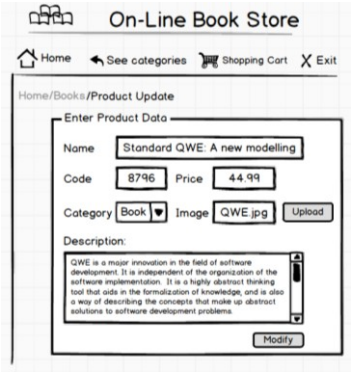
ProductList

ProductDetails

ProductEntry Form



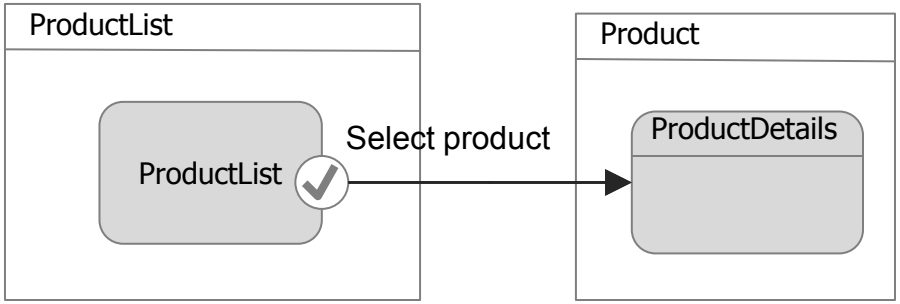


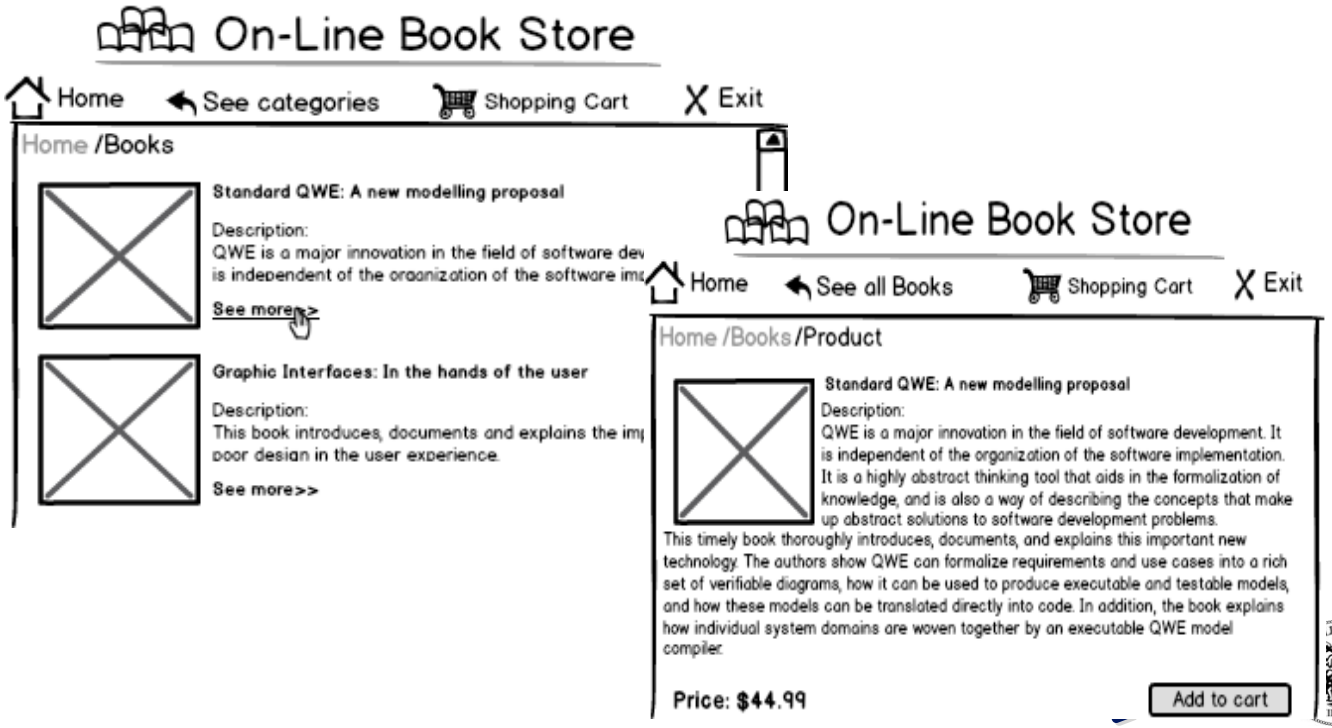


WEB RATIO

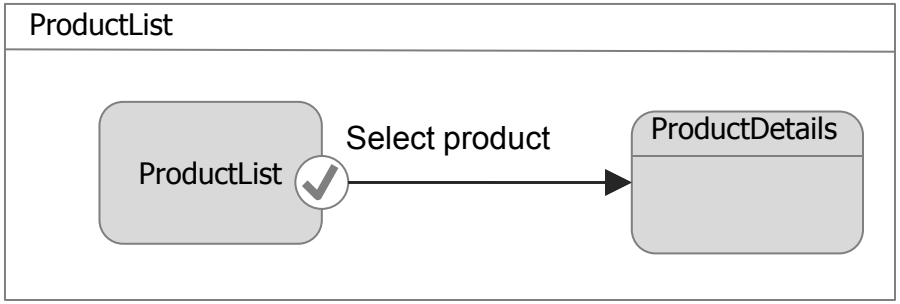
POLITECNICO MILANO

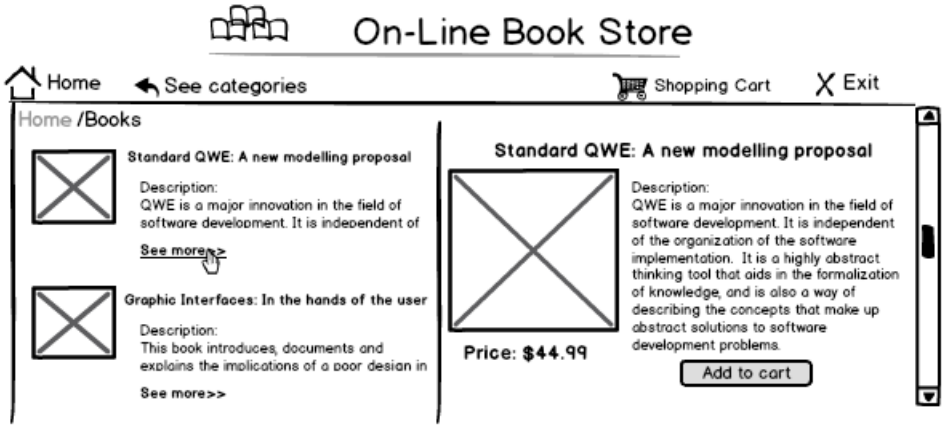
IFML Multiple containers and navigation



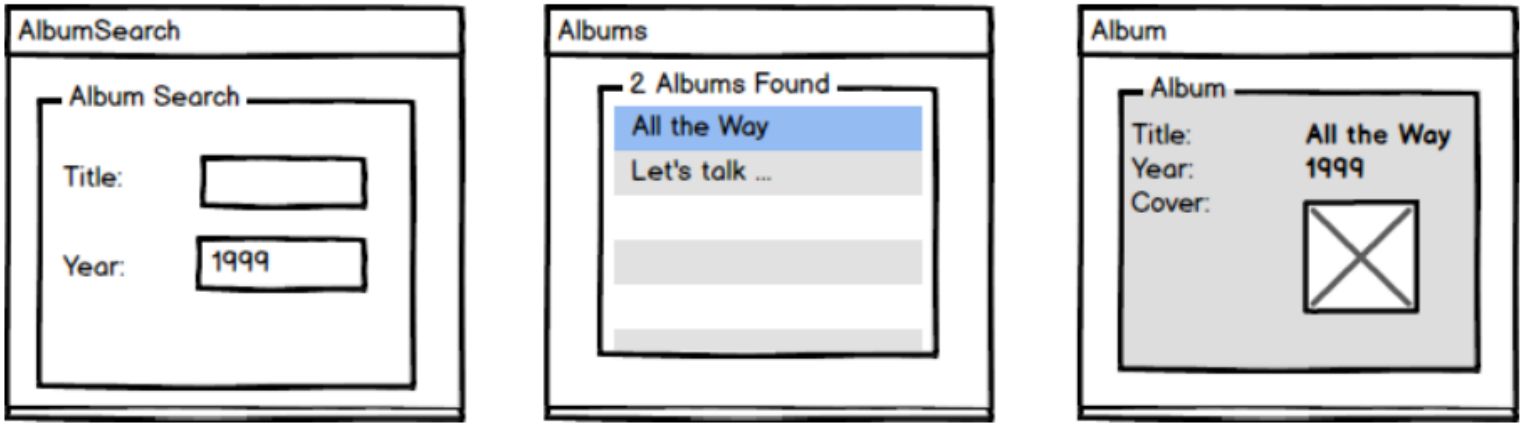


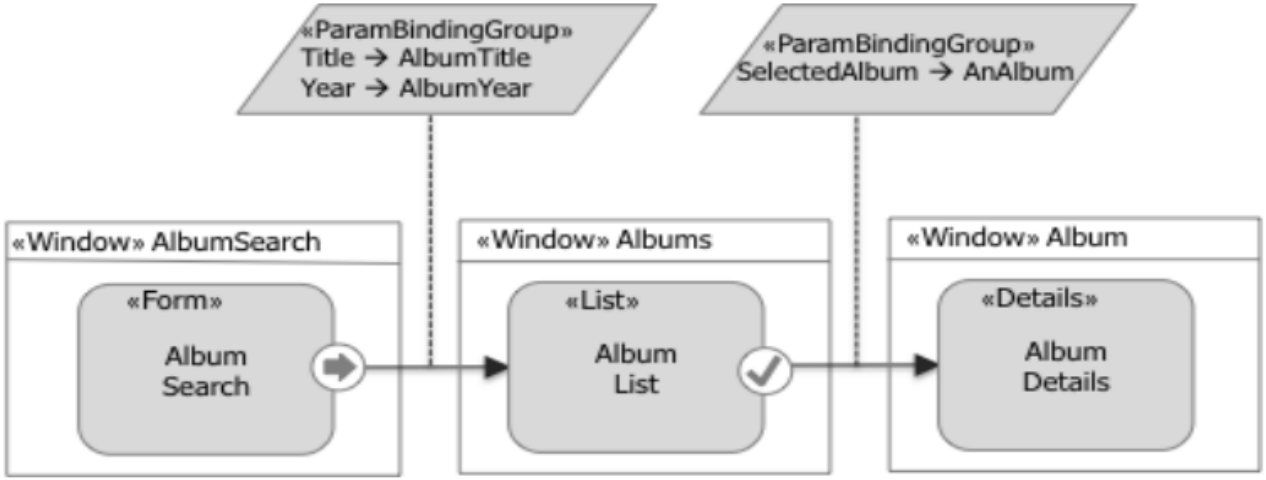
IFML Single container and navigation





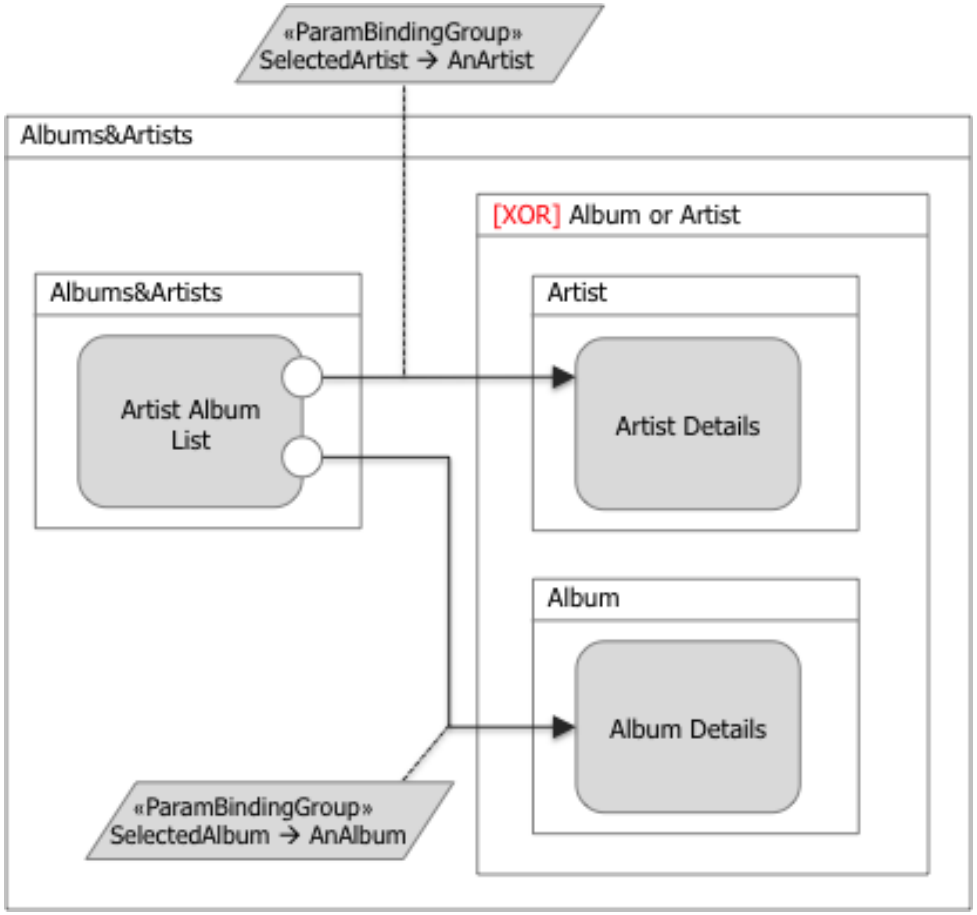
IFML by example





WEB RATIO

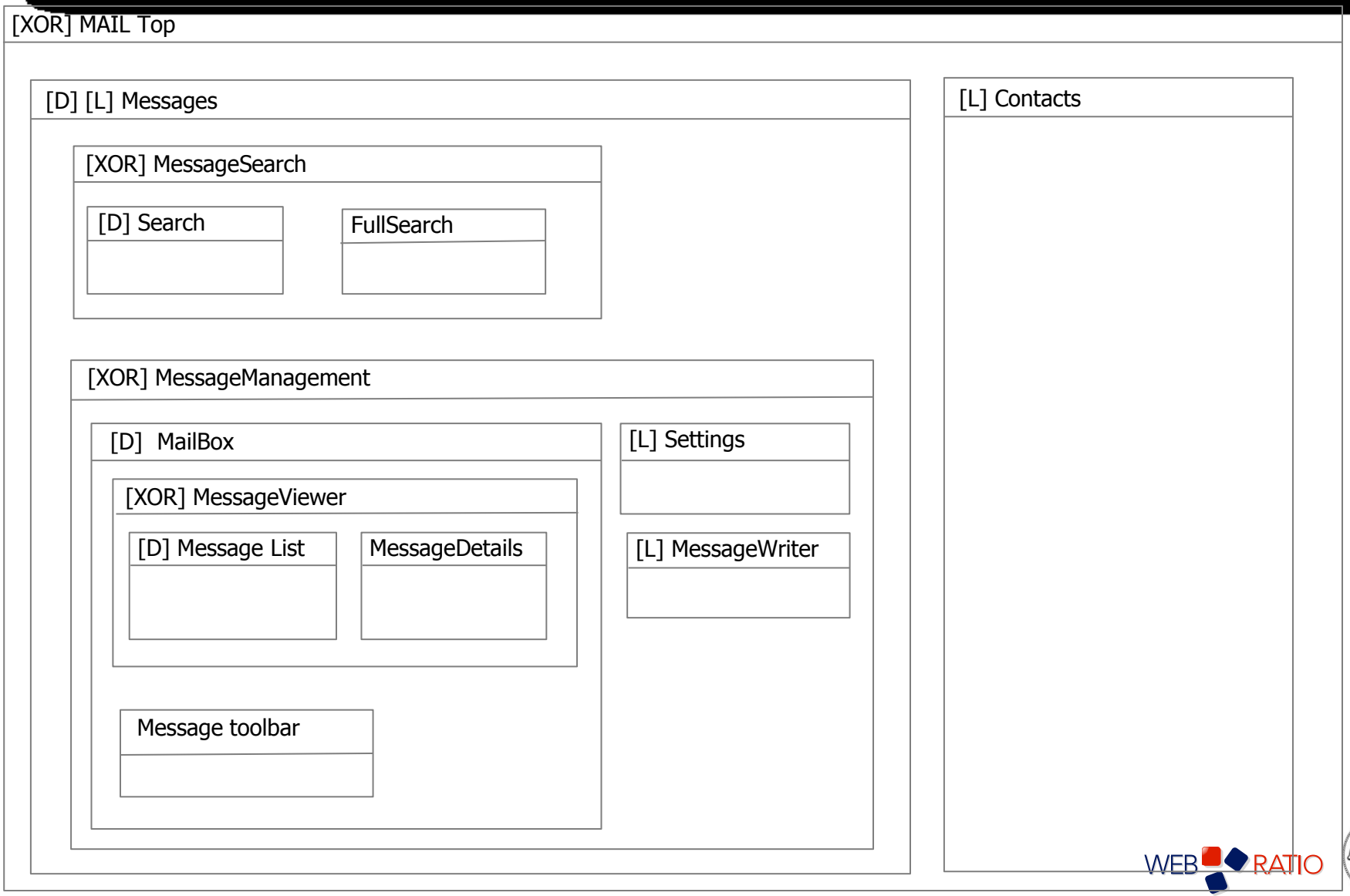
IFML by example



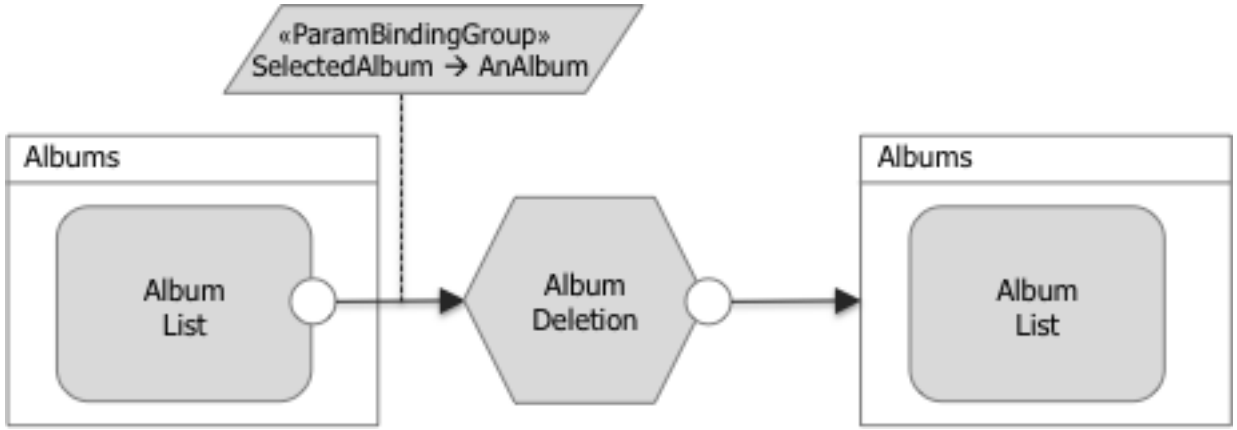
Nesting of ViewContainers

Tagged ViewContainers (XOR, L, D, Modal, Modeless)

IFML ViewContainers

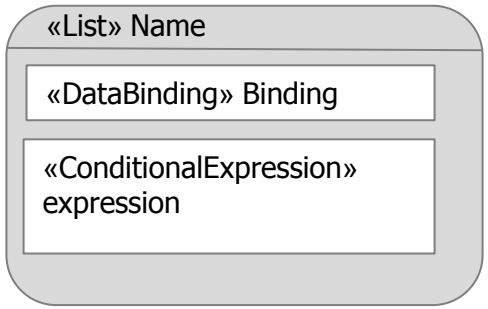


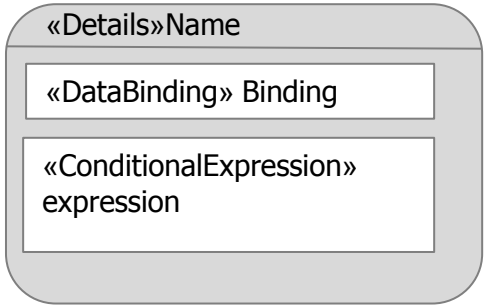
IFML by example

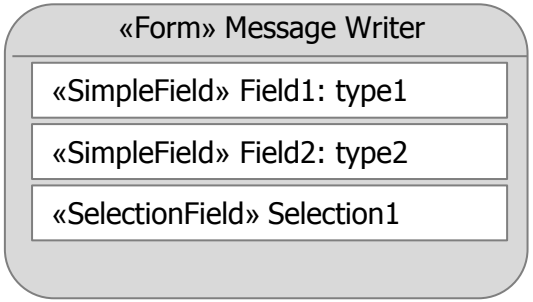


Actions

IFML – adding details to ViewComponents







ViewComponentParts:

- Data binding
- Parameters

Types of ViewComponents (<<List>>)

