Lightweight (Meta-)Modelling: MDD? We’re doing it wrong!

Sébastien Mosser, Philippe Collet, Mireille Blay-Fornarino, WMM’18
Sébastien Mosser
"geek, snowboarder & composition-driven guy"

- 12-....: Ass. Prof.: Soft. Composition
- 11-12: RS: CloudML & IoT / CPS
- 10-11: Postdoc: Business processes, SPL
- 2010: PhD: Soft. Composition & SOA
Who am I are WE?

Sébastien Mosser
Mireille Blay-Fornarino
Philippe Collet
Cyril Cecchinel
Ivan Logre
Benjamin Benni
Sami Lazreg
Clément Duffau
Model-Driven Development?

We’re doing it wrong!
Once upon a time
Teaching Reverse-Engineering @UCA

Reverse-Engineering

Maintenance

Software Evolution

(aka “RIMEI” in French)
Students are asked to answer questions

Does TDD improve Code quality?

How the Docker community follows the engine evolution?

...
In the MDE kingdom
1. Create a meta-model
2. Implement model transformations
3. Create a model-driven workbench
4. Use it
Dockerfile(s) → Parser → Analyser → Classifier

Dashboards → Users

Tools: BASH, Python, Docker
No models, No transformations
Are we such terrible supervisors?
Who are we?
Software developers!

What do we want?
To develop software!
Model-Driven Development?

We’re doing it wrong!
Who cares about models?

Devs care about services!
(or libraries)
Structure ≠ Behaviour

As a dev, one assemble services (or libraries) to solve a particular problem
The Service neighbour
Service Integration is **not** about designing beautiful systems.

It is about **dealing with existing** and **crappy** ones.
Integrating (Distributed) Systems

- portal
- shared functions
- data
- service bus

[eip]
Integrating (Distributed) Systems

- Portal: Weak
- Data: Dangerous
- Shared Functions: So-so
- Service Bus: Mature
Let’s zoom on the dangerous one...

\[
\text{model'} \quad \text{transformation} \quad \text{model}
\]
Model-Driven Development?

We’re doing it wrong!
Express deployment descriptors

Create “stuff” one can query

Run queries, aggregate results

Store

Display results

[SAC’18]
And they lived happily ever after!
Focus on models if you like so
Do not decide for devs how they should work
Provide a service, not a structure.
Integrating Model-based approaches as services

Legend

Domain
Service
MMConcept
Operation

Domain d
Domain d'

Operation

Bus
Engine

\{ \begin{align*}
    \text{Op1} & \Rightarrow \text{d.Op1} + \text{d'.Op4} \\
    \cdots & \Rightarrow \cdots \\
    \text{OpN} & \Rightarrow \text{d.OpN} + \cdots^*
\end{align*} \}

\text{operation1 (messageOp1) : answerOp1}
\text{operation2 (messageOp2) : answerOp2}
\text{operation4 (messageOp4) : answerOp4}
\text{operation5 (messageOp5) : answerOp5}

MM

MM'

\text{part of Logre’s PhD, 2017}
Let’s all give it up for Monolith, who is about to retire. I’d also like to take this opportunity to proudly introduce...

THE MICROSERVICES

CLAP CLAP CLAP CLAP CLAP CLAP HUHUUU CLAP CLAP CLAP CLAP YEAH CLAP CLAP CLAP YEAH HUHUUU YEAH

Gimme five, buddy.

Talk to my API gateway please...

I’m not sharing a database with you!

Well, they might need some ORCHESTRATION first.

Daniel Stori {turnoff.us}