Formalising Metamodel Evolution based on Category Theory

Florian Mantz\textsuperscript{1}, Alessandro Rossini\textsuperscript{2}, Gabriele Taentzer\textsuperscript{3}, Yngve Lamo\textsuperscript{1}, Uwe Wolter\textsuperscript{2}

\textsuperscript{1}Department of Computer Engineering, Bergen University College, Norway
\textsuperscript{2}Department of Informatics, University of Bergen, Norway
\textsuperscript{3}Philipps-Universität Marburg, Germany

27 October 2011
Nordic Workshop on Programming Theory, Västerås, Sweden
Model-driven engineering
Model-driven engineering (MDE)

- (Diagrammatic) Models
  - reason at a higher level of abstraction
- Metamodels
  - definition of modelling languages
- Model transformation
  - automatic generation of systems
Challenges in MDE

- Complex evolution of models and metamodels
  - Need for techniques and tools to support this evolution
Metamodel evolution
Metamodel evolution

- Metamodels describe modelling languages
- Models conform to the metamodel
Metamodel evolution

- Metamodels evolve
• Models have to be migrated
• Model migration should reflect metamodel evolution
Challenges in metamodel evolution

- Description of metamodel evolution
- Migration of dependent artefacts
  - Models
  - Model transformations
  - Documentation
  - ...
- Lack of formalisation
Diagram Predicate Framework
Diagram Predicate Framework (DPF)

- Formal diagrammatic specification framework founded on category theory and graph transformation
  - Diagrammatic modelling
  - Metamodelling
  - Model transformation
  - Model versioning
  - Deep metamodelling
Metamodelling in DPF

- Models
  - Graph + set of atomic constraints
- Metamodels
  - Graph + set of atomic constraints
- Model-Metamodel Relation
  - Morphisms
Running example
Servers and services

- Simple network metamodel and conforming model
**Servers and services**

- **SecurityPolice** is moved from **Server** to **Service**
- Model does not conform any more to its metamodel
Servers and services

- Model has to be migrated
Formal approach
Idea

1. Specify metamodel evolution change as rule
2. Derive the metamodel migration rule from this rule
Cospan DPO approach

- Metamodel evolution rule as graph transformation rule
Generated model migration rule as an amalgamated rule consisting of a set of isomorphic pattern of the metamodel evolution rule
Application of migration rule

LHS

Model

(PO1)

Model_1

(PO2)

Model'

RHS

n*pattern

(1) (4) (2)

(5) (2)

(3)

(7) (10)

(11) (8)

(9)

(1)

(4) (2)

(5) (2)

(3)

(7) (10)

(11) (8)

(9)

(1)

(4) (2)

(3)

(7) (10)

(8)

(9)

(12) n*pattern

(1)

(4) (2)

(3)

(7) (10)

(8)

(9)

(12) n*pattern
Application of migration rule

- One match of a sub-pattern of the migration rule
- There are as many patterns as possible matches
Application of migration rule

• A second match
A third match
Application of migration rule

- Complete match indicated by mapping numbers
Extension of metamodel evolution rule

- Arrow (5) is deleted in the metamodel
- All corresponding arrows must be deleted in the model
- The migration rule must migrate all elements deleted by the metamodel evolution rule
- The metamodel evolution rule application is restricted using constraints
We create a metamodel evolution rule using the Cospan DPO approach.
The rule conforms to a specification $M_3$
From this rule we derive a model migration rule.
The migration rule is also a Cospan DPO rule.
• We apply the metamodel evolution rule and the model migration rule interleaved to keep models permanently well-typed

• First we extend the metamodel, then we migrate the model and finally we reduce the metamodel
- Model $M_1$ stays well-typed during the whole migration process
The migration rule is represented by an amalgamated rule resembling the metamodel evolution rule.
• All morphisms between metamodel, model, evolution- and migration rule
Conclusion
Future Work

• Improve rule derivation strategy
• Consider further constraints on the metamodel level
• Guide rule derivation where alternative migrations are possible
Thank you!

Questions?

For more information visit: http://dpf.hib.no/