MDE
to the people
Jordi Cabot
jordi.cabot@inria.fr
Barcelona – UPC
22/03/2013
AtlanMod
AtlanMod
The team
Our Research
Research

- MDE as a software engineering paradigm to improve software production, evolution and operation.

- MDE based on the rigorous use of software models and model manipulation operations.

- AtlanMod researches core MDE techniques and their adaptation to specially relevant industrial challenges.
Research

- **Application-driven research**
  - Constant collaboration with companies

- **Open source community via Eclipse**
  - MoDisco, AM3, EMF Facet, ATL, AMW, etc
We have advanced a lot on the core techniques

- UML and profiles
- DSLs & Language workbenches
- Model-to-model and model-to-text transformations
- Model management and evolution
- ...

...
But it’s clearly not enough

- Modeling will be commonplace in 3 years time – S. Mellor
  Though he is giving the same answer for the last 20 years
Our place in MDE

- Model-Driven Reverse Engineering
- Human Factor
- Formal Methods for MDE
- Scalability in MDE
- Core MDE

AtlanMod
Core MDE

- **Model Transformations**
  - Refactoring of transformations
  - Bidirectional transformations
  - Reactive ATL

- **Model management**
  - EMF Profiles
  - DSL for querying and manipulating model repositories
Model to Model transformations (M2M)

```plaintext
let j2dNet : Transformation = Transformation::allInstances()
  ->any(t | t.identifier = 'j2dNet')
in
Model::allInstances()
  ->select(m | m.conformsTo.kind = 'Java')
  ->collect (jModel | j2dNet.applyTo(jModel))

TransformationRecord::allInstances() -> collect (tr | tr.run())
```
MDE-based software development process

Errors in models will lead to errors in the resulting software
Quality - EMFtoCSP

EMFtoCSP

Properties to check
- Weak Satisfiability
- Strong Satisfiability
- Liveliness
- Constraints Redundancies
- Constraints Subsumptions
...

Model and Properties to CSP Translator

CSP Code

CSP Engine (EclipseCS)

Valid Instantiation Sample

VALID

NO VALID

© AtlanMod - atlanmod-contact@mines-nantes.fr
Quality

- Verification of MT
  - With CSPs but also SMTs

- Testing of MT
- As old as CS itself. Always relevant
- First level models: zero information loss
Reverse Engineering

- Reverse engineering of security policies
- Reverse engineering of business rules
- Moving applications to the cloud
Scalability

- Scalability important both at the model (loading very large models) and model manipulations level (executing complex transformations on large models)

- Key problem in industrial scenarios but far from a trivial one
Virtual Models (i)

- “a virtual model is a model whose (virtual) model elements are proxies to elements contained in other models”

Diagram:
- Virtual Model VMab
- Virtual EMF
- Model Ma
- Links
- Model Mb
- Transformations
- Editors
Scalability

- Scalability important both at the model (loading very large models) and model manipulations level (executing complex transformations on large models)

- Key problem in industrial scenarios but far from a trivial one
Scalability for MT

- Incremental ATL
- Lazy ATL
- Parallel ATL
Human Factors (the DSL case)
In General

- Research is not aligned with the real needs of end-users
  - Architects and NFRs
  - Ikerlan

- This is a very important problem when creating DSLs
  - Quality of DSL = user experience
  - Evaluating user experience is a challenging task
  - We cannot develop specific techniques for each different DSL
  - Need the participation of users
Quality

- We know what quality properties make sense for models (e.g. satisfiability) but they do not translate well to DSLs.

- Quality of DSL = user experience

- Evaluating user experience is a challenging task
  - We cannot develop specific techniques for each different DSL
  - Need the participation of users
Quality: Dealing with users is not easy
The DSL case - Before
Process

- DSLs are domain-specific but still it’s a non domain-expert who creates the DSL

- **Collaboro** aims to enable a more collaborative process
Motivation

Developers

End-Users
Motivation

Developers

End-Users
Group of people involved with the DSML under development, which includes both technical level users and domain expert users
Anatomy

Concepts & relationships
Well-formed rules
Textual
Graphical
Denotational
Pragmatic
Translational
Operational

Abstract Syntax
Concrete Syntax
Semantics

DSL
Current development process

Decision
- Is it really necessary to provide a new language?
- Can we take advantage of existing languages?

Analysis
- What is the domain?

Design
- What is the most suitable syntax?
- How to define semantics?

