Model-Driven Software Evolution: The Delft Research Agenda

Arie van Deursen
Eelco Visser, Jos Warmer
The Delft Software Engineering Research Group

SWERL Software Evolution
- SPCI
- AMR
- TFA
- MoVES
- MoDSE
- Ideals

ESL Embedded Software
- Merlin
- Trader
- Finesse
- Poseidon
- Tangram
The Delft Software Engineering Research Group

SWERL Software Evolution
- SPCI
- TFA
- MoVES
- MoDSE
- Ideals

ESL Embedded Software
- Merlin
- Reconstructor
- Trader
- Finesse
- Poseidon
- Tangram
Software Architecture Reconstruction

Research
- Model transformations
- Software Visualization

Directions:
- Archive splitting
- Repository mining
- Header file remodularization
- Conformance, consistency
Software Architecture Reconstruction

Research
- Model transformations
- Software Visualization

Directions:
- Archive splitting
- Repository mining
- Header file remodularization
- Conformance, consistency
12.5 mloc C; 3-month release cycle
The Ideals Project

Objectives
- Effort reduction
- Lead time reduction
- [Increased dependability]

Means
- Address way crosscutting concerns are handled
- Investigate potential for using aspects
- Analyze migration issues

Some results
- Evaluated clone detection for aspect mining
- Analyzed error handling:
  - 20% of locs
  - two faults per kloc
- ASML investing in (home grown) weaver for C.

See our TSE’05, ICSE’06, AOSD’07 papers
Abstraction

- Abstraction in programming is
  - the process of identifying common patterns
  - that have systematic variations;
- An abstraction
  - represents the common pattern and
  - provides a means for specifying which variation to use.
The MoDSE Project, 2007-2012

- ICT Consultancy, MDE
- ICT Consultancy, OCL, MS-DSL
- ICT Consultancy, MDA
- MDA tool provider
- Finance, problem owner

- Program transformations
- Domain-specific languages
- Sw. evolution & rev. eng.
The MoDSE Project, 2007-2012

- ICT Consultancy, MDE
- ICT Consultancy, OCL, MS-DSL
- ICT Consultancy, MDA
- MDA tool provider
- Finance, problem owner 🇧🇪

- Program transformations
- Domain-specific languages
- Sw. evolution & rev. eng.
2012: And, what about that MDE application?

On line cigar shop

2007: started happily creating models

2008: 1.0 generated from models using MGen, customized using C#, deployed on .NET

2009: new EU cigar selling regulations, new payment system, TNT instead of UPS, adding pipes, ...

2010: Microsoft announces D-flat, MGen taken over by Computer Associates, shop turned into Albert portal, ...

2011: MGen migration to D-flat needed but delayed, portal integration failed, loosing market share, ...

2012: Maintenance night mare
MoDSE (MoVES?)
Looking Back, Looking Ahead

• CASE & 4GL tools:
  – what you liked
  – what you hated

• MDE
  – What you hope
  – What you fear

• MoDSE
  – Work on what you hate & fear ...
Model-Driven Evolution Dimensions
Model-Driven Evolution Dimensions

- Regular
  - Changes expressed in models
Model-Driven Evolution Dimensions

- Regular
  - Changes expressed in models
- Meta-model evolution
  - Improve expressiveness of modeling language
Model-Driven Evolution Dimensions

- Regular
  - Changes expressed in models
- Meta-model evolution
  - Improve expressiveness of modeling language
- Platform evolution
  - Code generators, application framework
Model-Driven Evolution Dimensions

- Regular
  - Changes expressed in models
- Meta-model evolution
  - Improve expressiveness of modeling language
- Platform evolution
  - Code generators, application framework
- Abstraction evolution
  - Further modeling languages for other concerns
Love & Hate (I)

- Generator maintenance nightmare
  - Strengthen meta-programming support
  - Reuse through extensibility
- Model interaction
  - Model-to-model and model-to-code
  - Mechanisms for separation of concerns not yet sufficiently powerful
  - Explicit interfaces & extension points
  - Methods for the identification of overlap and inconsistencies
Love and Hate (II)

• Inappropriate abstractions
  – Search for “patterns that have systematic variations”.
  – Clone detection, data mining, program analysis, software visualization, ...

• Support for “abstraction refactoring”
  – Meta-model changes lead to model migration
  – Creating a new modeling language should be as easy as creating a new class?
Collaboration Forms

- Joint papers
- Joint workshops
- Common (open source) cases?
- PhD student exchange
- Master student exchange
Outlook

- Model-driven software evolution dominant Delft theme in next 5 years
  - Reverse engineering & pattern mining
  - Program transformation
  - Model manipulation
  - Industrial participation
- Existing Belgian connections very fruitful
- Looking forward to MoVES collaborations
More Information

- M. Bravenboer, E. Visser: Concrete syntax for objects: domain-specific language embedding and assimilation without restrictions. *OOPSLA’04*
- T. Reus, H. Geers and A. van Deursen. Harvesting Software Systems for MDA-Based Reengineering. *ECMDA-FA’06*
- Mariano Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, T. Tourvé: A Qualitative Comparison of Three Aspect Mining Techniques. *IWPC’05*
- Bruntink, A. van Deursen, M. d’Hondt and T. Tourvé. Simple crosscutting concerns are not so simple. *AOSD’07.*
- B. Cornelissen, A. van Deursen, L. Moonen, and A. Zaidman. Visualizing Testsuites to Aid in Software Understanding. *CSMR’07*