Towards a Structured Workflow Language for Model Management

Sahar Kokaly[1] , Zinovy Diskin[1,2] , Tom Maibaum[1] , Richard F. Paige[3]

[1] NECSIS, McMaster University, Canada *fdiskinz|kokalys|maibaum@mcmaster.ca*

[2] University of Waterloo, Canada *zdiskin@gsd.uwaterloo.ca*

[3] University of York, UK *paige@cs.york.ac.uk*

The Problem

1. *Need for **workflows** in MDE.*
2. *Workflows should be **modelled**.*
3. *Models of workflows should be **structured**.*
4. *Workflow models should use the **appropriate modeling primitives** [1].*
5. *Traceability mappings **first class citizens**.*
6. ***Verification and Validation** of Workflows.*

The Solution: Structured Workflows for Model Management

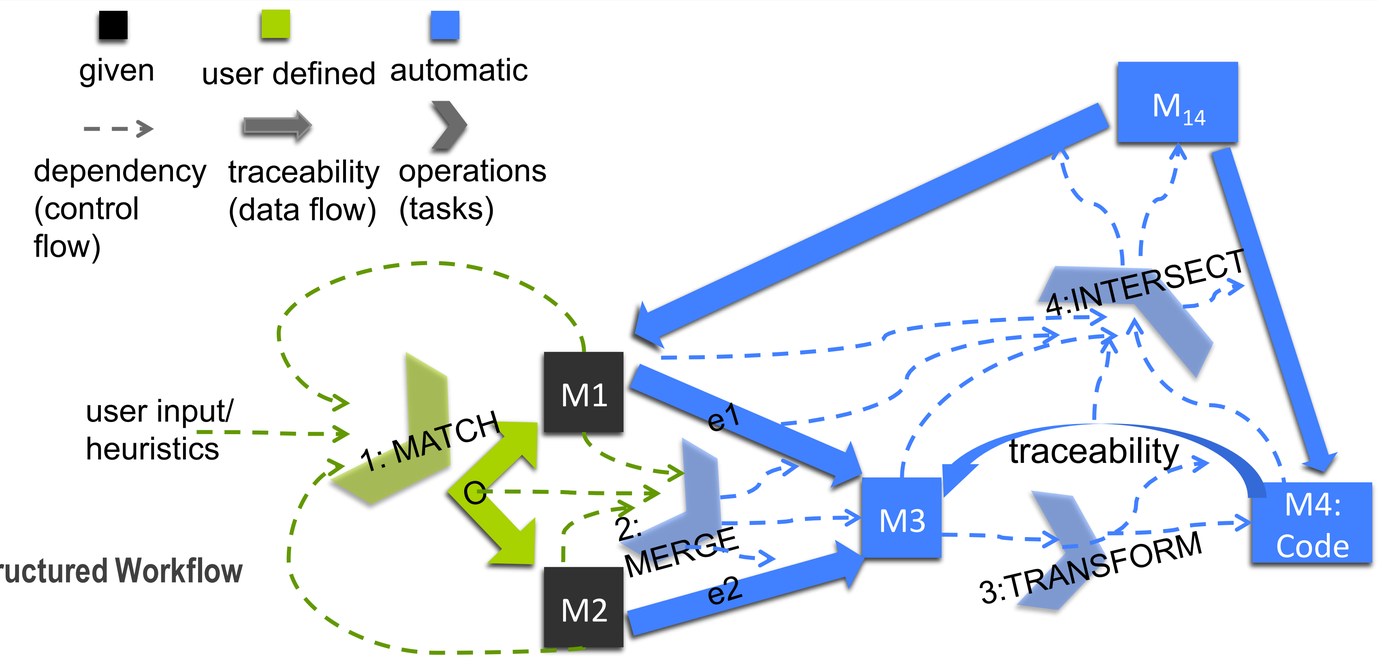

Figure 3: Structured Workflow

An Example

Requirement: "I have two models: M1 and M2. I would like to merge them and then generate code from the result. I would then like to trace back to see what part of the code came from M1."

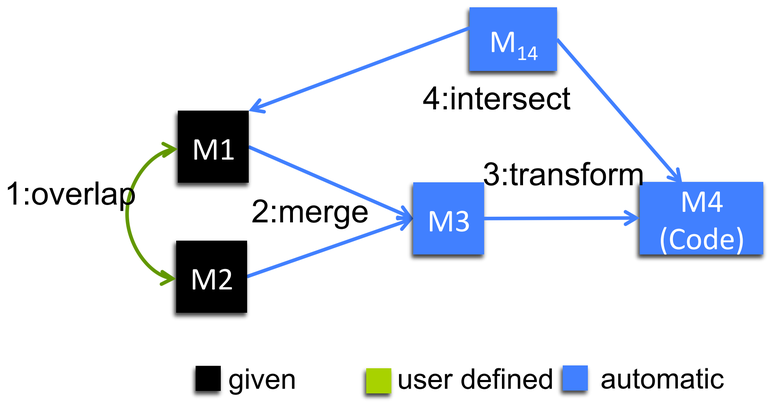

Figure 1: The workflow

```
import models M1, M2
import overlap O=O(M1, M2)
merge (M1,M2) into M3 using O
import transformation t
apply t to M3 to produce Code
Intersect (M1,Code) into M14
```

Figure 2: The code implementing it

The State of the Art

Issues identified in the state-of-the-art of *workflow languages for model management*:

- Modeling
  - not all workflows are modelled
- Expressiveness
  - sequential composition only
  - transformations are the only model management operation considered
- Traceability
  - non-existent or implicit
- Verification and Validation
  - focus on execution and not V&V

