A Focused Approach to Software Architectural Recovery

http://sunset.usc.edu/~neno/Focus

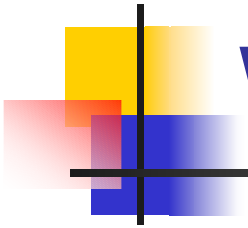
Nenad Medvidovic, Vladimir Jakobac

Center for Software Engineering

University of Southern California

Los Angeles

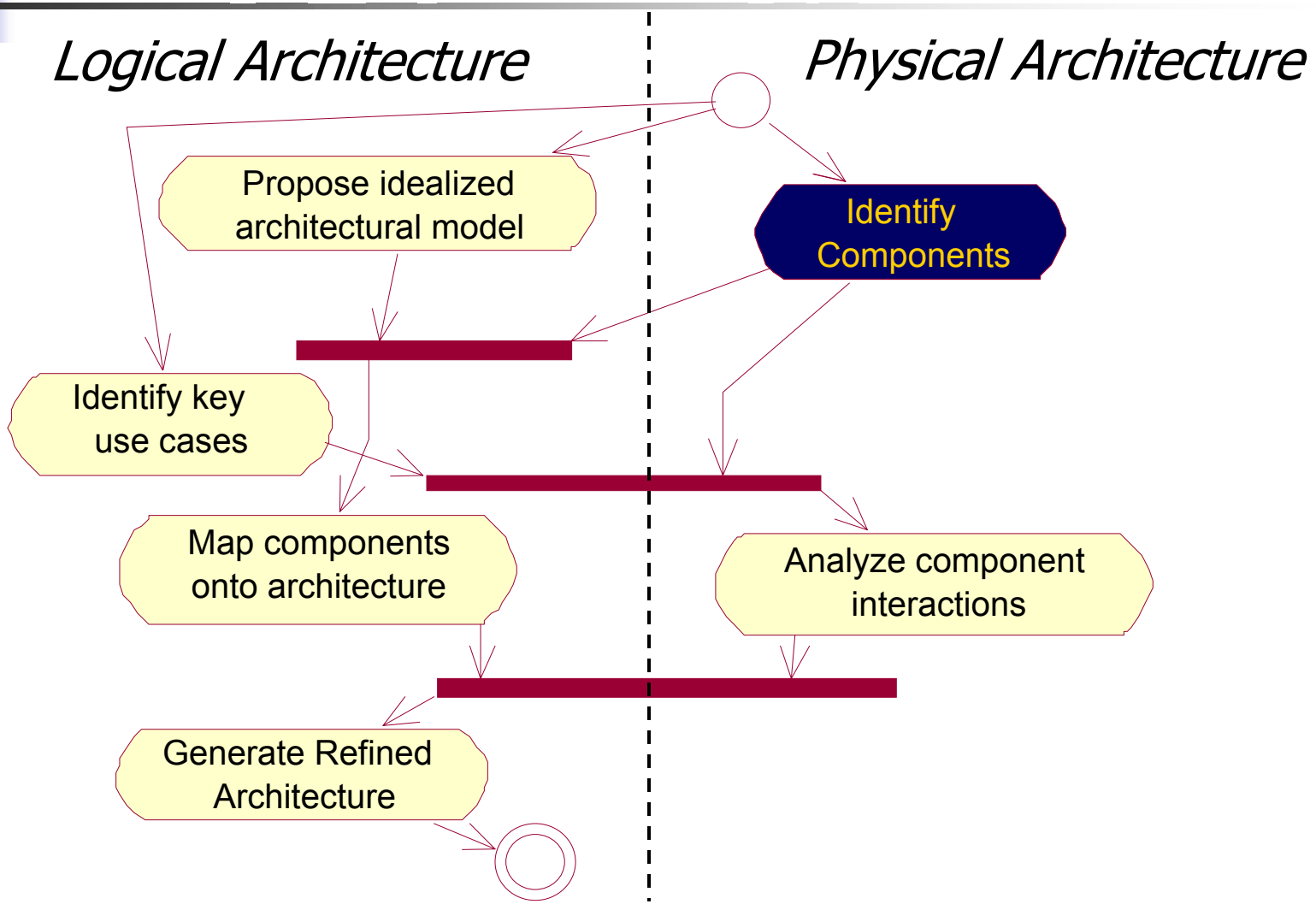
{neno,jakobac}@usc.edu

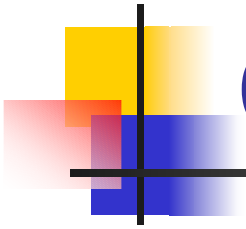


Why Recovery?