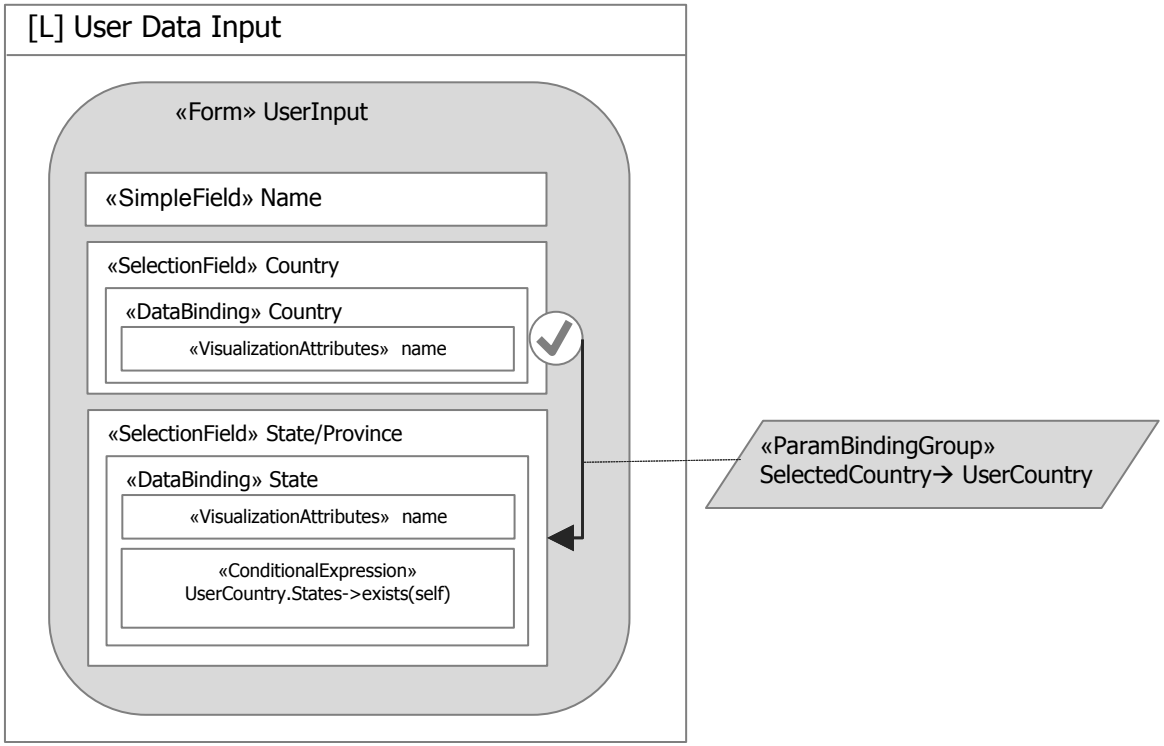
Data binding

- Joint use of IFML and other modeling languages:
  - DataBinding to classes and attributes of UML Class Diagrams
  - Upcoming: also with other content models, such as: Entity-Relationship, Ontologies, …

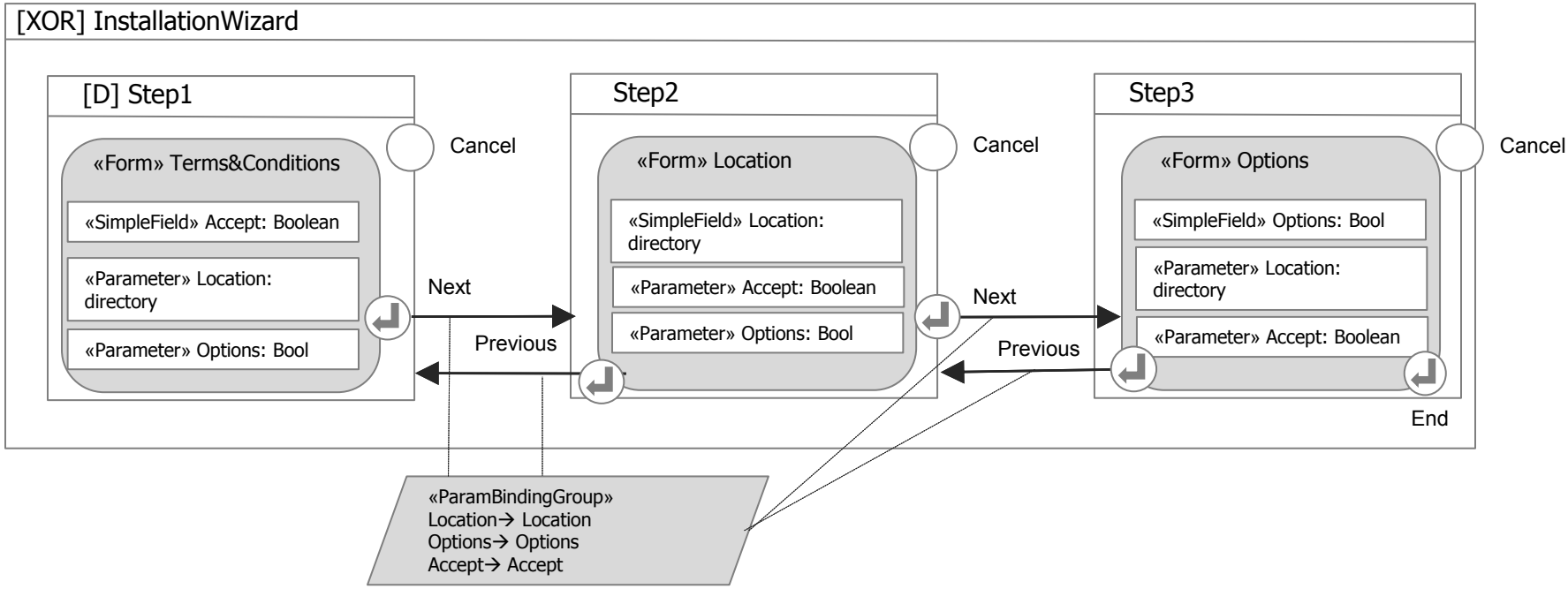
Dynamic Behaviour

- Joint use of IFML and other modeling languages
- Connection of Actions to back-end business logic as
  - UML methods of classes
  - whole UML dynamic diagrams
    - activity diagram, sequence diagram, state chart diagram, …