Switching Abstraction Levels

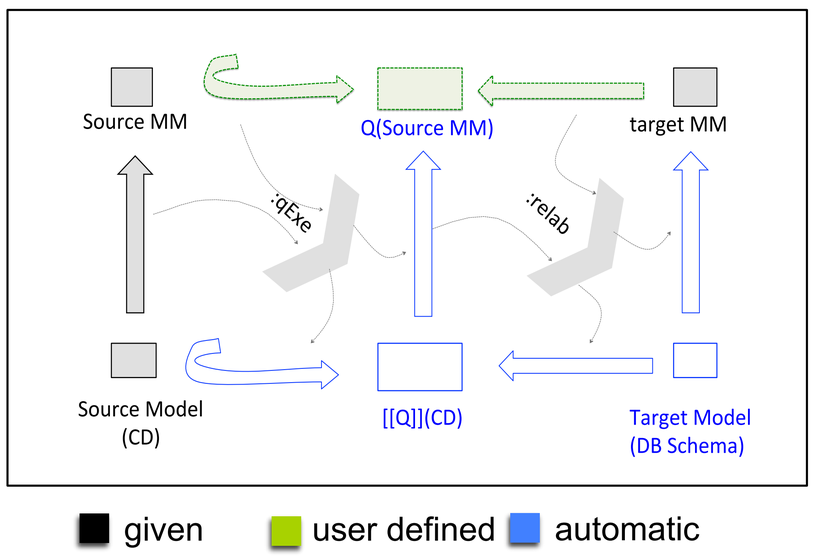

Figure 4: Zooming into the "transform" operation

Typing System

Name	Input Arity	Output Arity
Match	A B	R (p, q) A B
Merge	R (p, q) A B	R (p, q) A r s B C
Transform	A	A t B
Intersect	R (p, q) A B	R (p, q) A r s B C

Figure 5: Operation Types

The Workflow Language

- **Signature of model management operations S_{MMT}**
  - Transform, Merge, Match, Intersect, etc.
- **Signature of workflow combinators S_C**
  - ; , || , Branching , etc.
- Workflow is then a term constructed from S_{MMT} and S_C. *E.g.,*

W = (Match ; Merge ; Transform; Intersect)

Visualization via a DAG

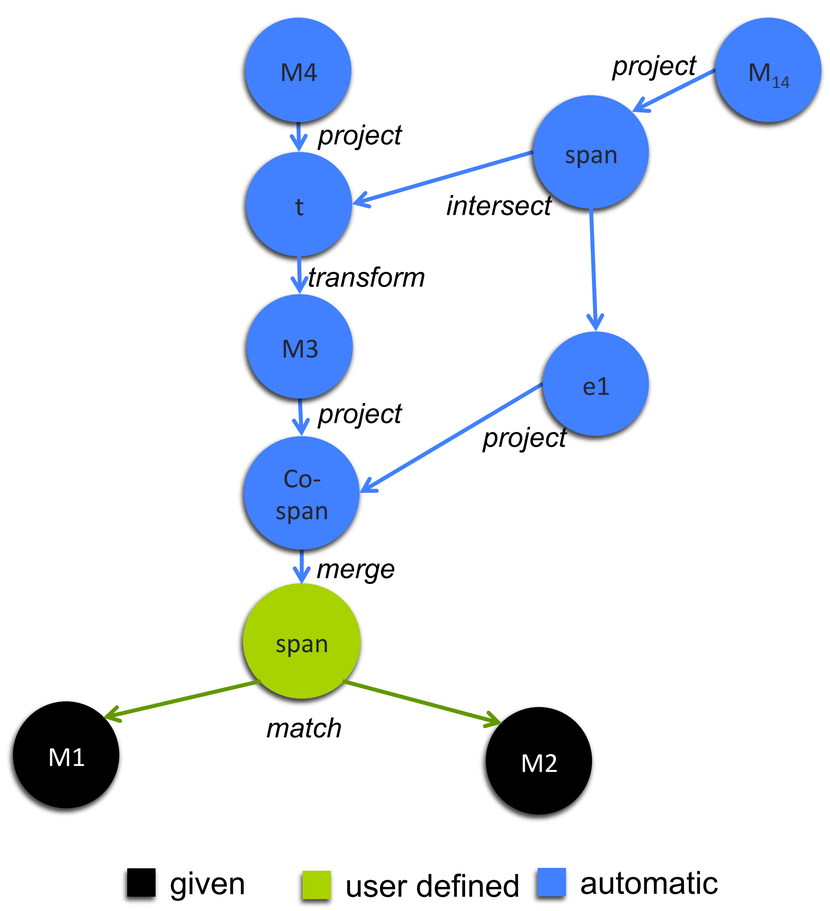

Figure 6: Visualization via a Directed Acyclic Graph for parsing

The Expected Contributions

1. A structured specification of megamodels:
   - Models and mappings as first class citizens modelled via **graphs** and **graph mappings.**
   - constraints on them modelled via **diagram predicates.**
   - operations over them modelled via **diagram operations**.
2. A workflow language for model management that is built via the composition (tiling) of diagram operations.
3. Workflow comparison and optimization.
4. Evaluation of the workflow language and a set of suggested improvements to the state-of-the-art in the area of workflows for model management.

References

1. Diskin, Z., Kokaly, S., Maibaum, T.: Mapping-Aware Megamodeling: Design Patterns and Laws. SLE '13.
2. Lucio, L., Mustaz, S., Denil, J., Vangheluwe, H., Jukss, M.: FTG+PM: An Integrated Framework for Investigating Model Transformation Chains. In: SDL Forum '13.
3. Diskin, Z.: Model Synchronization: Mappings, Tiles, and Categories. In: GTTSE '09

Acknowledgements

This work is part of the NECSIS project funded by NSERC and Automotive Partnership Canada.