A Relationship-Driven Approach to View Merging

Mehrdad Sabetzadeh Shiva Nejati Steve Easterbrook Marsha Chechik
Department of Computer Science
University of Toronto, Canada
{mehrdad,shiva,sme,chechik}@cs.toronto.edu

Motivation

➜ **Collaborative Model-Based Development**

- Models built and manipulated by distributed teams
  - Each team works on a partial view of the system
  - These views may express different perspectives, features, concerns, etc.
- Some reasons for merging partial views:
  - Creating a unified global perspective
  - Exploring interactions between different features
  - Performing different kinds of analysis
    - . . . such as verification and validation

Challenges

- Conceptual Overlaps
  - Views may have concepts in common but these may be represented differently in each view
- Incompleteness and Inconsistency
  - Views may be ambiguous, uncertain, or conflicting
- Evolution
  - Views evolve over time, so merges need to be updated if the source views change
- Property Preservation
  - Merges should preserve properties that are essential to the source views' intended use

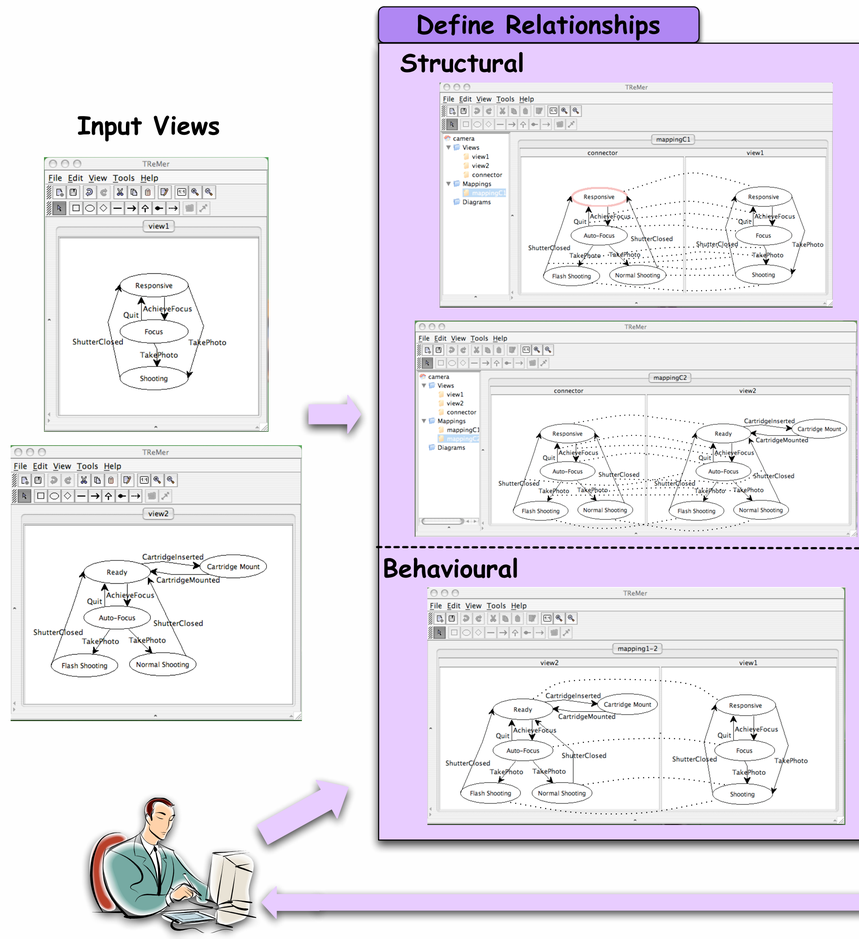
Applications of Merge

➜ **Requirements and Design**

- Merging goal models and use cases
- Merging UML class diagrams
- Merging state-machines and scenarios