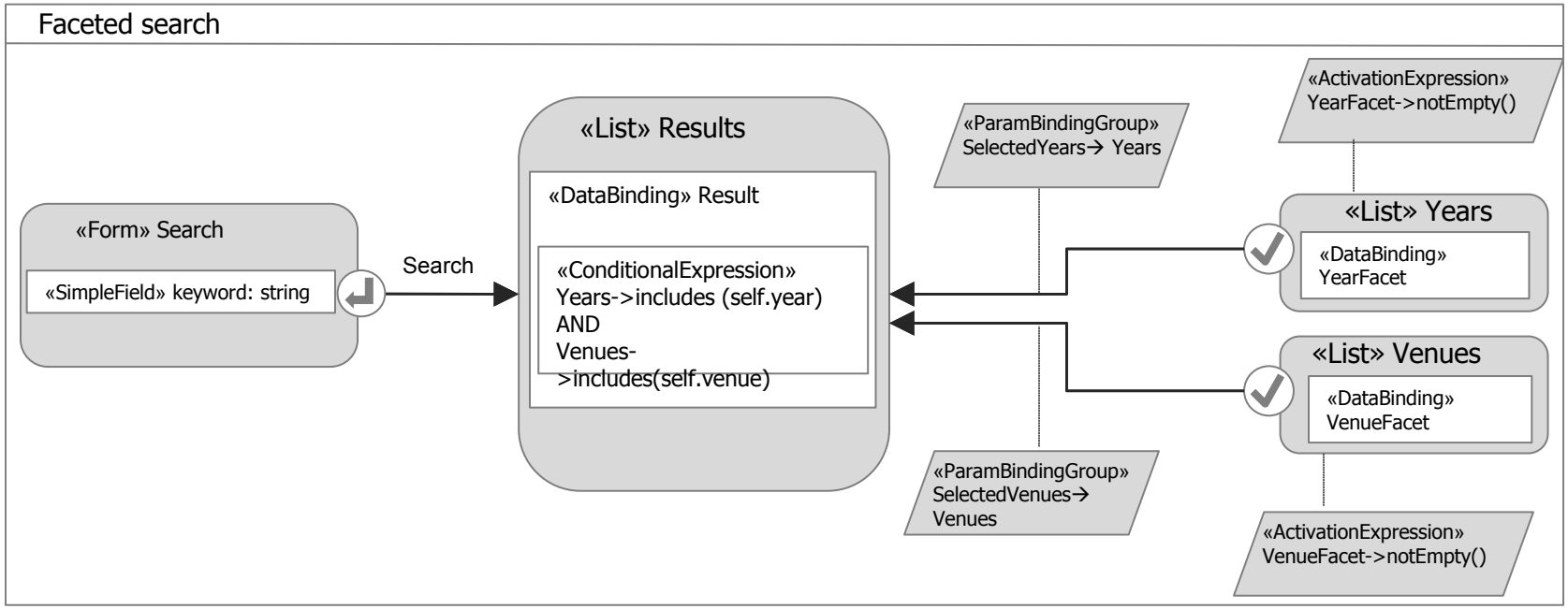
Dynamic Form Behavior



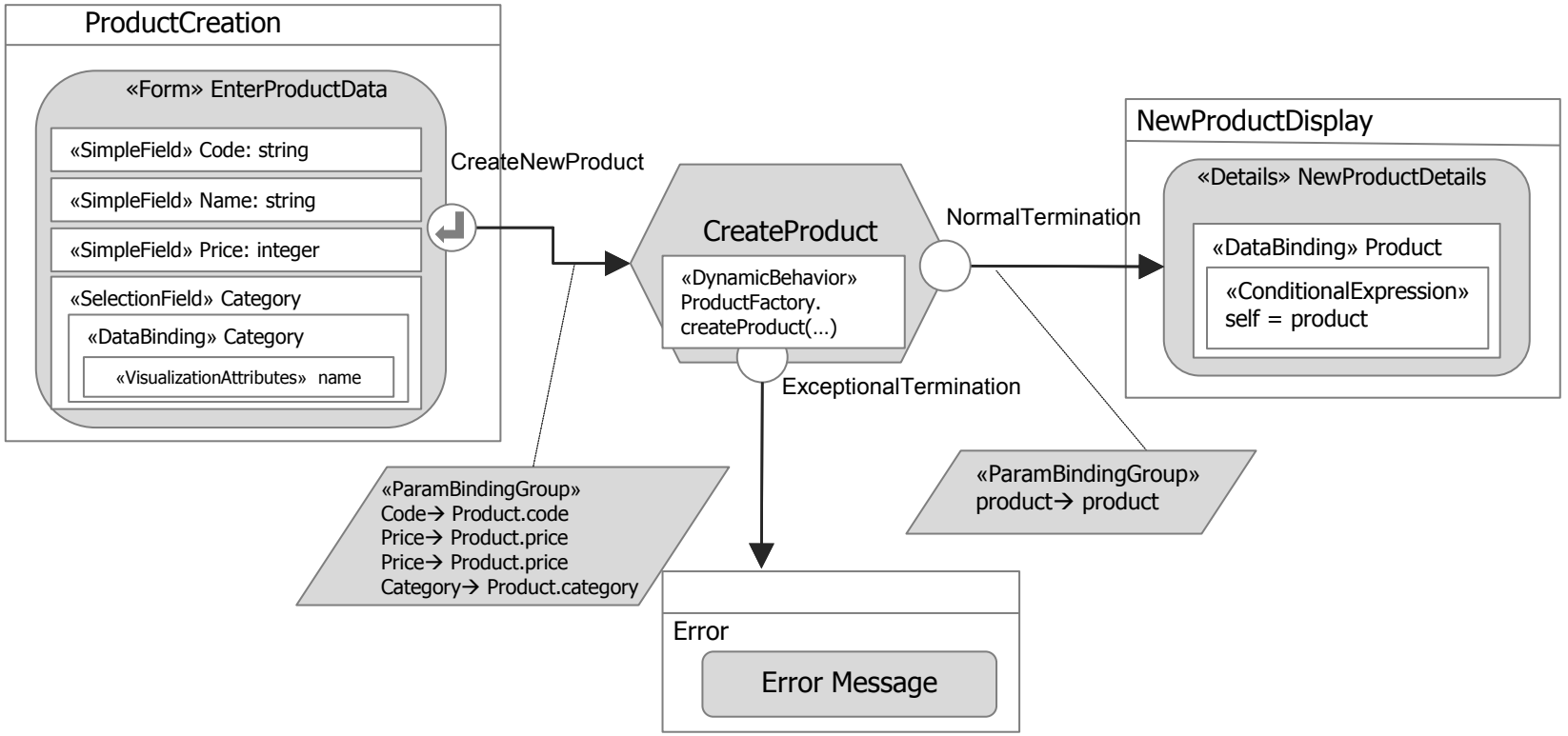
Example: Wizard



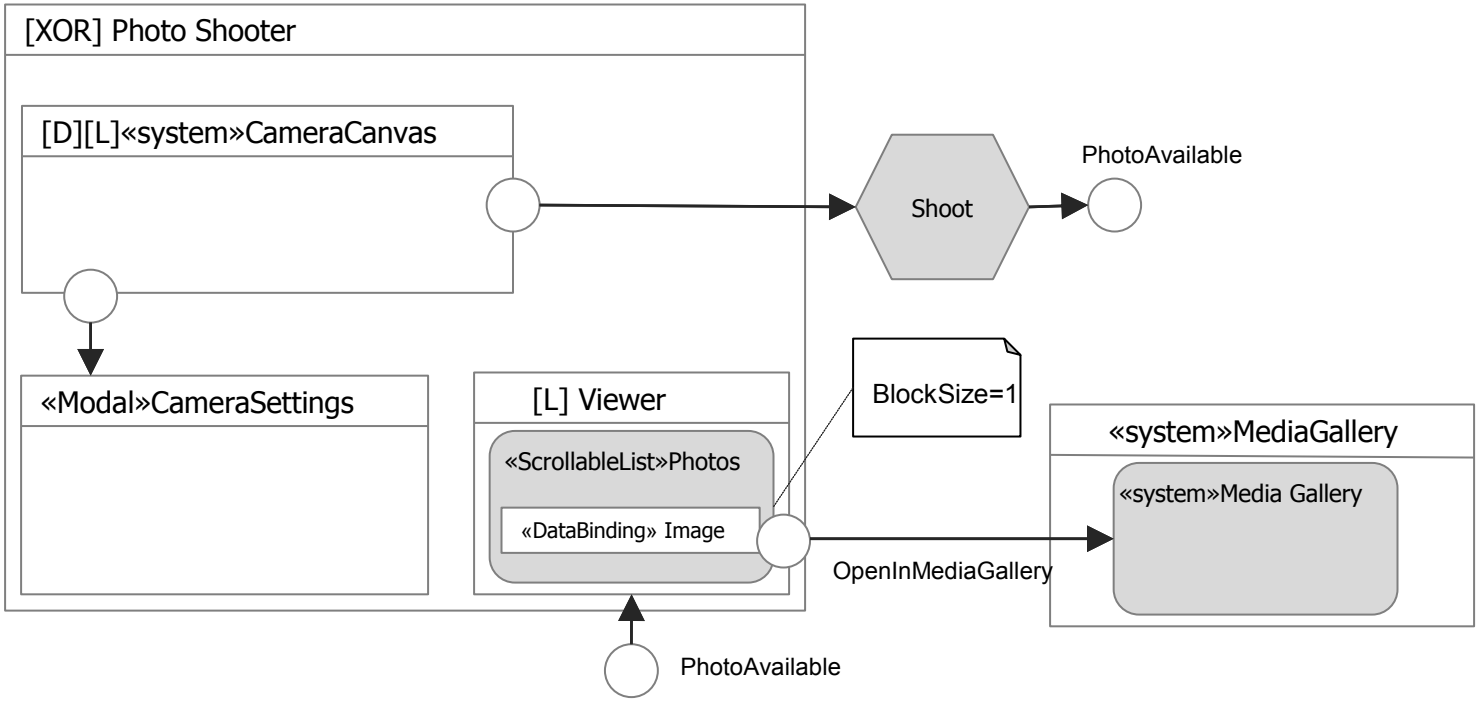
Example: Faceted Search



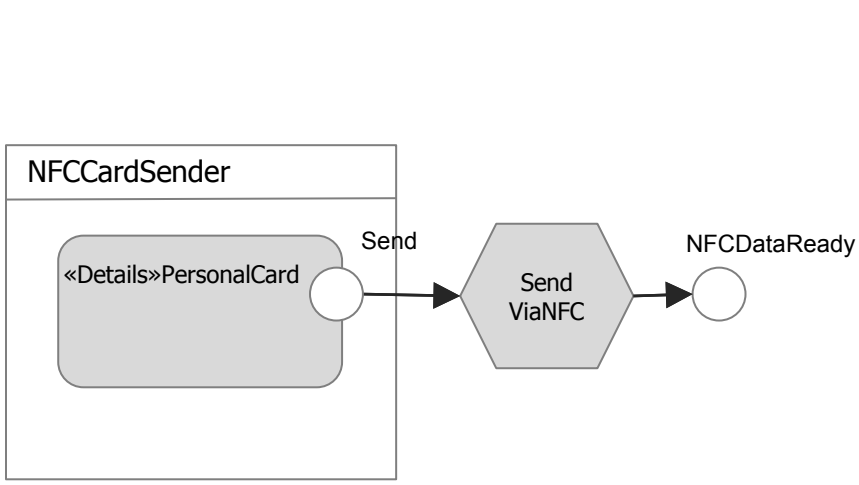
Example: Details on Actions

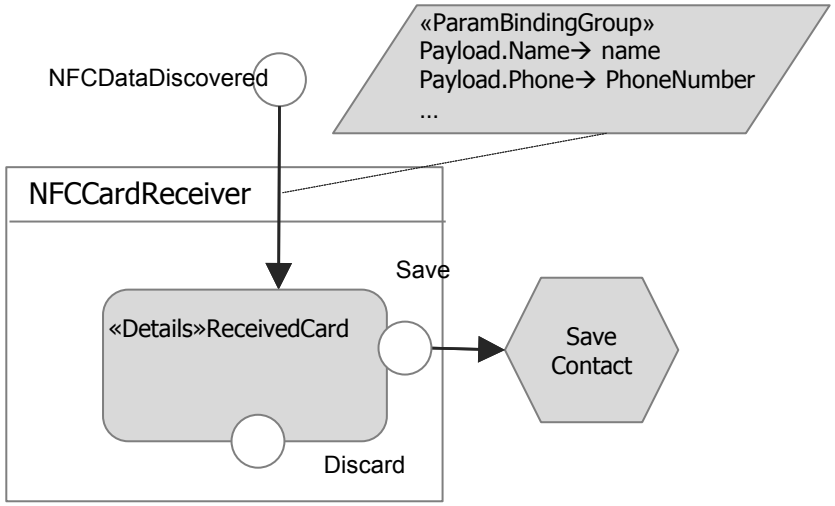


Example: Mobile Device, Camera Controls



Example: NFC Controls

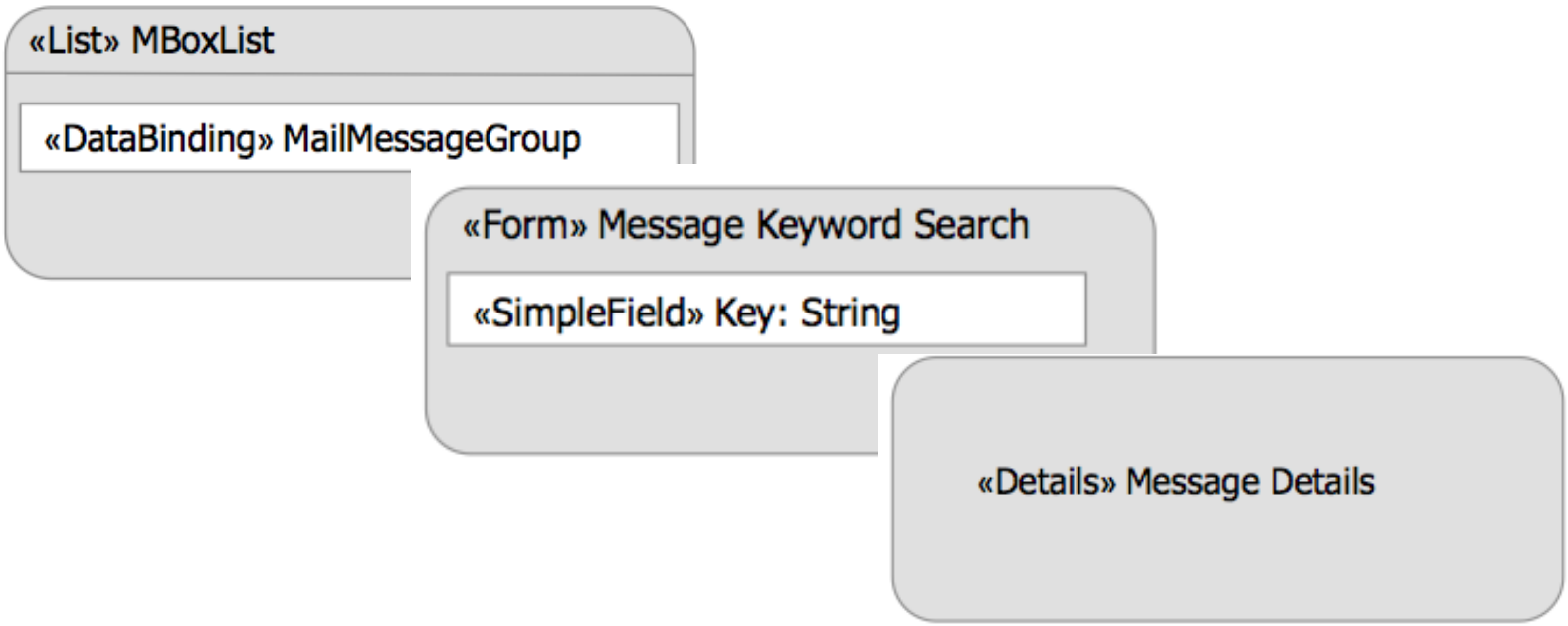




IFML – subtyping components and events

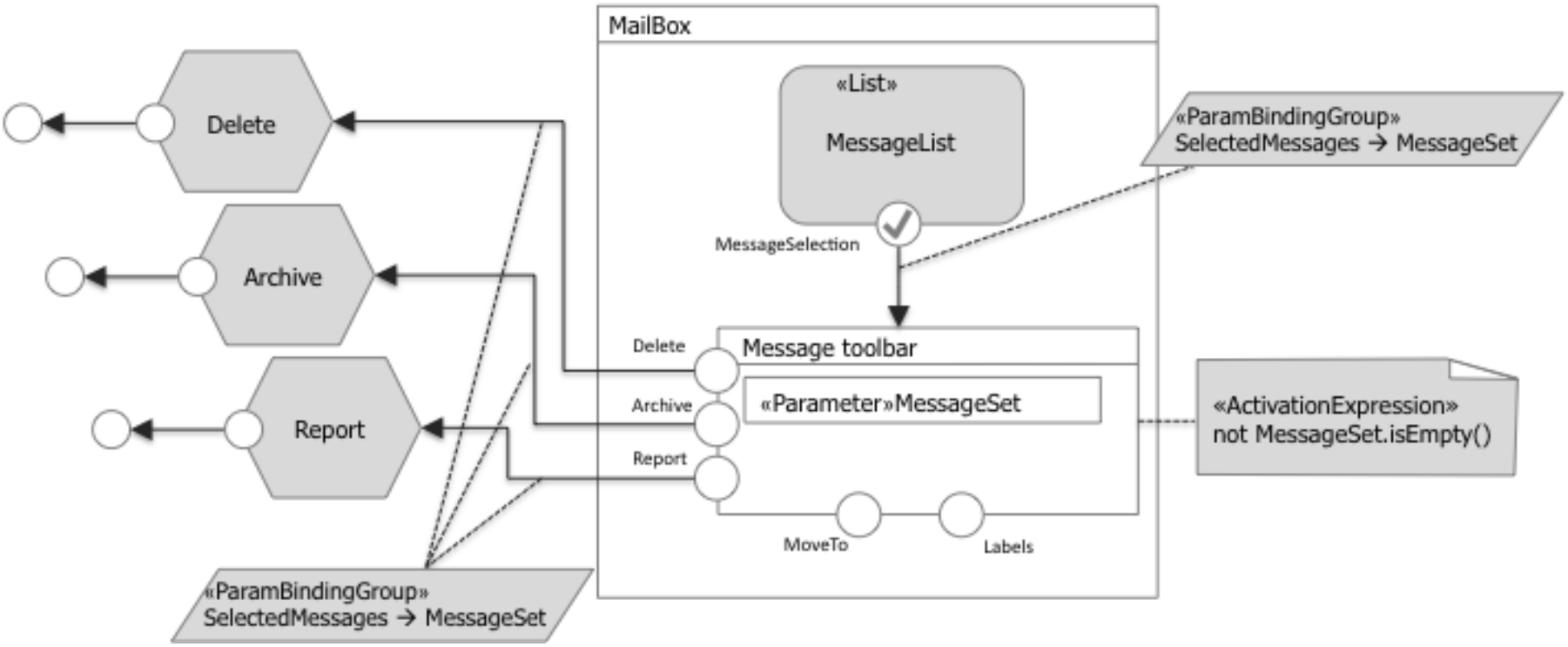
Selection event

Submit event



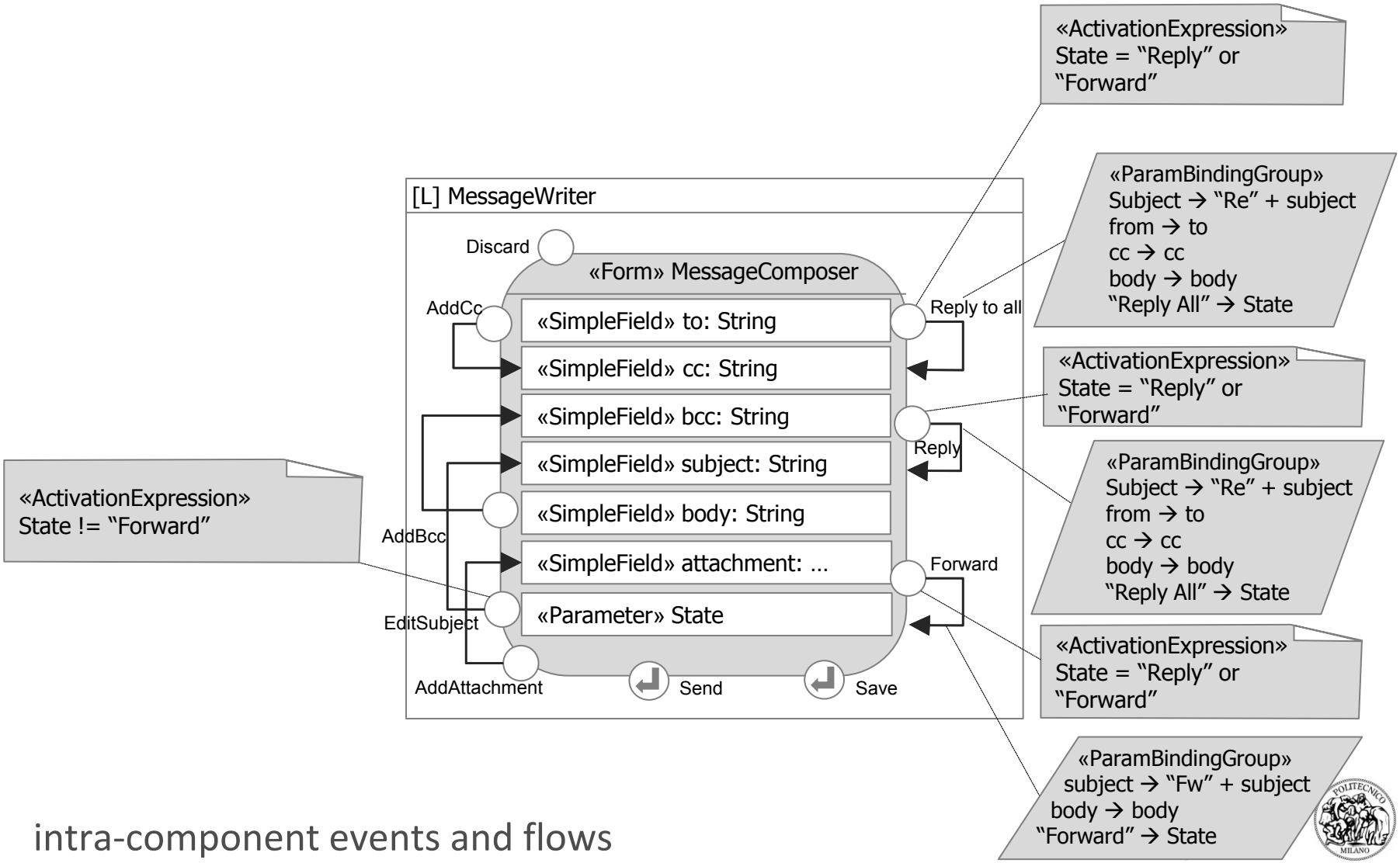
.. And as many others as you want!

IFML by example



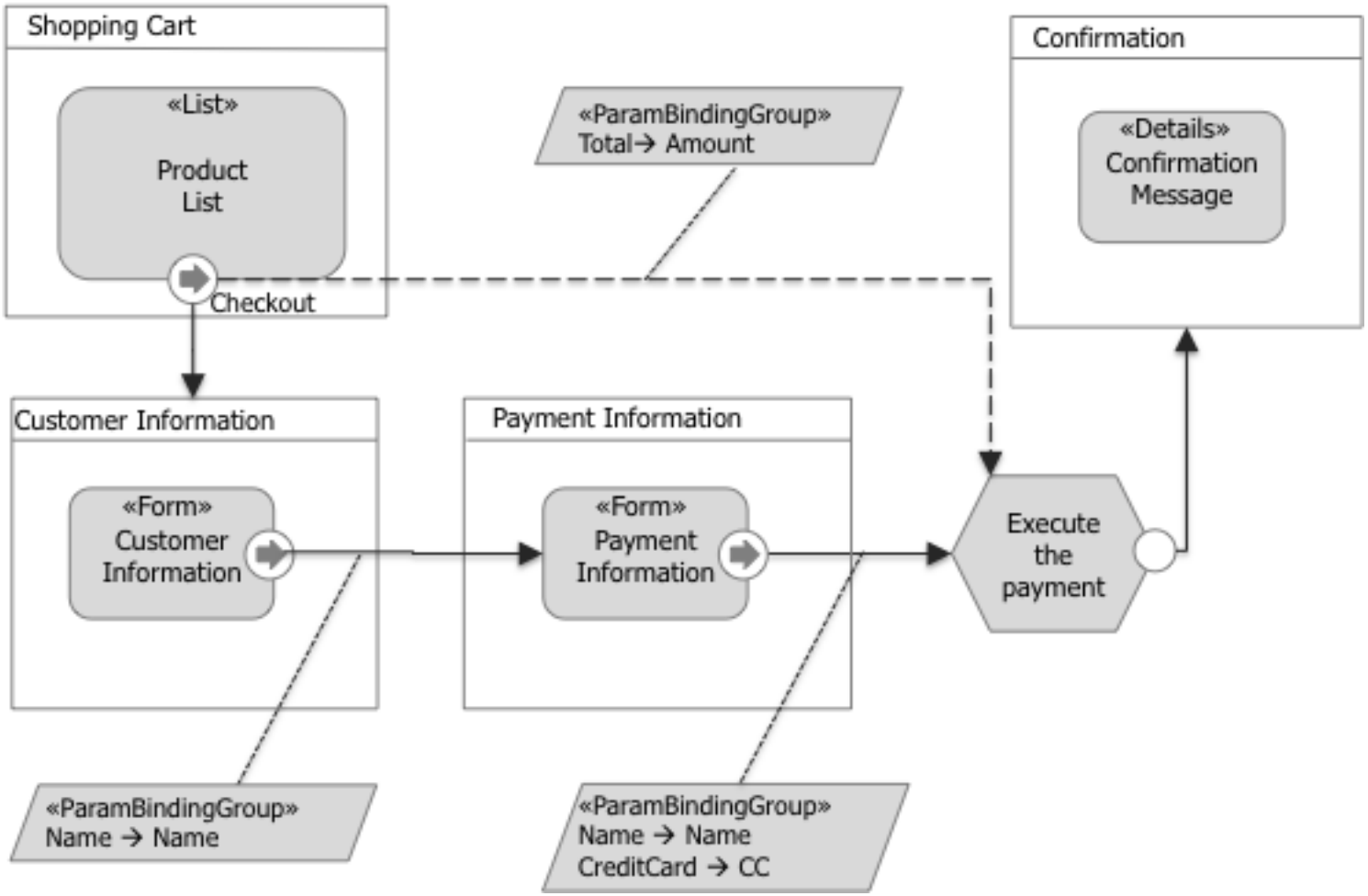
ActivationExpression, SubmitEvent, Event generation

IFML concrete syntax by example



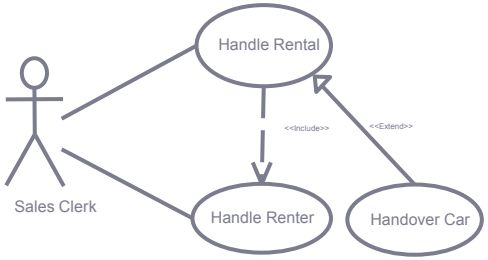
intra-component events and flows

IFML example – online payment

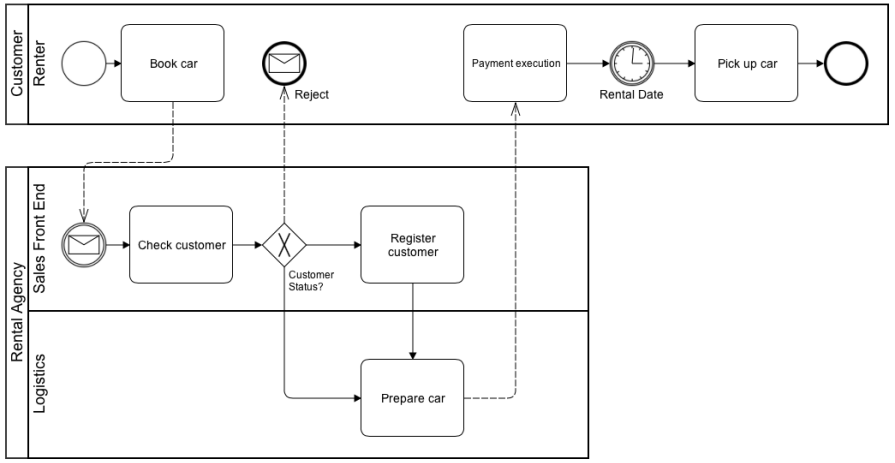


Multiple aspects modeling – 1
(business and requirements)

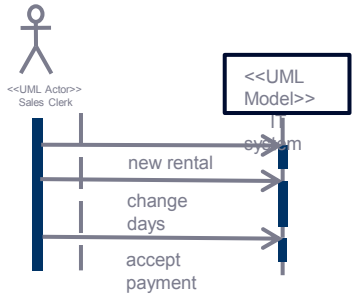
UML Use Case



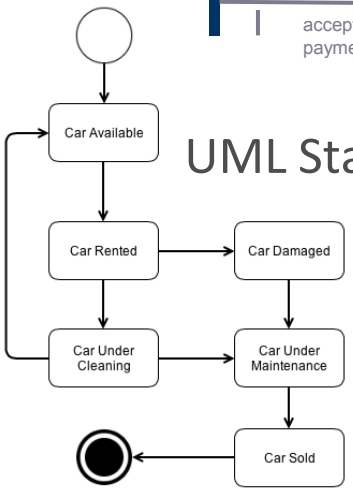
BPMN process



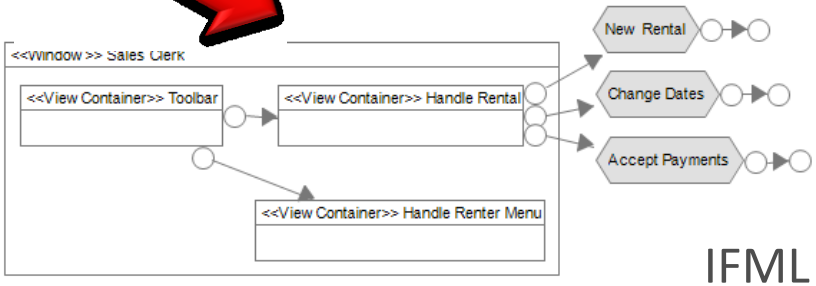
UML Sequence



UML Statechart



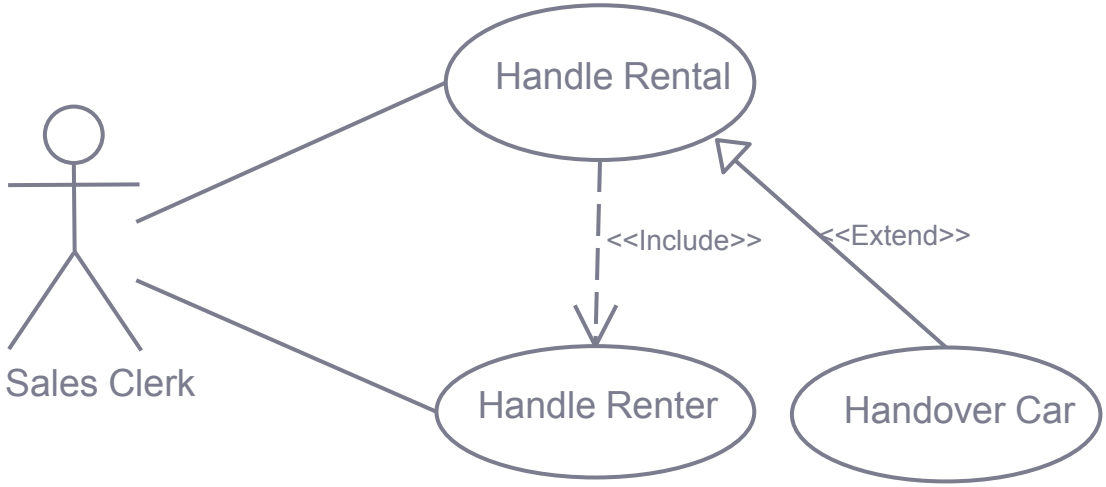
IFML



WEB RATIO

POLITECNICO MILANO

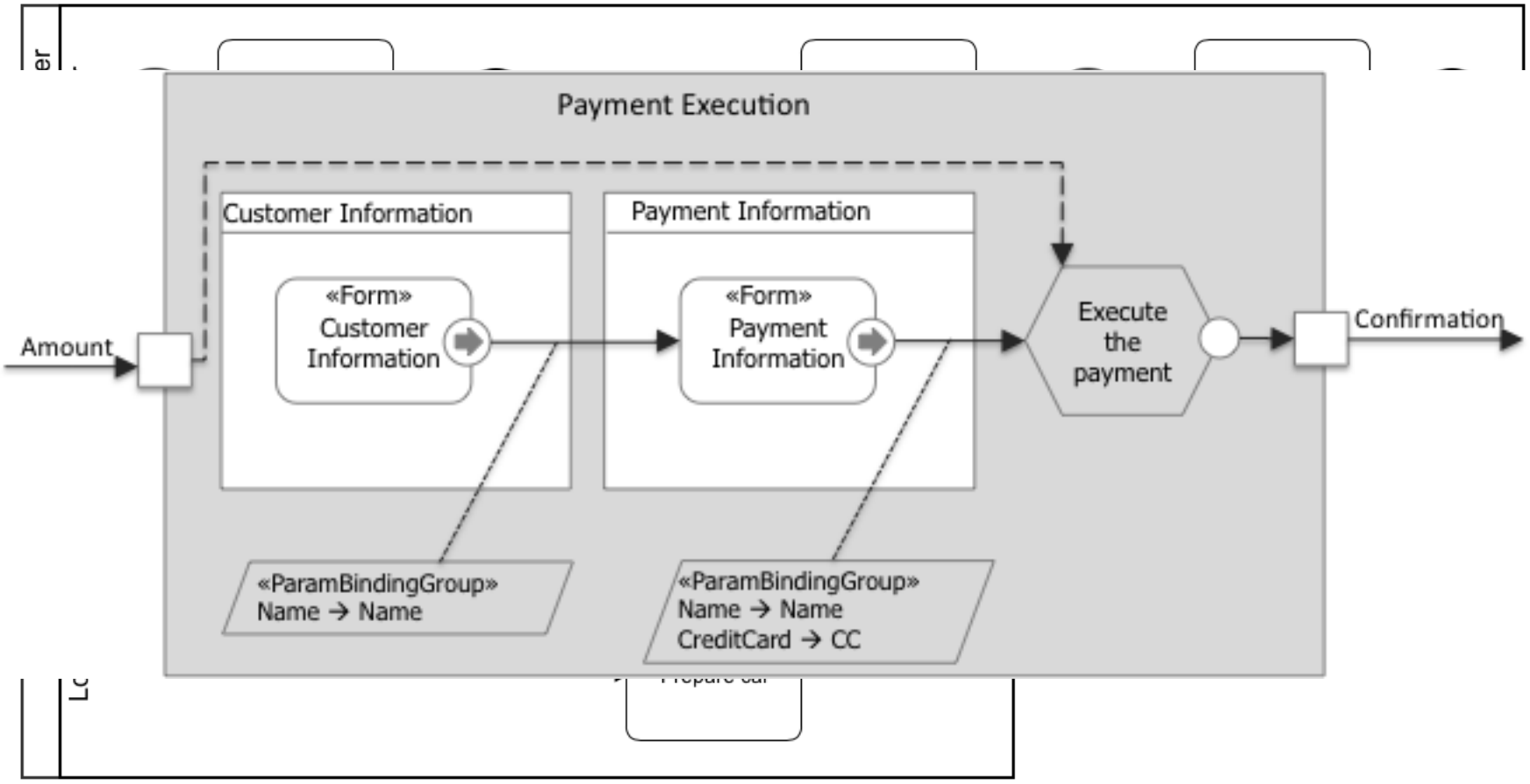
Integration with UML Use Cases



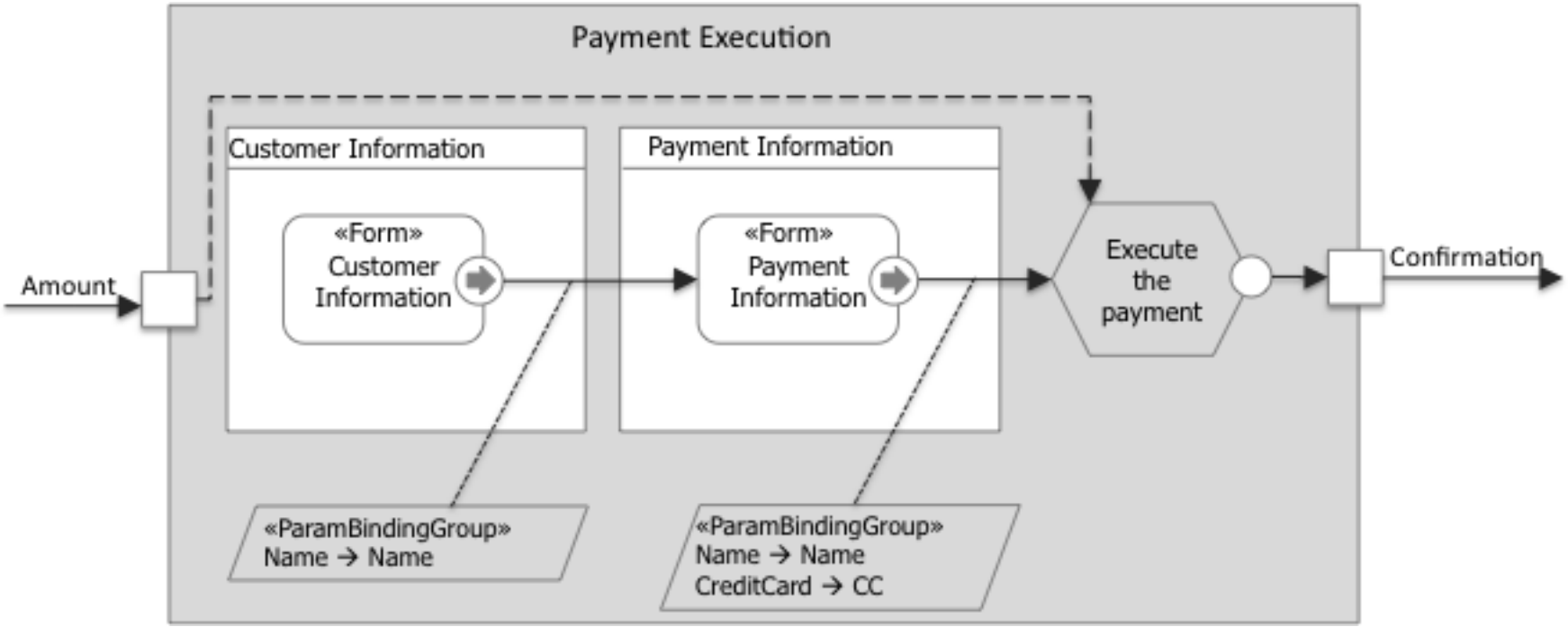
Each use case can be described by

- A business process
- A plain UI description in IFML
- Some UML dynamic diagrams (e.g., activity, sequence, …)

