Industrial Experiences on Domain-Specific Modeling
Contents

- Domain-Specific Modeling Languages: Introduction
- Review of industry cases
  - Touch screen
  - Home automation control
  - Sports computer
  - Military radio
- Summary and discussion
A rise in productivity is overdue

- "The entire history of software engineering is that of the rise in levels of abstraction"

- New general-purpose programming languages have not increased productivity

- Abstraction of development can be raised above current level...

- ... without losing control or accepting substandard results

*Software Productivity Research & Capers Jones, 2002
What data is available behind the statements* like:

"5-fold productivity increase when compared to standard development methods"

"The quality of the generated code is clearly better, simply because the modeling language rules out errors"

"The DSM solution makes development significantly faster and easier than the old manual coding practices"

* source: www.metacase.com/cases/
Four cases in more detail

- Touch screen device (Panasonic)
- Home automation (Ouman)
- Sports computer (Polar Electro)
- Military radio (Elektrobit)
Case 1: Panasonic’s touch screen devices (Safa, 2007)

- Home automation solution installed by construction firms
- Features for controlled
  - Lights
  - Heating
  - Air-conditioning
  - Electricity
  - Alarms
    - Burglar
    - Gas
    - Smoke
  - Reporting (energy saving)
Evaluation method

- Compare DSM to current manual coding practice
- Two evaluation methods
  1. Implemented an existing product with DSM
  2. Implemented the existing features to a different target platform
     - The same models, a new generator
Tool-chains covering:
- Generate
- Compile
- Upload
- Boot
- Run

Generators for:
- PC simulation
- Touch screen
- Microcontroller
Measuring development time

- Build the same application with both approaches
  - Measure the time used

- Results:
  - DSM is 425% faster

- Implemented the same product to a new platform
  - 3 days for generator development
  - 0 days for modeling
Return on investment: Panasonic

- DSM solution developed in 15 days
- Product development with:
  - the current approach: 17 days
  - DSM: 4 days
Case 2: Home automation (Puolitaival, 2011)
Home automation, remote control via mobile phone

- Ouman manufactures home automation systems
- Products are focused mainly on temperature control with many different heating systems
- SMS interface for remote control
A language for specifying remote control applications
One model for Ouman EH-60 product remote controller
Return on investment

- DSM solution developed: 2 weeks
- Product development: 1-2 days
  - Comparison to earlier development effort not possible since outsourced
    - Cost was 6 figure number
- DSM allows a non-programmer to develop applications
Case 3: Polar’s Sport computers

- Heart rate measuring, analysis and visualization
- Calorie calculation, like current, cumulative, expenditure rate, active time
- Speed: current, average, maximum
- Distance, based on interval, trip, recovery
- Altimeter, vertical speed, altitude alarms, slope counter, graphical trend
- Cycling information like pedaling rate and cycling power
- Barometer, pressure drop alarm, graphical trend
- Exercise diaries
- Sensor connectivity (heart rate, speed, cadence, power, GPS)
- Compass, Temperature, Odometer, Logbooks, etc.
About the product development (Kärnä et al. 2009)

- Polar focused on UI application development
  - Single largest piece of software
    - Takes 40-50% of the development time
    - Typically always vary among products

- Software development is constrained by limited resources:
  - Memory, processor speed and battery life

- Polar created the needed languages and generators internally
Sample of UI application design
Evaluation methods

- Compare the use of DSM and the current practice

- Two research methods
  - Laboratory study
    - 6 current developers, 6 implementations
    - Implement a small, typical feature
  - Pilot project
    - Implement large portion of a whole product
    - 1 person
Results of the studies

- Laboratory study
  - Measuring time: at least 750% faster
  - Asking opinions: results (scale 1-5, 5 best):

- Pilot
  - Measuring time: >900% faster
Return on investment: Polar

- DSM solution developed in 60 hours
- Product development with:
  - the current approach: 23 days
  - DSM: 2.3 days
Case 4: Military radio
(Puolitaival et al., 2011)
EB Tough VoIP Features

- Tough VoIP is a wired phone that is using UDP/IP network for connection
- Manufacturer: Elektrobit
- Main features:
  - Easy configuration
  - Point-to-Point call
  - All call
  - War-proof device
Testing problem
Two language solution

- Modeling one test case
- Generating one test case
- Executing the test case
- Modeling a test logic
- Model-Based Testing generates multiple test cases
- Executing test cases
Model example 1: test case
Model example 2: test logic
Return on investment: Elektrobit

- About 10 times faster with modeling
- Set-up time estimation:
  - 2 weeks for the first version, +1 week to make it better

Other benefits:
- Test coverage dramatically increase
- Easy test configuration
Economics of DSM

- **Repetition:**
  - # of product variants
  - # of similar features
  - # of developers
  - "outsourcing" to domain experts

- **Investment:**
  - Effort needed to implement DSLs
Steps for implementing DSM

1. Concepts
2. Rules
3. Symbols
4. Generators
Summary

- Domain-Specific Modeling languages provide:
  - Better productivity
  - Quality improvements
  - Easier use and introduction of new developers

- MetaEdit+ makes moving to DSLs feasible
  - Expert can focus on language design, not on creating tooling
  - Models update and work when languages changes

- Creation of languages does not take much time!
More details on the cases

Thank you!

For more cases, customer stories, testimonials visit www.metacase.com