An Algebraic Framework for Merging Incomplete and Inconsistent Views

Mehrdad Sabetzadeh       Steve Easterbrook

Department of Computer Science
University of Toronto, Canada.

Email: {mehrdad, sme}@cs.toronto.edu

September 2, 2005
Background and Motivation

- What is view merging?
  - Putting a set of views together to produce a new view encompassing all the given views

- Why is view merging important?
  - Creating a unified perspective
  - Exploring interactions between views
  - Performing various kinds of analyses

- What is a “view”?
  - A model delineated in a particular notation
    - . . . can be a stakeholder’s perspective, a feature, a concern, etc.
Applications of View Merging

→ Requirements Engineering
  → Merging goal models
  → Merging behavioral models

→ Databases
  → Schema integration

→ The Semantic Web
  → Ontology integration
Example: Merging $i^*$ views

Overview

Example, revisited

Adding support for traceability

Conclusion
Example: A Meeting Scheduler

→ Problem: Mary and Bob want to elaborate their requirements with the help of an analyst, Sam

→ Initial views of Mary and Bob:

Mary:
- Email requests to participants
- Available dates be obtained
- Agreeable slot be found

Bob:
- Effcient
- Plan meeting
- Send request letters
- Responses be gathered
- Consolidate results
Different Aspects of the Problem

Different vocabularies

Merged View

+ Efficient
Send request letters
Bob
results
Consolidate
Email requests
to participants

Available dates be obtained
Agreeable slot be found

Mary

Efficient
Plan meeting

Send request letters
Responses be gathered
Consolidate results

Bob

Schedule meeting

Meeting requests be sent
Available dates be obtained
Agreeable slot be found

Send requests by snail mail
Send requests by email
Consolidate results

Merged View

Different vocabularies
Different Aspects of the Problem

**Merged View**

**Structural discrepancies**
Different Aspects of the Problem

Different vocabularies

Merged View

Efficient
Send request letters
Bob
results
Consolidate
Email requests
to participants
Mary
Available dates
Agreeable
be obtained
slot be found

Efficient
Plan
meeting

+

Bob
Send request letters
Responses
be gathered
Consolidate
results

Available dates
be obtained

Agreeable
slot be found

Efficient
Schedule
meeting

Meeting requests
be sent

Send request letters
by snail mail

Send requests
by email

Consolidate
results

Mary
Agreeable
slot be found

Conceptual disagreements
Also, need to be able to:

- record how sure stakeholders are about their decisions
- ... and how their decisions can evolve
- ... differentiate between the assumptions made and generated merges
- distinguish between the contributions of different stakeholders to the merged view

Our framework addresses all these ...
Overview of the Approach

Roadmap

- Defining a representation formalism (Views)
- Defining a notion for embedding a view into another (Mappings)
- Specifying how view merges can be hypothesized (Interconnection Diagrams)
- Devising a view merging algorithm

Assumptions

- Specification notations are graph-based
- It is possible to be precise about the areas of uncertainty and disagreement
Each view is represented as a directed graph

View elements are *annotated*

Each annotation

- describes a stakeholder’s *belief* about the element it is attached to
- is a value drawn from a fixed lattice

Lattices provide a flexible framework for modeling incompleteness and inconsistency [Belnap, Ginsberg]
We use a 4-valued lattice for annotations:

- !: *proposed* but not certain to be well-conceived
- ✗: known to be ill-conceived (*repudiated*)
- ✓: known to be well-conceived (*affirmed*)
- ✸: both repudiated and affirmed, hence *disputed*
Annotations – Example
Where are we?

Basic constructs in the framework:

- Views
- Mappings
- Interconnection Diagrams

What is a view mapping?

- An embedding of a view into another
  - Mappings preserve graph structure
  - Mappings respect view annotations:
    - The value annotating the image of an element must be at least as specific as the value annotating the element
View Mapping - Example

A

B

\( n_1 \)

\( e_1 \)

\( e_2 \)

\( e_3 \)

\( n_2 \)

\( n_3 \)
Capturing Merge Hypotheses

→ Where are we?
  ➡ Basic constructs in the framework:
     ✓ Views
     ✓ Mappings
     ↔ Interconnection Diagrams

→ What is an interconnection diagram?
  ➡ A set of views and a set of mappings
  ➡ . . . and describes a merge hypothesis
Correspondence identification

Annotations have been omitted for simplicity

Pattern:
Incorporating a concern

Annotations have been omitted for simplicity

Pattern:
View Merging

→ Where are we?