Integration with BPMN

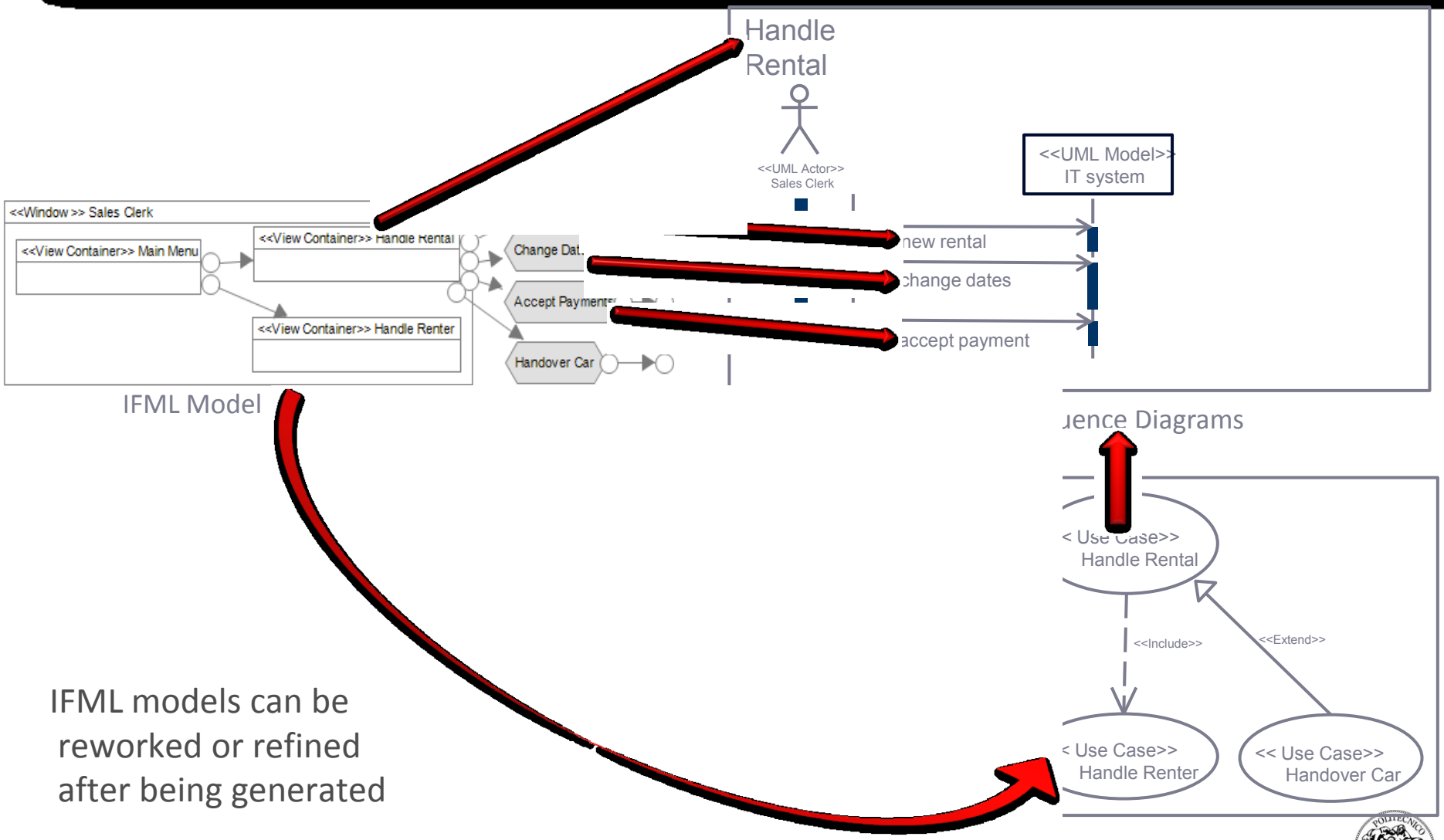


IFML concrete syntax by example



IFML Modules - definition

Example of UML - IFML mapping

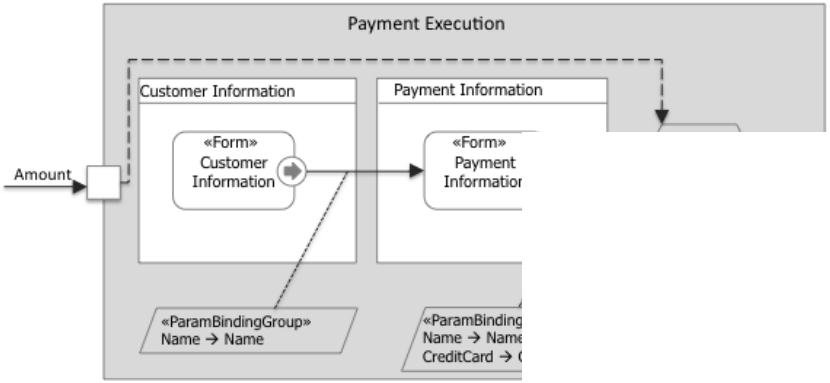


IFML models can be reworked or refined after being generated

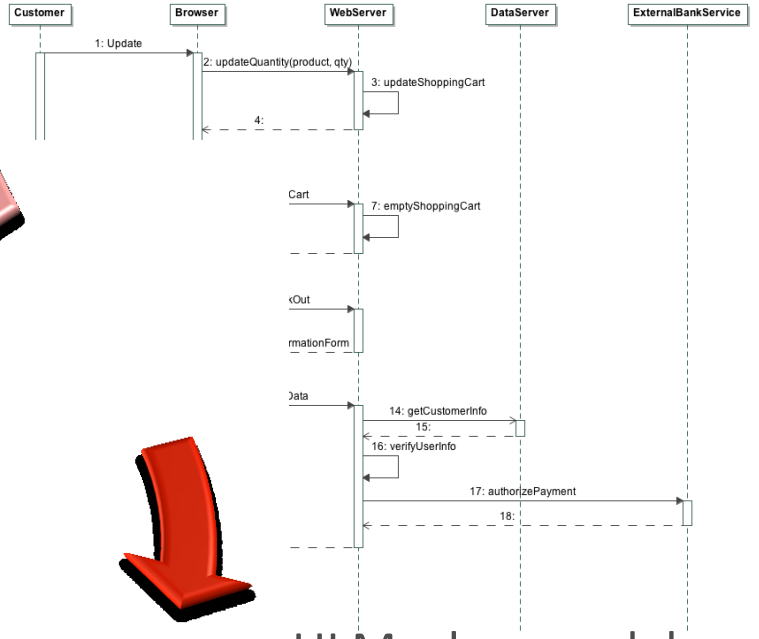
WEB RATIO

Multiple aspects modeling – 2
(implementation and architecture)

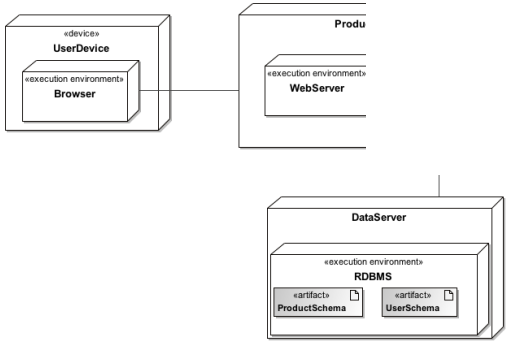
IFML



UML Sequence



UML Deployment



UI Mockup models

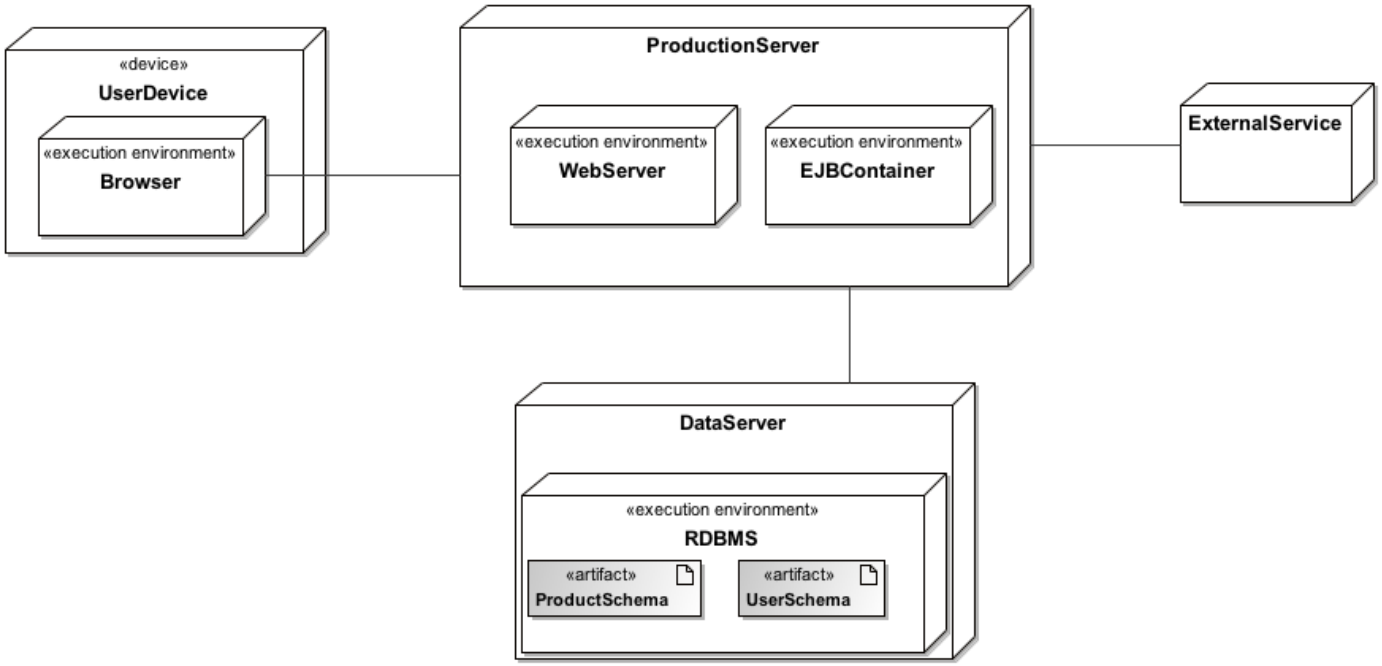
WEB RATIO

POLITECNICO MILANO

Integration with UML

Description of deployment architecture

- UI is just one facet of system design
- Often need to position it in a broader architectural vision

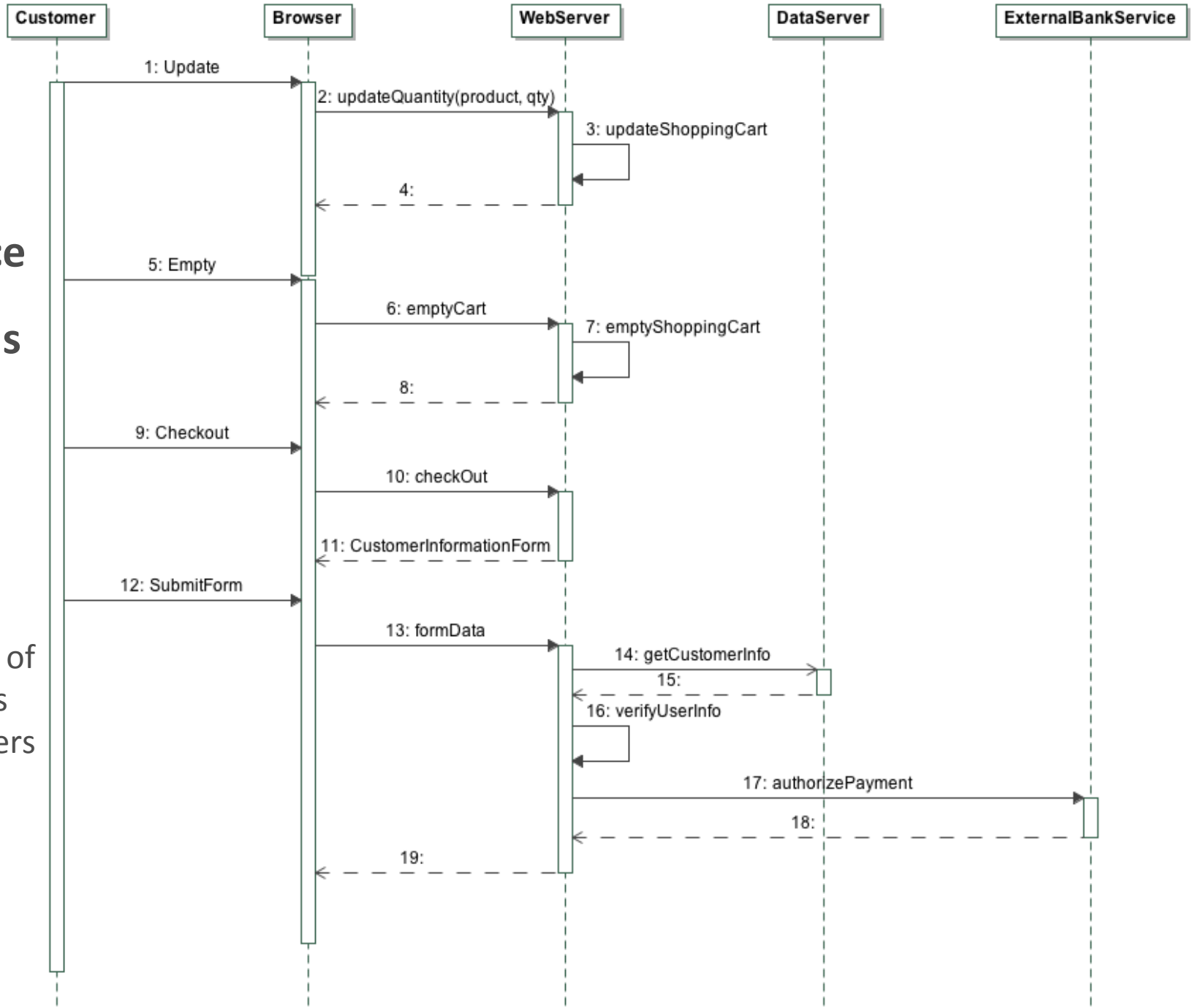


- UML deployment diagram

UML Sequence Diagrams

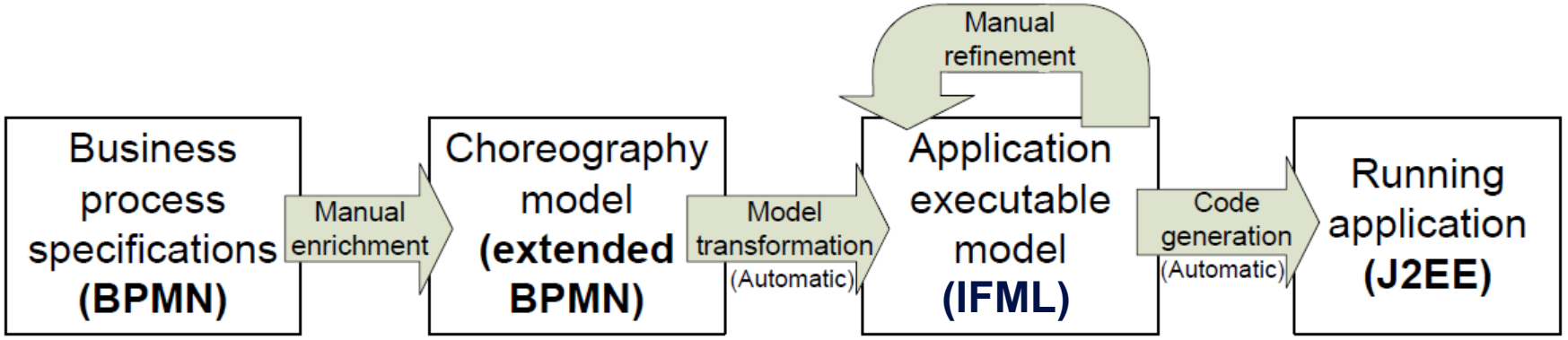
Tiers and calls

Explicit description of interactions between tiers

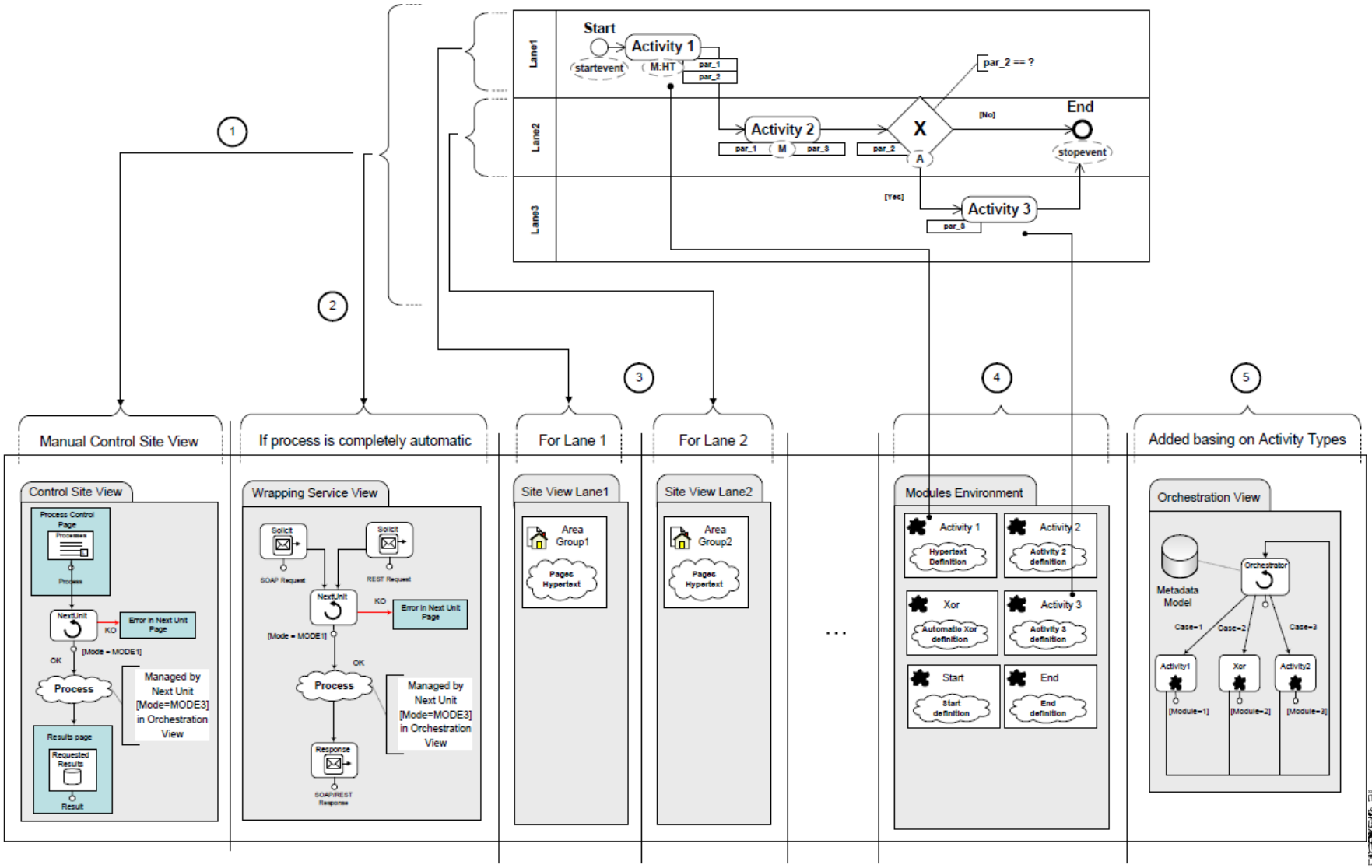


Model-driven Development Process

- Manual specification of BPMN process model
- Automatic transformation of BPMN to IFML
- Possible manual refinement of IFML models
- Automatic running code generation on J2EE platform
- Virtuous development cycle



