Model-Driven Development of Software Front Ends

Marco Brambilla
Politecnico di Milano and WebRatio

@marcobrambi
marco.brambilla@polimi.it
The modeling approach
A model is based on an original (=system).

A model only reflects a (relevant) selection of the original’s properties.

A model needs to be usable in place of an original with respect to some purpose.

Purposes:
• descriptive purposes
• prescriptive purposes
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.
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*
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
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
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.
Eg., business process

- New customer arrives to counter
- Check customer identity
- Retrieve account number
- Ask customer about operation to perform
- Execute operation on account
Eg., business object description and constraints

```
«business entity»
Account
- number : Integer {unique}
- balance : Float
+ getNumber() : Integer
+ setNumber(number : Integer)
...
```

-- English
Account number must be between 1000 and 9999

-- OCL
context Account inv:
number >= 1000 and number <= 9999
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
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
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

- Binding to Persistence Layer
- Navigation Path
- Content
- Event
- Binding to Business Logic
- Binding to Persistence Layer
IFML Objectives: Content
Designing Data-Intensive Web Applications (The Morgan Kaufmann Series in Data Management Systems) [Paperback]

- Stefano Ceri (Author), Piero Fraternali (Author), Aldo Bongio (Author), Marco Brambilla (Author), Sara Coma (Author), Marianna Katsarou (Author)

- Rating: (1 customer review)

- List Price: $93.95
- Price: $72.54 & FREE Shipping. Details
- You Save: $21.41 (23%)

- In Stock.
- Ships from and sold by Amazon.com. Gift-wrap available.

- Want it tomorrow, Sept. 19? Order within 5 hrs 46 mins and choose One-Day Shipping at checkout. Details

- 30 new from $24.50  22 used from $8.01

- Kindle Edition
- Read instantly on your iPad, PC, Mac, Android tablet or Kindle Fire
- Buy Price: $60.86
IFML Objectives: Navigation Path

Best Books of the Month

September Spotlight:
The Lowland by Jhumpa Lahiri

"Her most accessible – and most profound – book yet."
– Sara Nelson, Amazon Editorial Director

Amazon Editors’ Top Picks for the Best Books of September

We’re happy to share with you the unique mix of books that our editors have hand picked as this month’s best.
IFML Objectives: Events
IFML Objectives: Events
IFML Objectives: Binding to business logic
IFML Objectives: Binding to business logic
IFML Objectives: Binding to persistence
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 Multiple containers and navigation

ProductList

Select product

Product

ProductDetails

On-Line Book Store

Home ➔ See categories ➔ Shopping Cart ➔ Exit

Home / Books

Standard QWE: A new modelling proposal
Description:
QWE is a major innovation in the field of software development. It is independent of the organization of the software implementation.

See more>

Graphic Interfaces: In the hands of the user
Description:
This book introduces, documents and explains the important new tool that aids in the formalization of knowledge, and is also a way of describing the concepts that make up abstract solutions to software development problems.

See more>>

Price: $44.99

Add to cart
IFML Single container and navigation

ProductList

Select product

ProductDetails

On-Line Book Store

Standard QWE: A new modelling proposal
Description:
QWE is a major innovation in the field of software development. It is independent of the organization of the software implementation. It is a highly abstract thinking tool that aids in the formulation of knowledge and is also a way of describing the concepts that make up abstract solutions to software development problems.
Price: $44.99

Add to cart
IFML by example
IFML by example

Nesting of ViewContainers

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

[XOR] MAIL Top

[D] [L] Messages

[XOR] MessageSearch

[D] Search

FullSearch

[XOR] MessageManagement

[D] MailBox

[XOR] MessageViewer

[D] Message List

MessageDetails

Message toolbar

[L] Contacts

[L] Settings

[L] MessageWriter
IFML by example

Actions
IFML – adding details to ViewComponents

**ViewComponentParts:**

- Data binding
- Parameters

**Types of ViewComponents (<<List>>)**

- «List» Name
  - «DataBinding» Binding
  - «ConditionalExpression» expression

- «Details» Name
  - «DataBinding» Binding
  - «ConditionalExpression» expression

- «Form» Message Writer
  - «SimpleField» Field1: type1
  - «SimpleField» Field2: type2
  - «SelectionField» Selection1
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, ...
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

[L] User Data Input

«Form» UserInput

«SimpleField» Name

«SelectionField» Country

«DataBinding» Country

«VisualizationAttributes» name

«SelectionField» State/Province

«DataBinding» State

«VisualizationAttributes» name

«ConditionalExpression»
UserCountry.States->exists(self)

«ParamBindingGroup»
SelectedCountry -> UserCountry
Example: Wizard

[XOR] InstallationWizard

[D] Step1
- «Form» Terms&Conditions
  - «SimpleField» Accept: Boolean
  - «Parameter» Location: directory
  - «Parameter» Options: Bool

Step2
- «Form» Location
  - «SimpleField» Location: directory
  - «Parameter» Accept: Boolean
  - «Parameter» Options: Bool