- Eroded architectures
  - Architectural model out of sync with the current implementation
- Reducing maintenance costs
  - Modifications should not cause other unforeseen problems
- System evolution
- Reuse

*Focus*ing Architectural Recovery

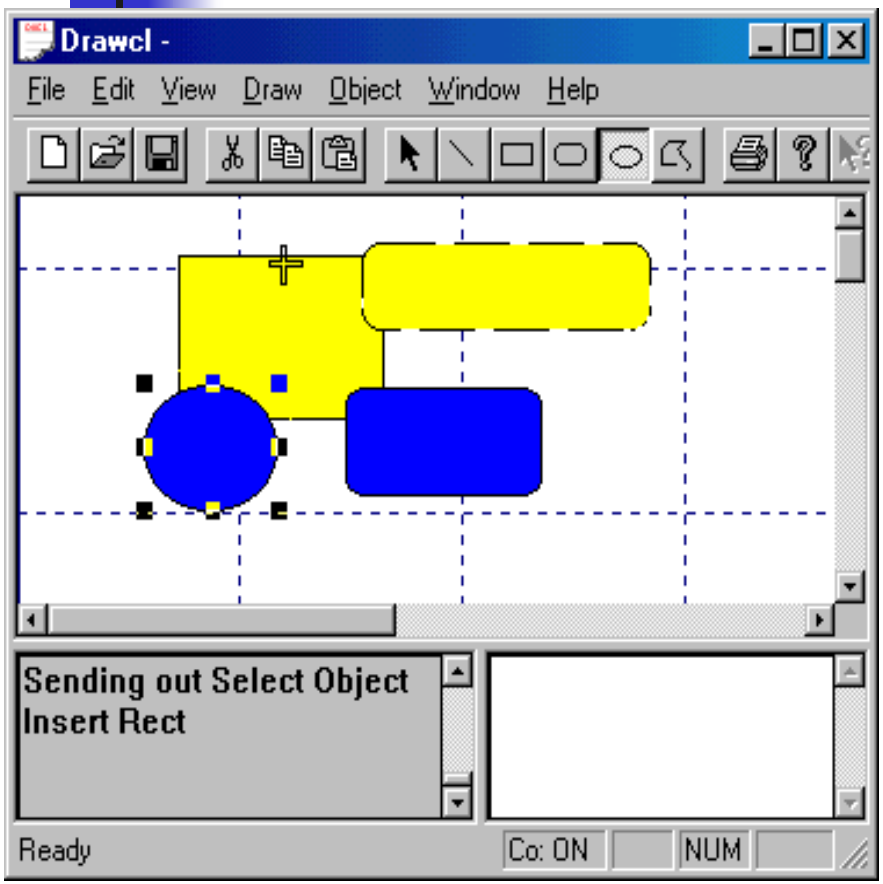


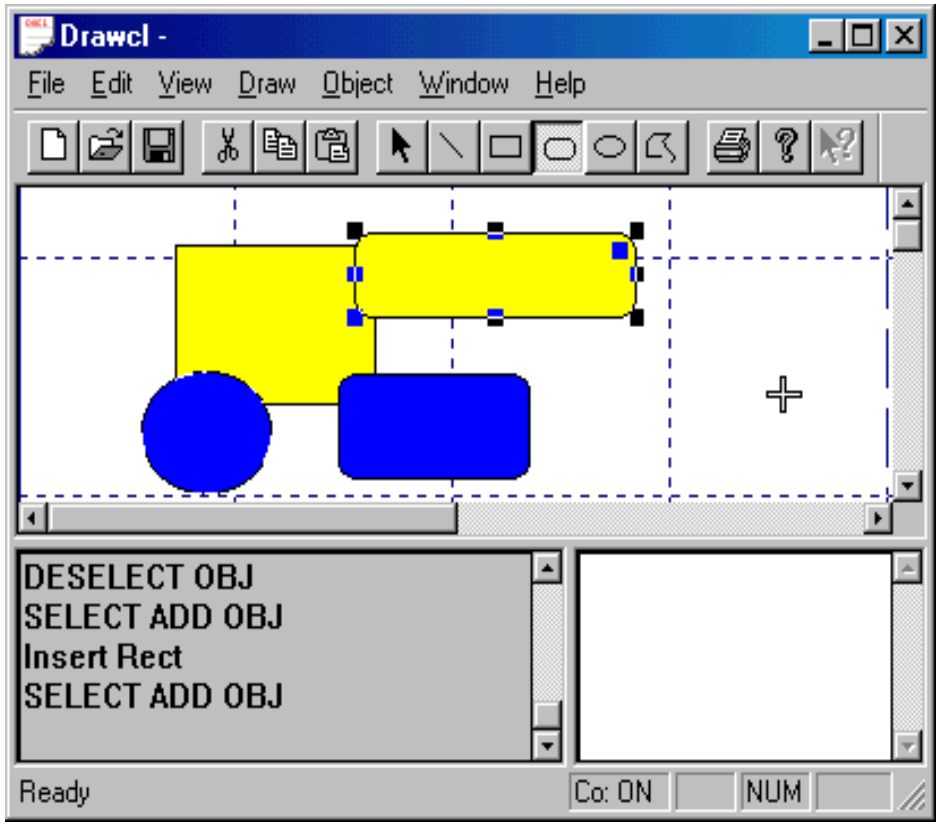


Component Recovery

