Exploiting the Internet of Things to teach DSLs and Modeling

S. Mosser, P. Collet, M. Blay-Fornarino
Université Nice - Sophia Antipolis, I3S
MDE Teaching, Graduate level

Mireille Blay-Fornarino

Philippe Collet

Software engineering

SPL (variability)

Model-driven engineering (generation + running project)

Model-driven engineering (profile, embedded)

Factories

Sébastien Mosser

Julien DeAntoni
This is a meta-course where we meta-do-not-understand any meta-thing.

- Student feedback, 2011.
(Brief) History
Models

Meta-models

Model transformation

Project: Class-based code generation
It would be much more interesting to read about MDE scenarios that don’t involve the infamous UML2Java transformation.

- Lies, Damned Lies and UML2Java [Paige&Rose]
Sébastien Mosser, Ivan Logre, Nicolas Ferry, Philippe Collet. “From Sensors to Visualization Dashboards: Need for Language Composition” in (GEMOC 2013, co-located with MODELS 2013), IEEE, pages 6, Miami, USA, 29 sept 2013

Data collected from sensors

Model graphical dashboards
Franck Fleurey's bike
Pleading Guilty. Again.

Students overwhelmed by Eclipse

"The tools are ugly, but the project was cool."
- Student feedback, 2012

KEEP CALM AND WALK THIS WAY
Objectives

01 Emphasize the benefits of using models on non-trivial cases

02 Support hands-on lab sessions

03 Decouple project description from tools
ArduinoML
(since 2013)
Brice Morin, Franck Fleurey, Arnor Solberg

Modeling systems

Internet of Things

DSL

ArduinoML
Arduino Uno + Electronic Bricks (~ $70)
What the user wants to express

(a) State Machine

(b) Decision Table

<table>
<thead>
<tr>
<th>Button</th>
<th>LED</th>
<th>ON</th>
<th>OFF</th>
<th>ON</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
</tbody>
</table>

when \( \text{button} \land \text{led} \) do \( \text{switch(led, off)} \)

when \( \text{button} \land \neg \text{led} \) do \( \text{switch(led, on)} \)
int state = LOW; int prev = HIGH;
long t = 0; long debounce = 200;

void setup() {
    pinMode(8, INPUT);
    pinMode(11, OUTPUT);
}

void loop() {
    int reading = digitalRead(8);
    if (reading == HIGH && prev == LOW && millis() - t > debounce) {
        if (state == HIGH) {
            state = LOW;
        } else {
            state = HIGH;
        }
        time = millis();
    }
    digitalWrite(11, state);
    prev = reading;
}
ArduinoML

- DSL
- Meta-model
- Code Generation
- V&V

IoT apps
Step #1: ArduinoML Kernel (3 weeks)

Write Arduino programs to understand the domain. Build a meta-model to support the task. Feedback from staff supports Design Models.
Step #2: Complete Language (5 weeks)

Design the **language** associated to the meta-model

Add new **features**

**Validation** with **domain users**

**Code generation**

**Validation** with micro-controllers
Coverage
We cannot cover every MDE topic for each student. Not enough time.

Choices must be made
Our choice about choices

Let the students choose!

We selected a few features that cover basic MDE topics. Others were "a la carte"

16 features available
Domain Modeling

Model execution frequency
(introduce time)

Support digital and analogical bricks
(complex transitions, thresholds)

when \( tp > 54^\circ C \), activate the buzzer

checker frequency: 2Hz
Domain-Specific Language

DSL to express queries, e.g., missing states (models querying)

Morse Alphabet compiled to executable signal (DSL → DSL mapping)

Konami Code recognizer (DSL & abstraction level)
Constraints, V&V

Specify **special rules** to check a model, e.g., LED must not be used for visually impaired people (on-demand constraints)

Project a model into a **formal specification** language, e.g., TINA (temporal model checking)
Model Transformation, Code Generation

Execute **two apps** on the same board without **interference**

*(model composition)*

Produce the **blueprint** associated to an app

*(doc. generation)*
Variability

The project is *variable by itself*: "a la carte" features

Model **hardware kits** with different contents
(hardware / software variability)
Conclusions & Perspectives
Feedback from 2013 Fall term

Some extensions do not work well together and implies to work more than the other groups

The workload is really important to implement the complete assignment, my group was badly organized and the final rush was tough

As the project focuses on a single domain, if you do not like the case study, you spend the whole course working on something you do not really appreciate
But mainly: strongly positive feedback!

interesting, useful and not boring

very interesting project

DSLs are fun!

a very good way to bind theory to practice

The extension-based approach allows us to focus on what we found interesting in the domain
New term started last monday

Evaluate more precisely the impact of the project on student’s evaluations

http://www.i3s.unice.fr/~mosser/teaching/dsl/start