➜ **Information Systems**

- Merging Entity-Relationship diagrams
- Merging ontologies



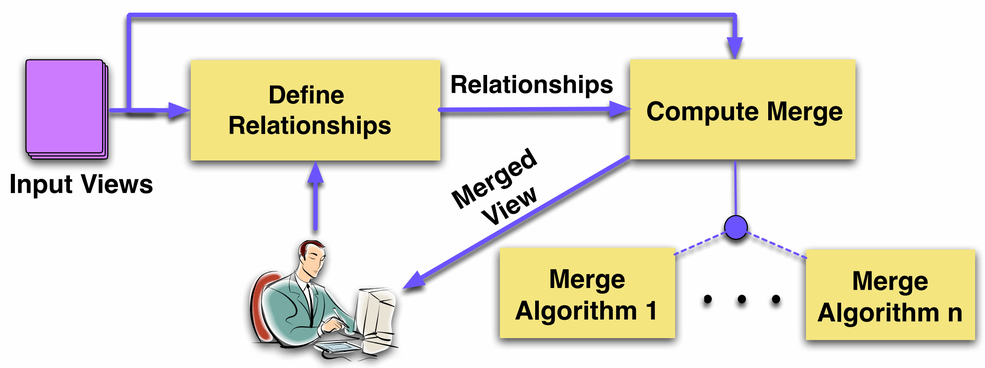
Requirements of a Merge Framework

(1) Allow to hypothesize alternative ways of mapping different concepts
  - Requires explicit means for defining view relationships

(2) Support different merge algorithms
  - These algorithms may require different species of relationships to be built between views

➜ **Our Approach**

We treat relationships between views as "first-class artifacts"



Observations

(1) Merge is exploratory
  - Views are developed by distributed teams
  - Can't be entirely sure how different concepts are related

(2) Merge has different goals at different stages of development
  - Early development: terminology unification, negotiation, making overall design decisions
    - Merge is often "syntactic" because views have loose or undefined semantics
  - Late development: semantic transformation, synthesis, system operationalization
    - Merge is usually "semantic" because views are more rigorously defined

Tool Support

➜ **TReMer**

- A Tool for Relationship-Driven Model Merging

➜ **Components**

(1) GUI for defining views and hypothesizing their relationships

(2) A library of merge algorithms

➜ **Current Support**

- ✎ Structural Merge
- ✎ Behavioural Merge

Structural Merge

➜ **Highlights**

- Is applicable to many graph-based notations
  - e.g., goal models, ER diagrams, class diagrams, simple state-machines
- Employs sub-graphs and structural mappings to define view relationships
- Uses categorical "colimits" for computing merges
- Tolerates incompleteness and inconsistency

Behavioural Merge

➜ **Highlights**

- Is applicable to behavioural models such as
  - Statecharts, Finite State-Machines (FSM's), etc.
- Employs behavioural mappings to define view relationships
- Preserves temporal behaviours of the source views in their merges
- Captures behavioural variabilities between the source views using parameterization

References

[1] S. Nejati, M. Sabetzadeh, M. Chechik, S. Easterbrook, and P. Zave. Matching and merging of statecharts specifications, 2006. Submitted for publication.

[2] G. Brunet, M. Chechik, S. Easterbrook, S. Nejati, N. Niu, and M. Sabetzadeh. A manifesto for model merging. In *1st International Workshop on Global Integrated Model Management*, pages 5–12, 2006.

[3] M. Sabetzadeh and S. Easterbrook. View merging in the presence of incompleteness and inconsistency. *Requirements Engineering*, 11(3):174–193, 2006.

[4] S. Nejati and M. Chechik. Let's agree to disagree. In *21st IEEE/ACM International Conference on Automated Software Engineering (ASE'05)*, pages 287–290, 2005.

[5] S. Uchitel and M. Chechik. Merging partial behavioural models. In *12th International Symposium on Foundations of Software Engineering (FSE'04)*, pages 43–52, 2004.

[6] M. Sabetzadeh and S. Easterbrook. Analysis of inconsistency in graph-based viewpoints: A category-theoretic approach. In *18th International Conference on Automated Software Engineering (ASE'03)*, pages 12–21, 2003.