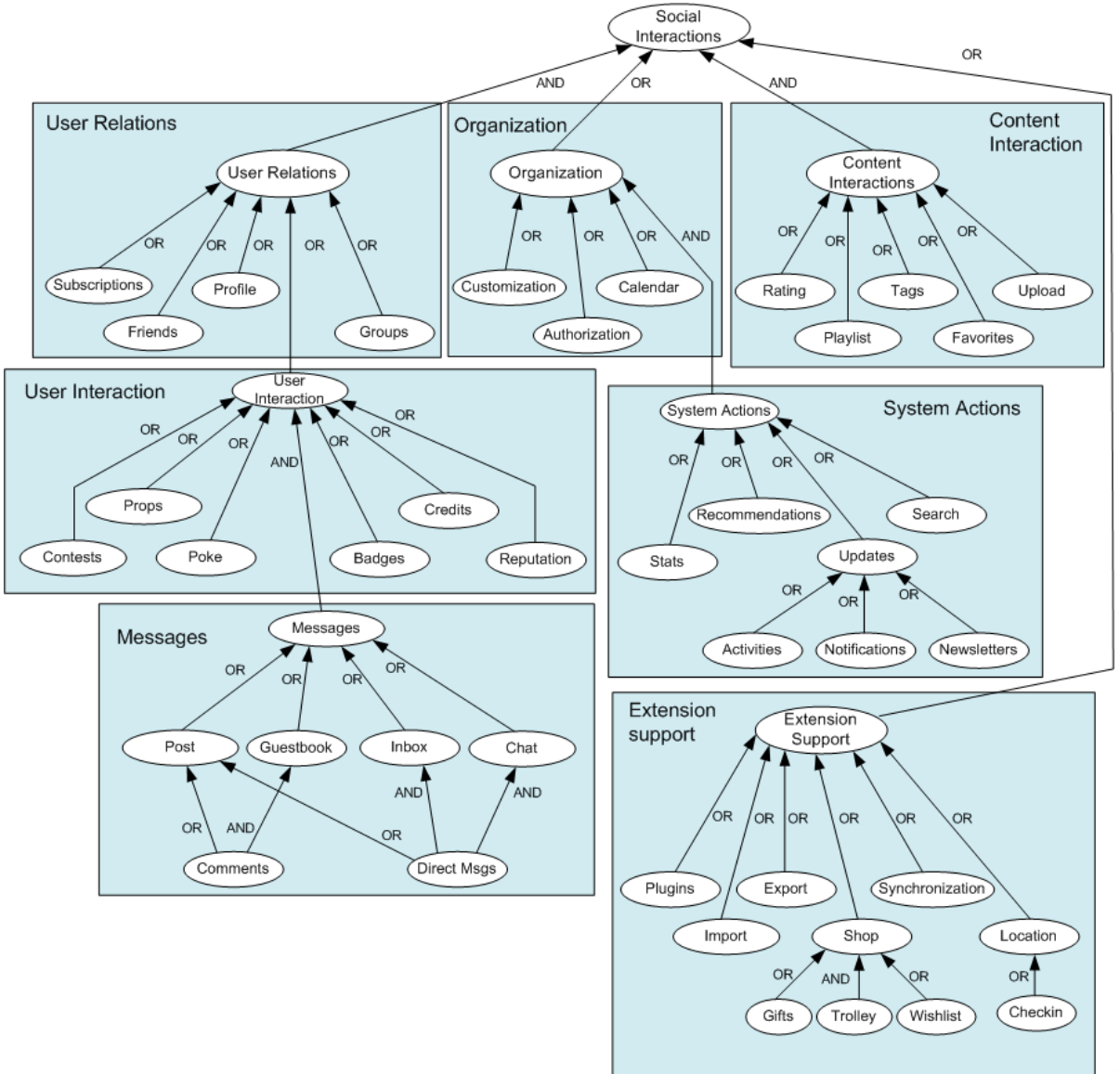
The generated model artifacts



Example: from social networking goals..

Goal taxonomy

Interleaving with enterprise values



.. to design patterns

As in the tradition of BPM design patterns, they capture reusable solutions to recurrent socialization requirements:

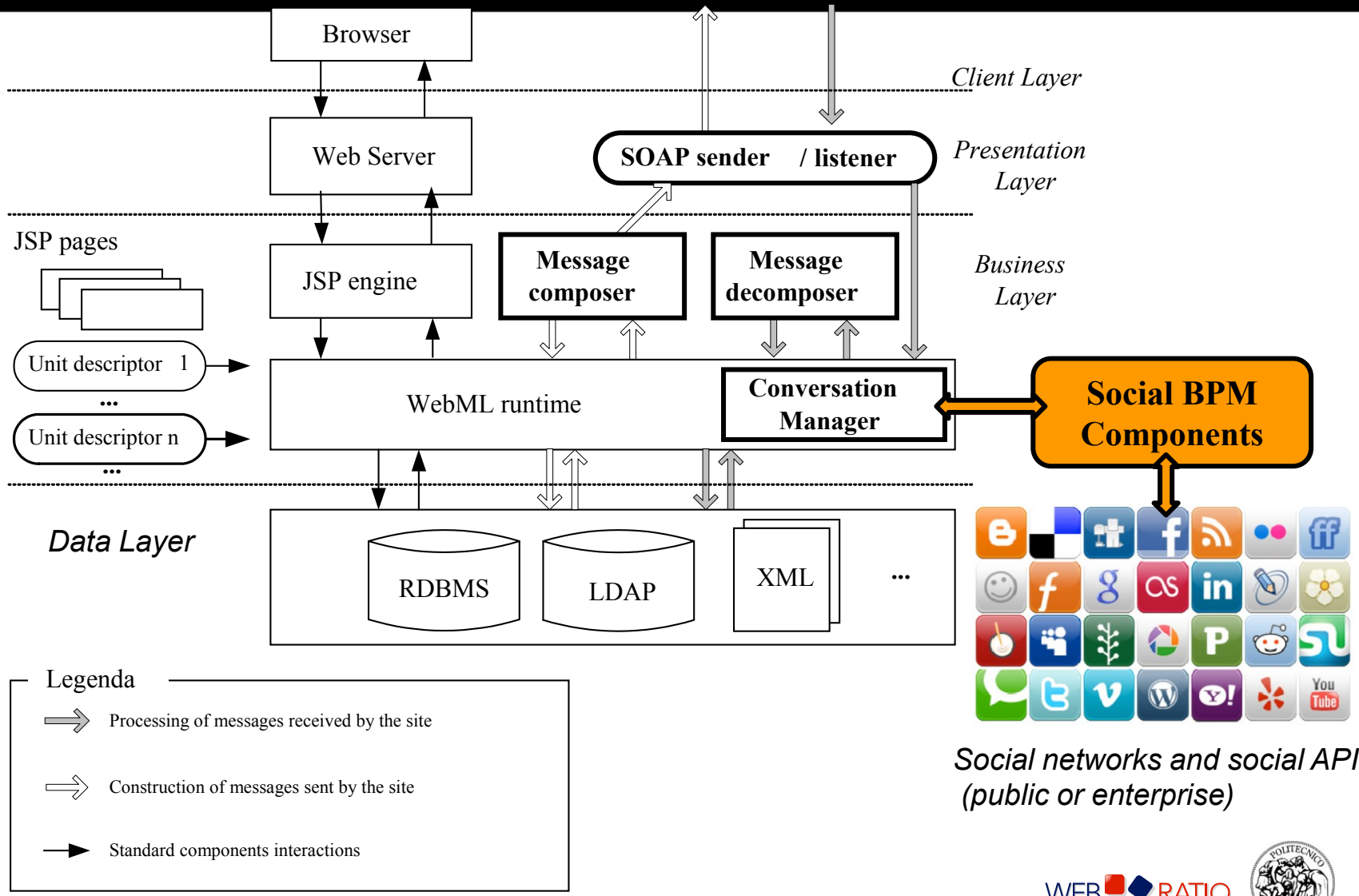
- Dynamic enrollment
- Poll
- People / Skill search
- Social content publication
- Social sourcing (vs. crowdsourcing)
- Progress notification
- Ranking and commenting

… and business objectives

Socialization goals can be used as drivers for the selection of the social BPM design patterns that are more relevant to a process socialization effort

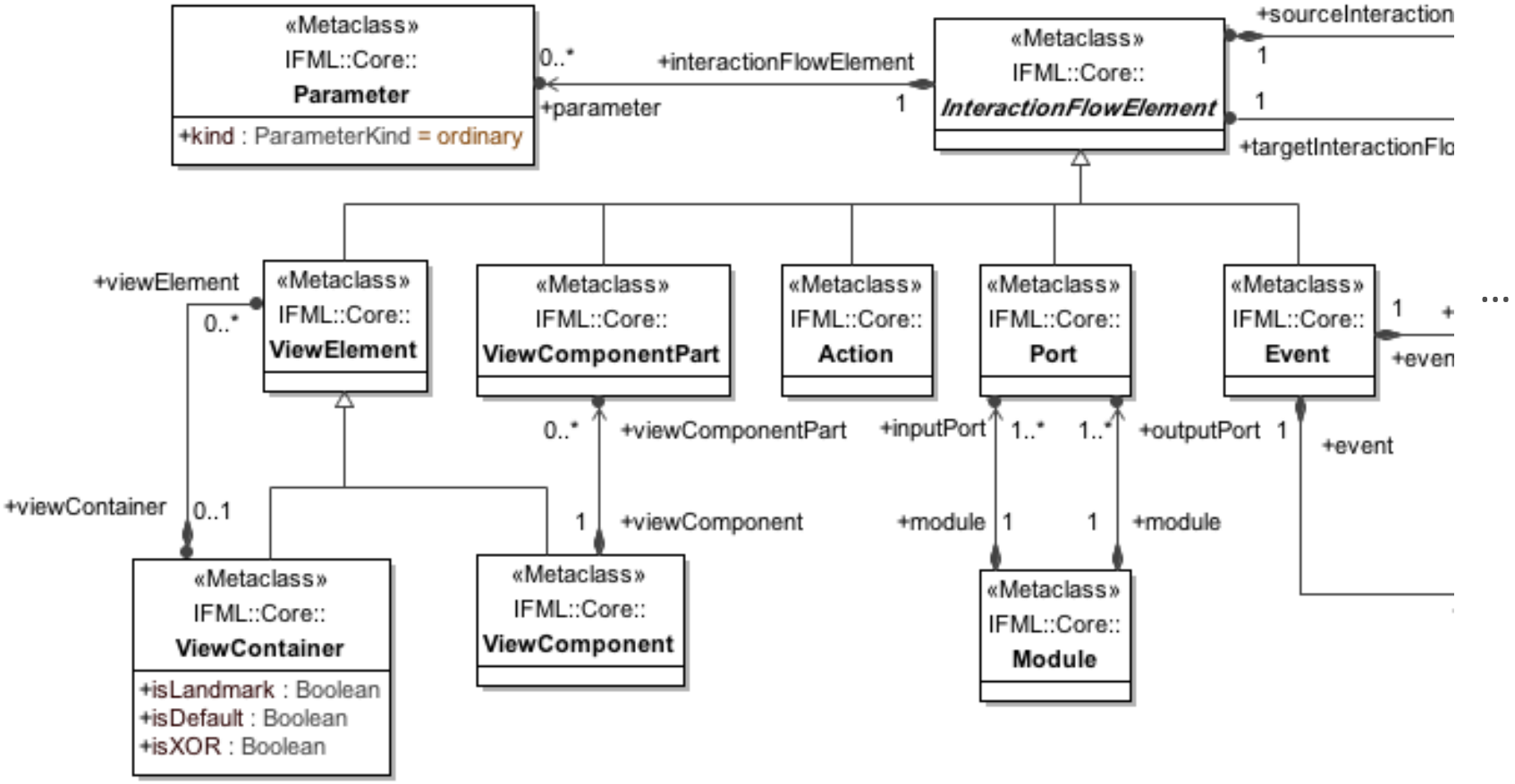
	Weak Ties / Tacit Knowledge	Transparency	Participation	Activity distribution	Decision distribution	Social f.back	Knowledge sharing
Dynamic enrollment			X				
Poll					X	X	
People / Skill search	X			X	X		
Social content publication		X					X
Social sourcing				X			
Progress notification		X					
Ranking and commenting	X				X	X	X

WebRatio runtime architecture and extension for Social Business Logic



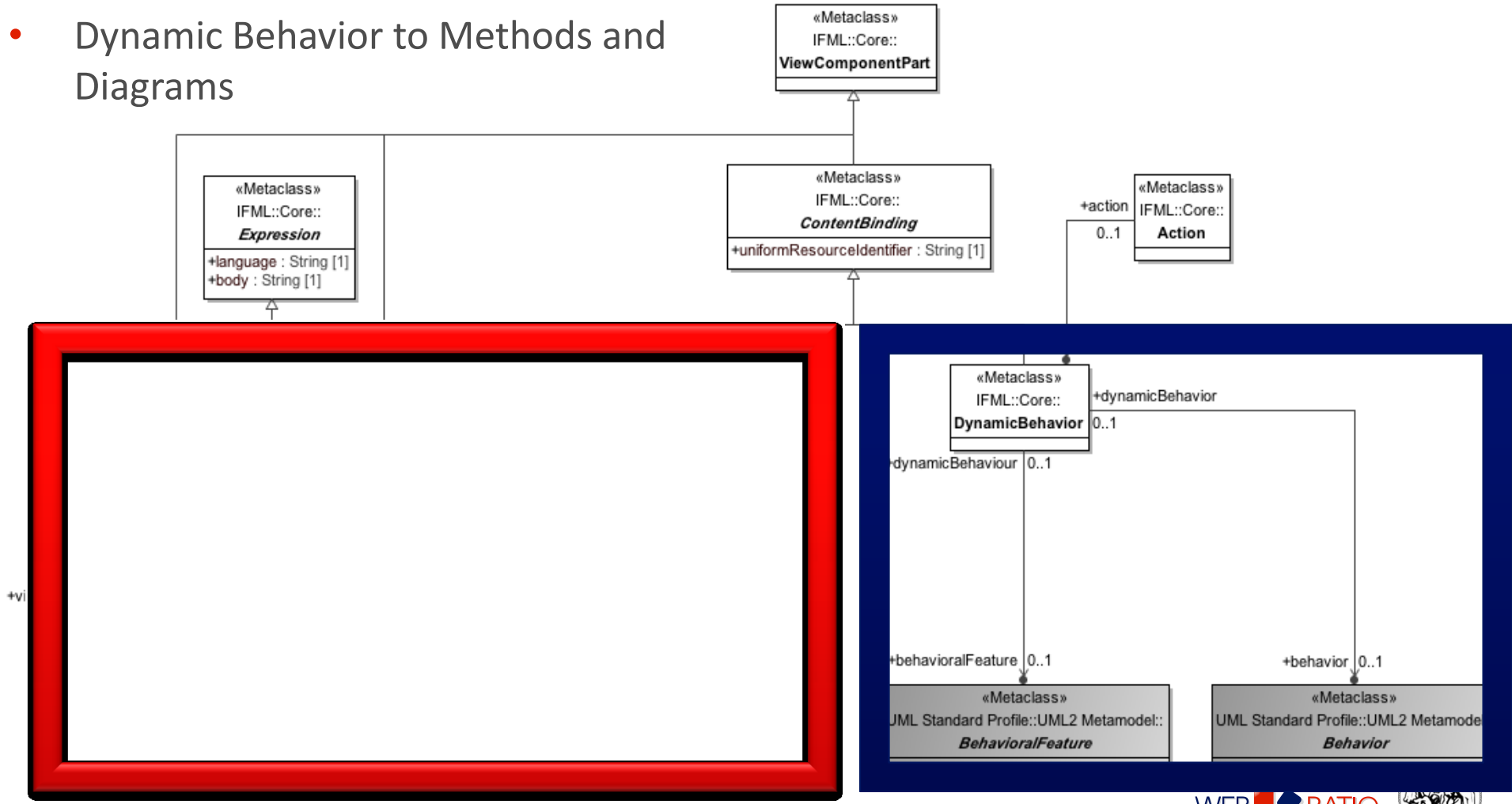
How does it work? IFML metamodel (1)

IFML is defined through a metamodel



IFML metamodel (2): Content Binding

- Data binding to Classes and Attributes
- Dynamic Behavior to Methods and Diagrams



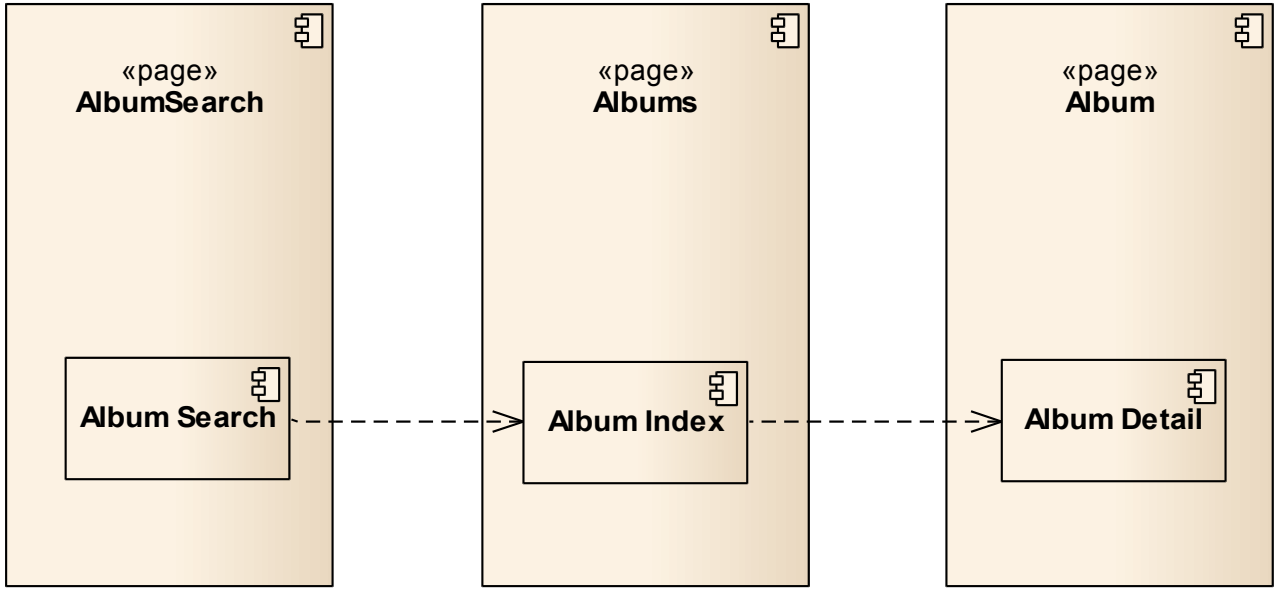
Practical results of having a standard

- An official **metamodel of the language** which describes the semantics of and relations between the modeling constructs
- A **graphical concrete syntax for the interaction flow notation** which provides an intuitive representation of the user interface composition, interaction and control logic for the front-end designer
- A **UML Profile** consistent to the metamodel
- An **interchange format** between tools using XMI

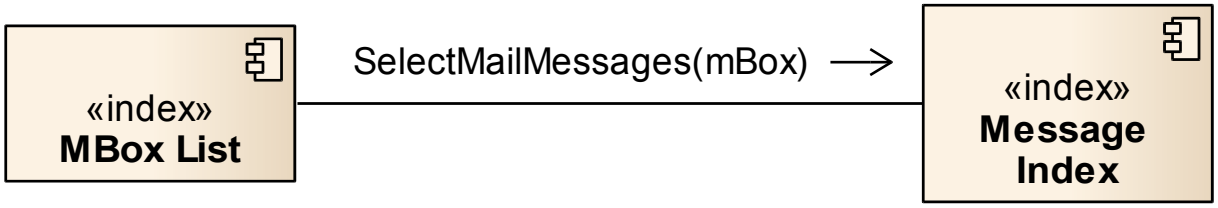
- All this, specified through standard notations themselves

Also: interchange with profile-based diagrams. The UML Profile for IFML

Static aspects

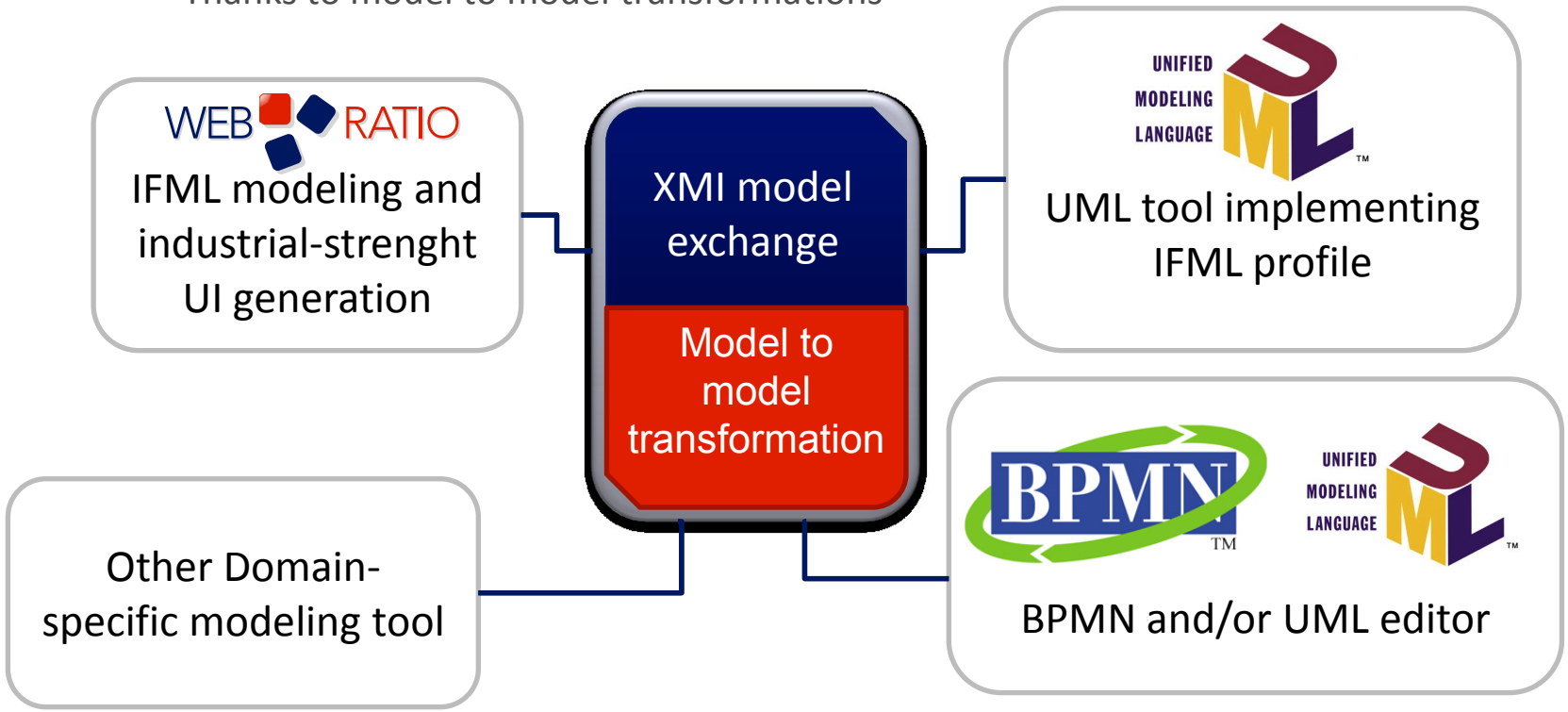


Dynamic aspects



Model integration and interchange

- Tight and seamless integration between different modeling tools
  - Thanks to XMI interchange format, UML profiles, vendor-specific notation implementations
  - Thanks to model to model transformations