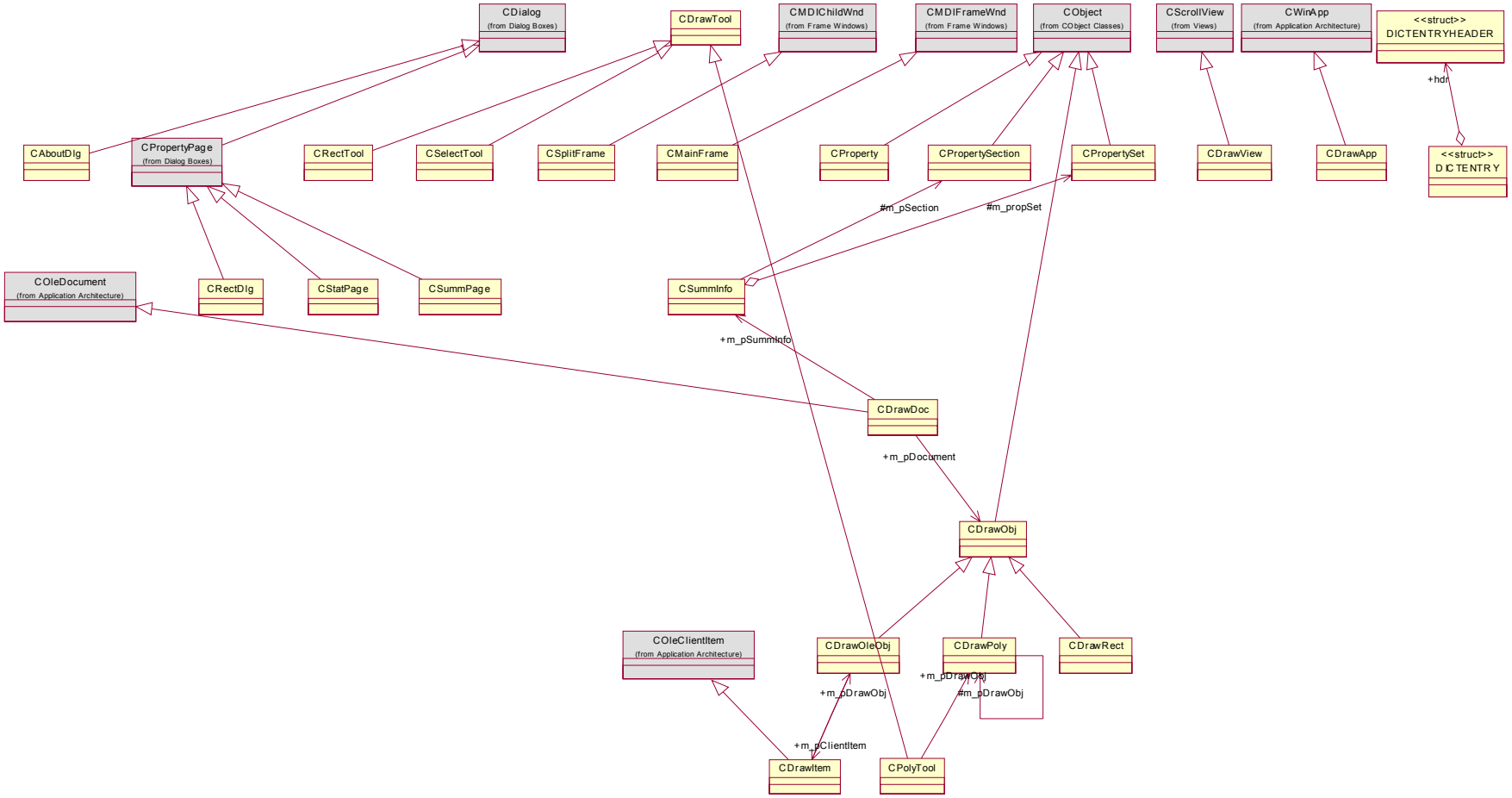
- Four-step process
  - Generate class diagram from implementation
  - Group related classes
    - Isolated classes
    - Aggregation, generalization, composition
    - Two-way associations
  - Package grouped classes into architectural elements
  - Determine partial system configuration

