



Design of Functionality



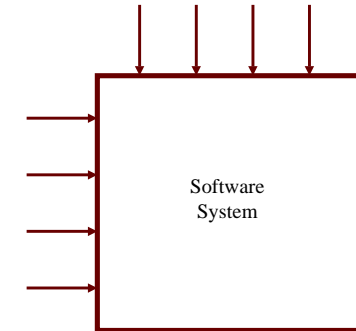
Orthogonal views

Structure View:

- Subsystems
- Classes
- Source Files
- Processes

Feature View:

- Abilities
- Functionality
- Use Cases



PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

2



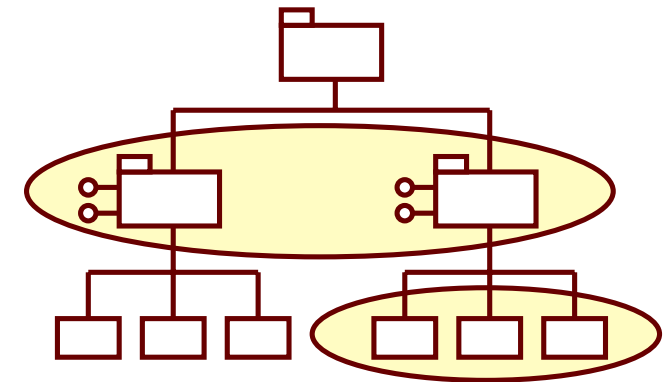
Designing Functionality

◆ Paradigm: Functionality is realized by a set of collaborating parts, modeling behaviour as well as state.

- ❑ Objects
- ❑ Processes
- ❑ ...



Collaboration Scopes



PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

3

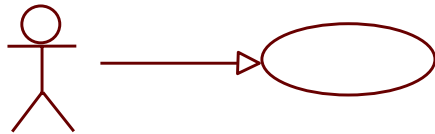
PVK--HT00

Copyright © 1997-1999, jubo@cs.umu.se/epltos@epl.ericsson.se

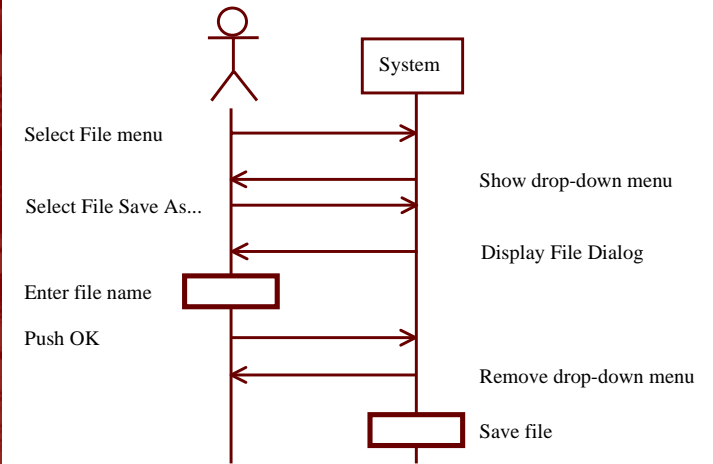
4



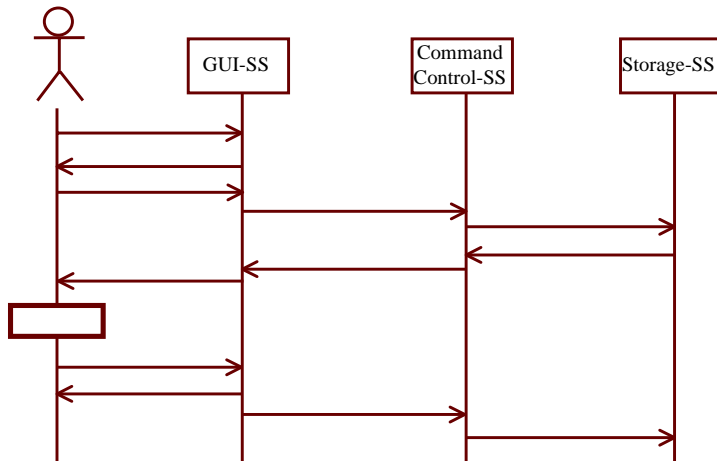
Example: Save File As....



Successful Scenario



Top Level Realization



Use Case Mapping Matrix

	Module 1	Module 2	Module 3	Module 4	Module ...
UC 1	X	X		X	
UC 2	X		X		X
UC 3				X	
UC 4	X				
UC ...		X	X	X	



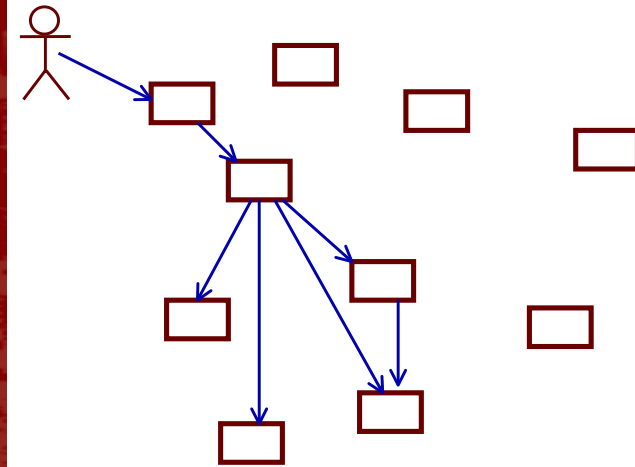
Realization Overlap?

