Practicing DSLs: From Code to Models

Laure Gonnord / Sébastien Mosser
Lyon 1, LIP / UCA, I3S
EduSump’18, 16.10.2018
Exploiting the Internet of Things to Teach Domain-Specific Languages and Modeling
The ArduinoML project

Sébastien Mosser, Philippe Collet, and Mireille Blay-Fornarino
Univ. Nice Sophia Antipolis, I3S, UMR 7271, 06900 Sophia Antipolis, France
{mosser, collet, blay}@i3s.unice.fr

Practicing Domain-Specific Languages: From Code to Models

Laure Gonnord
Univ Lyon, Université Claude Bernard Lyon 1
LIP, CNRS, ENS de Lyon, Inria
F-69342, LYON Cedex 07, France
Laure.Gonnord@ens-lyon.fr

Sébastien Mosser
Université Côte d’Azur, CNRS, I3S, France
Sophia Antipolis
mosser@i3s.unice.fr
First things first: Why a new version?

Thanks to the national railway company for their strike, leading to this situation.
First things first: Why a new version?

Thanks to the national railway company for their strike, leading to this situation.
Terrible news!

Students consider Modelling as Meta-bla-bla-bla-bla

Complicated tools

Compilation ≠ parsing

MDE is not “UML2RDMS”

Who cares about the UML in real life situations?
Terrible news!

Students consider Modelling as Meta-bla-bla-bla

Complicated tools

Who cares about the UML in real life situations?

MDE is not “UML2RDMS”

Blatant Advertisement:

Survey paper this afternoon!
Objectives: Make modelling fun again!

1. Abstract code into models;
2. Operationalise model;
3. Concepts ≠ Tools;
Finding the Right Application Domain

- The domain cannot be a “toy example”
  - The domain should not be too much complicated
- The domain should illustrate nicely different modelling approaches
  - The domaine should also illustrate common challenges in compilation

Reactive systems & Micro-controllers
Approximate cost = 30€
“Clean” version
Devil’s plan: Simple → Complicated

Provide a blinking LED example

Led + Button / On & Off

7-segment counter / reset

The altogether!
A journey into abstraction levels

Step-back
questions
Where do we start?

```c
void change_led_state()
{
    PORTB ^= 0b00100000; // Change digital 13 on->off->on (xor is life!)
}

void init(void)
{
    // DDRB is the configuration register for digital 7 to 18
    DDRB |= 0b00100000; // Digital 13 "outputmode"
    // TODO : enable write for digital 1 to 7 (7seg)
    // TODO : initialize global state values
}

int main(void)
{
    init();
    while(1) // infinite loop
    {
        // display_7seg(0); // uncomment to test the 7-seg when DDRD is configured
        change_led_state();
        _delay_ms(1000); // 1Hz period
    }

    return 0;
}
```
```c
int main(void)
{
    setup();
    while(1)
    {
        // displayDigit(0);
        change_state_led();
        _delay_ms(1000);
    }
    return 0;
}

void change_state_led()
{
    if (led_on){
        digitalWrite(led, LOW);
    } else
        digitalWrite(led, HIGH);
    led_on = !led_on;
}
```
Programming a model

```c
int main(void)
{
    setup();
    state_on(); // initial state
    return 0;
}

void state_on() {
    digitalWrite(led, HIGH);
    _delay_ms(1000);
    state_off();
}

void state_off() {
    digitalWrite(led, LOW);
    _delay_ms(1000);
    state_on();
}
```
Modelling a Program
Tooling the model: Composing Apps

• Challenge: Express the LED example and the Counter one separately
  • Compute the expected one instead of programming it

• How to create such a composition?
  • All the FSMs at the very same place (runtime composition semantic)
  • Creating an app containing all the FSMs separately
  • Merging the FSMs to assess global properties
  • ...

Designing a DSL
Course Outline

Please read the paper published at EduSymp'18 (workshop of the MODELS conference) that describe the course: Practicing Domain-Specific Languages: From Code to Models, L. Gonnord and S. Mosser.

Phase #1: ArduinoML as a sandbox (Padawan)

- Lab support: ArduinoML at * levels of abstraction

- Week 37: Friday, 1:30PM -> 5:30PM
  - Lecture (3h): Introduction to Model-driven Engineering & Domain-specific Languages (SM)
  - Lab (1h): Kickstarting the ArduinoML lab

- Week 38:
  - No Lecture
  - Supervised lab

- Week 39: Friday, 1:30PM -> 5:30PM
  - Lecture (2h): Implementing a DSL, Using Groovy to implement embedded DSL (SM)
  - Lab (2h): Presentation by students of their lab status (steps 1 -> 4) + lab work

- Week 40:
  - Supervised lab

- Week 41: Friday, 1:30PM -> 5:30PM
  - Lecture (1h): Verification & Validation, Lustre as example.
  - Lab (3h): Presentation by students of their lab status (steps 5 + one another according to student's tastes) + lab work

- Week 42:
  - No Lecture
  - Supervised lab

- Week 43: Friday, BAM -> 10AM
  - Evaluation (2h): Final presentations for the ArduinoML project

Phase #2: Sensor Simulation engine (Master)

- Lab support: Sensor Simulation Language
Implementation (grad)

- ~10 students
- Research-oriented institution
- Practical labs + bibliographic study

- >40 students
- Software Engineering
- 8 weeks project + exam
Key takeaways: No Pain, No Gain

Starting at a low level is “painful”

Pain helps to accept the modelling overhead

Students defend the MDE approach at the end!
Practicing Domain-Specific Languages: From Code to Models

Laure Gonnord
Univ Lyon, Université Claude Bernard Lyon 1
LIP, CNRS, ENS de Lyon, Inria
F-69342, LYON Cedex 07, France
Laure.Gonnord@ens-lyon.fr

Sébastien Mosser
Université Côte d'Azur, CNRS, I3S, France
Sophia Antipolis
mosser@i3s.unice.fr