Example Application: ShareDraw

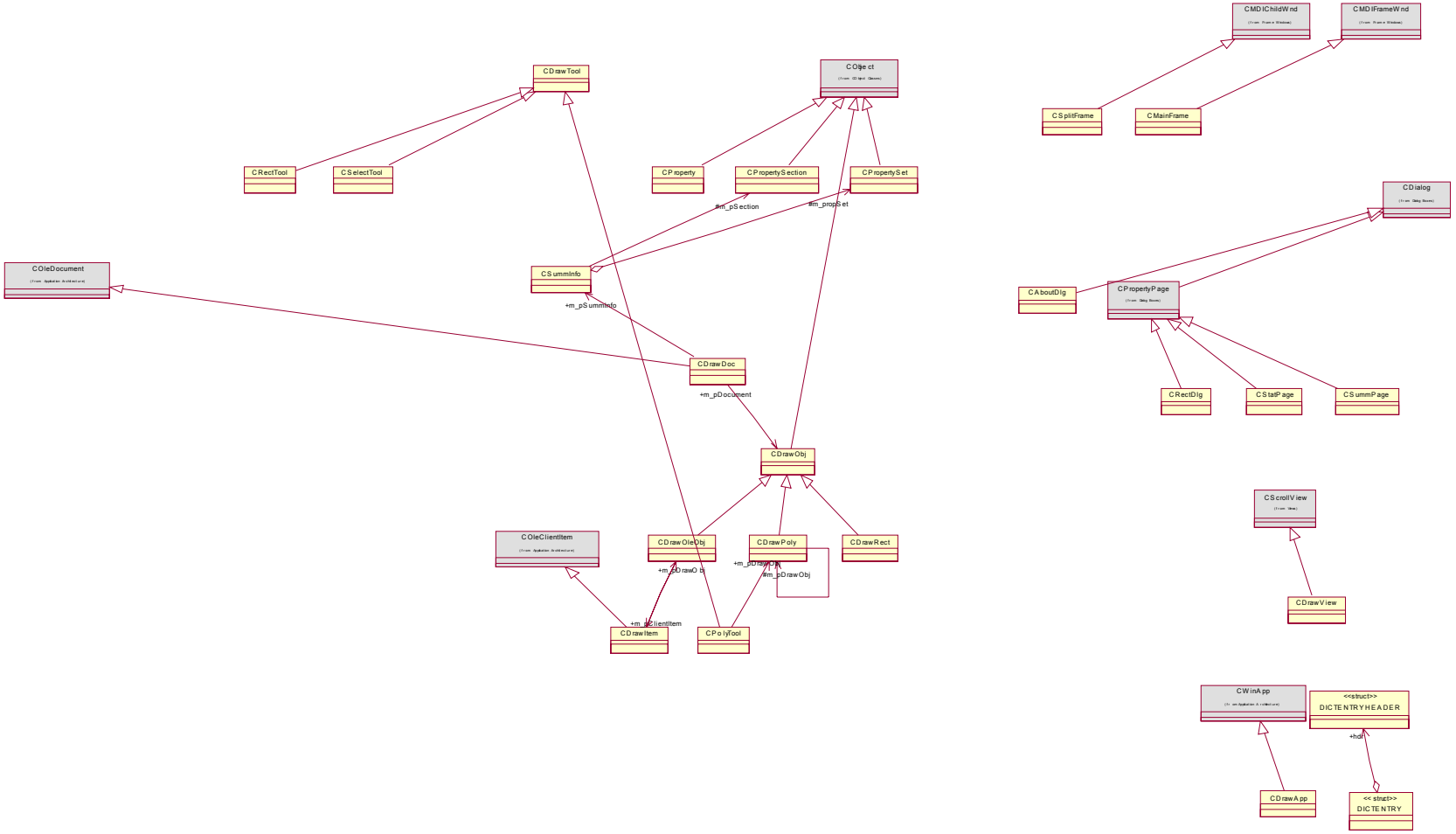




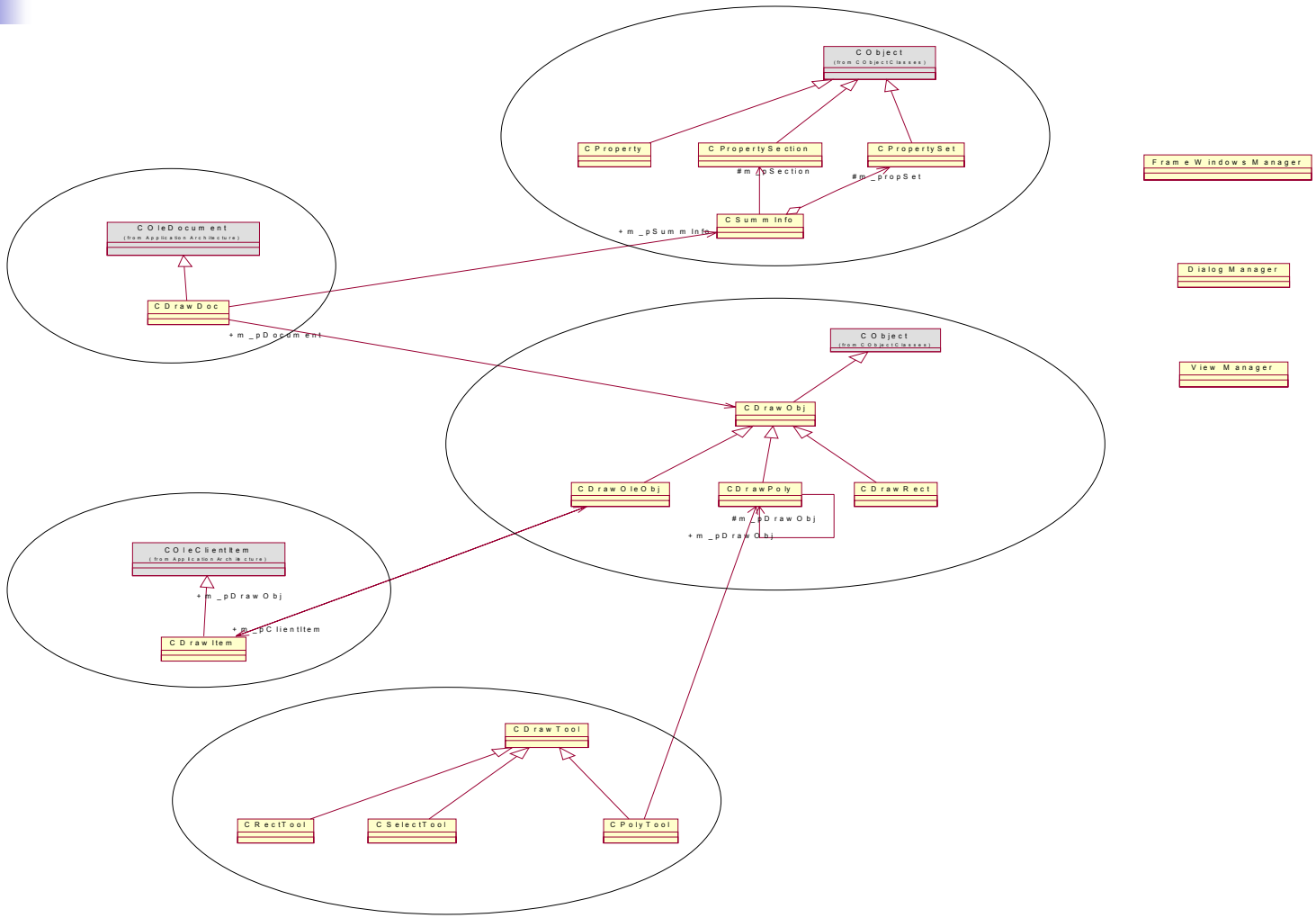