Step3
- «Form» Options
  - «SimpleField» Options: Bool
  - «Parameter» Location: directory
  - «Parameter» Accept: Boolean

End

«ParamBindingGroup»
Location → Location
Options → Options
Accept → Accept

Next
Previous
Cancel
Example: Faceted Search

Faceted search

«Form» Search

«SimpleField» keyword: string

Search

«DataBinding» Result

«ConditionalExpression»
Years->includes(self.year)
AND
Venues-
>includes(self.venue)

«List» Results

«List» Years

«ActivationExpression»
YearFacet->notEmpty()

«List» Venues

«ActivationExpression»
VenueFacet->notEmpty()

«ParamBindingGroup»
SelectedYears-> Years

«ParamBindingGroup»
SelectedVenues-> Venues
Example: Details on Actions

ProductCreation

- «Form» EnterProductData
  - «SimpleField» Code: string
  - «SimpleField» Name: string
  - «SimpleField» Price: integer
  - «SelectionField» Category
    - «DataBinding» Category
    - «VisualizationAttributes» name

NewProductDisplay

- «Details» NewProductDetails
  - «DataBinding» Product
    - «ConditionalExpression» self = product

CreateNewProduct

CreateProduct

- «DynamicBehavior» ProductFactory.createProduct(...)

NormalTermination

ExceptionalTermination

Error

Error Message
Example: Mobile Device, Camera Controls

[XOR] Photo Shooter

[D][L]<system>CameraCanvas

«Modal»CameraSettings

[L] Viewer

«ScrollableList»Photos

«DataBinding» Image

PhotoAvailable

Shoot

PhotoAvailable

BlockSize=1

«system»MediaGallery

OpenInMediaGallery

«system»Media Gallery
Example: NFC Controls

NFCCardSender

Send via NFC

NFCCardReceiver

NFCDataDiscovered

Payload.Name → name
Payload.Phone → PhoneNumber

«Details» ReceivedCard

Save Contact

Save

Discard

«Details» Personal Card
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

IFML

UML Statechart
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, ...)

![Diagram showing a use case model with a Sales Clerk interacting with Handle Rental, Handle Renter, and Handover Car.]
Integration with BPMN

The UI of each activity can be described by an IFML module. Some UML dynamic diagrams (e.g., activity, sequence, …)
IFML concrete syntax by example

IFML Modules - definition
Example of UML - IFML mapping

IFML models can be reworked or refined after being generated.
Multiple aspects modeling – 2
(implementation and architecture)
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
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
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.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Weak Ties / Tacit Knowledge</th>
<th>Transparency</th>
<th>Participation</th>
<th>Activity distribution</th>
<th>Decision distribution</th>
<th>Social f.back</th>
<th>Knowledge sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic enrollment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poll</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>People / Skill search</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Social content publication</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Social sourcing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Progress notification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ranking and commenting</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
WebRatio runtime architecture and extension for Social Business Logic

Client Layer
- Browser
- Web Server

Presentation Layer
- SOAP sender / listener

Business Layer
- JSP engine
- Message composer
- Message decomposer
- Conversation Manager

Data Layer
- WebML runtime
- RDBMS
- LDAP
- XML
- Other data sources

Legenda
- Processing of messages received by the site
- Construction of messages sent by the site
- Standard components interactions

Social BPM Components
- Social networks and social APIs (public or enterprise)
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
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
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
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 mgmt
- Truly multi-role model-driven development
Some numbers

WebRatio is
- now at 7\textsuperscript{th} 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
You capture business requirements in abstract, technology independent models
You customize the environment by defining your own generation rules
You get a tailored, yet standard, Java Web application with no proprietary runtime.
Get the application
Involve business users in the development process and converge quickly to the target.
Our innovation environment

- Research (research group at Politecnico MI)
- Tool vendor (ourselves founding WebRatio company)
- Teaching (university courses)
- WebML (Design method, language, and platform)
- Customers (requirement providers)
  - Software houses and integrators
  - Final customers
  - (EU) Research Project consortia
Evolution of tool (and language)

- Web pages and content publishing components
- Basic content management components
- Language extensibility and plugin units, custom page layouts and component look & feel
- Web service model
- Reusable modules
- Extended interaction model for RIA

WebML

- Proprietary IDE
- Java servlet / JSP + SQL code generator
- Integration of Web frameworks in code generation: Struts
- Integration of external components in code generation: custom units
- SOA code generator
- AJAX code generation
- Eclipse integration
- Business Process Modeling with BPMN, generation of WebML models from BPMN

WebRatio

- WR 6: Cloud deployment, App store, HTML 5 and RIA code generation
The final picture
Agility + MDD

Iterative, agile development

Zero-cost, 1-click, prototype generation

Model
Generation Engine
Generation Rules
Different deployment configurations

Development / Testing environment

Development Prototype
Test User

Final Application
Final App User

Final execution environment
The MDE Virtuous Cycle

Do not change the generated application code
Touch the generation rules instead
Case Studies
Kinds of application