Implementation
- Which tool will be used?
- Will wizards be included?

Deployment
- Is the end-user happy?
- Do they need anything else?
Boehm’s graph
Current end-user participation

![Graph showing the increase in effort over time. The effort increases exponentially with time.](image-url)
The sooner the better
How? Existing tools
Existing tools

No support for DSLs
Our proposal

Participation

Collaboration

Collabororo
But… technically?

**Participation**
- Providing means to discuss about language elements
- Overcoming technical barriers

**Collaboration**
- Suitable environment
- Fostering end-user discussion
- Facilitating voting processes
Example: before

Abstract Syntax

```
IDEElement
  id : String

Vehicle
  follows
  0..* isFollowedBy

Route
  0..* isContained

Bus

Tram

Stop
  contains
  lat : double
  lon : double
```

Concrete Syntax Example

```
tram 1:    route A:    stop 001:
route A:  stops : 001, 002;  lat: 23.1082
          ...          lon: 12.9883
          ...
```
Example

Change Proposal

It is necessary to add support for handicapped people in each Stop.

Agreement: Developer 1  Developer 2

End-User 1

Developer 1
Example

Change Proposal

It is necessary to add support for handicapped people in each Stop.

Agreement: ✅ Developer 1 ✅ Developer 2

End-User 1

Developer 1
Example

Change Proposal

It is necessary to add support for handicapped people in each Stop.

Agreement: ✔️ Developer 1 ✔️ Developer 2

ACCEPTED

End-User 1

Developer 1
Example

Change Proposal
It is necessary to add support for handicapped people in each Stop.
Agreement: ✔️ Developer 1 ✔️ Developer 2
ACCEPTED

Solution
Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"
Agreement: Developer 2 End-User 1
Example

Change Proposal

It is necessary to add support for handicapped people in each Stop.

Agreement: ✅ Developer 1 ✅ Developer 2

Solution

Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"

Agreement: ✅ Developer 2 ✗ End-User 1
Example

**Change Proposal**

It is necessary to add support for handicapped people in each Stop.

Agreement:  ✔️ Developer 1  ✔️ Developer 2  
ACCEPTED

**Solution**

Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"

Agreement:  ✔️ Developer 2  ❌ End-User 1

**Comment**

The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)

Agreement:  ✔️ End-User 1  ✔️ Developer 1
Example

**Change Proposal**

It is necessary to add support for handicapped people in each Stop.

Agreement: ✅ Developer 1 ✅ Developer 2

ACCEPTED

**Solution**

Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"

Agreement: ✅ Developer 2 ✗ End-User 1

**Comment**

The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)

Agreement: ✅ End-User 1 ✅ Developer 1
**Example**

**Change Proposal**

It is necessary to add support for handicapped people in each Stop.

Agreement: ✅ Developer 1 ✅ Developer 2

ACCEPTED

---

**Solution**

Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"

Agreement: ✅ Developer 2 ✗ End-User 1

REJECTED

---

**Comment**

The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)

Agreement: ✅ End-User 1 ✅ Developer 1

ACCEPTED
Example

Change Proposal
It is necessary to add support for handicapped people in each Stop.
Agreement: ✔️ Developer 1 ✔️ Developer 2
ACCEPTED

Solution
Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"
Agreement: ✔️ Developer 2 ✗ End-User 1
REJECTED

Comment
The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)
Agreement: ✔️ End-User 1 ✔️ Developer 1
ACCEPTED
Example

Change Proposal
It is necessary to add support for handicapped people in each Stop.
Agreement: ✔️ Developer 1 ✔️ Developer 2
ACCEPTED

Solution
Stop should include a new integer attribute called "handicap". The representation will include the new keyword "H support" with numeric values
Agreement: ✔️ End-User 1 ✔️ Developer 2

Comment
The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)
Agreement: ✔️ End-User 1 ✔️ Developer 1
ACCEPTED

Solution
Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"
Agreement: ✔️ Developer 2 ✗ End-User 1
REJECTED
Change Proposal

It is necessary to add support for handicapped people in each Stop.

