An Algebraic Framework for Merging Incomplete and Inconsistent Views

Mehdrad 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

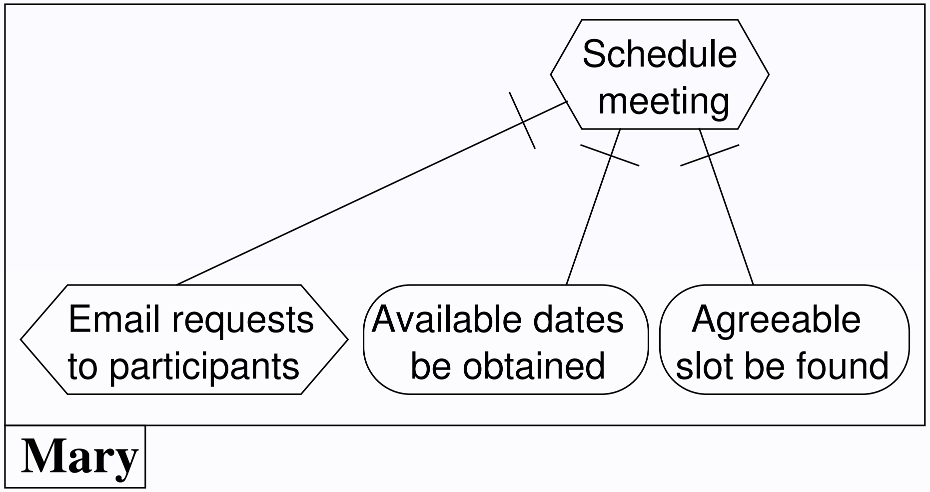
Outline

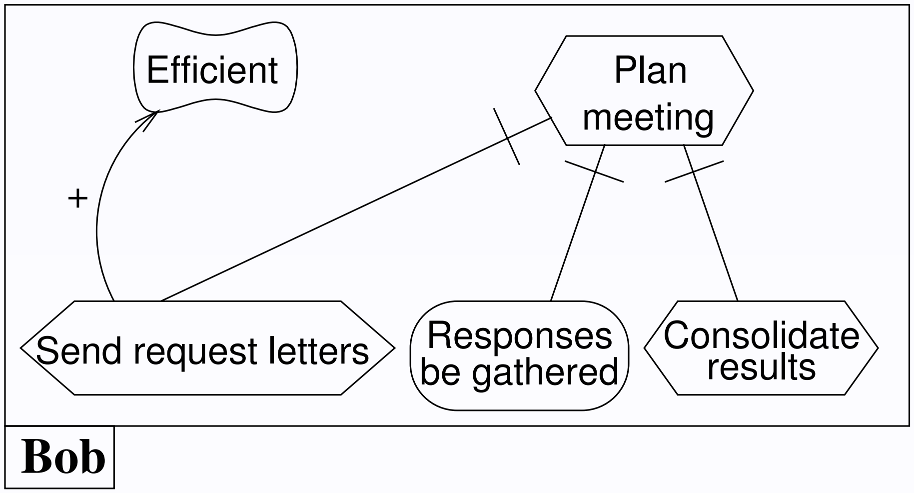
- ➜ **Example: Merging i^* views**
- ➜ **Overview**
- ➜ **Example, revisited**
- ➜ **Adding support for traceability**
- ➜ **Conclusion**

Example: A Meeting Scheduler

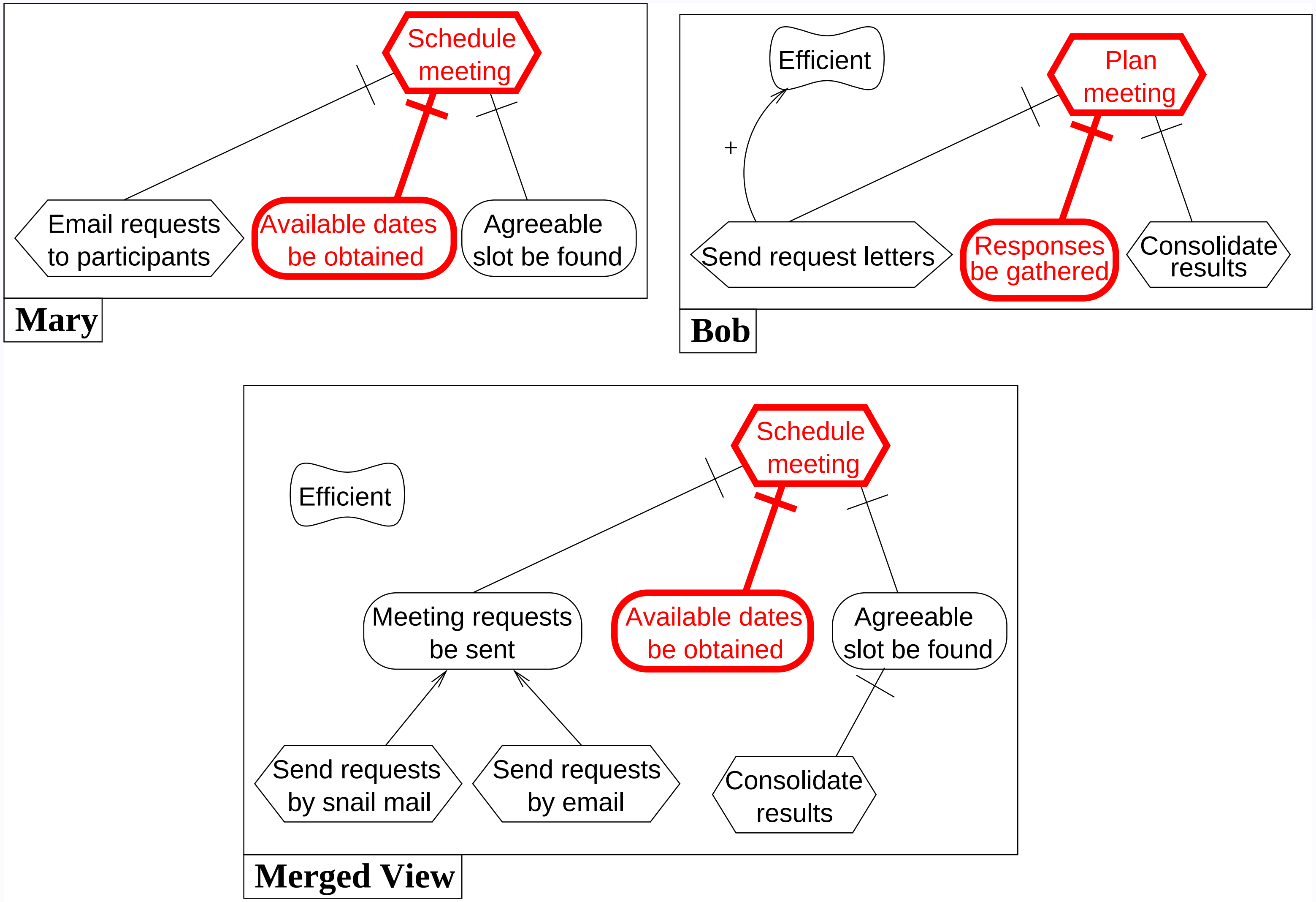
➜ **Problem**: Mary and Bob want to elaborate their requirements with the help of an analyst, Sam

➜ **Initial views of Mary and Bob**:



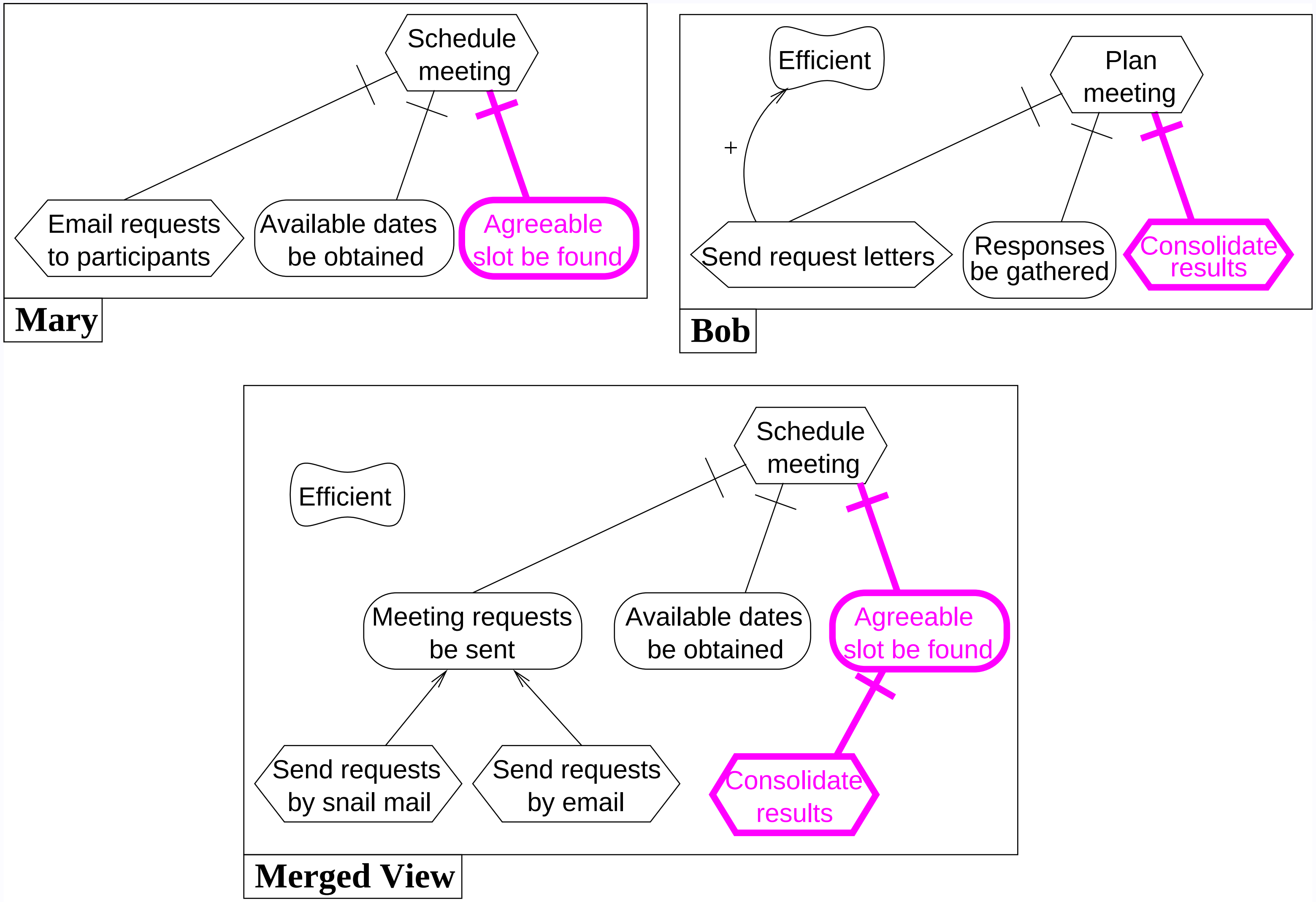


Different Aspects of the Problem



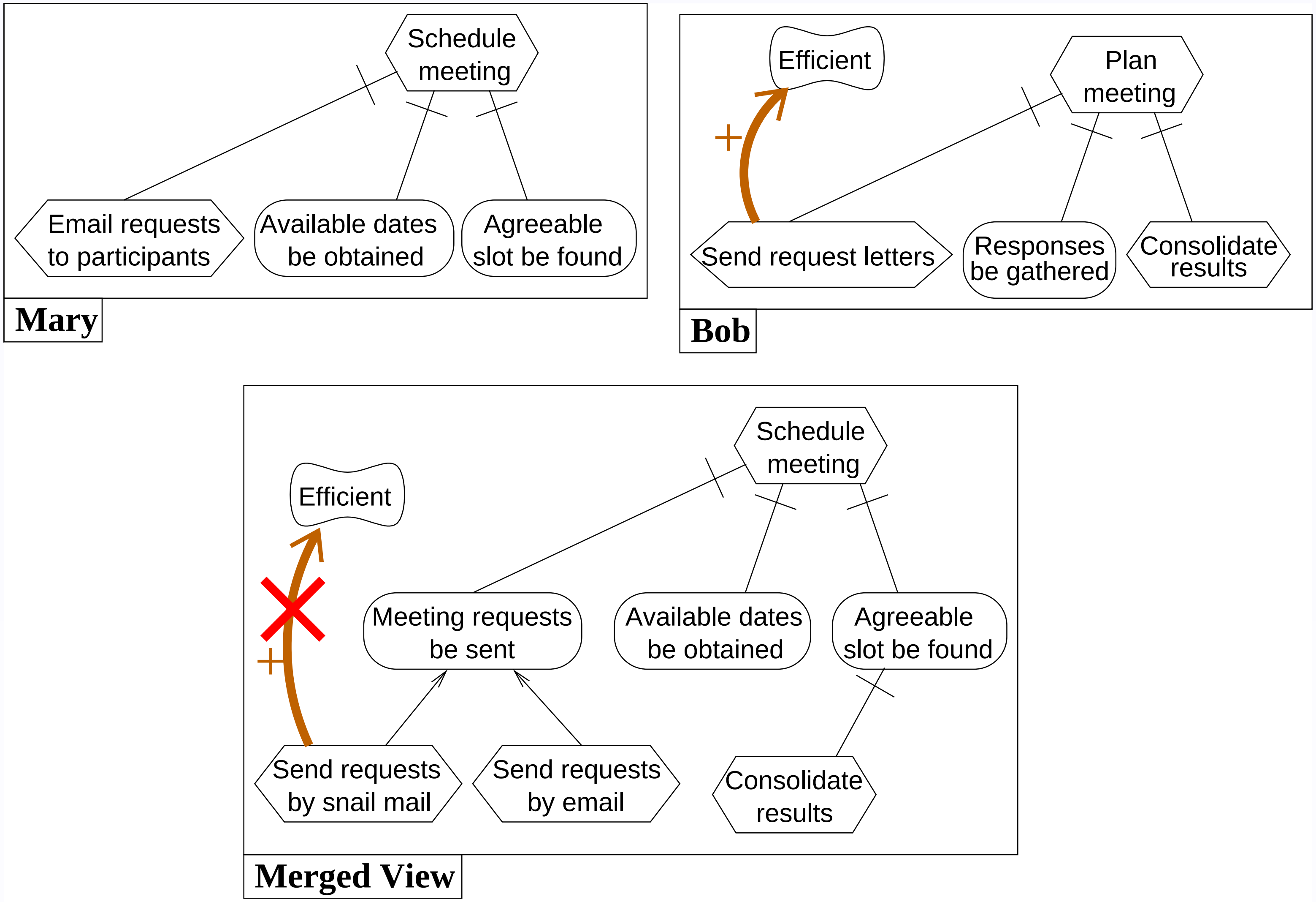
Different vocabularies

Different Aspects of the Problem



Structural discrepancies

Different Aspects of the Problem



Conceptual disagreements

Different Aspects of . . . – Cnt'd

➜ **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

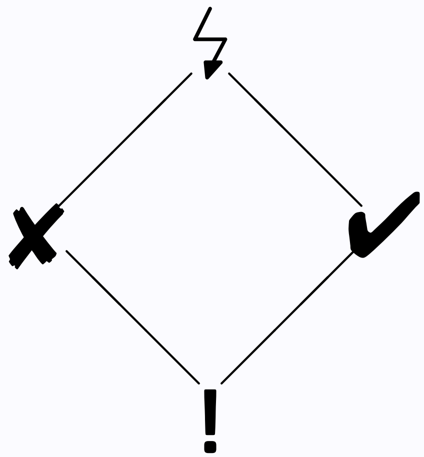
- **Roadmap**
  - Defining a representation formalism (**Views**)
  - Defining a notion for embedding a view into another (**Mappings**)
  - Specifying how view merges can be hypothesized (**Interconnection Diagrams**)
  - Devising a view merging algorithm
- **Assumptions**
  - Specification notations are graph-based
  - It is possible to be precise about the areas of uncertainty and disagreement

View Representation

- 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]

View Annotations

➜ We use a 4-valued lattice for annotations:



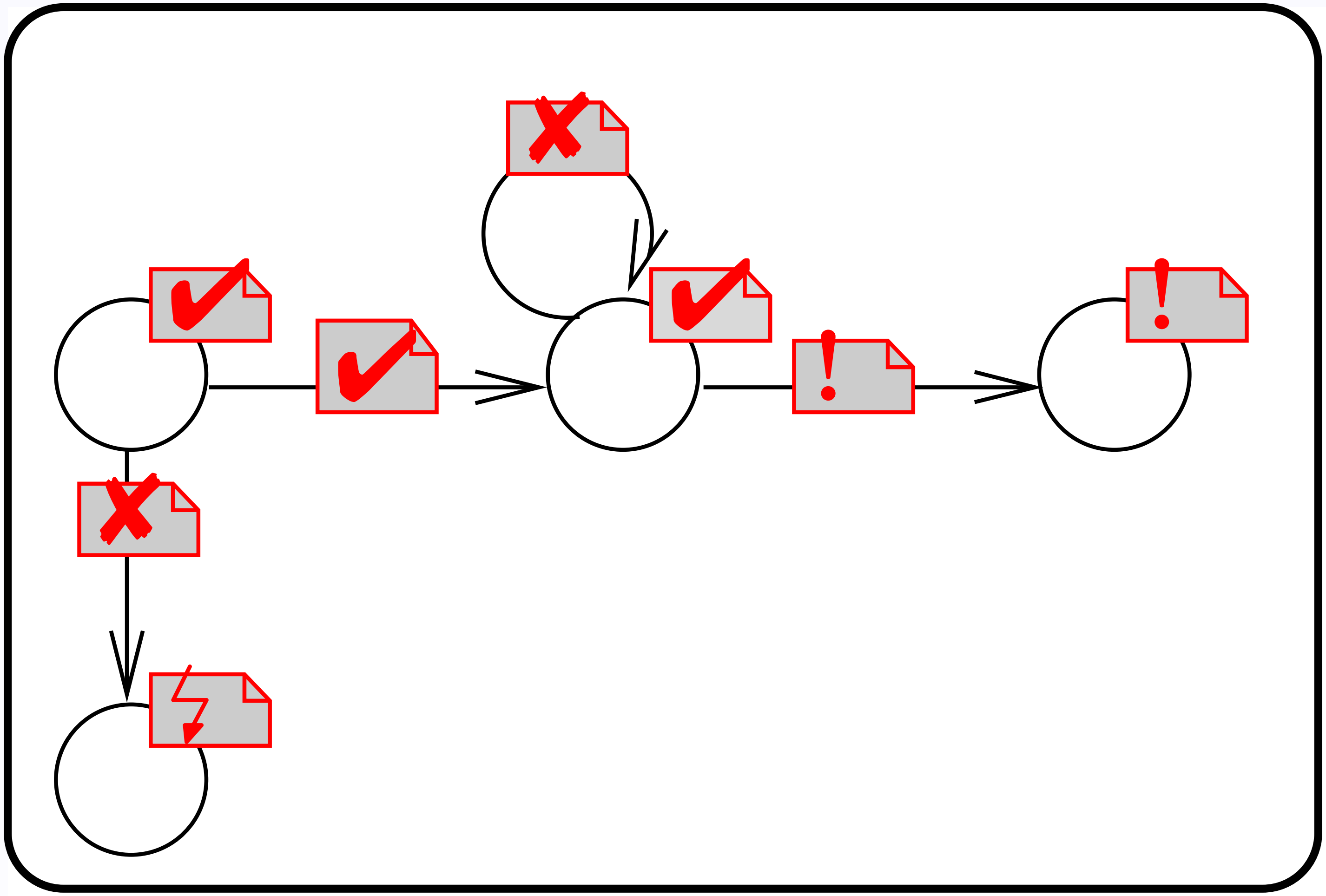
➥ !: *proposed* but not certain to be well-conceived

➥ ✘: known to be ill-conceived (*repudiated*)

➥ ✔: known to be well-conceived (*affirmed*)