Recovered Class Diagram



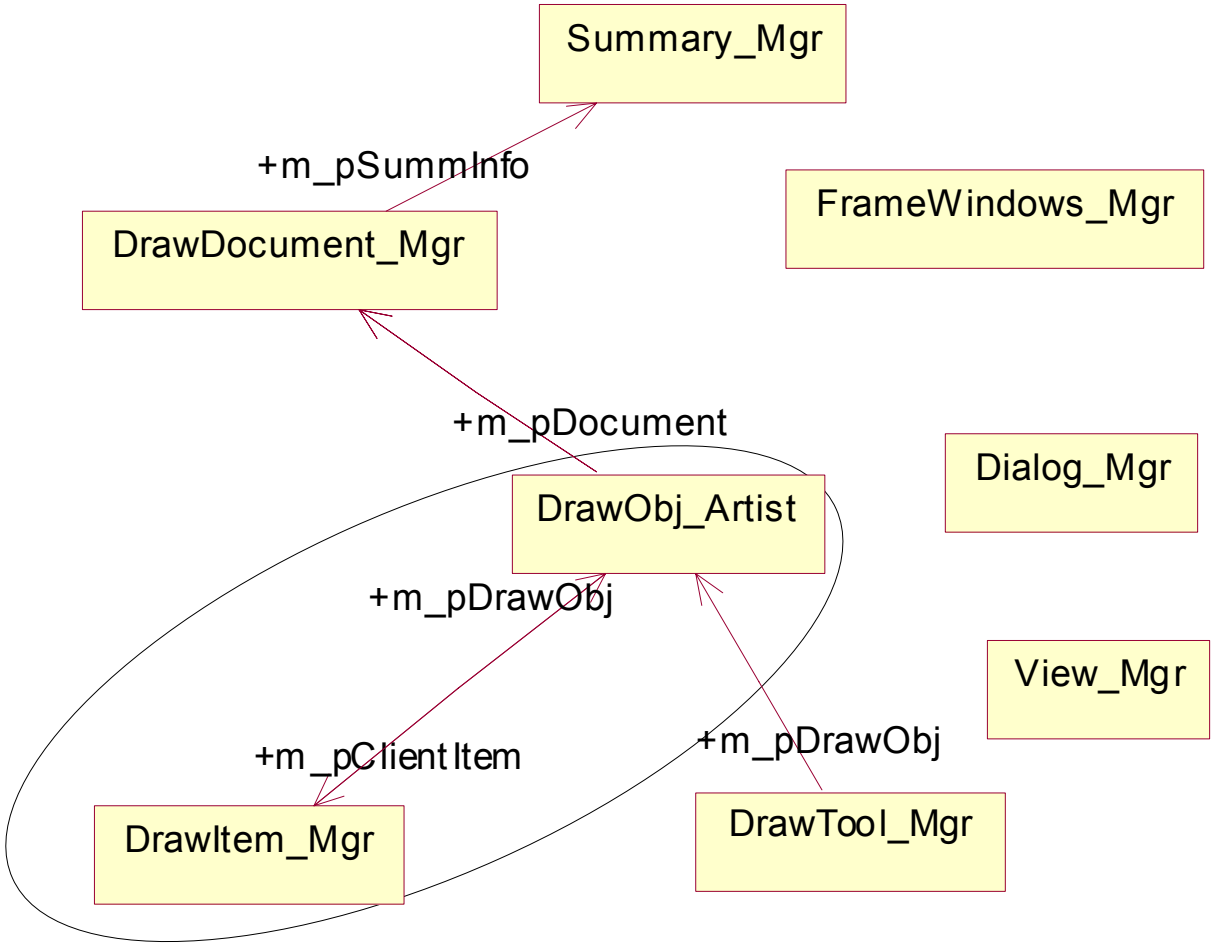
Component Recovery - Isolated Classes