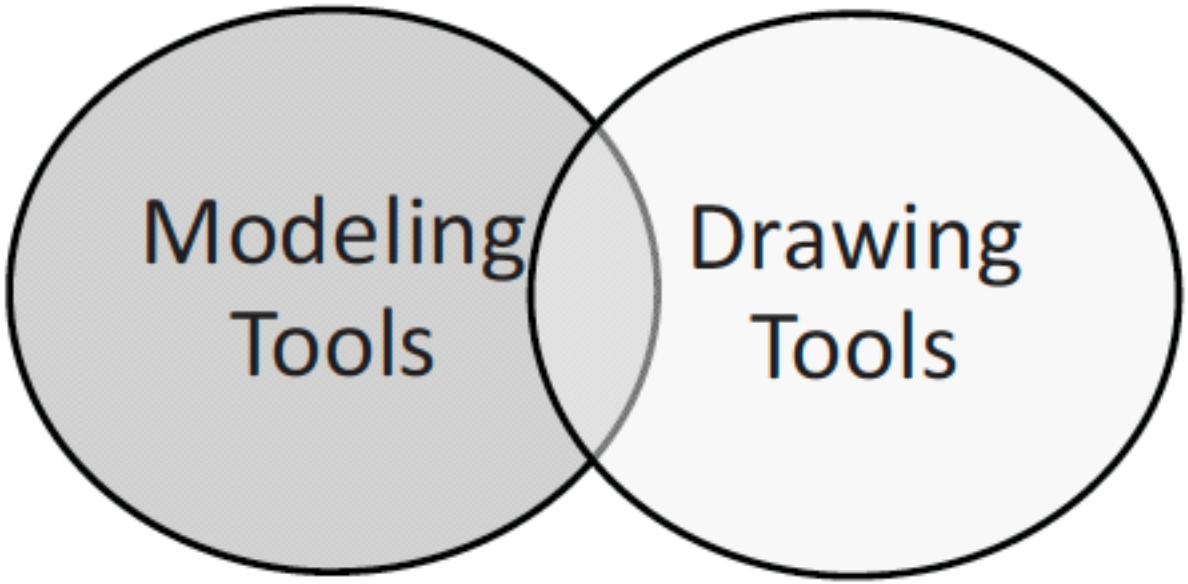
Broader, enterprise-wide system modeling

- Joint usage of IFML with other MDA languages can be devised:
  - SysML
  - SoaML
  - …
- … and also with other frameworks (e.g., Model Driven Enterprise Engineering)

The tool

Tool support for MDE/MDD

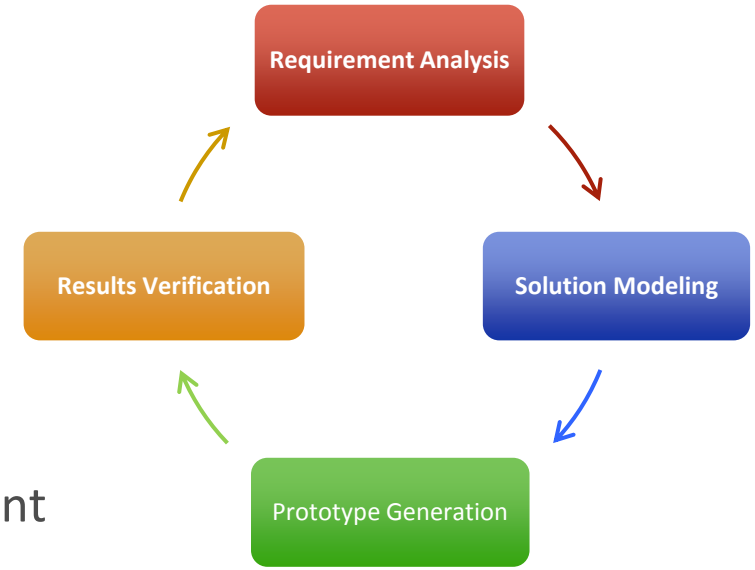
Drawing vs. modeling



What is WebRatio

An Eclipse-based development environment allowing:

- Modeling: ER + **IFML** + BPMN
- **100% code generation** of standard JEE applications
  - Clear separation between design time and run time
  - No proprietary runtime
- Quick and agile development cycles
- Extending the generation rules
  - Defining new presentation styles
  - Defining new components
- Versioning, teamwork, full lifecycle mgt
- Truly multi-role model-driven development



WEB RATIO

Some numbers

WebRatio is

- now at 7th release
- on the market since 2001

WebRatio customers

- 130+ companies and 500+ commercial users
- mainly Italy, USA, Europe and Latin America

WebRatio adoption

- 15,000+ users of the free edition
- Used in hundreds of universities all over the world

WebRatio partners

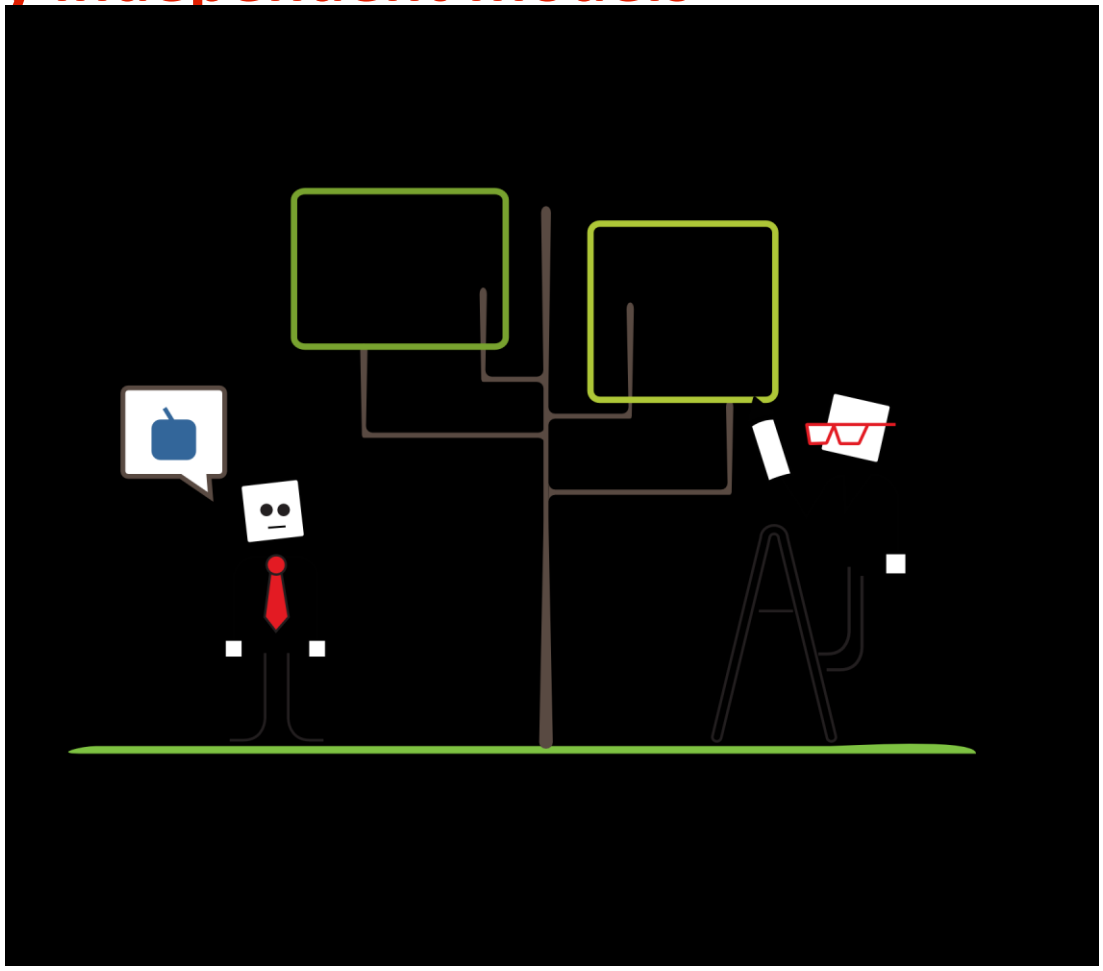
- 40+ software houses and system integrators
- 300+ universities worldwide, 13.000+ students