➥ ϟ: both repudiated and affirmed, hence *disputed*

Annotations – Example

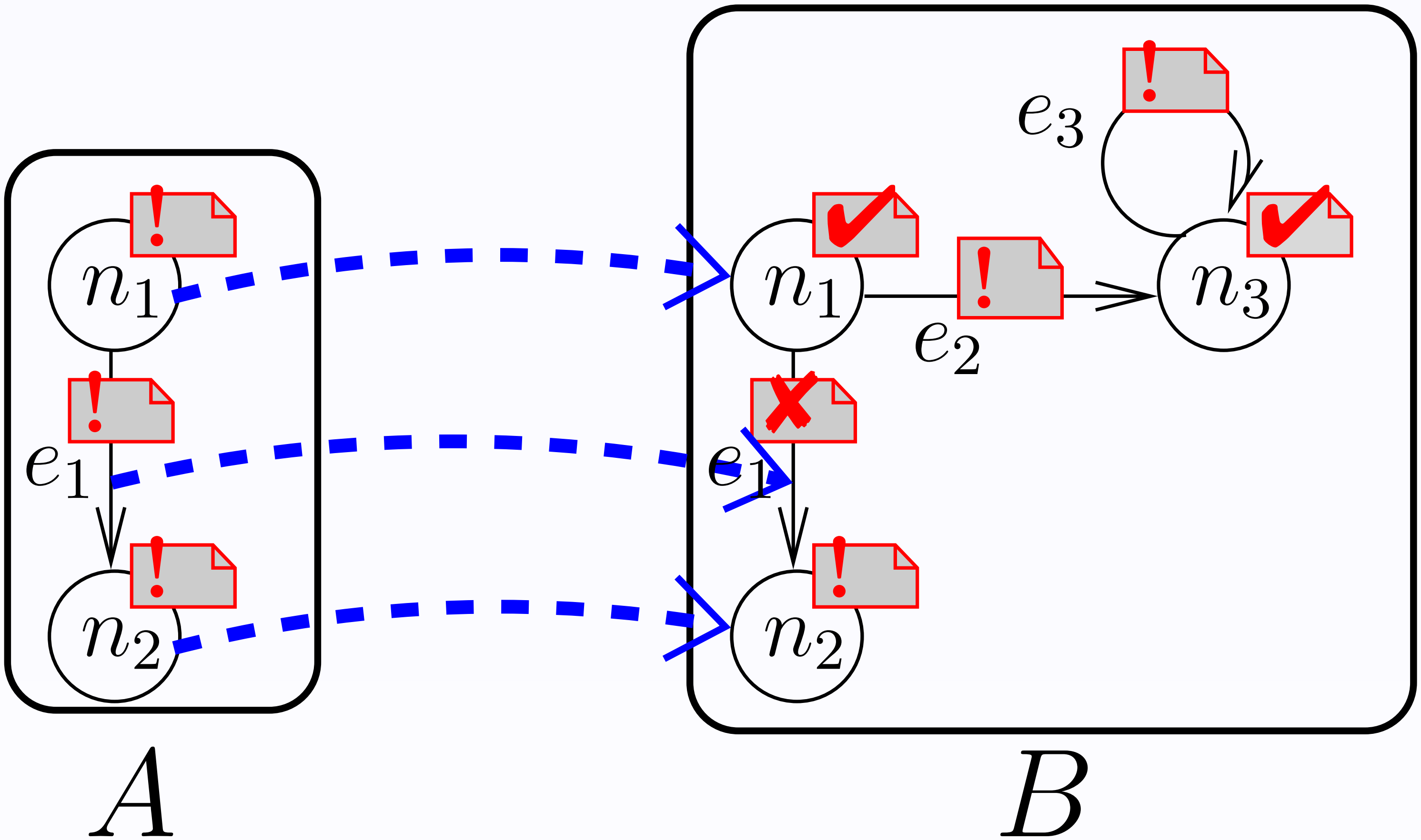


View Mapping

- **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



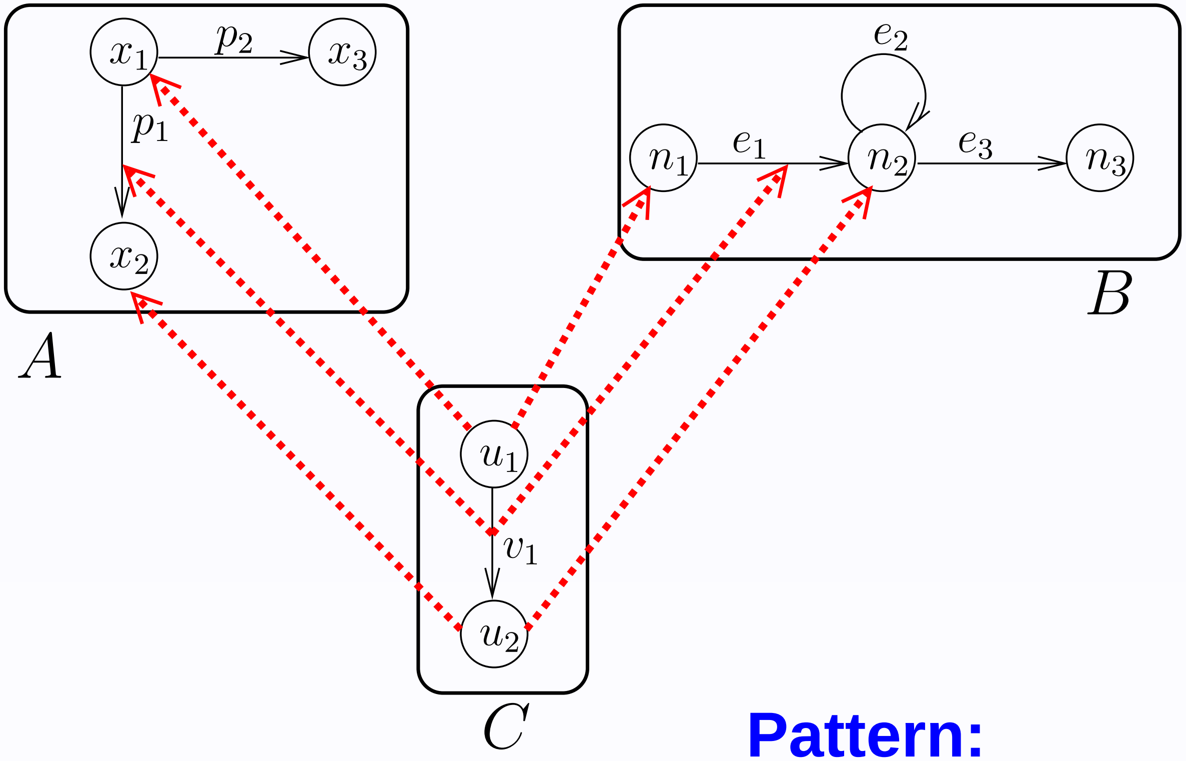
Capturing Merge Hypotheses

- **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

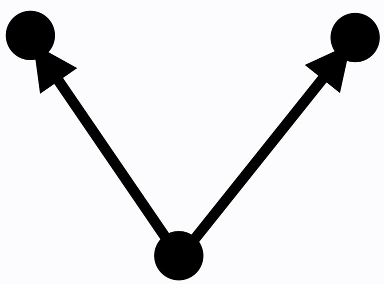
Merge Hypotheses – Examples

➜ Correspondence identification

▷ Annotations have been omitted for simplicity



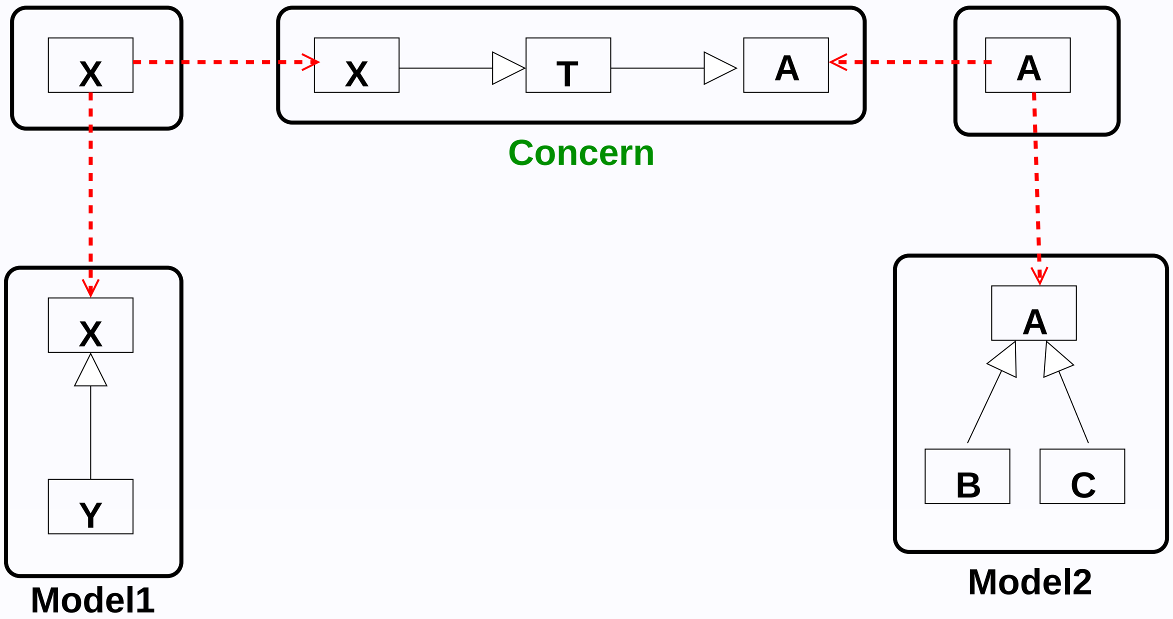
Pattern:



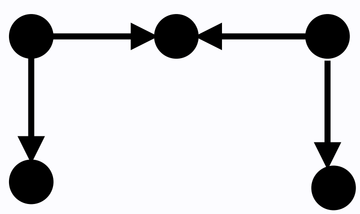
Examples – Cnt'd

➜ 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 <u>annotated sets</u>**

Merging Annotated Sets

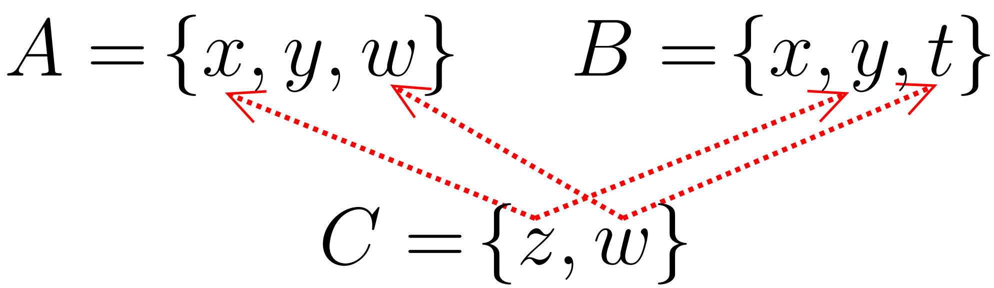
➜ **Algorithm:**

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)