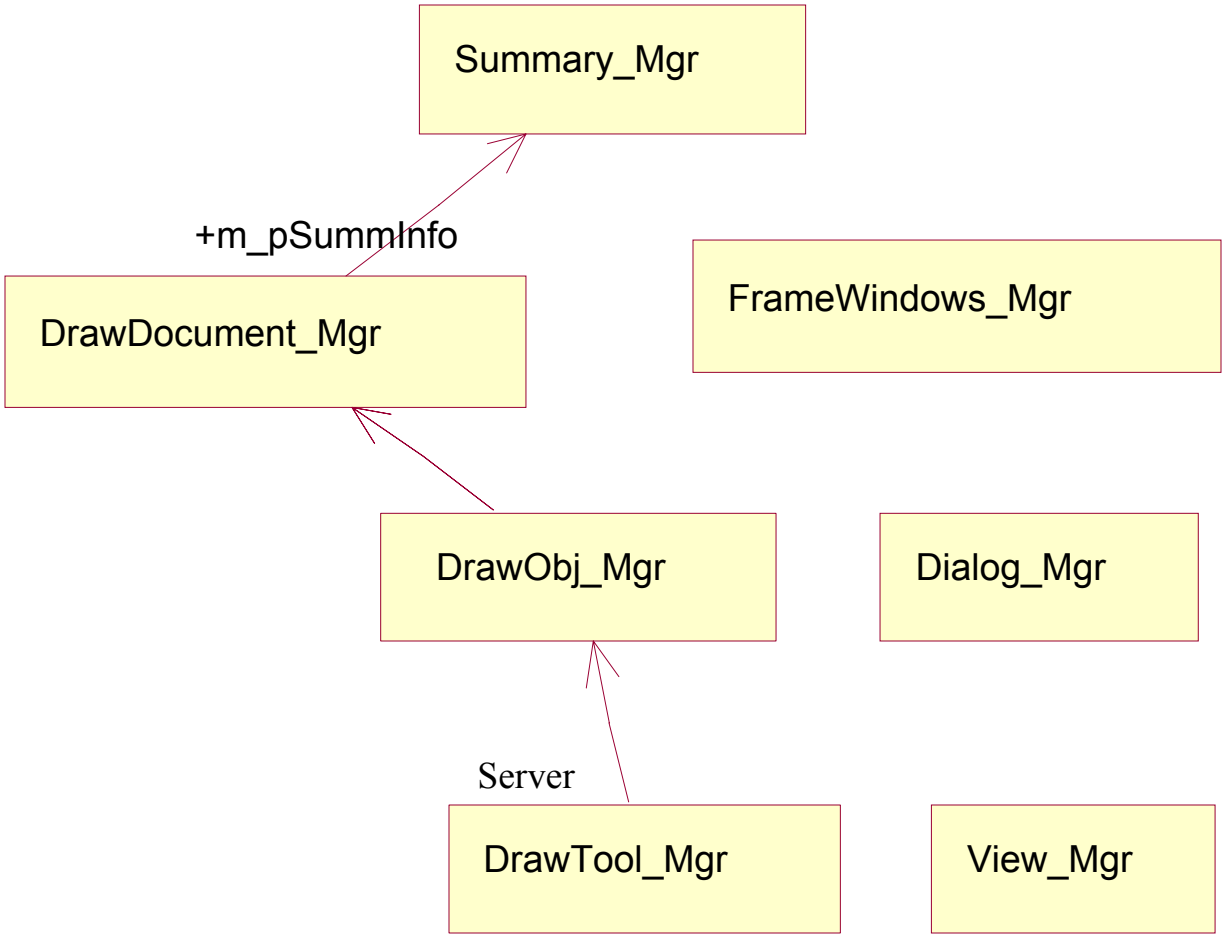
Component Recovery –
Generalization, Aggregation, Composition