✓ Basic constructs:
  ➢ Views, Mappings, Interconnection Diagrams

↔ View merging

→ The merge process

↔ Input: An interconnection diagram $D$

↔ Output: A merged view combining all the views in $D$ w.r.t. to the view relationships described by the mappings in $D$
The Merge Algorithm

Intuition:

- Assume all view elements are distinct
- Unify elements deemed equal by the interconnections

View nodes and edges are merged component-wise

... hence, we only need an algorithm for merging annotated sets
Merging Annotated Sets

⇒ Algorithm:

Step 1. Disregard the annotations
Step 2. Merge the resulting plain sets
Step 3. Compute an annotation for each element of the merged set
Merging Sets (Step 2)

\[ A = \{x, y, w\} \quad B = \{x, y, t\} \]

\[ C = \{z, w\} \]

The given interconnection diagram
Merging Sets (Step 2)

\[ A = \{x, y, w\} \quad B = \{x, y, t\} \]

\[ C = \{z, w\} \]

\[ w_C \]

\[ x_A \quad y_A \quad w_A \quad x_B \quad y_B \quad t_B \]

\[ z_C \]

Compute the disjoint union
Merging Sets (Step 2)

$$A = \{x, y, w\} \quad B = \{x, y, t\}$$

$$C = \{z, w\}$$

Connect up the related pairs
Merging Sets (Step 2)

\[
A = \{x, y, w\}, \quad B = \{x, y, t\}
\]

\[
C = \{z, w\}
\]

Find the connected components
Merging Sets (Step 2)

\[ A = \{x, y, w\} \quad B = \{x, y, t\} \]

\[ C = \{z, w\} \]

\{ \{x_A, y_B, z_C\}, \{y_A\}, \{w_A, t_B, w_C\}, \{x_B\} \}

The merged set
Computing the Annotations (Step 3)

- Take the *least upper bound* of the annotations of the elements in each connected component.
  - l.u.b. denotes the least specific (and yet admissible) belief.

**Example:**

```
xA  yA
ωA
zC
wC
xB  yB
wA
xB  tB
```

Computing the Annotations (Step 3)

- Take the *least upper bound* of the annotations of the elements in each connected component
  - l.u.b. denotes the least specific (and yet admissible) belief

Example:
Computing the Annotations (Step 3)

- Take the *least upper bound* of the annotations of the elements in each connected component
  - l.u.b. denotes the least specific (and yet admissible) belief

Example:
Computing the Annotations (Step 3)

- Take the least upper bound of the annotations of the elements in each connected component
- l.u.b. denotes the least specific (and yet admissible) belief

Example:

\[ \text{lub}\{!, \checkmark, \times\} = \checkmark \]
Example – Revisited

→ Where are we?

✓ Basic constructs: Views, Mappings, Diagrams
✓ The merge algorithm
↔ Merging $i^*$ views – an example

→ Initial views of stakeholders

Mary

Bob

Email requests to participants
Available dates be obtained
Agreeable slot be found
Schedule meeting

Efficient
Plan meeting

Send request letters
Responses be gathered
Consolidate results
View Interconnections

Email requests to participants
Available dates be obtained
Agreeable slot be found
Meeting
Schedule
Plan
Meeting
Send request letters
Responses be gathered
Consolidate results

Initial views
View Interconnections

Initial correspondences

Connector I

Schedule meeting
Available dates be obtained

Email requests to participants
Available dates be obtained
Agreeable slot be found

Mary

Bob

Efficient
Send request letters
Responses be gathered
Consolidate results

Plan meeting
View Interconnections

Schedule meeting

Available dates be obtained

Connector I

Mary

Schedule meeting

Email requests to participants

Available dates be obtained

Agreeable slot be found

Bob

Plan meeting

Efficient

Send request letters

Responses be gathered

Consolidate results

Mary Revised

Schedule meeting

Send requests by email

Meetings requests to be sent

Mary Revised

Bob Revised

Revising the initial views
View Interconnections

1. Schedule meeting
2. Available dates be obtained
3. Email requests to participants
4. Available dates be obtained
5. Agreeable slot be found
6. Connectors I

Mary

1. Schedule meeting
2. Available dates be obtained
3. Agreeable slot be found
4. Email requests to participants
5. Connectors II

Bob

1. Efficient
2. Plan meeting
3. Send request letters
4. Responses be gathered
5. Results Consolidate

Mary Revised

1. Schedule meeting
2. Available dates be obtained
3. Agreeable slot be found
4. Meeting requests be sent
5. Connectors II

Bob Revised

1. Efficient
2. Plan meeting
3. Send requests by snail mail
4. Responses be gathered
5. Agreeable slot be found

New correspondences
The Merged View

Result of the merge operation
The Merged View

Merged View

If Bob insisted on his contribution link
The Merged View

... with repudiated elements removed
Stakeholder Traceability

→ **Issue:**

- Merges do not reflect the decisions of individual stakeholders
  - What if we want to attach a priority or credibility factor to each stakeholder?

→ **Solution:**

- Using a more elaborate annotation scheme:
  - Attach multiple annotations to view elements
  - . . . one value for each involved stakeholder
The annotation of each element reflects the conceptual contributions made to it by the involved stakeholders.
Recap

→ Summary

⇒ A formal framework for merging incomplete and inconsistent views
  ➢ independent of any particular modeling formalism
  ➢ customizable to many graph-based notations

→ Limitations and weaknesses

⇒ based purely on syntactic mappings
⇒ still not clear how the identification of interconnections can be automated
Future Work

- Automating the identification of interconnections (ongoing)
- A more comprehensive analysis of traceability concerns (ongoing)
- Adding model-based semantics
- Investigating how the framework can facilitate negotiation during requirements analysis
Thank You!

Questions?

Acknowledgments: John Mylopoulos, Renée Miller, Linda Liu, Shiva Nejati, and the members of the Formal Methods, Database, and EarlyRE groups at the University of Toronto. Financial support was provided by NSERC, MITACS, and BUL.