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

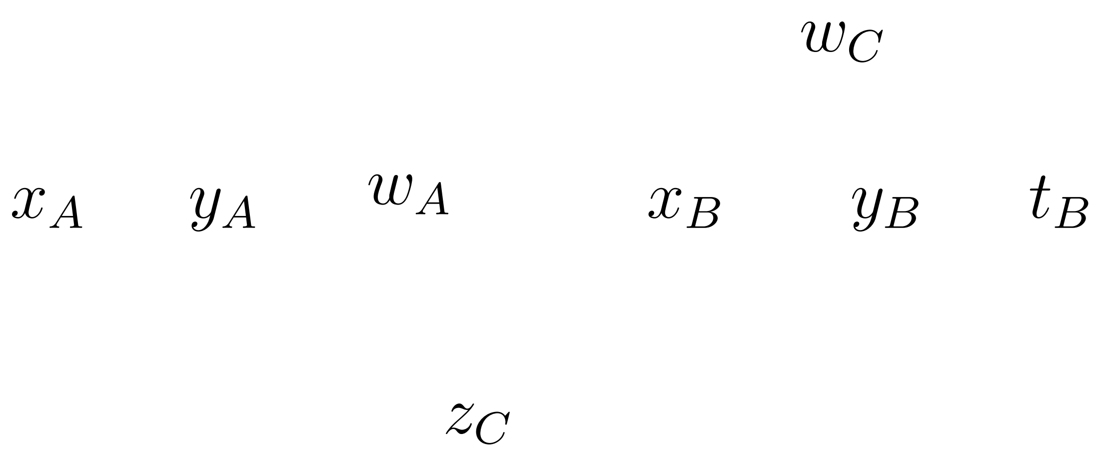
$$C = \{z, w\}$$

The given interconnection diagram

Merging Sets (Step 2)

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

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

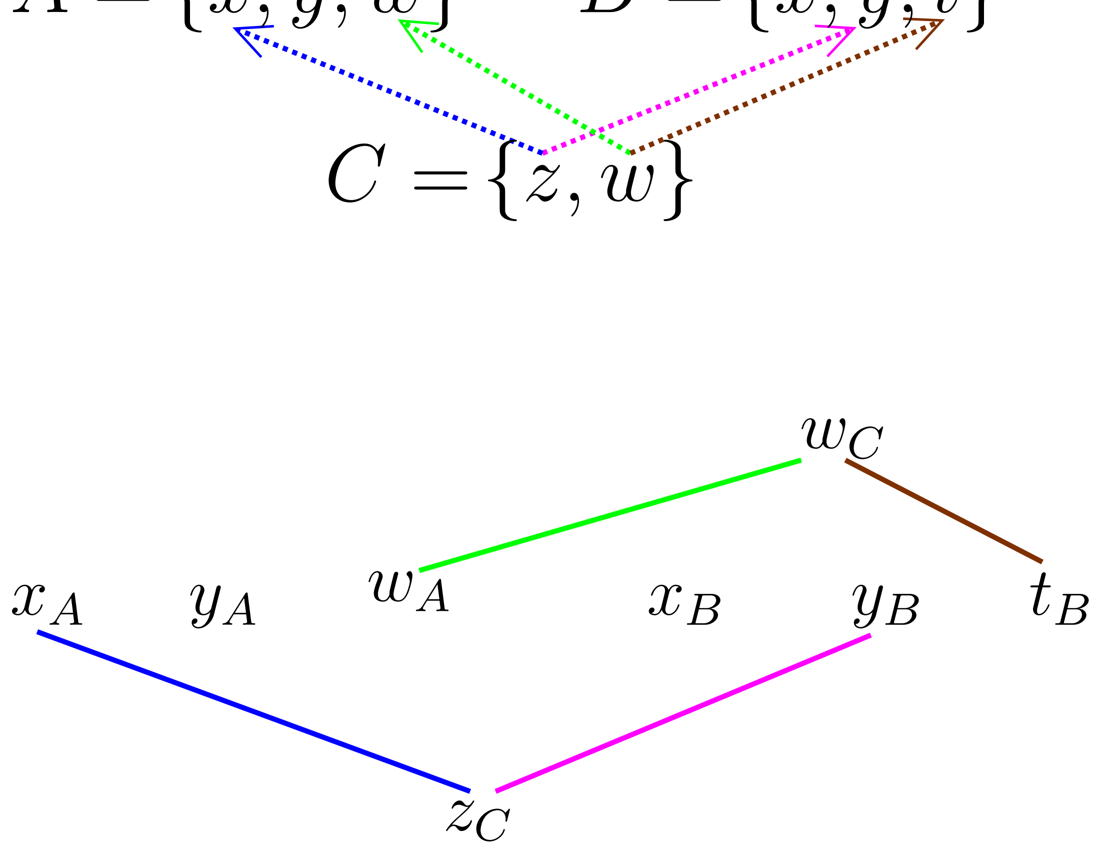


Compute the disjoint union

Merging Sets (Step 2)

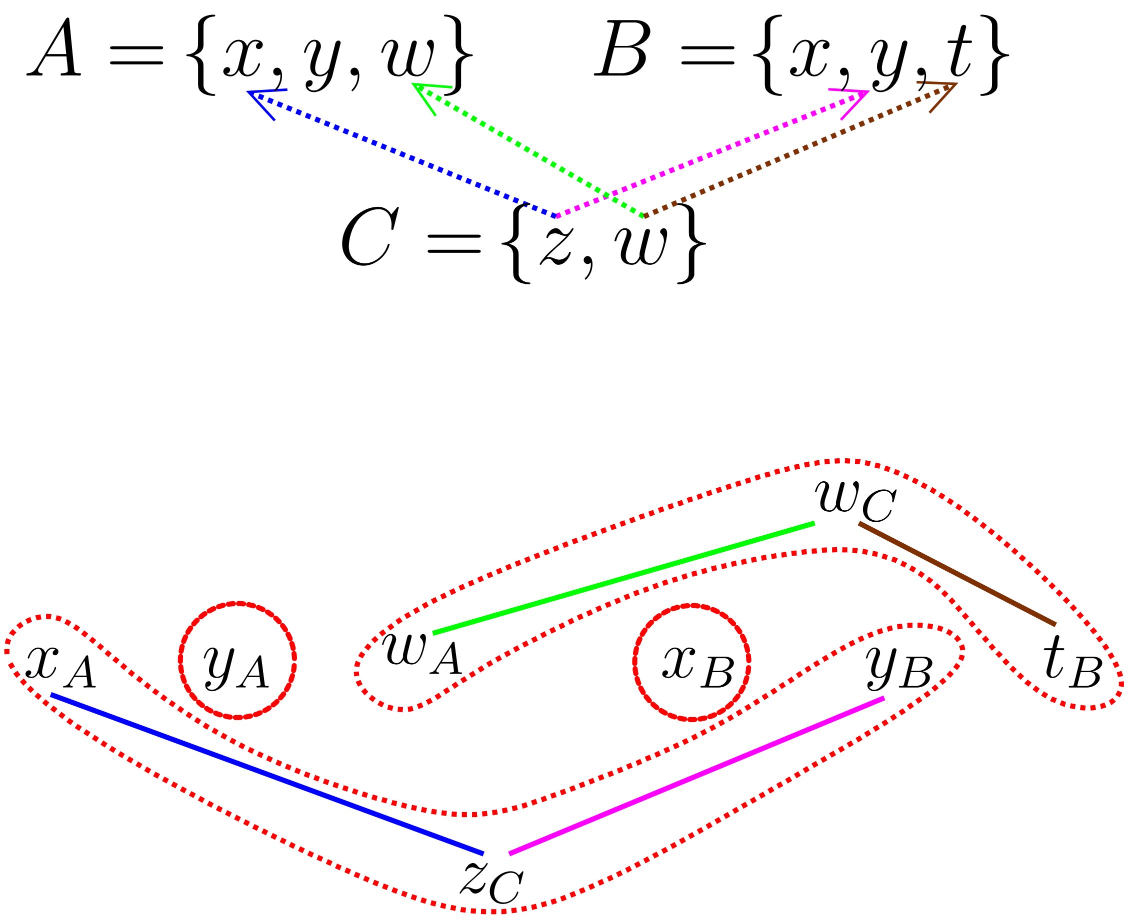
$$A = \{x, y, w\} \qquad B = \{x, y, t\}$$