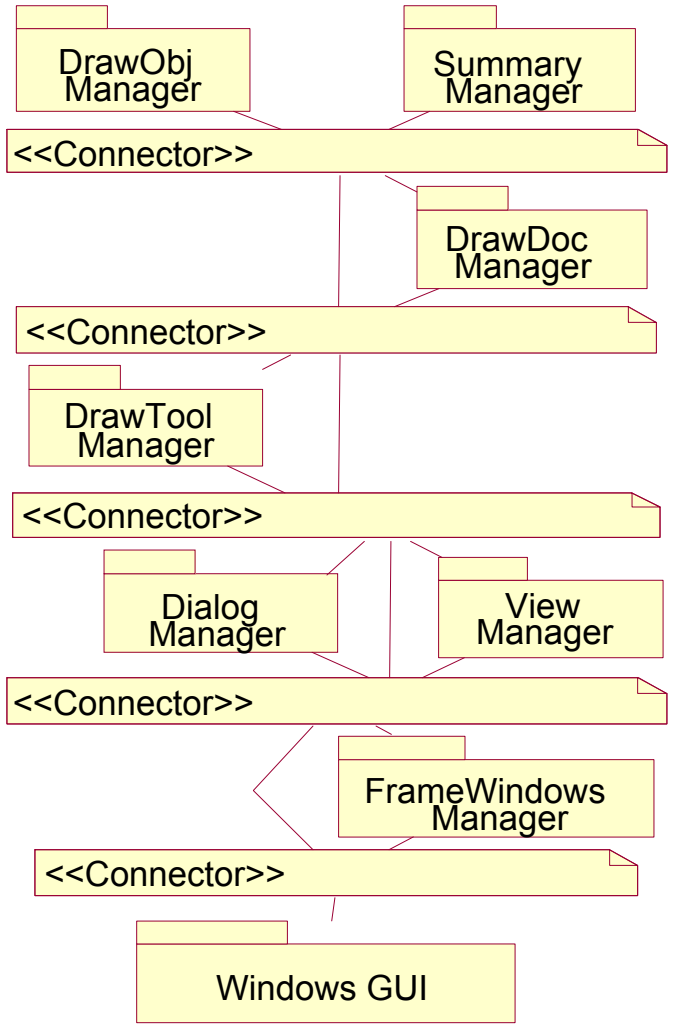
Component Recovery – Two-Way Associations



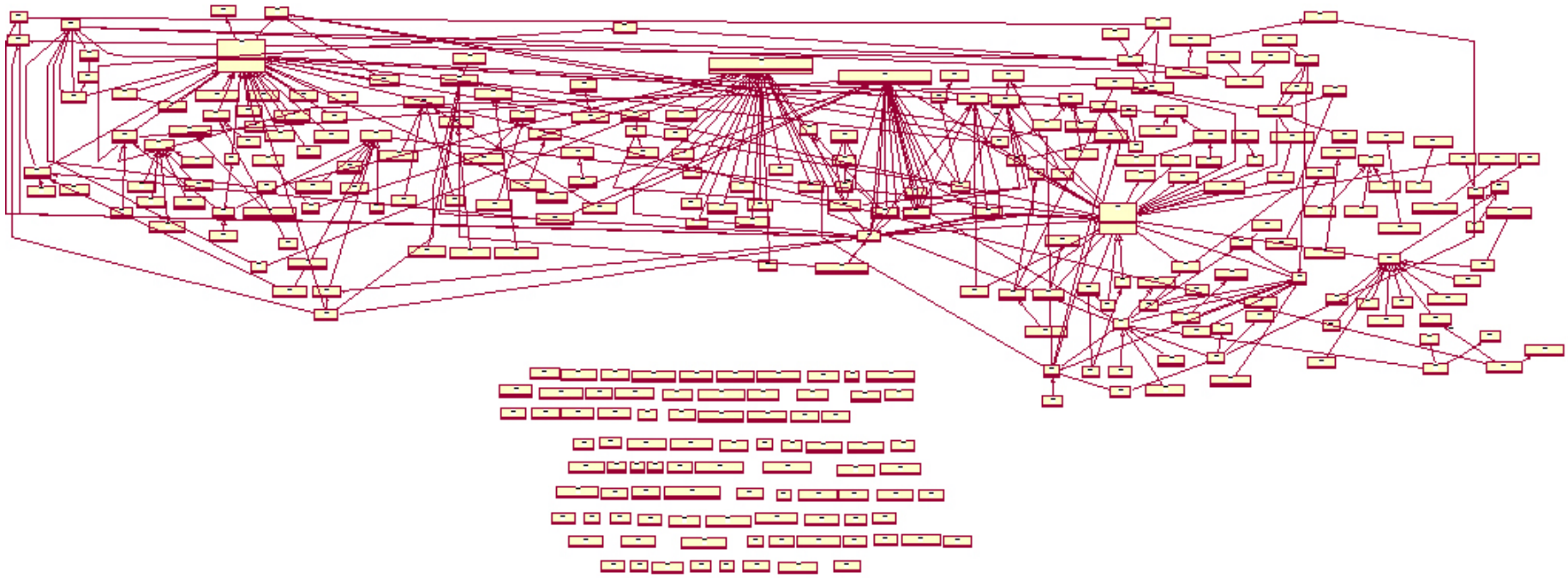
Component Recovery – Partial Configuration