WEB RATIO

POLITECNICO MILANO

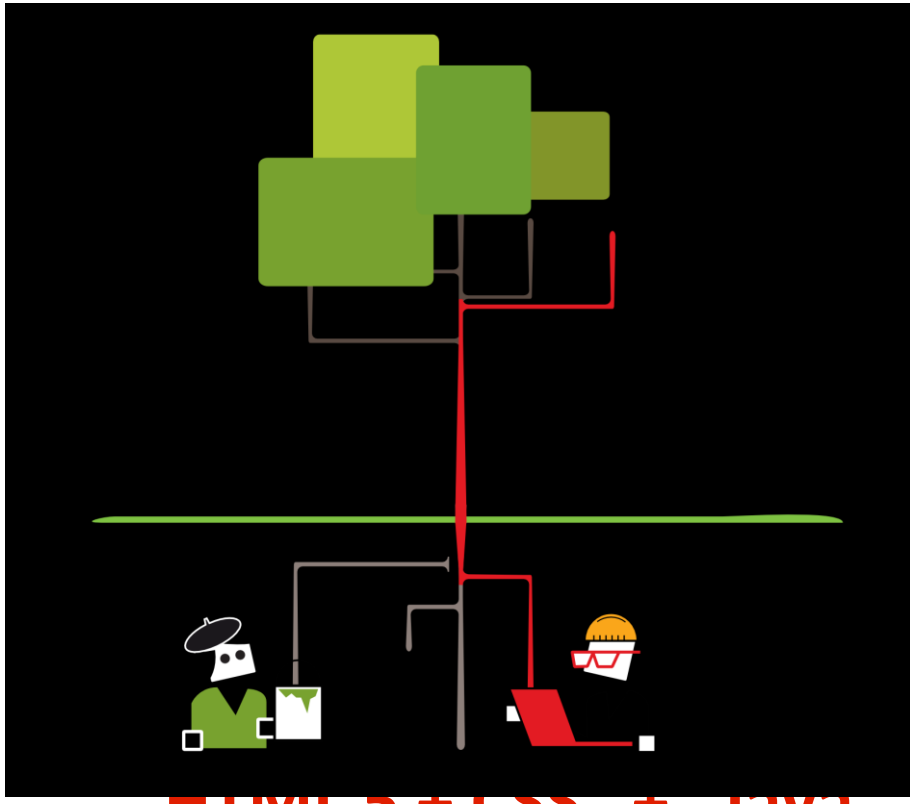
WebRatio – Step 1

You capture business requirements in abstract, technology independent models



WebRatio – Step 2

You customize the environment by defining your own generation rules



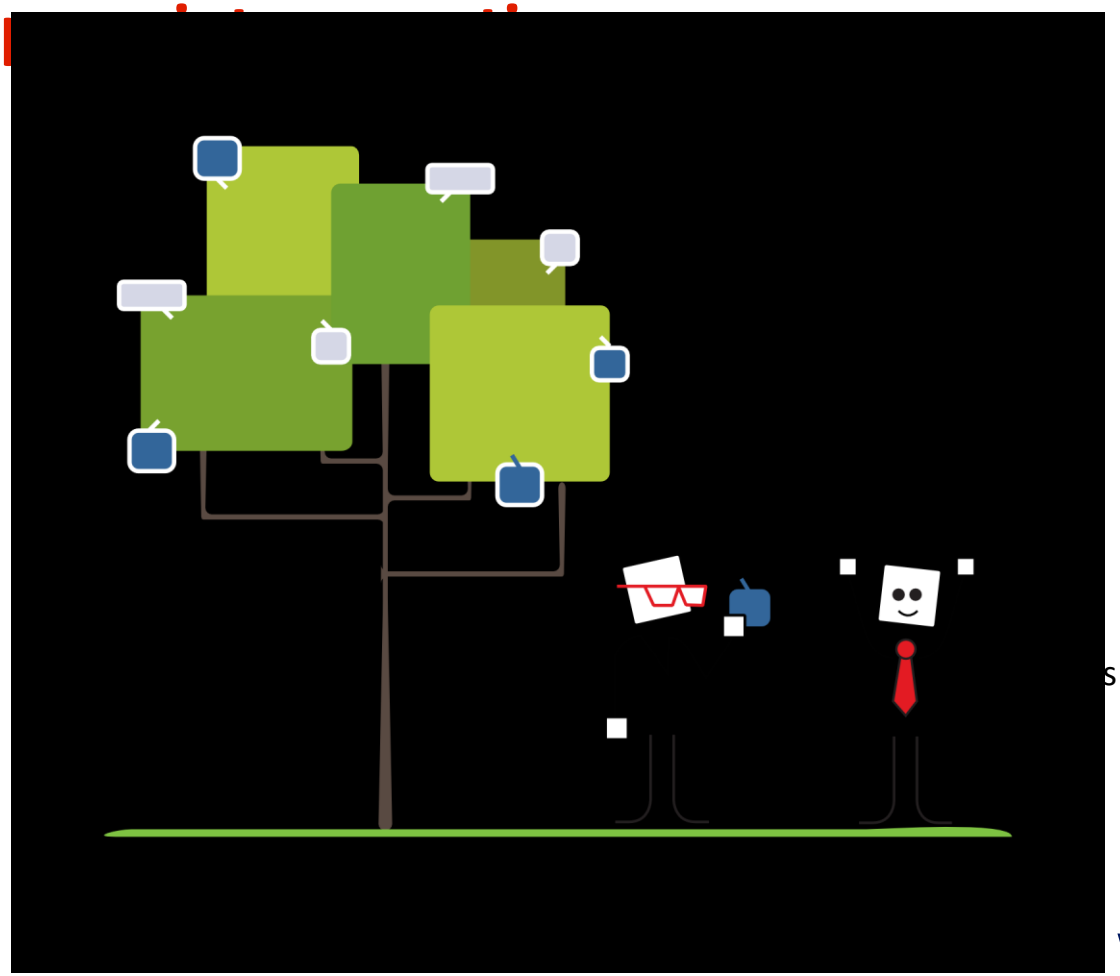
HTML 5 + CSS + Java

WEB RATIO

POLITECNICO MILANO

WebRatio – Step 3

You get a tailored, yet standard, Java Web application with no p

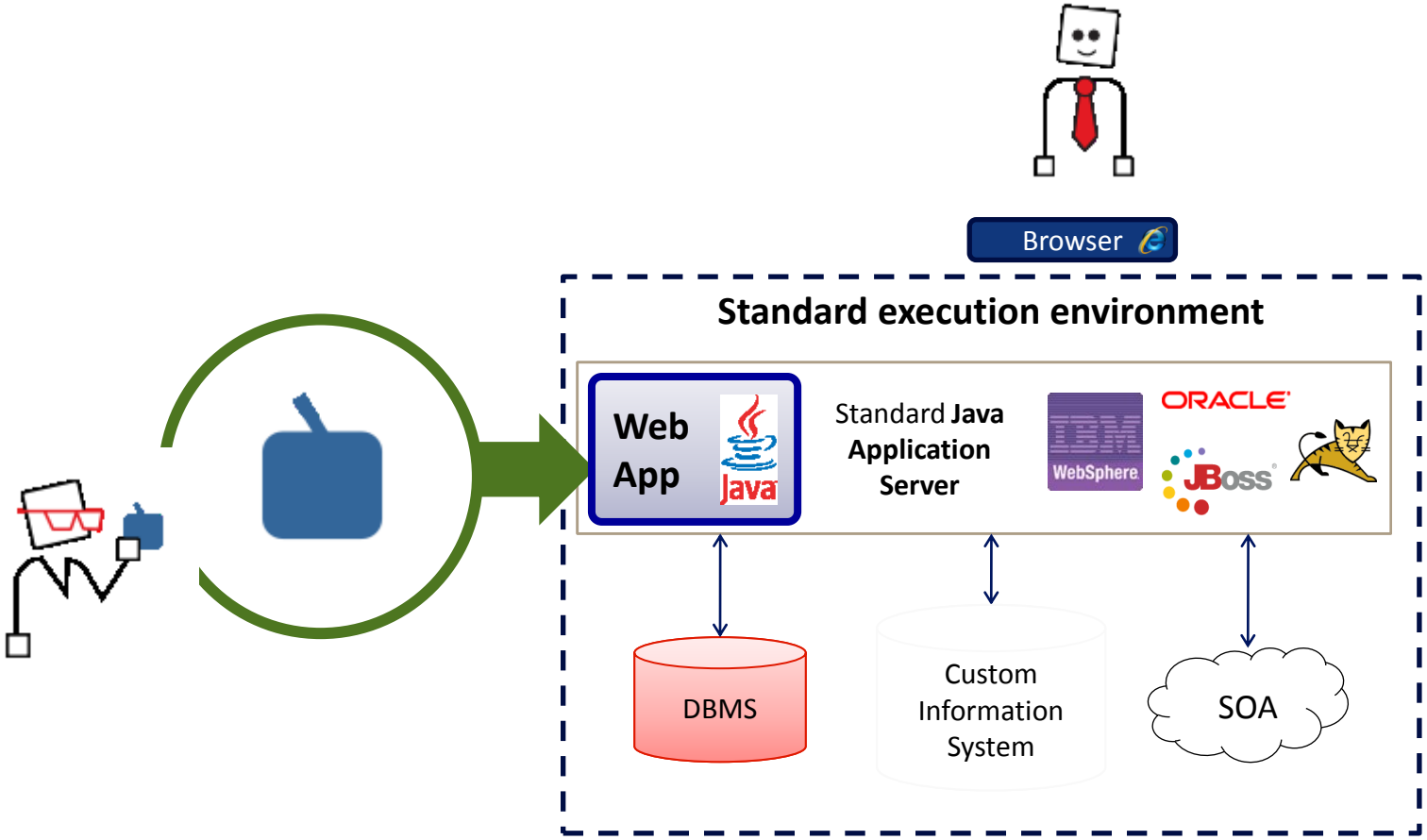


s

WEB RATIO

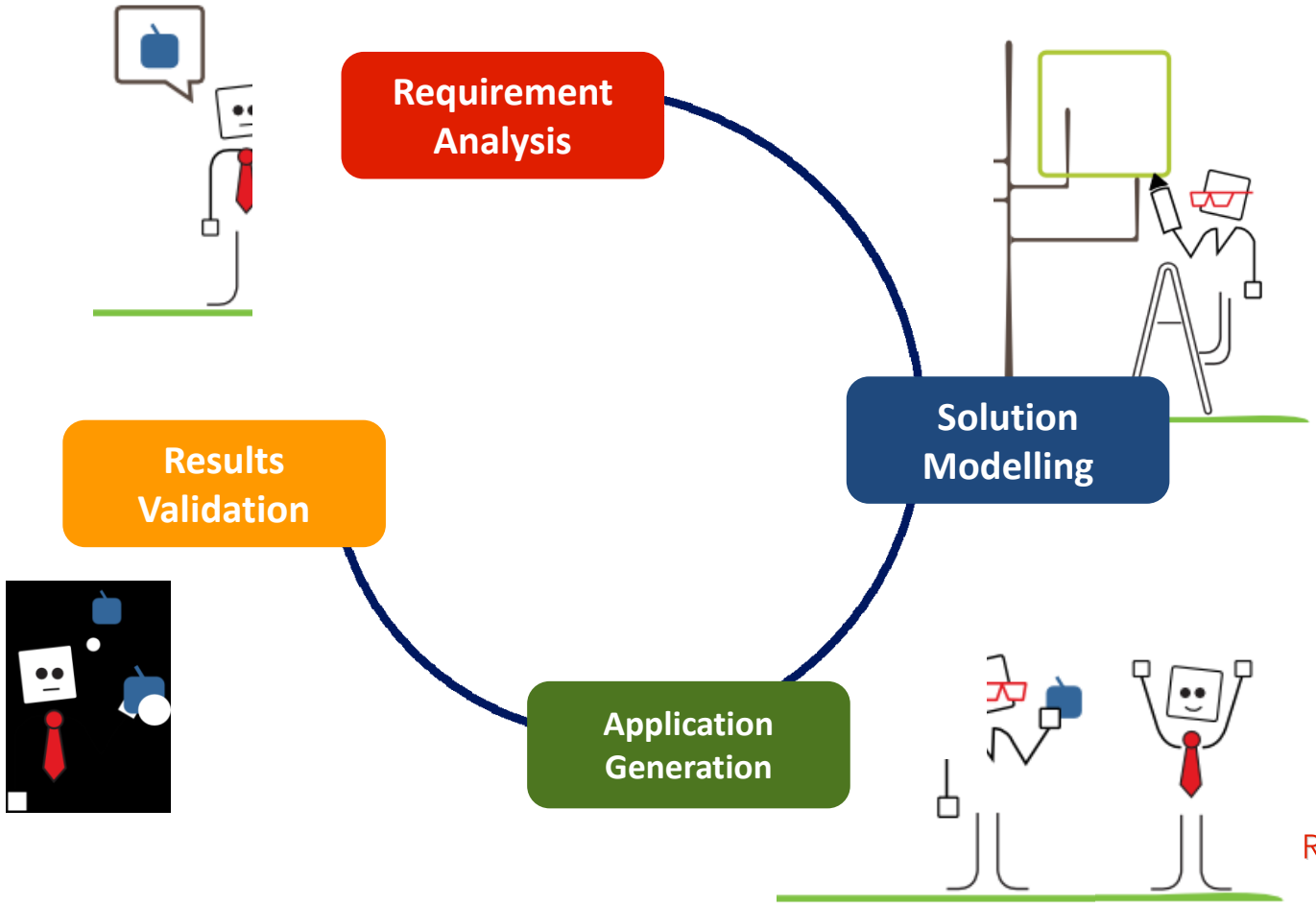
POLITECNICO MILANO

Get the application



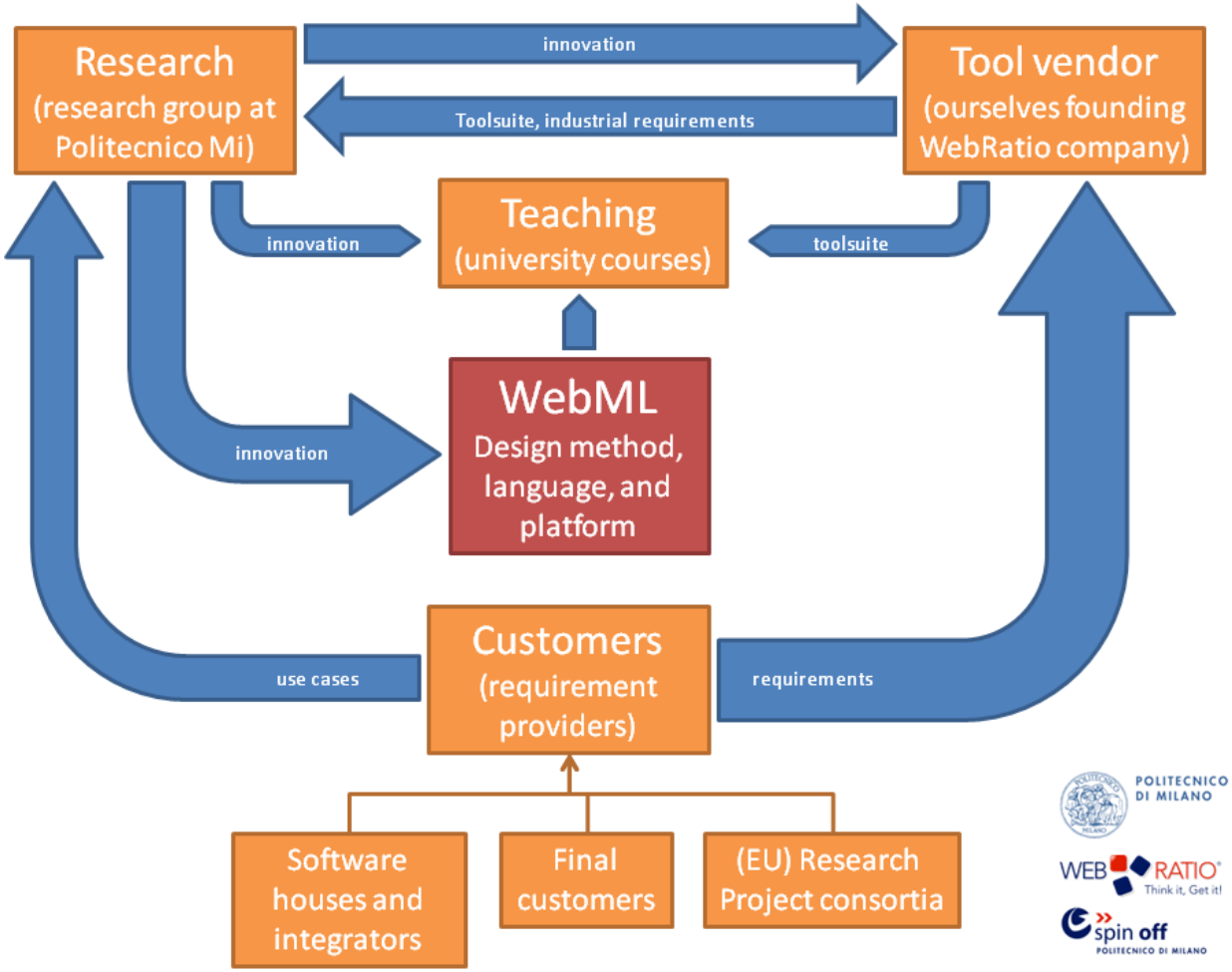
Agile, quick prototyping

Involve business users in the development process and converge quickly to the target



RATIO

Our innovation environment



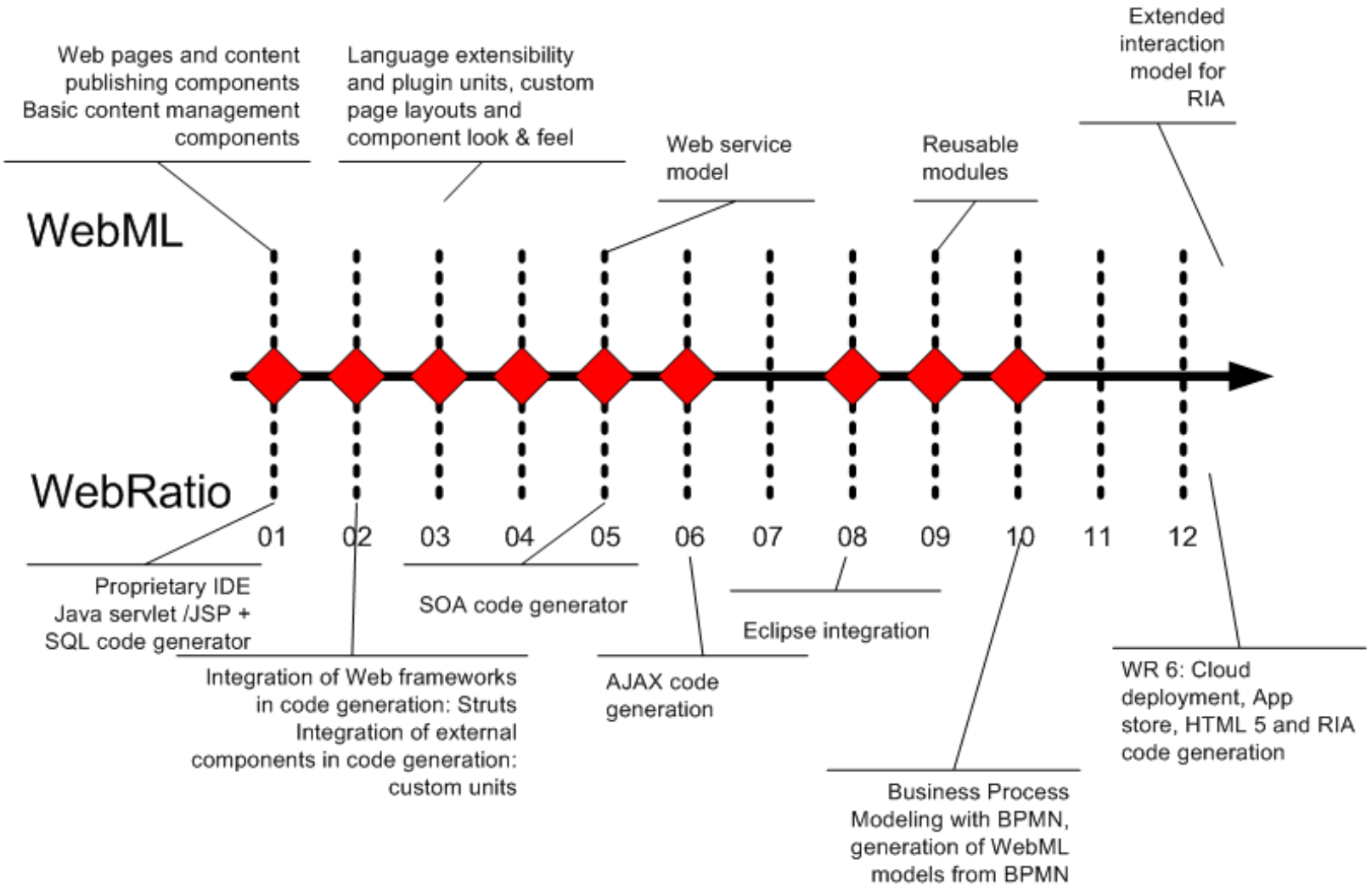
POLITECNICO DI MILANO

WEB RATIO Think it, Get it!

spin off POLITECNICO DI MILANO

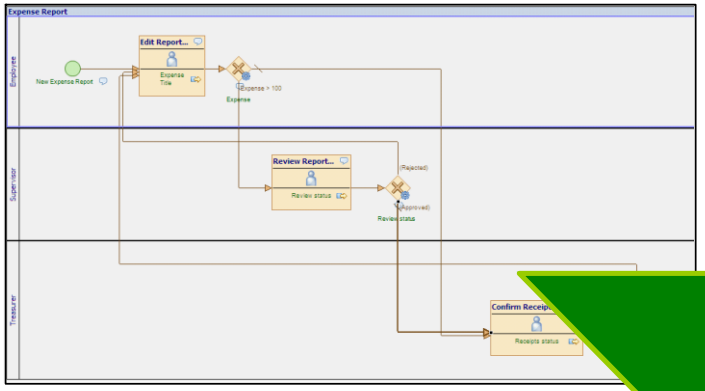
WEB RATIO

Evolution of tool (and language)



WEB RATIO

The final picture



Iterative refinement of the process enactment applications

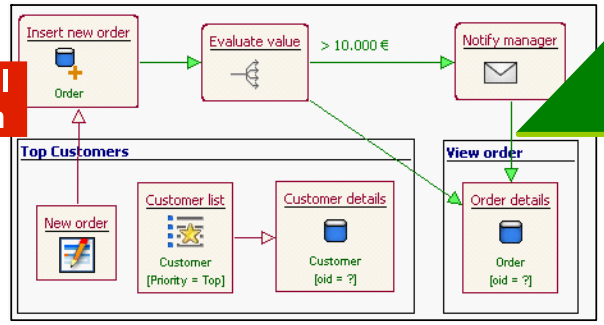
Business Process Model

Application model

Model to model transformation

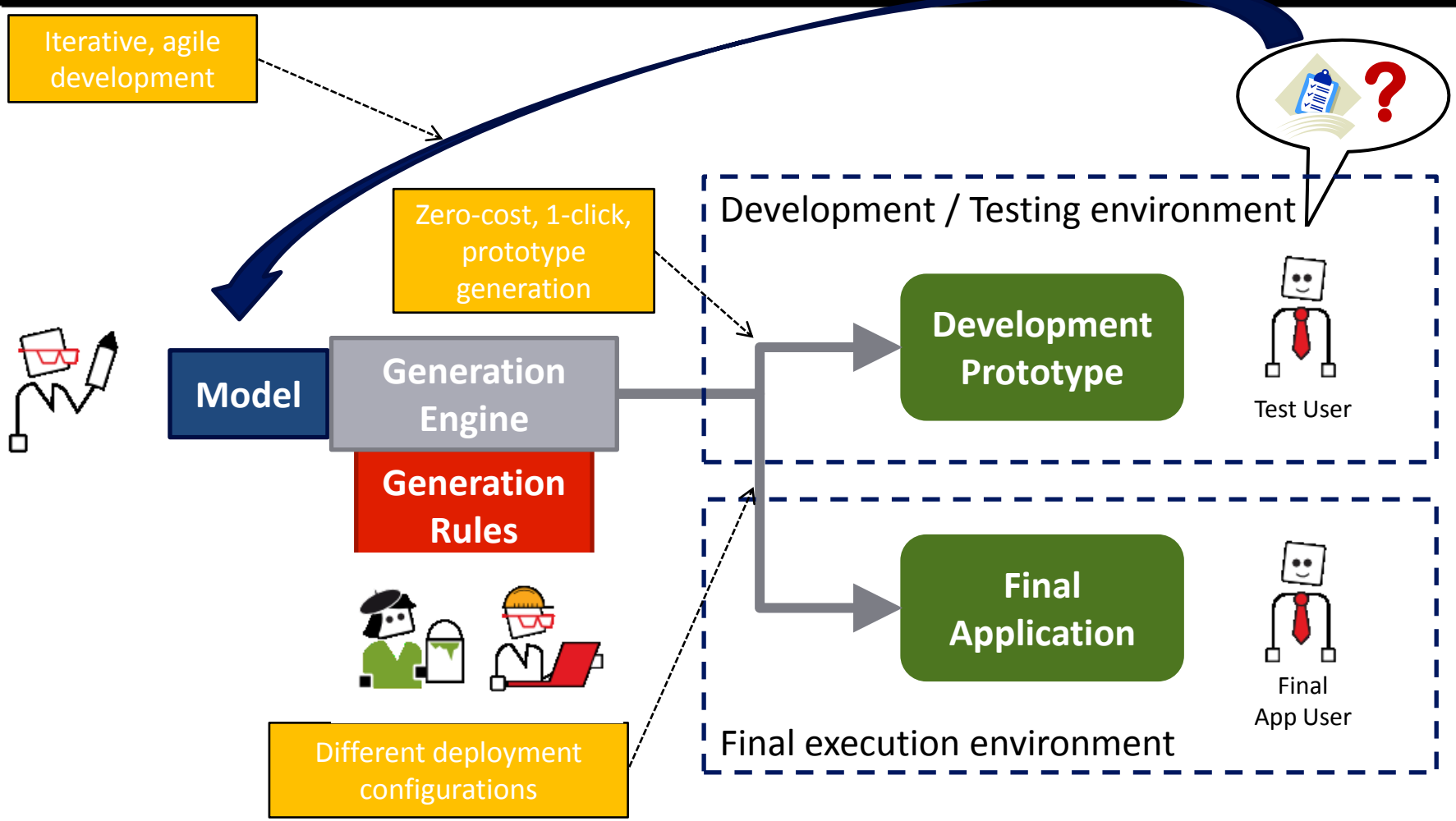
Model to code transformation

Standard Java application



Iterative refinement of the activity execution applications

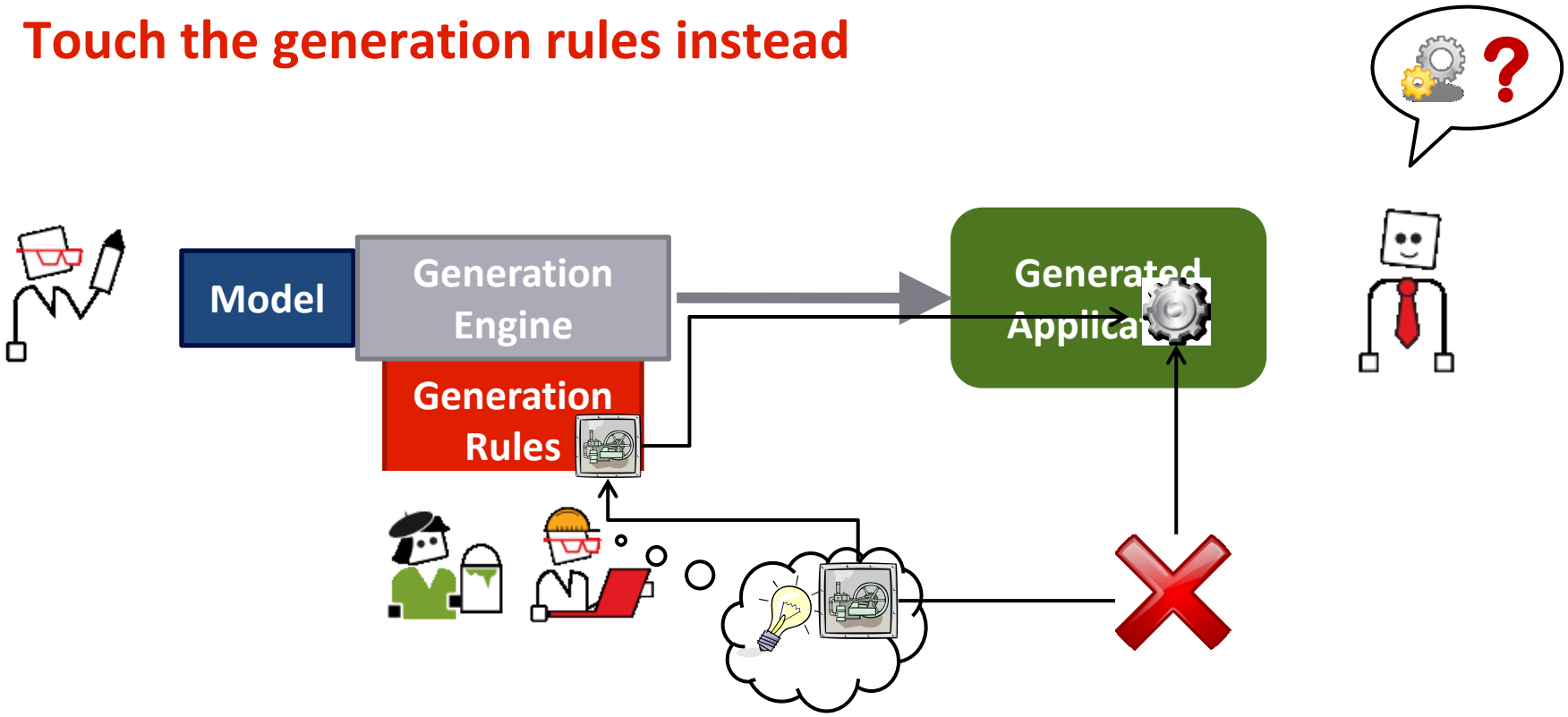
Agility + MDD


Iterative, agile development
Zero-cost, 1-click, prototype generation
Development / Testing environment
Development Prototype
Test User
Model
Generation Engine
Generation Rules
Final Application
Final App User
Final execution environment
Different deployment configurations

The MDE Virtuous Cycle

Do not change the generated application code

Touch the generation rules instead



Case Studies

Kinds of application

Acer

B2C + CMS Web applications initially for 14 EU countries

Corporate news, Product technical & commercial data, Service & Partner area, Where to Buy…

Multilingual, multi-actor, distributed workflows for local and central PMs, local and central MarCom managers

… and a: very limited Time to Market (7 weeks!!)

Size & effort

Class	Dimension	Value
Size	Number of localized B2C web sites	14
	Number of main CMS applications	4 (Admin, News, Product, Other content)
	Number of supported languages	12 for B2C Web sites, 1 for CMS
	Number of data entry masks	39
	Number of automatically generated database tables	46
	Number of automatically generated database views	82
	Number of automatically generated database queries	279 for data extraction, 89 for data update
	Number of automatically generated JSP page templates	48
	Number of automatically generated or reused Java classes	250
	Number of automatically generated Java lines of code	12500 Non commented lines of code
Time & effort	Number of elapsed workdays	49
	Number of development staff-months (analysts and developers)	6 staff-months (6 weeks x 4 persons)
	Total number of prototypes	9
	Average elapsed man days between consecutive prototypes	5,4
	Average number of development man days per prototype	15,5

Size & effort

DEGREE OF AUTOMATION	
Number of manually written SQL statements	17(SQL constraints)
Percentage of automatically generated SQL code	96%
Number of manually written/adapted Java classes /JSP templates	10% JSP templates manually adapted
Percentage of automatically generated Java and JSP code	90% JSP templates, 100% Java classes
COST AND ROI	
Total cost of software development of first version	75.000 €
HW, SW licenses, and connectivity cost of first version	70.000 € (db server license)
Return on investment of first version	12-15 months
Average effort of extension to one additional country	0,5 staff-months
Average cost of extension to one additional country	7.500 €
Average ROI of extension to one additional country	2 months
PRODUCTIVITY	
Number of function points	177 (B2C web site) + 612 (CMS) = 789
Average number of function points delivered per staff-month	131,5

Comments

On the positive side:

- Almost 80% of the delivery effort concentrates in the phases of data design, hypertext design and prototyping:
  - more development time is spent with the application stakeholders

MDD allows a more flexible distribution of responsibilities between the IT department and the business units

The peak productivity rates has reached five times the number of delivered function points per staff-month of a traditional programming language like Java

Comments (continued)

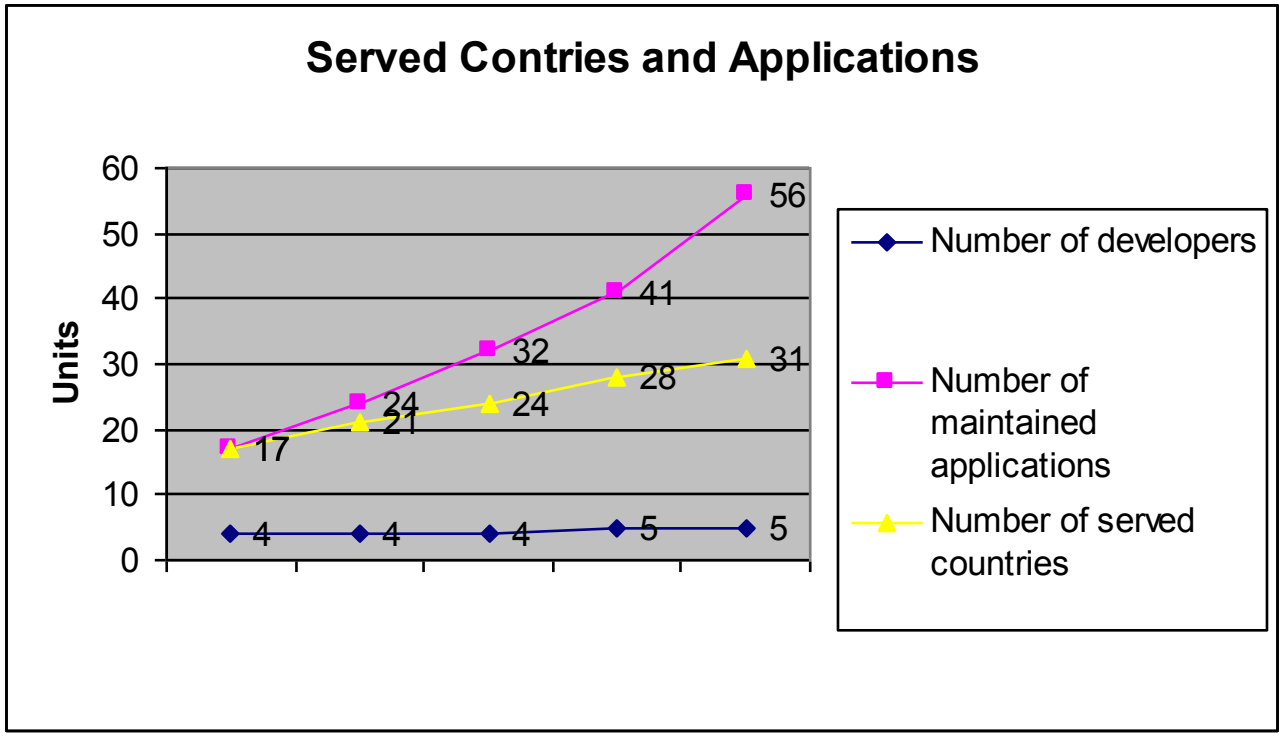
On the negative side..