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



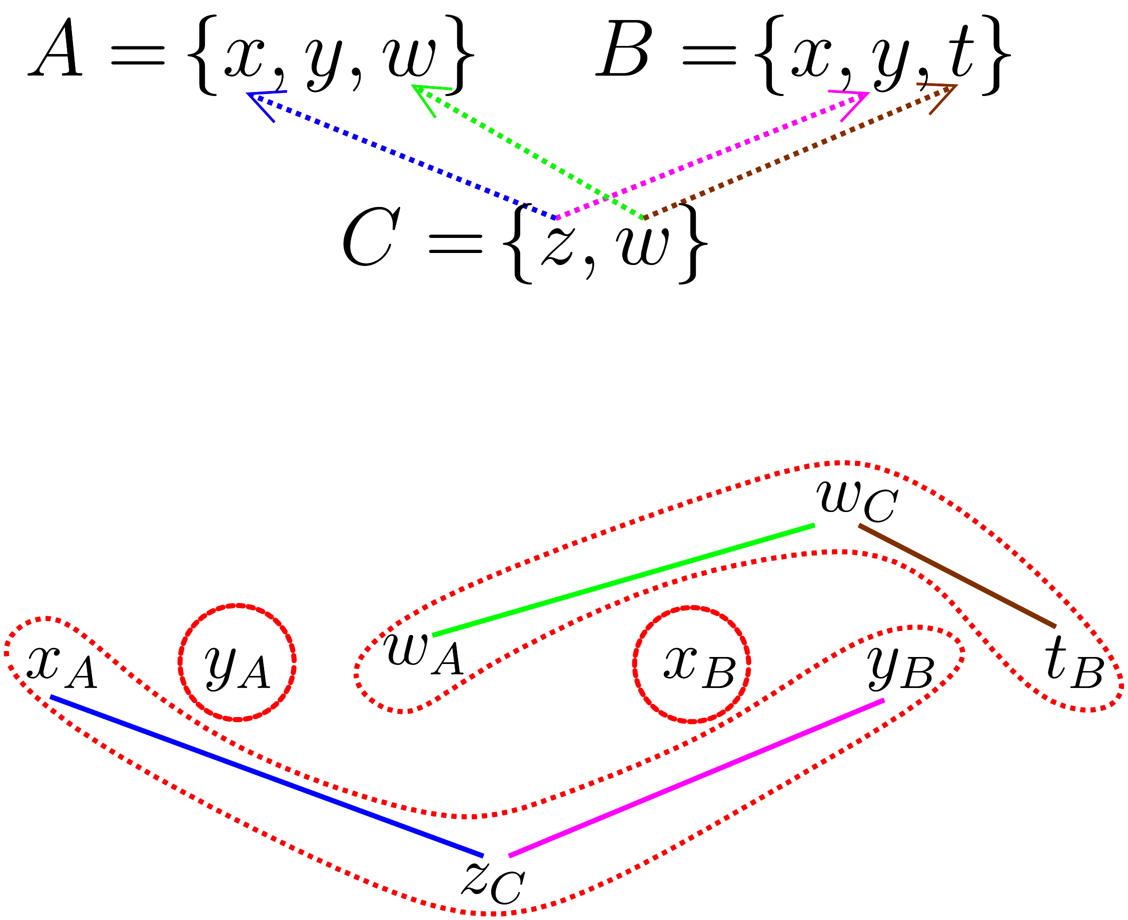
Connect up the related pairs

Merging Sets (Step 2)



Find the connected components

Merging Sets (Step 2)

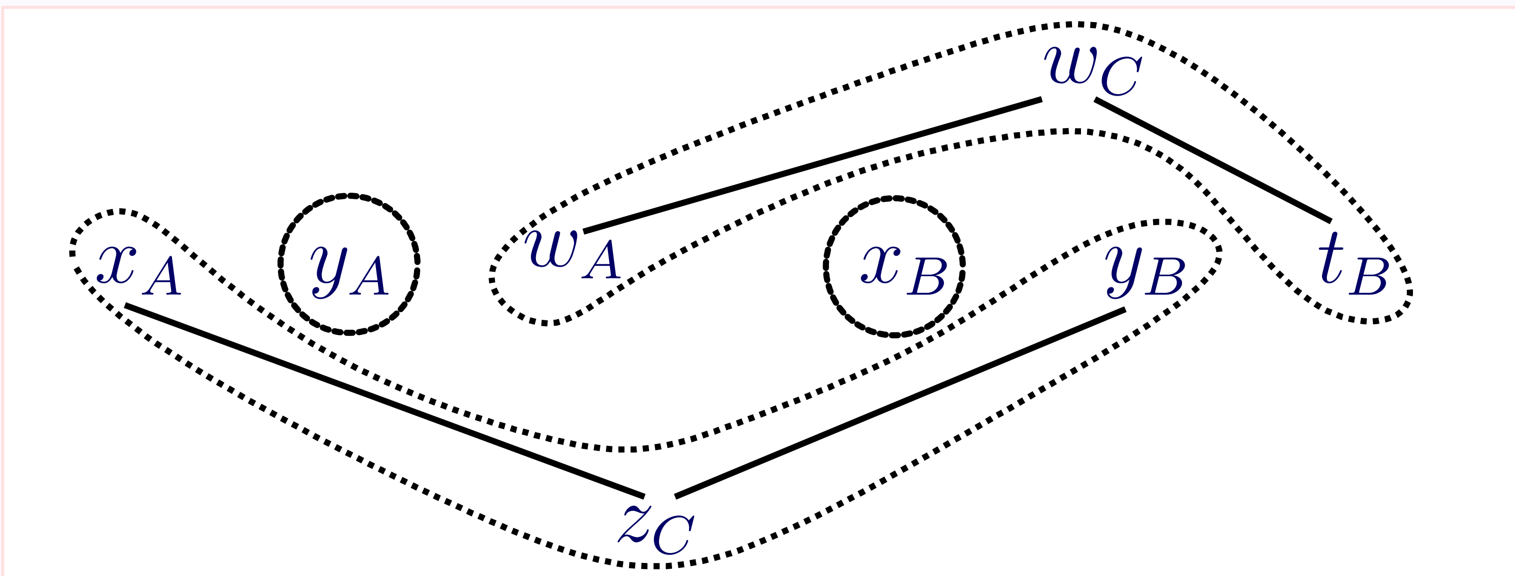


$$\{\underline{\{x_A, y_B, z_C\}}, \underline{\{y_A\}}, \underline{\{w_A, t_B, w_C\}}, \underline{\{x_B\}}\}$$

The merged set

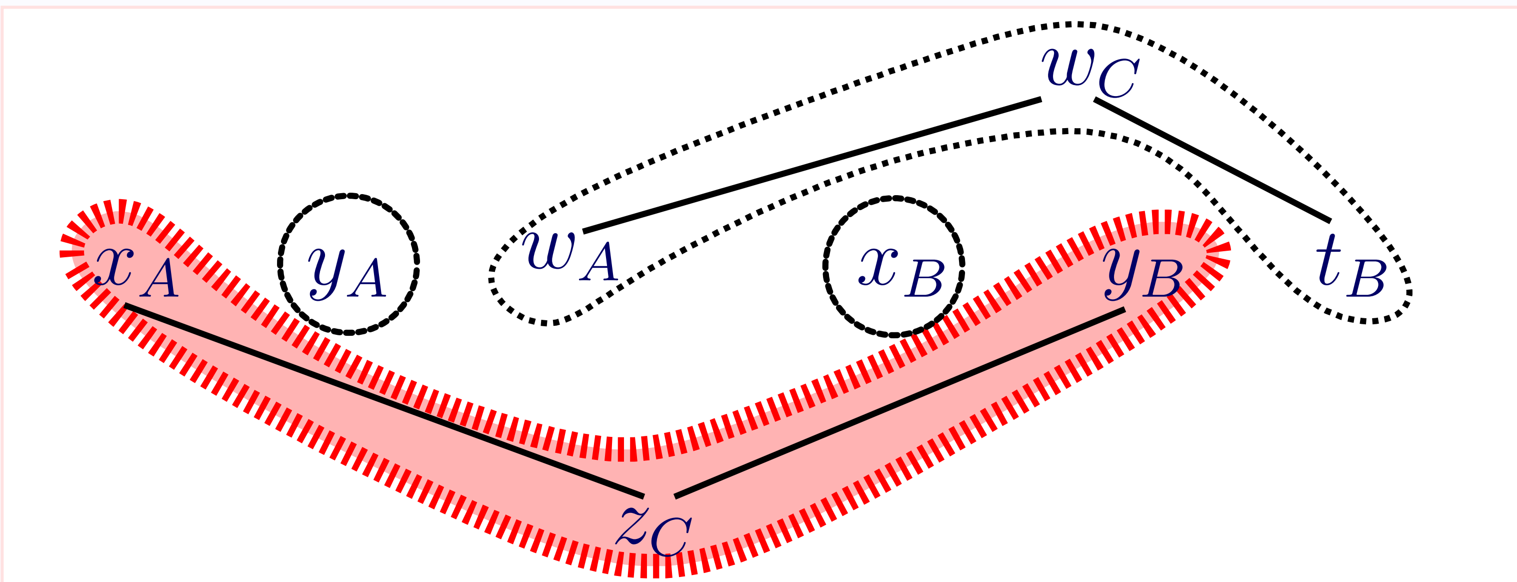
Computing the Annotations (Step 3)

- 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**:



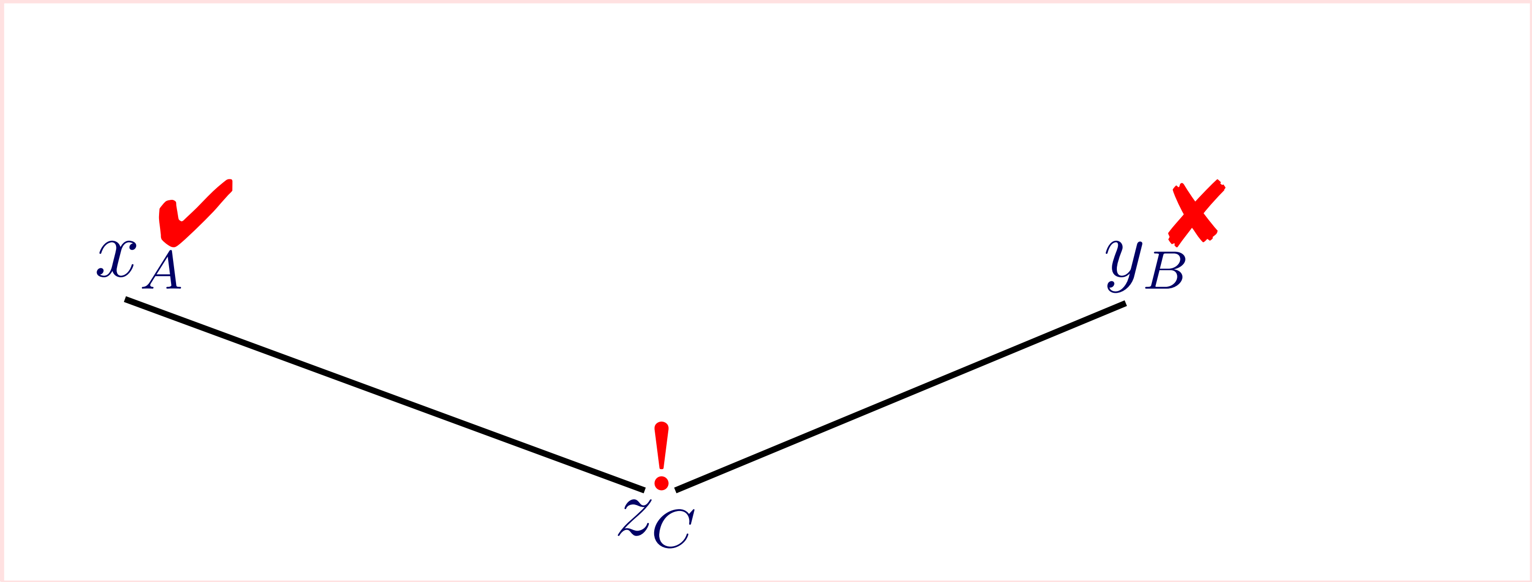
Computing the Annotations (Step 3)