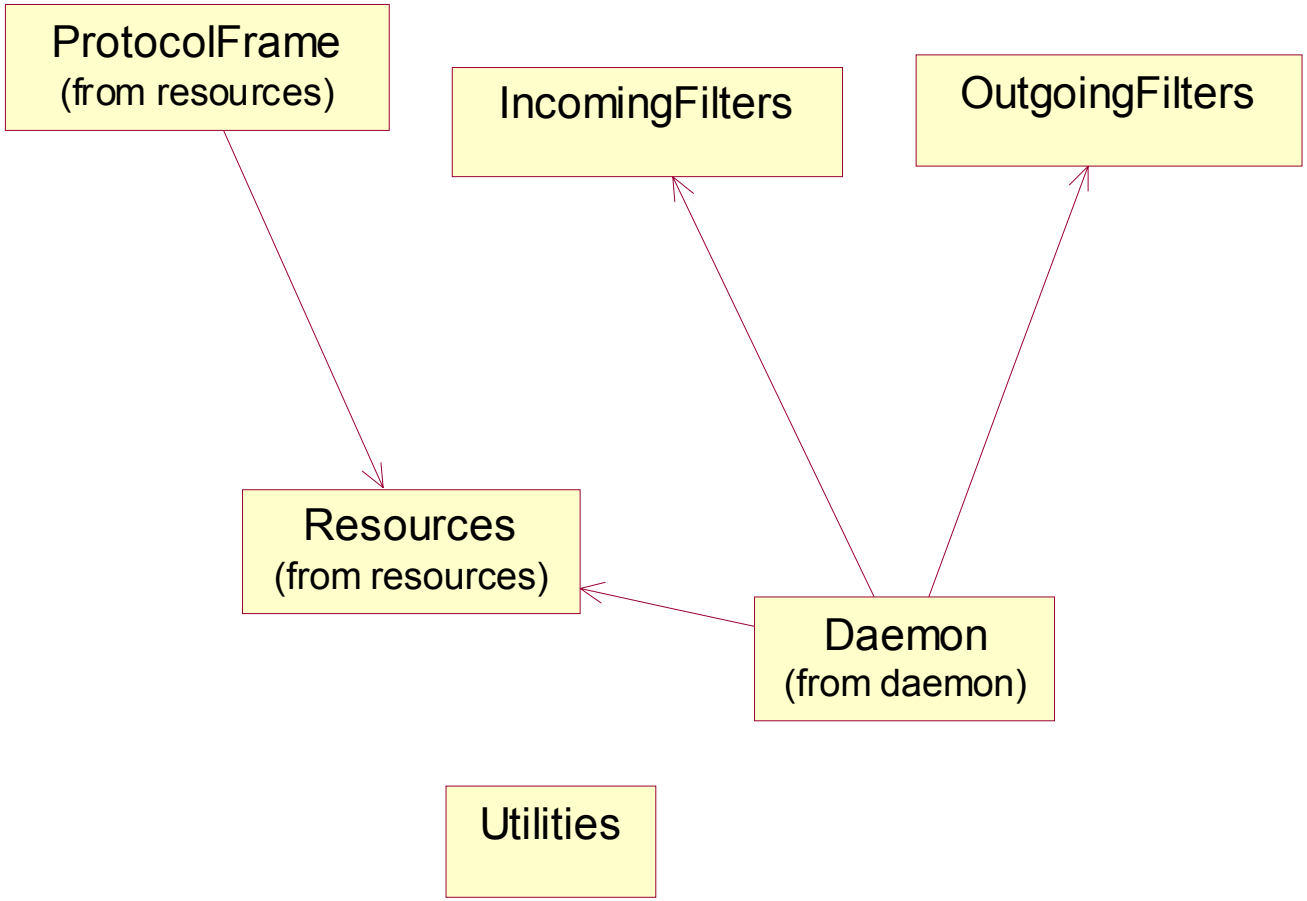
Architecture – C2

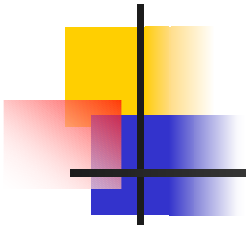


Jigsaw Web Server – from ~300 Classes…



Jigsaw Web Server – … to Partial Configuration





Further enhancements

- More precise architectural models can be obtained by:
  - Discovering the effective architectural style(s) for the system
  - Recovering connectors using the taxonomy of software connectors
  - Other sources of information
    - Domain knowledge
    - Dynamic behavior of a system