

Requirements Engineering with Viewpoints

Anna Hanstorp, Adrian Gasparini
10.11.2014

Definitions

“View” (or “viewpoint”):

- Different classes of users
- The contexts in which different roles are performed
- To distinguish between stakeholder terminologies
- To encapsulate knowledge about a system into loosely-coupled objects

Inconsistency:

- Relationship between views does not hold

Using ViewPoints for Inconsistency Management - Motivation

- **Multiple development participants** often hold multiple inconsistent views
- Considerable effort is spent handling **recurrent inconsistencies**
- The evolution of a specification may affect both **resolved** and **unresolved inconsistencies**

ViewPoint Framework

- Software development knowledge is partitioned into multiple views called **Viewpoints**
- **Inconsistencies** between Viewpoints are **managed**:
 - Relationships between Viewpoints are represented
 - Resolved/unresolved inconsistencies are recorded
- Assumption: Viewpoints **often inconsistent**
→ keep complete **work record**, detailing any detected inconsistencies and actions taken to resolve them

ViewPoint Objects

Viewpoints are objects encapsulating:

- Representation style
- Domain
- Specification
- Work plan
- Work record

Reasons for Inconsistencies

- Mistakes
- Missunderstandings
- Lack of information
- Infeasible or impractical requirements
- Conflicts between knowledge sources

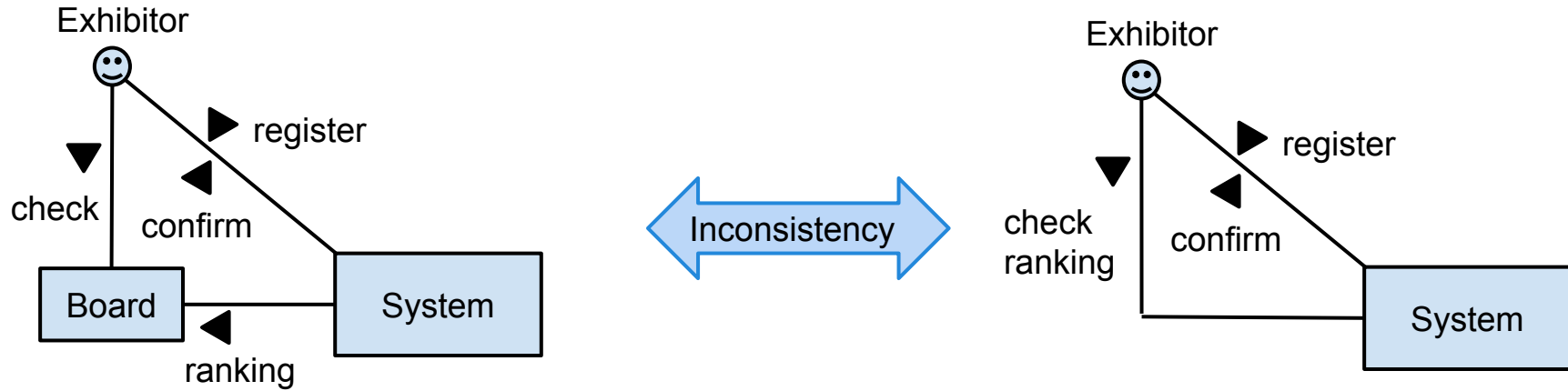
View Merging in the Presence of Incompleteness and Inconsistency - Motivation

- View merging **key problem** in conceptual modeling
- Large models often constructed and accessed by **manipulating individual views**
- Important to **consolidate** a set of views
- View merging is complicated by **incompleteness** and **inconsistency**

View Merging Framework

- Merging **incomplete** and **inconsistent** views
- Applicable to a variety of **graphical modeling languages**
- **Trace contributions** in the merged view back to sources and to assumptions

Case: Exhibition of Inventions



- **ViewPoint:** State relationship, rules → Detected Inconsistency
- **Merge Views:** Define connector → Merged View

Conclusions

- **These frameworks can be used for:**
 - Requirements Interaction Management (RIM)
 - Requirements Negotiation
- **ViewPoint framework facilitates:**
 - Separation of concerns
 - Partitioning of software development knowledge
- **View Merging framework facilitates:**
 - Identification of interconnections between views prior to the merge → no naming conventions

Questions

- Do you have any experience with different viewpoints and inconsistencies?
- How did you manage and resolve them?
- Did you use tools for this purpose?