- Acer estimates that it took from 4 to 6 months to have fully productive developers with MDD, IFML, and WebRatio
- Difficult to find skilled people

..but..

- The initial investment in human capital required by MDD pays off in the mid term
  - MDD benefits testing, maintenance, and evolution (which account for over 60% of the total lifecycle cost)
  - reasoning on the system is far more effective at the conceptual level

Maintenance effort



GTT: Turin Transportation Group

- Public company owned by the City of Turin in Italy
- Local public transport serving 190 million passengers every year.
- A new e-ticketing system (avail able at http://ecommerce.gtt.to.it and serving 64,000 daily passengers)
- published on-line in only 2 months.
- The application comprises 100 page templates (IFML pages) and 1215 IFML units.
- KEY: iterative and quick prototyping approach supported by WebRatio

A2A: Utility in Milan

- Multi-utility company buying and selling wholesale electric power.
- Integrated Energy Management System that replaced individual productivity tools used by traders for the management of electric power.
- KEY: quick prototyping approach and involvement of actual users in the development process.
- Deployment of final app in 6 months after the initial meeting with WebRatio (time to market that took one-third of the time estimated in case of adoption of a traditional development)

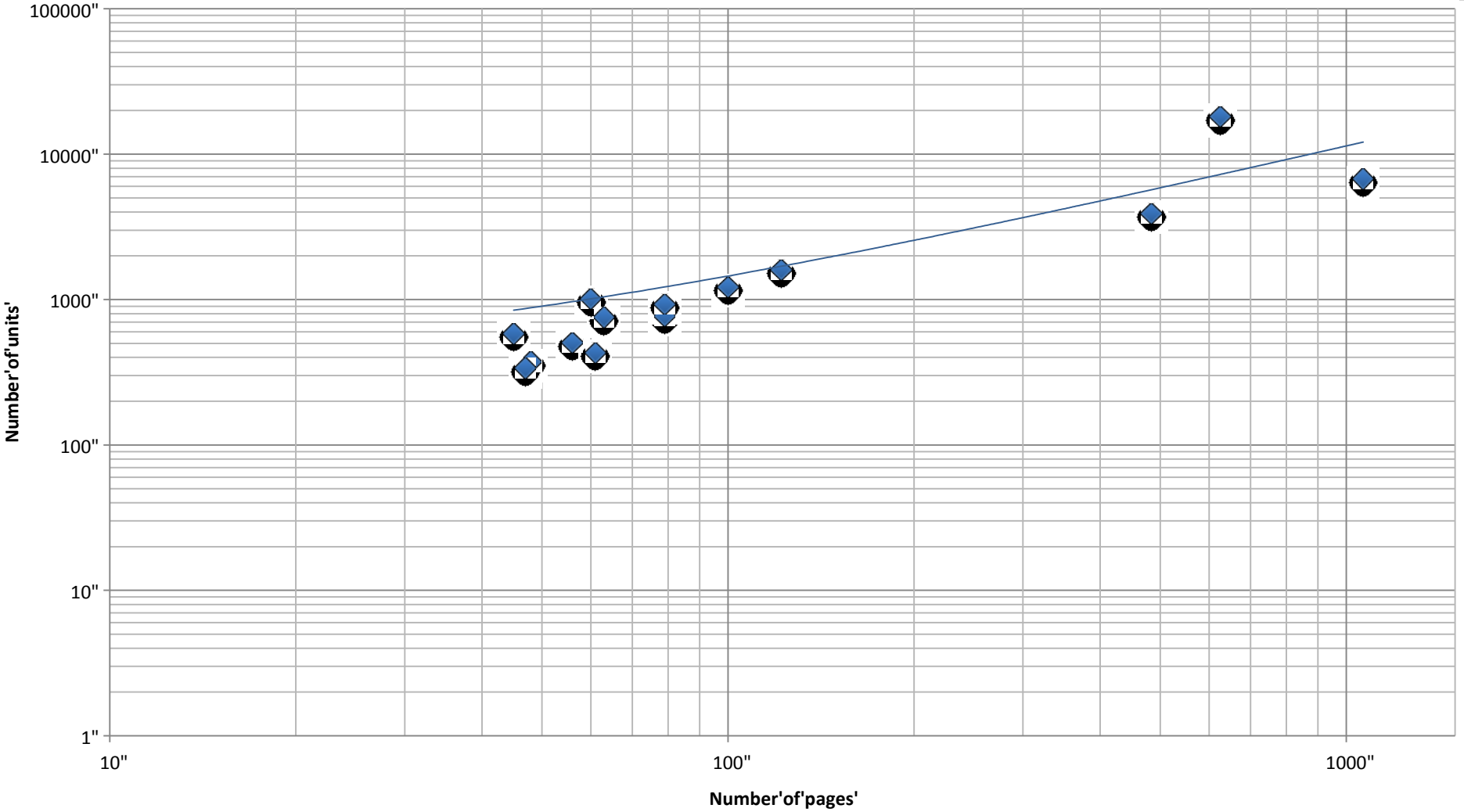
Other experiences

- Banking (UniCredit)
  - BPM + SOA + Web interfaces
  - Crucial points: modularization, multiple models integration, multiple tools integration, strict runtime platform requirements
- Banking (ABI)
  - System integration (Pure backend!)
  - Why IFML?
- Latin America
  - Cooperatives, banks, public bodies, central government
- Wholesale (IKEA)
- Financial / leasing (GE Capital)

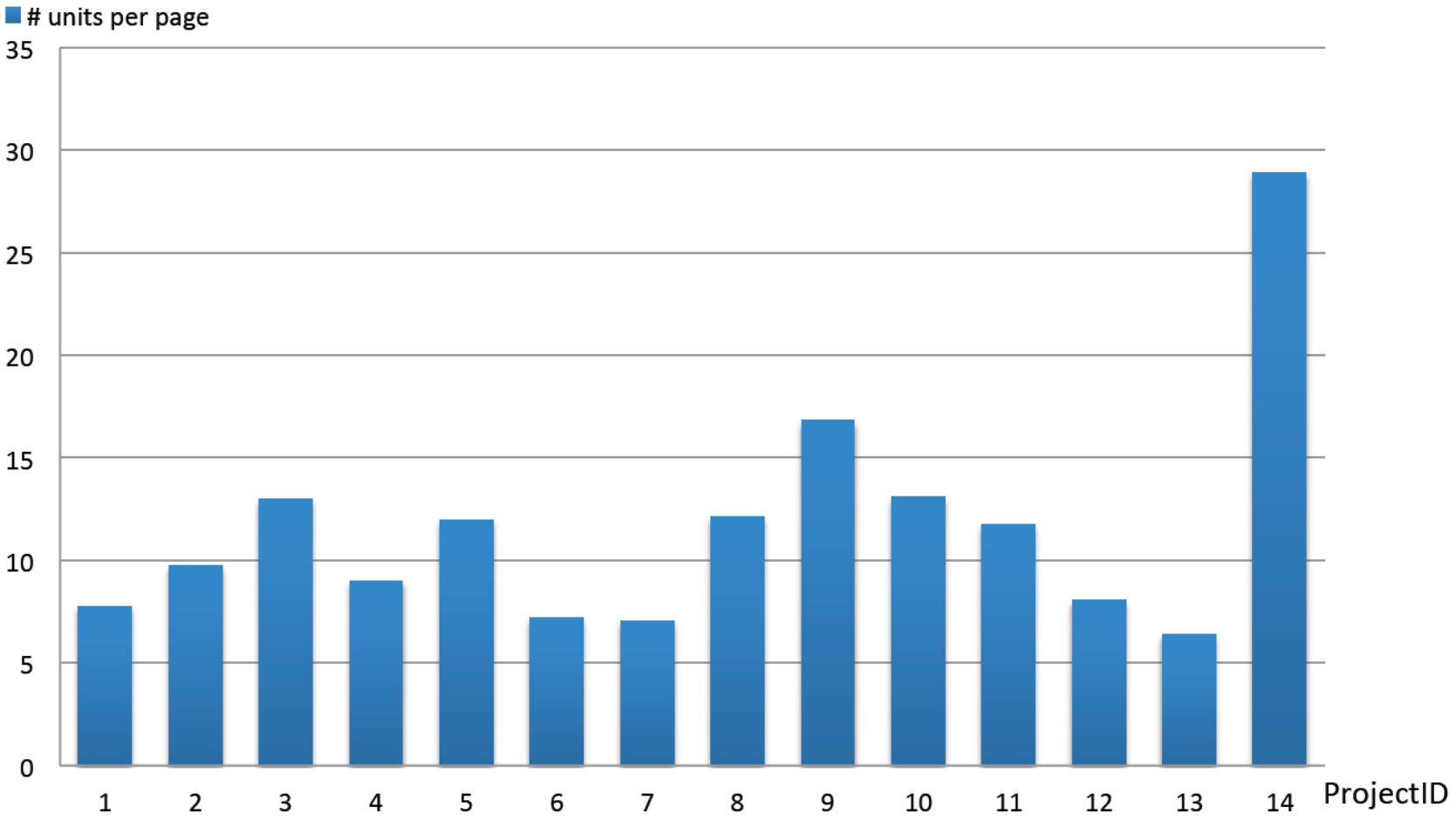
Where IFML works

- **Models integration**
- **Large applications with strong need for coherence and standardized paradigms**
  - Cooperatives, banks, public bodies, central government
- **Service orientation**
- **No pure modeling exists**
- **Code generation still win-win**

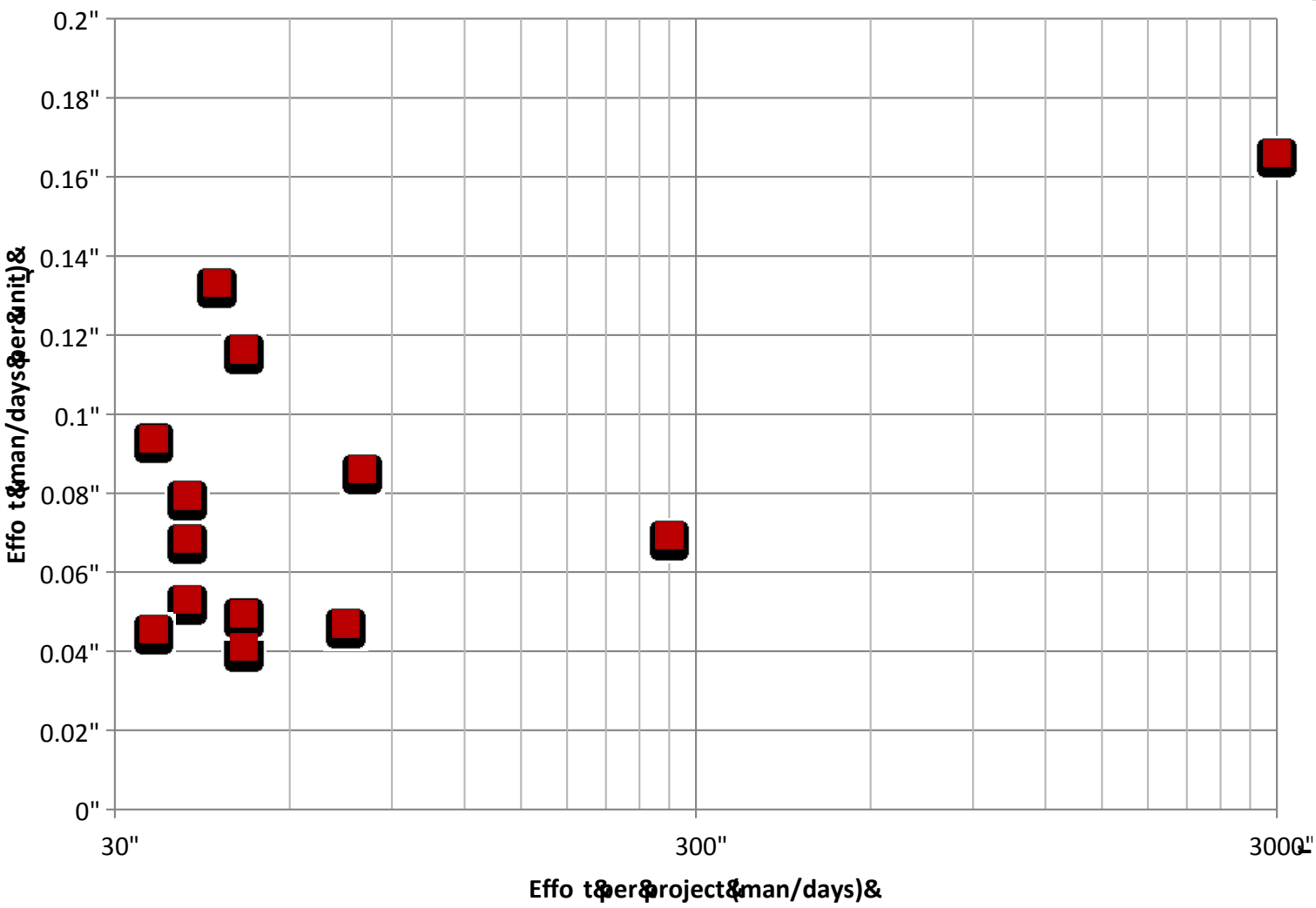
Components and pages per project



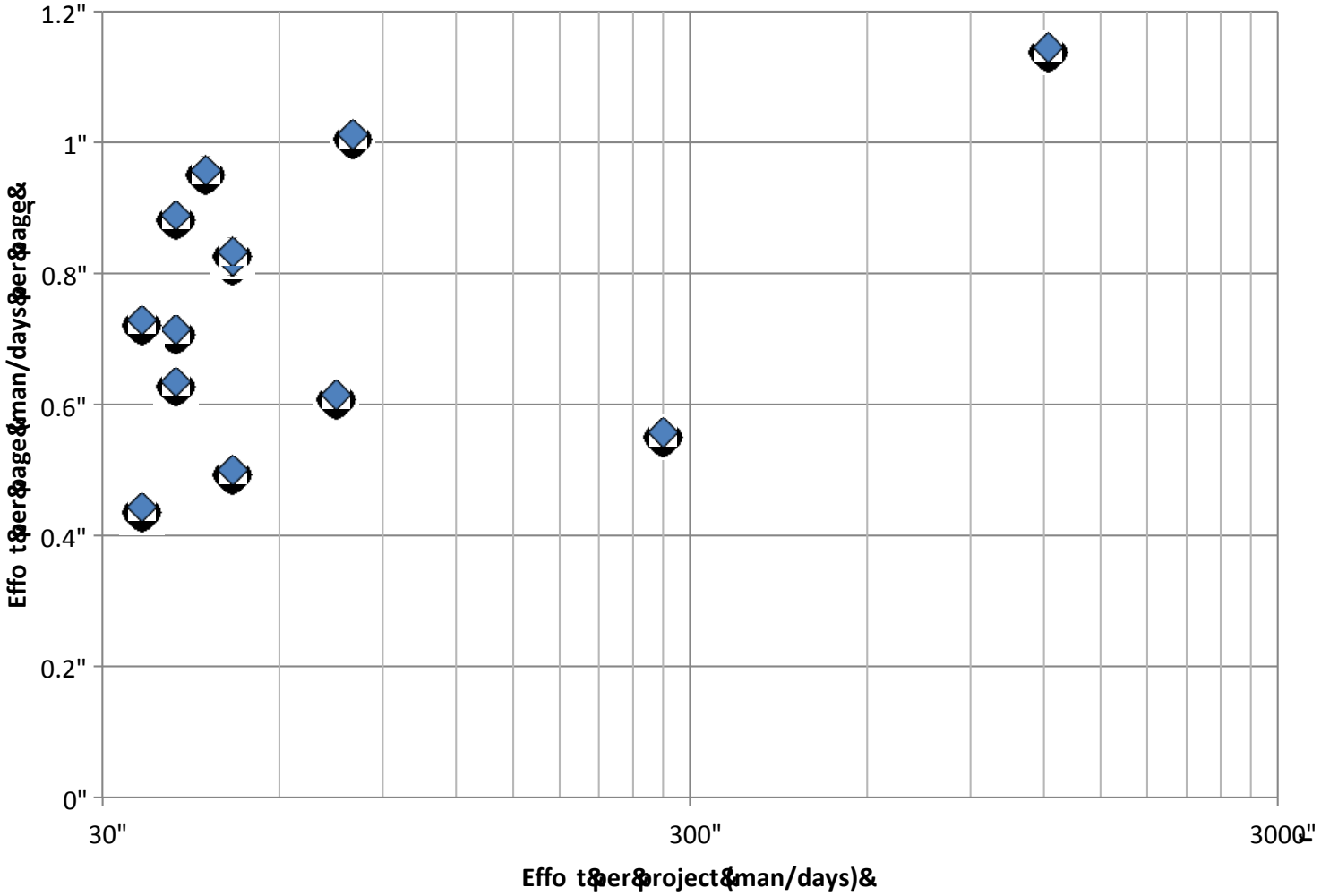
Components per page (avg)



Man/days per component



Man/days per page



WEB RATIO

POLITECNICO MILANO

Tool usage stats

Description	Value
# of times the tool has been opened daily	1.79
# of daily code generations	11.76
# of 1-click generation and publishing of the application	0.26
# of checks of the modelling warnings	2.09
# of checks of graphical layout warning	0.11
# of automatic generations of the documentation	0.02

WEB RATIO

POLITECNICO MILANO

(some) references

S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, M. Matera: Designing Data-Intensive Web Applications, Morgan-Kaufmann Publishers, San Francisco, ISBN 1-55860-843-5 (Series edited by Jim Gray, foreword by Adam Bosworth) 590 pages.

M. Brambilla, J. Cabot, M. Wimmer: Model Driven Software Engineering in Practice. Morgan & Claypool, USA, September 2012, foreword by Richard Soley (OMG), 184 pages. ISBN 978-1608458820.

Manolescu, M. Brambilla, S. Ceri, S. Comai, P. Fraternali: Model-driven design and deployment of service-enabled web applications. ACM Trans. Internet Technology (TOIT). 5(3), pp. 439-479 (2005).

M. Brambilla, S. Ceri, P. Fraternali, I. Manolescu: Process modeling in Web applications. ACM Trans. Softw. Eng. Methodol (TOSEM). 15(4), pp. 360-409 (2006).

M. Brambilla, I. Celino, S. Ceri, D. Cerizza, E. Della Valle, F. M. Facca: Model-Driven Design and Development of Semantic Web Service Applications, ACM Trans. on Internet Technology (TOIT). 8(1), pp.3:1 - 3:31 (2007).

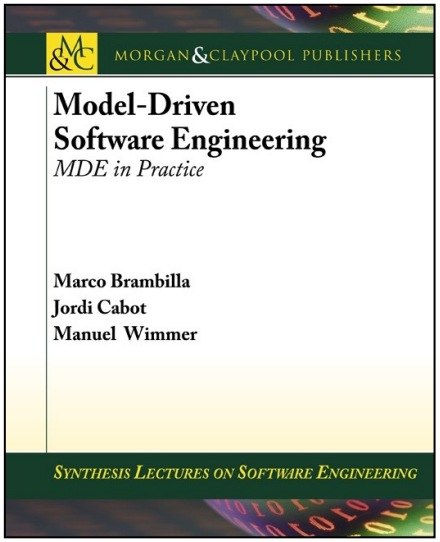
M. Brambilla: From Requirements to Implementation of Ad-hoc Social Web Applications: an Empirical Pattern-Based Approach. IET Software, 6(2), 2012, pp.114-126.

M. Brambilla, S. Ceri, S. Comai, C. Tziviskou. Exception Handling in Workflow-Driven Web Applications. WWW 2005 Int. Conference on World Wide Web. ACM, pp. 170-179.

Some Ads

"Model Driven Software Engineering in Practice". Brambilla, Cabot, Wimmer.

Morgan&Claypool, USA, 2012



MD* blog

www.modeldrivenstar.com

And the upcoming **IFML book!**

Morgan-Kauffman – Elsevier, USA, 2014

Thanks!

Marco Brambilla

marcobrambi

marco.brambilla@polimi.it