- 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**:

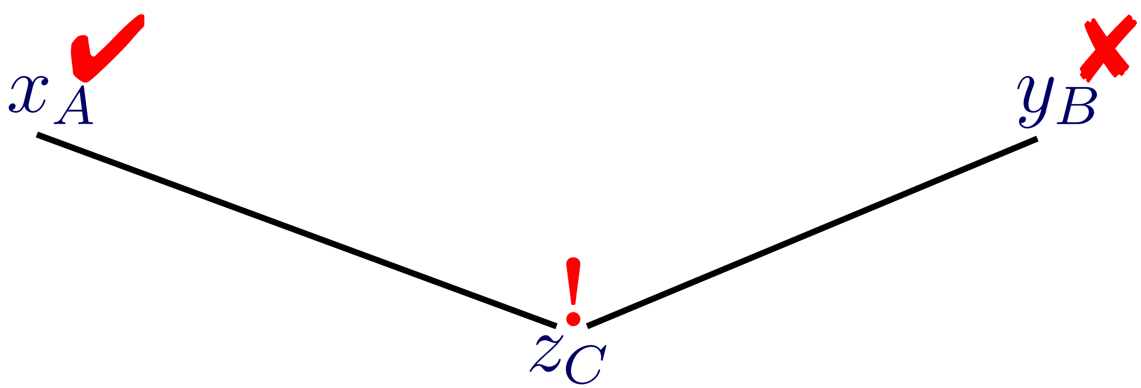


Computing the Annotations (Step 3)

- 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**:

$$\mathrm{lub}\{!, \text{✔}, \text{✘}\} = \text{↯}$$

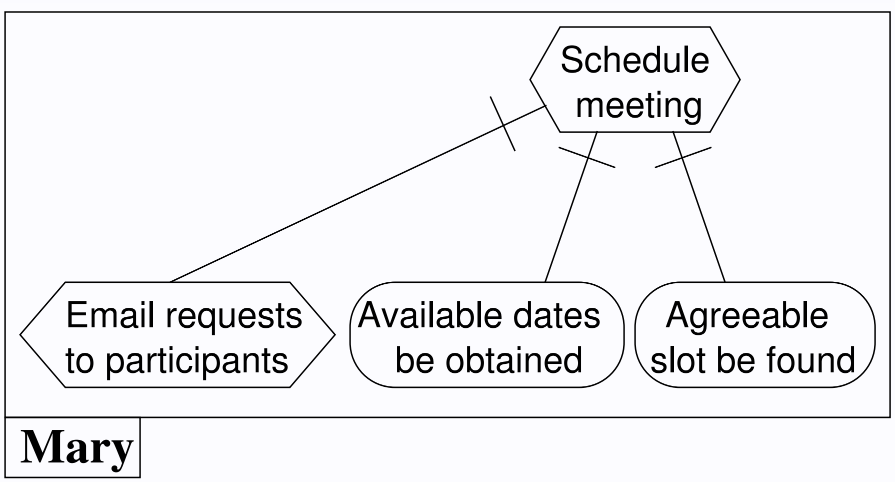


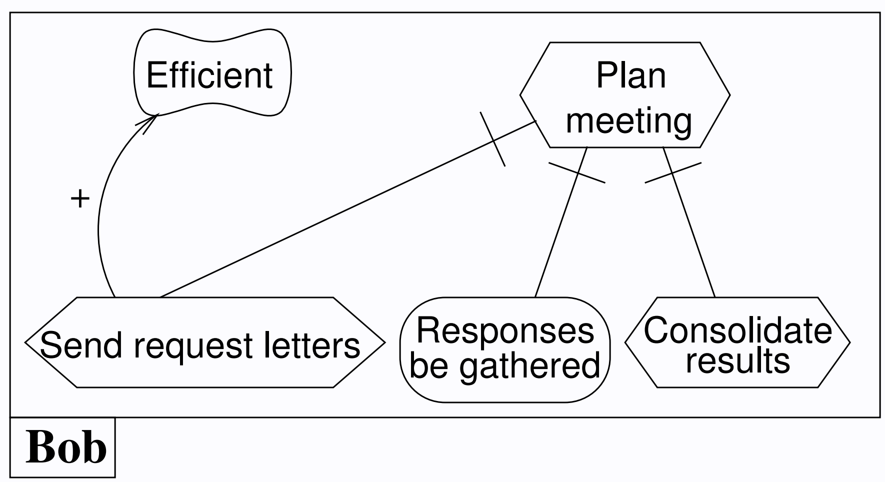
Example – Revisited

➜ Where are we?

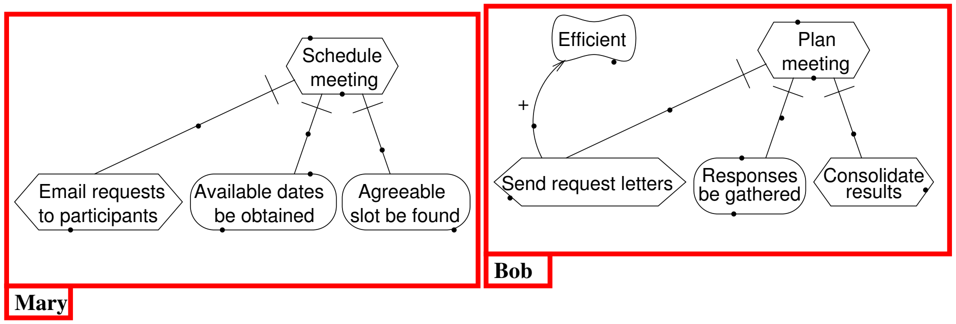
- ✔ Basic constructs: Views, Mappings, Diagrams
- ✔ The merge algorithm
- ➻ Merging i^* views – an example