Agreement: ✔️ Developer 1 ✔️ Developer 2

Solution

Stop should include a new integer attribute called "handicap". The representation will include the new keyword "H support" with numeric values

Agreement: ✔️ End-User 1 ✔️ Developer 2

Comment

The handicap support must conform to company regulations and ranged from 0 (no support), to 3 (full support)

Agreement: ✔️ End-User 1 ✔️ Developer 1

Example
Example

Change Proposal
It is necessary to add support for handicapped people in each Stop.
Agreement: ☑️ Developer 1 ☑️ Developer 2
ACCEPTED

Solution
Stop should include a new integer attribute called "handicap". The representation will include the new keyword "H support" with numeric values
Agreement: ☑️ End-User 1 ☑️ Developer 2
ACCEPTED

Solution
Stop should include a new boolean attribute called "handicap". The representation will include the new keyword "H support" with values "true" or "false"
Agreement: ☑️ Developer 2 ✗ End-User 1
REJECTED

Comment
The handicap support must conform to company regulations and range from 0 (no support), to 3 (full support)
Agreement: ☑️ End-User 1 ☑️ Developer 1
ACCEPTED
Example: after

Abstract Syntax

```
IDEElement
  id : String

Vehicle
  0.* follows
    isFollowedBy 0..1

Route
  0.* isContained
    contains 0..*

Bus
Tram

Stop
  lat : double
  lon : double
  handicap: integer
```

Concrete Syntax Example

```
tram 1: route A: stop 001: route A; stops : 001, 002; lat: 23.1082 lon: 12.9883
... ... H support: 3
... ...
```
Discussing the abstract syntax
Discussing the concrete syntax
Representing collaborations
Environment

collaboro.examples
  companies
  productionSystem
  transport
    example1.transport.xmi
      transport-evolved.textual.ecore
      transport.graphical.ecore
      transport.graphical.history
      transport.graphical.notation
      transport.textual.ecore
      transport.textual.history
      transport.textual.notation
      ModiscoWorkflow.ecore
      ModiscoWorkflow.history
      ModiscoWorkflow.notation

platform/resource/collaboro.examples
  Transport System
    Tram T1
    Route R1
    Stop stop1
    Stop stop2
    Stop stop3

tram T1:
  route R1
  Stop stop1, stop2, stop3
stop1
stop2
stop3
Not enough...

- Engagement is limited
  - End-users are required to express changes at high-level of abstraction
  - Solution: Example-driven collaboration

- Collaboration strategies
  - How to adapt the collaboration protocol?
  - Solution: Mechanism to define a democratic process

- Good notations
  - What is exactly a good notation?
  - Need of experimentation on this field

- Semantics
  - What happens with semantics?
  - Solution: Mechanisms to make easier the discussion about semantics
Example-driven Bottom-up

1. Import
2. Fragment-n
3. Induction
4. Manual refactoring
5. Conformance checking & reporting
6. Compilation into implementation meta-models

Purpose of use
- EMF
- Model-to-model Transformation
- Meta Depth
- Model-to-model Transformation

Technical Spaces
- Visual Language
- Textual Language
- Visual Language
- Textual Language
The DSL case - During
Automatic testing of the user experience (interactive)

- Specially for concrete syntaxes

- Reuse what we know from web interaction and design
  - Small changes can make a huge difference
  - Even seasoned designers fail to predict upfront what would work
  - Different user profiles may require different concrete syntaxes

- What about A/B testing for DSLs?
Automatic testing of the user experience (interactive)

- Which syntax for expressing transformation rules is better?
  - It’s up to the users to choose!!!

**Rule newMachine**

- **Extended (declarative) notation**
  - LHS:
    - c1 → m1
    - c2
  - RHS:
    - c1 → m1
    - c2 → m2

- **Compacted (operational) notation**
  - Equivalent:
    - c1 → m1
    - c2 → m2
The DSL case - After
Corpus analysis (post mortem)

- Analysis of repositories of DSL models (i.e. instances of the DSL under analysis)

- We can analyze:
  - (meta) classes that are never used <- irrelevant?
  - Clusters in the DSL <- two subDSLs?
  - Complex structures (clones) that appear often <- is the DSL missing an important element?
Corpus-based DSL Analysis

Domain specific languages (DSLs) – Languages tailored for a specific domain

Motivation

Has received limited focus compared to, for example, the implementation phase†

Our Focus

 DSL Quality Evaluation

Transformation Results Analysis
Individual Model Analysis
Runtime Analysis
Corpus Analysis
User Feedback Analysis

†Gabriel et al. (2010) Do software languages engineers evaluate their languages?
Corpus-based DSL Analysis

Clone Analysis
Identify duplication within the language

Relationship Analysis
Identify metamodel element relationships

Instance Analysis
Identify metamodel element usage

DSLs under evaluation/consideration:

- OCL