- Corporate Operations
- Human Capital Management
- Product Life Cycle Management
- Customer Relationship Management
- Supply Chain Management
- Enterprise Resource Planning
- Sales and Lead Management
- Marketing Resources Mgt
- Web Customer Services
- B2C/B2B E-Commerce
- Document Management
- Knowledge Management
- Learning Management
- Project Management
- Knowledge Support
- Business Intelligence
- Financial Services
- Enterprise Resource Planning
- Web Content Management
- Customer Relationship Management
- Supplier Relationship Mgt
- Front-Office Process Mgt
- Web Front-End of accounting sys.
- Payment Services Orchestration
- Financial Services
- Order Mgt
- Recruitment
- Training
- Workforce Management
- Enterprise Governance
- Risk and Compliance
- Supplier Relationship Mgt
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

<table>
<thead>
<tr>
<th>Class</th>
<th>Dimension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td><strong>Number of localized B2C web sites</strong></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td><strong>Number of main CMS applications</strong></td>
<td>4 (Admin, News, Product, Other content)</td>
</tr>
<tr>
<td></td>
<td><strong>Number of supported languages</strong></td>
<td>12 for B2C Web sites, 1 for CMS</td>
</tr>
<tr>
<td></td>
<td><strong>Number of data entry masks</strong></td>
<td>39</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated database tables</strong></td>
<td>46</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated database views</strong></td>
<td>82</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated database queries</strong></td>
<td>279 for data extraction, 89 for data update</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated JSP page templates</strong></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated or reused Java classes</strong></td>
<td>250</td>
</tr>
<tr>
<td></td>
<td><strong>Number of automatically generated Java lines of code</strong></td>
<td>12500 Non commented lines of code</td>
</tr>
<tr>
<td><strong>Time &amp; effort</strong></td>
<td><strong>Number of elapsed workdays</strong></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td><strong>Number of development staff-months (analysts and developers)</strong></td>
<td>6 staff-months (6 weeks x 4 persons)</td>
</tr>
<tr>
<td></td>
<td><strong>Total number of prototypes</strong></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td><strong>Average elapsed man days between consecutive prototypes</strong></td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td><strong>Average number of development man days per prototype</strong></td>
<td>15.5</td>
</tr>
</tbody>
</table>
## Size & effort

### DEGREE OF AUTOMATION

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of manually written SQL statements</td>
<td>17 (SQL constraints)</td>
</tr>
<tr>
<td>Percentage of automatically generated SQL code</td>
<td>96%</td>
</tr>
<tr>
<td>Number of manually written/adapted Java classes / JSP templates</td>
<td>10% JSP templates manually adapted</td>
</tr>
<tr>
<td>Percentage of automatically generated Java and JSP code</td>
<td>90% JSP templates, 100% Java classes</td>
</tr>
</tbody>
</table>

### COST AND ROI

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total cost of software development of first version</td>
<td>75,000 €</td>
</tr>
<tr>
<td>HW, SW licenses, and connectivity cost of first version</td>
<td>70,000 € (db server license)</td>
</tr>
<tr>
<td>Return on investment of first version</td>
<td>12-15 months</td>
</tr>
<tr>
<td>Average effort of extension to one additional country</td>
<td>0.5 staff-months</td>
</tr>
<tr>
<td>Average cost of extension to one additional country</td>
<td>7,500 €</td>
</tr>
<tr>
<td>Average ROI of extension to one additional country</td>
<td>2 months</td>
</tr>
</tbody>
</table>

### PRODUCTIVITY

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of function points</td>
<td>177 (B2C web site) + 612 (CMS) = 789</td>
</tr>
<tr>
<td>Average number of function points delivered per staff-month</td>
<td>131.5</td>
</tr>
</tbody>
</table>
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
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

- Number of developers: 4, 4, 4, 5, 5
- Number of maintained applications: 17, 24, 32, 41, 56
- Number of served countries: 17, 21, 24, 28, 31

Public company owned by the City of Turin in Italy

Local public transport serving 190 million passengers every year.

A new e-ticketing system (available 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
• 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

Number of units

Number of pages

Man days per page and per unit
Man/days per component

- Effort per component (man/days)
- Effort per project (man/days)
Man/days per page
# Tool usage stats

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># of times the tool has been opened daily</td>
<td>1.79</td>
</tr>
<tr>
<td># of daily code generations</td>
<td>11.76</td>
</tr>
<tr>
<td># of 1-click generation and publishing of the application</td>
<td>0.26</td>
</tr>
<tr>
<td># of checks of the modelling warnings</td>
<td>2.09</td>
</tr>
<tr>
<td># of checks of graphical layout warning</td>
<td>0.11</td>
</tr>
<tr>
<td># of automatic generations of the documentation</td>
<td>0.02</td>
</tr>
</tbody>
</table>
(some) references


Some Ads


MD* blog
www.modeldrivenstar.com

And the upcoming IFML book!
Morgan-Kauffman – Elsevier, USA, 2014
Thanks!

Marco Brambilla
marcobrambi
marco.brambilla@polimi.it