➜ Initial views of stakeholders



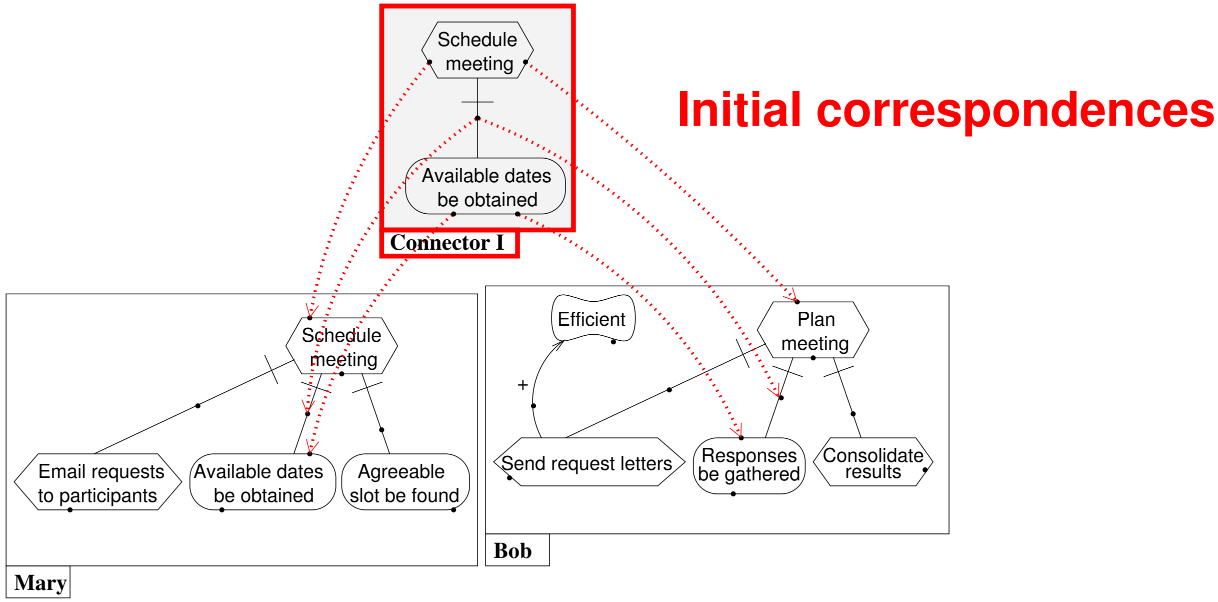


View Interconnections

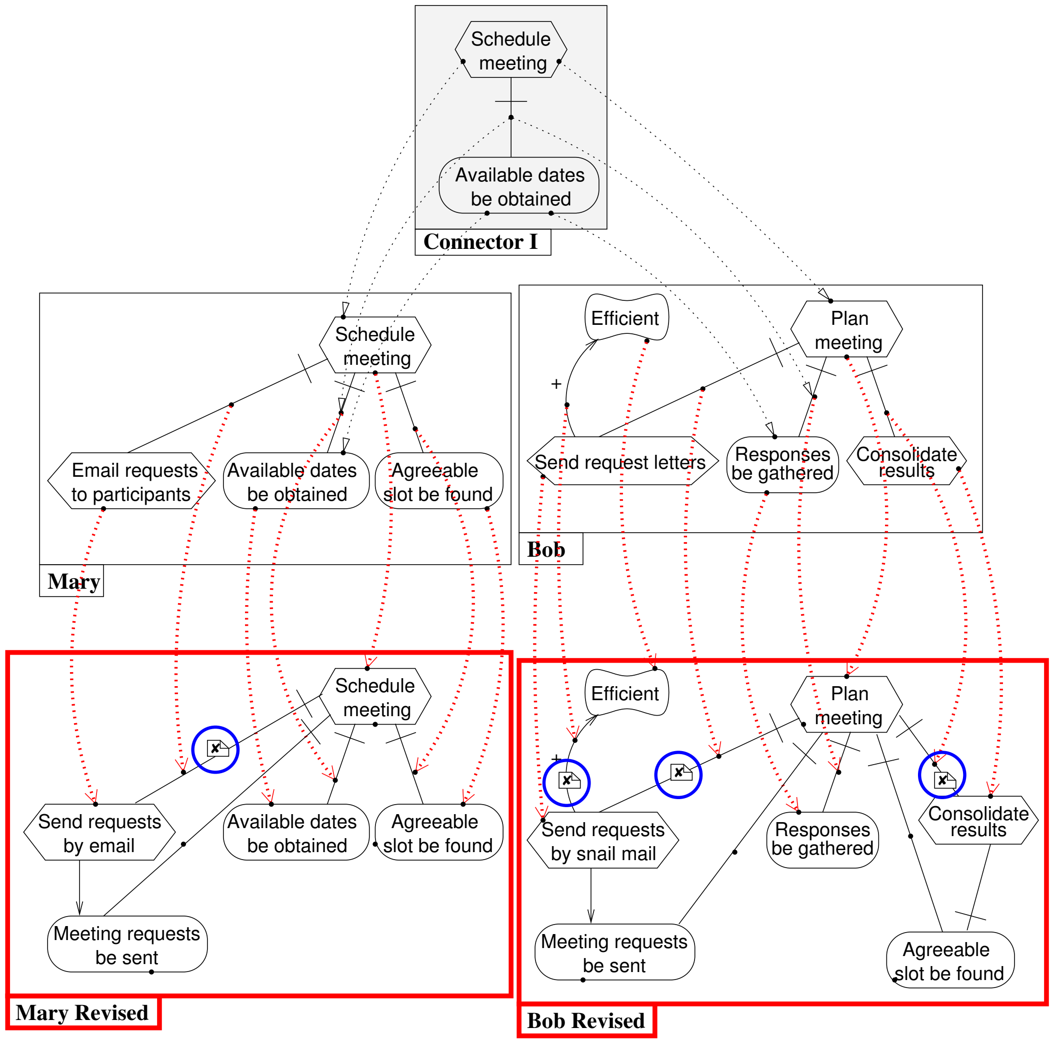


Initial views

View Interconnections

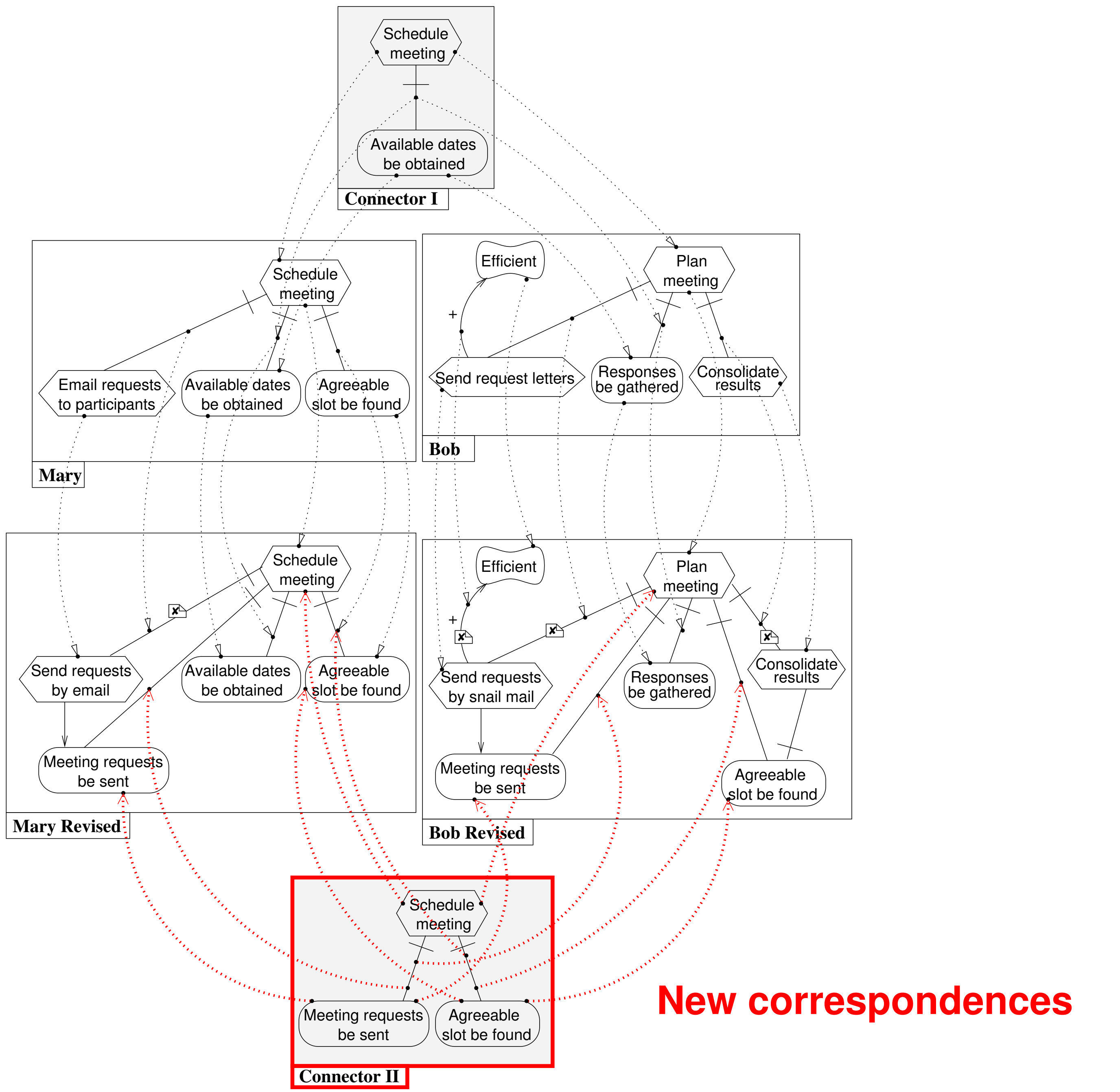


View Interconnections

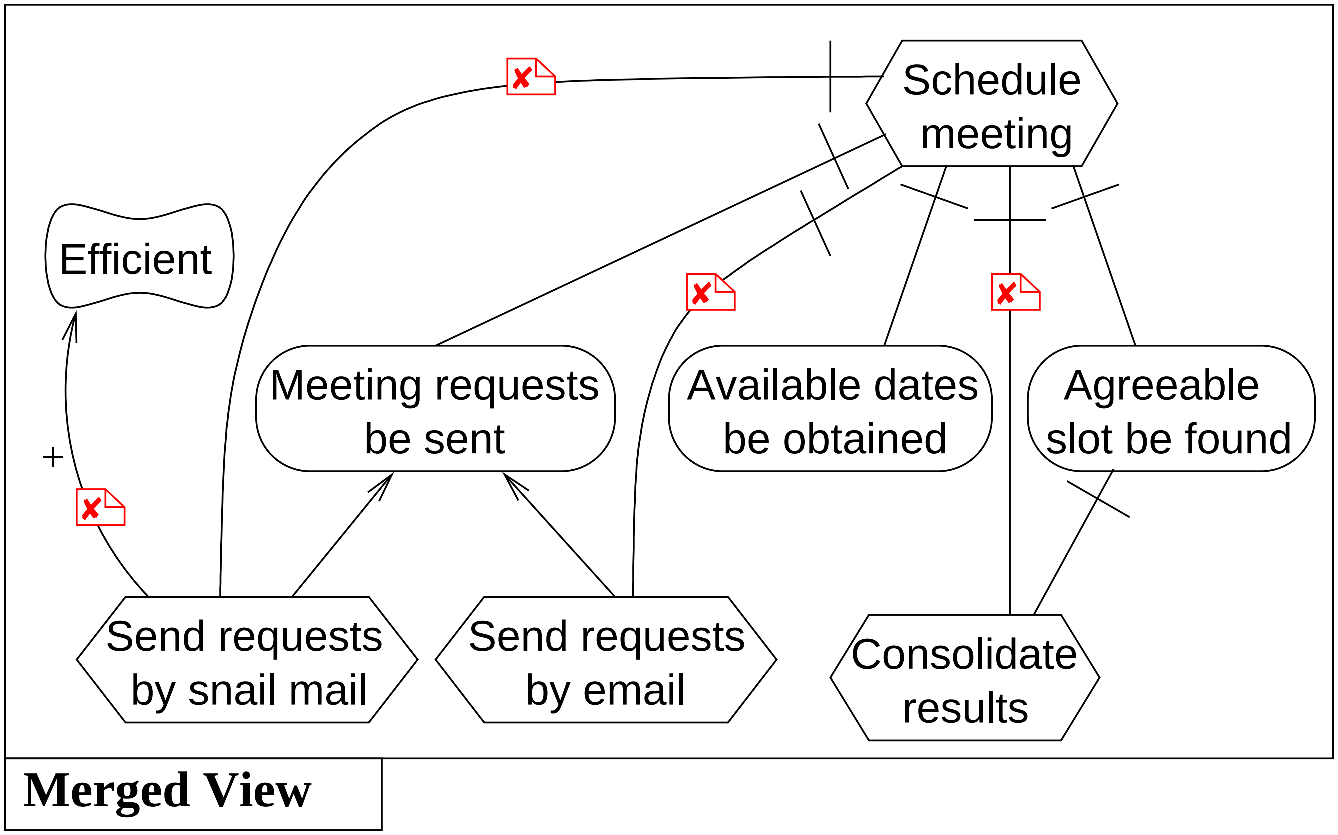


Revising the initial views

View Interconnections

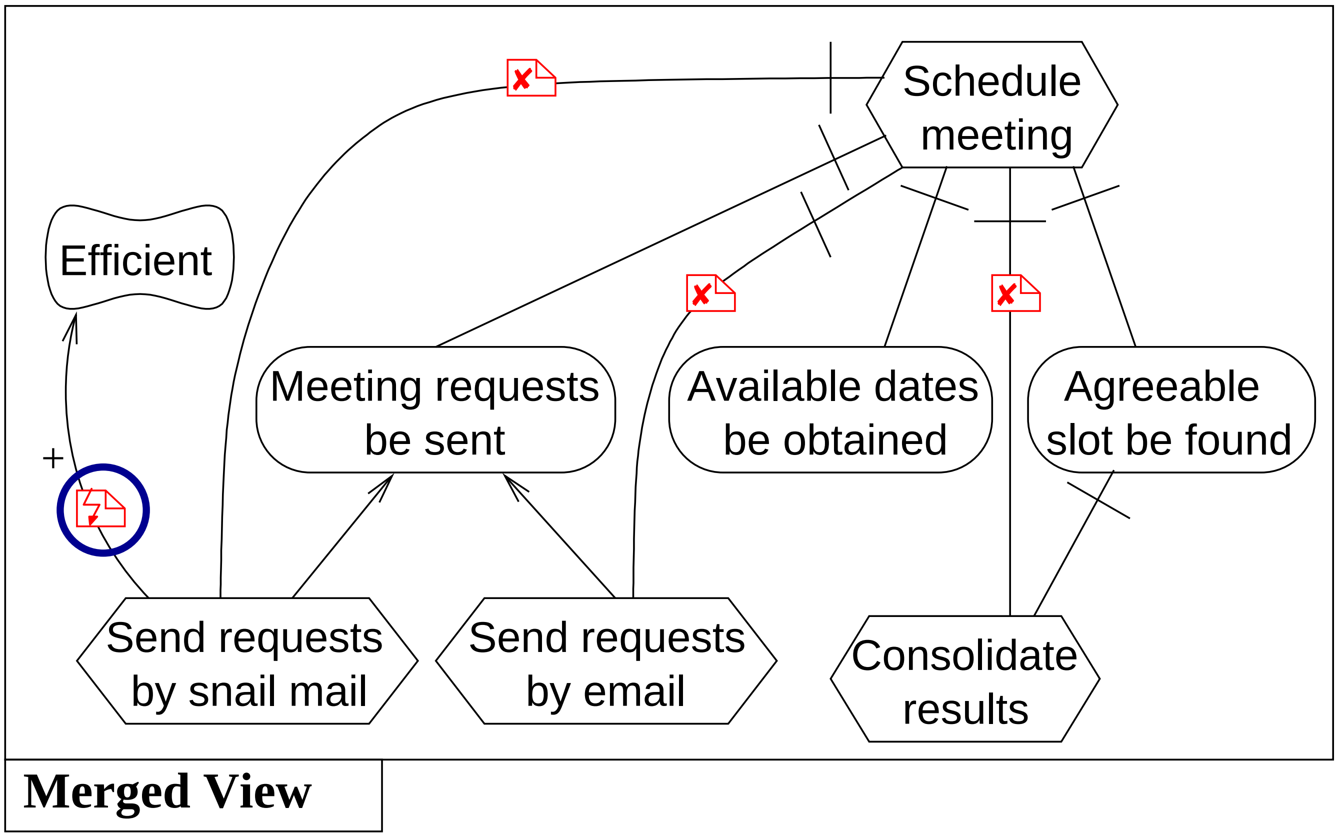


