

Requirements Engineering with Viewpoints

Group 1

Andrea Capobianco

ViewPoints Framework

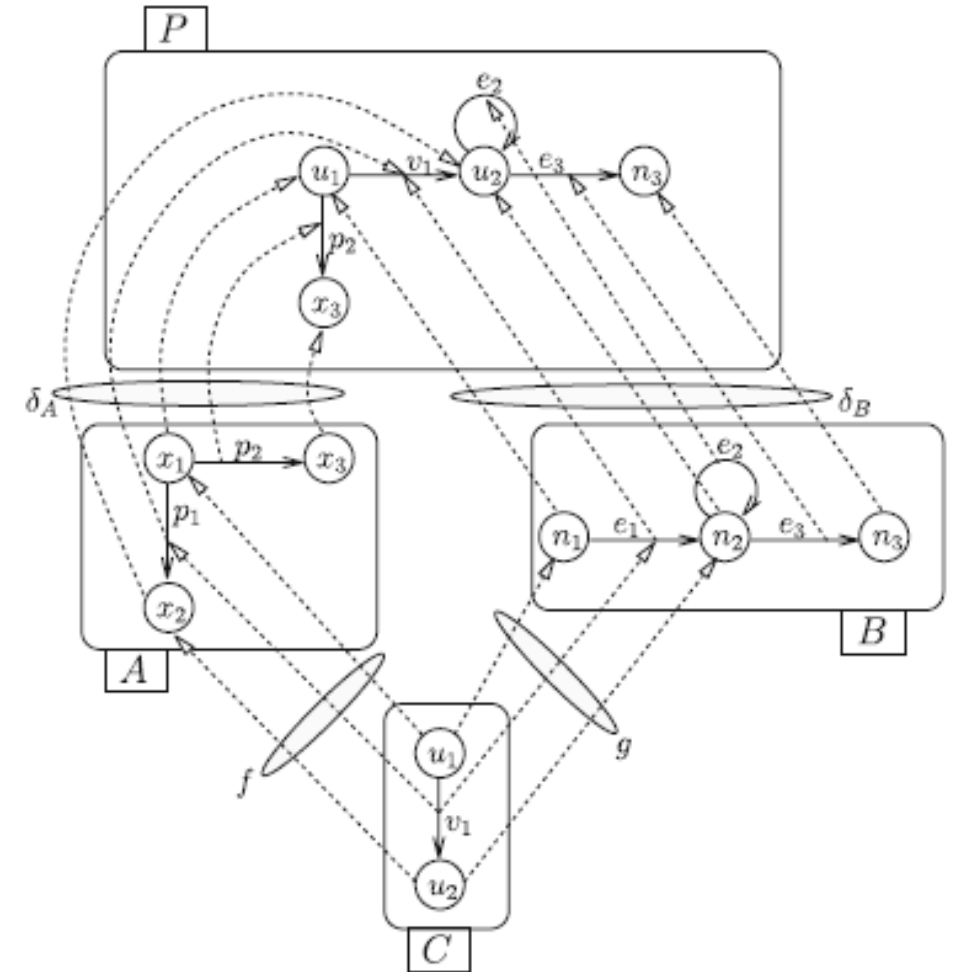
- ViewPoints framework for inconsistency management
- Inconsistency: «any situation in which two parts of a specification do not obey some relationship that should hold between them»
- Idea: Have internally consistent overlapping partitions (viewpoints) that may be inconsistent when merged: Allow changes, tolerate inconsistency.
- Inconsistency detection, classification (cause and kind) and handling (ignore, delay, circumvent, ameliorate, resolve)

ViewPoints Framework

- Views visual models (state diagrams, ERDs,...)
- Steps of ViewPoints framework:
 - Define grammatical notation
 - Partition into multiple Diagrams (e.g.: making call and receiving call)
 - Define consistency rules (e.g. Equal states in different diagrams need to have same transitions)
 - Analysis by regarding Views as different interacting instances (e.g. Two phones)
 - Define further consistency rules
- Changes to views check against rules defined and are recorded in work records (locally) → Record of resolved and unresolved inconsistencies

View Merging

- Merging incomplete/inconsistent views:
 - Views (Diagrams) as annotated graphs with nodes and edges
 - Annotations describe source (stakeholder) for source traceability and certainty
 - Merging by:
 - creating connector views (intersection of views to merge)
 - Finding homomorphisms (mappings)
 - Relation of views and merges expressed in an interconnection graph
 - Traceability of origin and assumption by creating named links between views



Application to RE 1 Case Study

- Shipment to Customer: View from Aviation Authorities and from Responsible for Drone Operations (Dave)
 - 2 Views (Activity Diagrams), multiple merge revisions
 - Actors have different domain knowledge and different focus (may cause difficulties for analyst)
 - What if merged view conflicts with other stakeholders views (pilots, project initiators, drone manufacturer) and how to handle increasing complexity?