Model Driven Engineering in High Tech Industry

Mark van den Brand

LUCAS-dagen, Lund, October 2016
Short introduction of myself

- 1992 - 1997: Assistant professor at UvA (NL)
- 1997 - 2005: Senior researcher at CWI (NL)
- 2006 - now: Full professor TU/e (NL)
  - Chair of Software Engineering and Technology (SET)
  - SET focuses on **Model Driven Software Engineering:**
    - Domain specific language design
      - dynamic semantics
    - Analysis of model transformations
      - metrics and visualization
      - verification
    - Modeling of Functional Safety in Automotive Domain
  - Industry motivated research
Introduction

• Software Engineering and Technology (SET) group has strong cooperation with High Tech industrial partners:
  • ASML (lithography systems for the chip industry)
  • DAF ((long distance heavy) trucks)
  • Océ (document handling)
  • Philips Healthcare (medical equipment: MRI, CT, X-ray, invasive surgery)
  • VanDerLande Industries (luggage handling, warehouses)
Introduction

Moore’s law for ASML software development

Software effort doubles every new generation

500my → 1000my → 2000my → ?

We need to stop this trend, it is not sustainable
Introduction

ASML software in numbers

- TWINSCAN software consists of
  - 40 million lines of code
  - 2000 components, 60,000 files

- 20+ computing nodes running more than 200 processes

- 1000+ engineers

- 40 million lines of code corresponds to several thousand ManYears of work
Introduction

- Software evolves, continuous growth in:
  - size of software (amount of LOC)
  - complexity of software
  - features in (software) systems
  - costs to build software
  - number of programming languages in software systems

- Software has become big data
The increase in software has raised:

- the need for efficient software development
- the awareness with respect to legacy software
- the opportunities of virtualization
Introduction

• Models are common practice when designing mechatronic systems
  • hardware
  • electronic design
  • physical models
  • Matlab/Simulink models

• Software has proven to be crucial but at the same time a challenge

• Model driven engineering has become very popular
Introduction

• Model driven engineering
  • considers models as first class citizens
  • increases level of abstraction because of the use of models
  • offers the choice between general purpose modeling
    languages or domain specific languages
    – the first may lead to a vendor lock-in
    – the second may involve a huge investment in language design, implementation, and tooling
Introduction

• A few (research/engineering) challenges:
  • Identification of common semantic concepts in a certain domain:
    – High-Tech Industry
      – real-time, state machines, supervisory control, material flow (paper, wafers, luggage), etc.
    – Capturing these concepts in semantic based workbenches
  • Quality and correctness of model transformations wrt
    – property preservation
    – underlying semantics
  • Modularity of meta-models and composition semantic building blocks
  • Evolution of meta-models and co-evolution of models
  • Mixing multiple domain specific languages
What is a Domain Specific Language (DSL)?

- A DSL is a formal, processable language targeting at a specific aspect of a system.
- Its semantics, flexibility and notation is designed in order to support working with that aspect as efficiently as possible.
- “A language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain”
Domain Specific Languages

- DSL design alternatives:
  - Use an existing language “as is”
    - add domain specific functionality via libraries
  - Adapt/extend an existing language
    - introduce new language constructs
  - Develop a new language from scratch
    - syntax, semantics and tooling
Domain Specific Languages

DSL meta-model

- code generation
- model transformation

Execution platform (C, Java - code)
Domain Specific Languages

Model transformations and code generation

C-code
Domain Specific Languages

- Model driven software engineering offers
  - Reduction of development time via increase of abstraction
  - Increase of robustness via verification of models and model transformations

- However,
  - efficient design of domain specific languages and corresponding tooling has to be implemented
Meta-modeling

- ASML has
  - 22 DSLs built using EMF + OCL
  - 5500 models created using the meta-models

- Two of these DSLs in more detail:
  - VPDSL
    - model driven engineering meets virtualization
  - LACE
    - exercise in defining the dynamic semantics
Domain Specific Languages

Software in a Loop Simulation

Test cases

Software

real function calls to influence actuators, query sensors

Simulators

replies on sensor values and positions, with realistic data

TWINSCAN

actuators, sensors
Domain Specific Languages

- **TWINSCAN Platforms and Machine Types**

- **TWINSCAN Platforms**
  - XT platform
    - Main specifications: down to 38 nm, up to 150 wafers/hr
    - Machine types: XT:1950Hi, XT:1900Gi, XT:1700Fi, XT:1450G, ...
  - NXT platform
    - Main specifications: below 38 nm, above 175 wafers/hr
    - Machine types: NXT:1950i, ...
  - NXE platform
    - Main specifications: below 27 nm, above 60 wafers/hr
    - Machine types: NXE:3100, ...

How to create wafer-flow simulator efficiently for this product family?

Common and unique HW (& SW) modules to maintain material (wafer) flow.
Domain Specific Languages

How to Build a Wafer Flow Simulator?

…try to re-create a virtual replica of the real world…

Actuators  Sensors  Holders  Transportable Material  Operator  Interactions  …

common and unique simulation ("virtual") modules to maintain material (wafer) flow
Domain Specific Languages

How to Build a Wafer Flow Simulator?
Error Scenarios

...however, the real world usually does not “behave” perfectly...

Error-injection Scenarios in Simulation

Transfer without wafer displacement

Transfer with wafer displacement
Domain Specific Languages

Model Driven Engineering Approach: VPDSL

- Test cases
  - Test nominal + non-nominal behavior
- Software
  - Automated + consistent
  - Wafer Flow config. with static variation points
- Hardware Simulation Models (.VPDSL)
  - Domain-specific language to configure wafer flow simulators of different products
  - Eclipse-based Editor + Design Checks
- Simulation Scenarios
  - interacting actuators, sensors
- Configuration: e.g. wafers, wafer size
  - Fault injection: e.g. wafer found/lost, wafer displacement

Configuration (models) are generated and deployed, interacting with hardware simulation models.
Design Workflow

1. Specify VPDSL model
   - X LoC

2. Generate simulation code
   - 12*X LoC
     - (8*X model-dependent)

3. Integrate simulation code with control software

Eclipse-based
Xtext DSL editor

Eclipse-based
Xtend/Xpand DSL to C++

Technologies in target code:
- Boost state machines
- Geometry library
- ASML sw. facilities

Domain Specific Languages

ASML

Technische Universität Eindhoven
University of Technology
Domain Specific Languages
Domain Specific Languages

• Another example of virtualization via a DSL
Questions