The Merged View



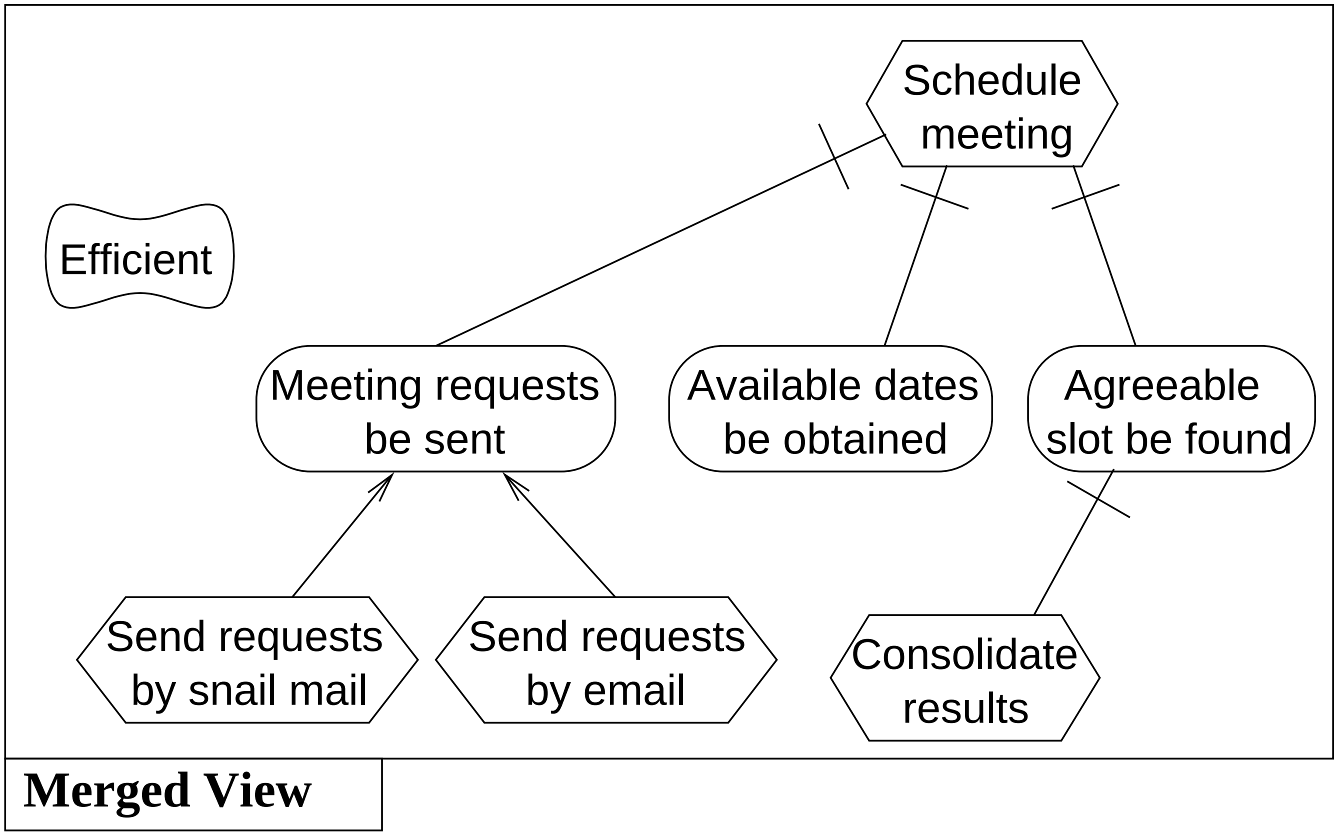
Result of the merge operation

The Merged View



If Bob insisted on his contribution link

The Merged View

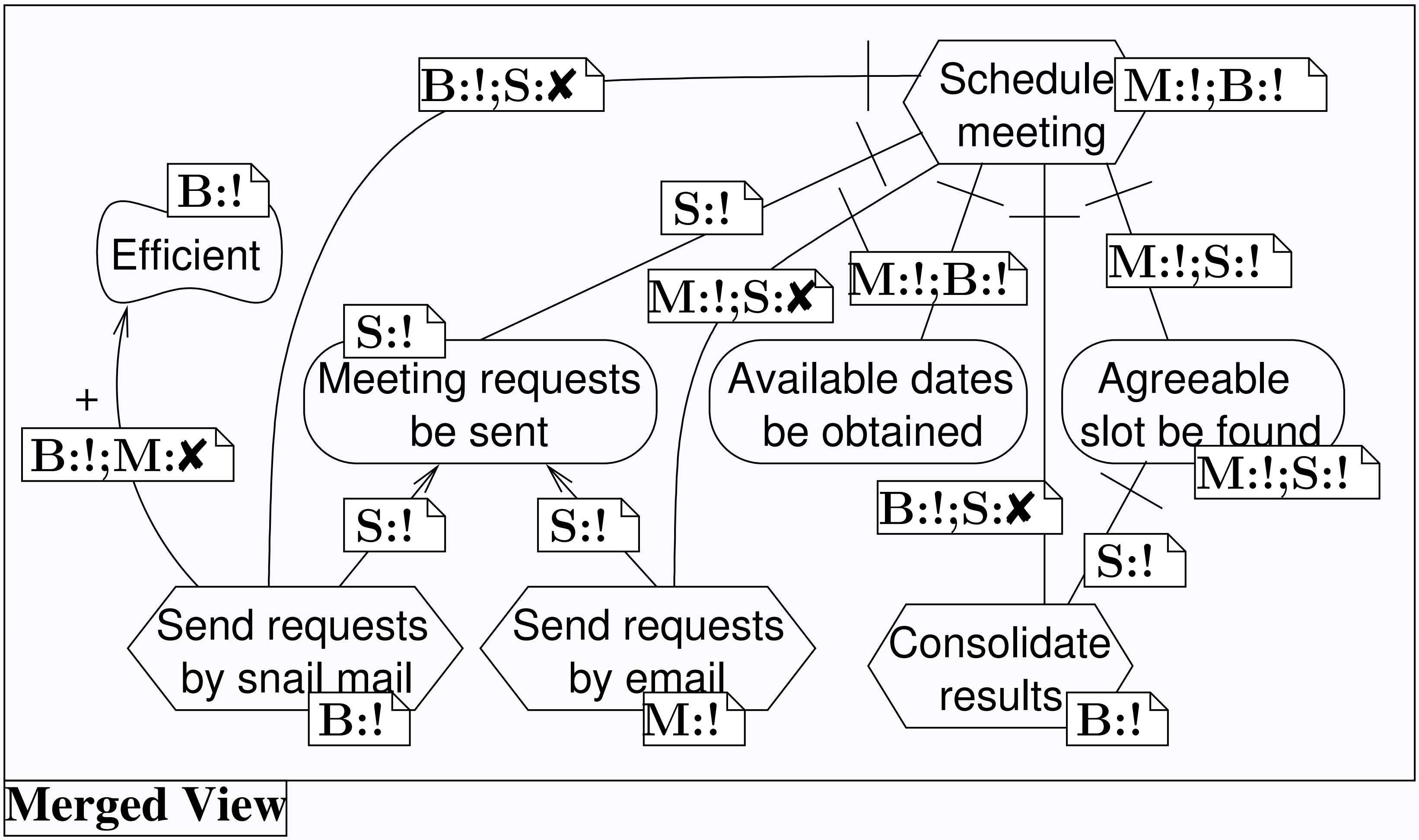


... with repudiated elements removed

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

Merged View with Traceability Info



✎ 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.