	Module 1	Module 2	Module 3	Module 4	Module ...
UC 1	X	X		X	
UC 2	X		X		X
UC 3				X	
UC 4	X				
UC ...		X	X	X	

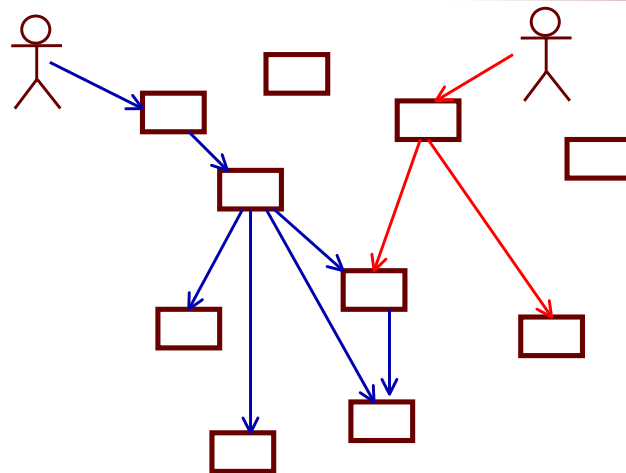
Same sequence?
Reuse opportunity!



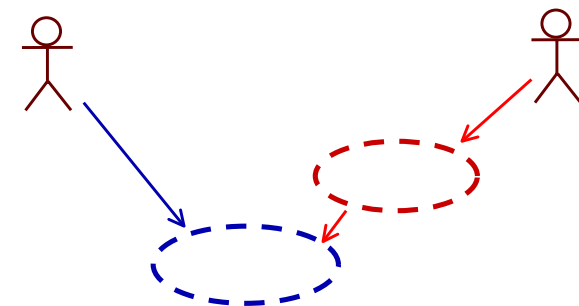
Collaboration (1)



Collaboration (2)

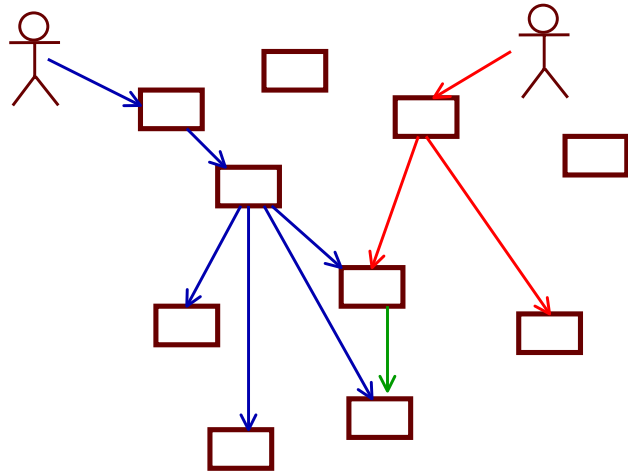


Collaboration (3)

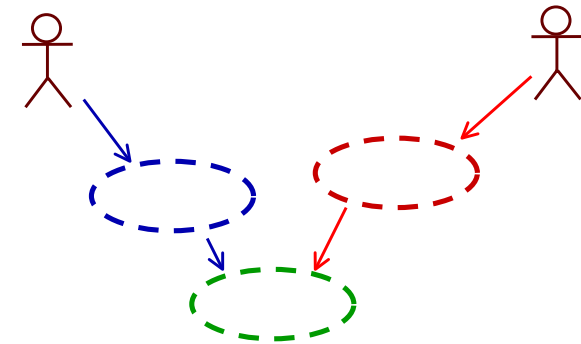




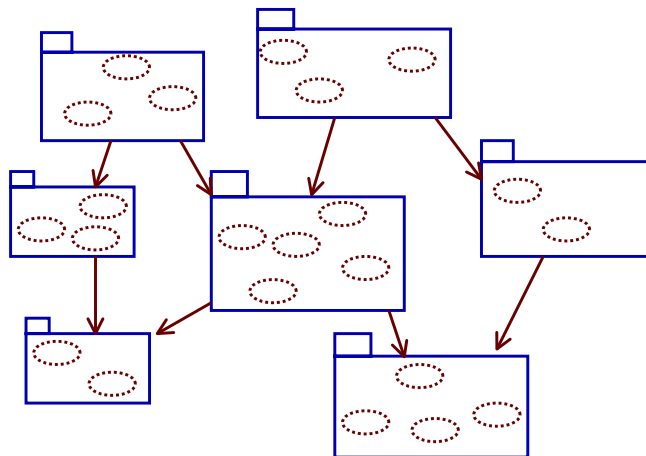
Collaboration (4)



Collaboration (5)



Anatomy*



*) Ericsson concept



Role of the Anatomy

- ◆ Traceability
 - Forward: Which code is executed by this certain scenario?
 - Backward: Which scenarios execute this certain code?
 - Backward traceability to all delivered systems!
- ◆ Planning:
 - PERT diagram directly deriveable from anatomy.



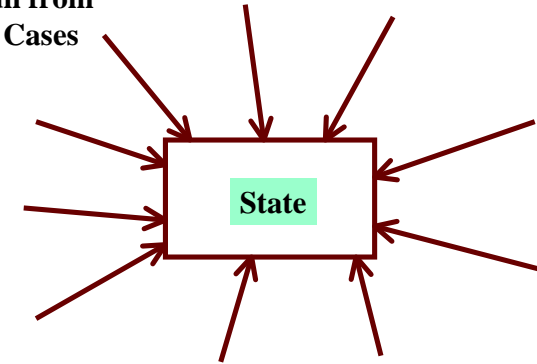
Interface Descriptions

- ◆ Assembles all requirements stated by use cases towards a certain module.
- ◆ Not mandatory. Helps identify reusable functionality.

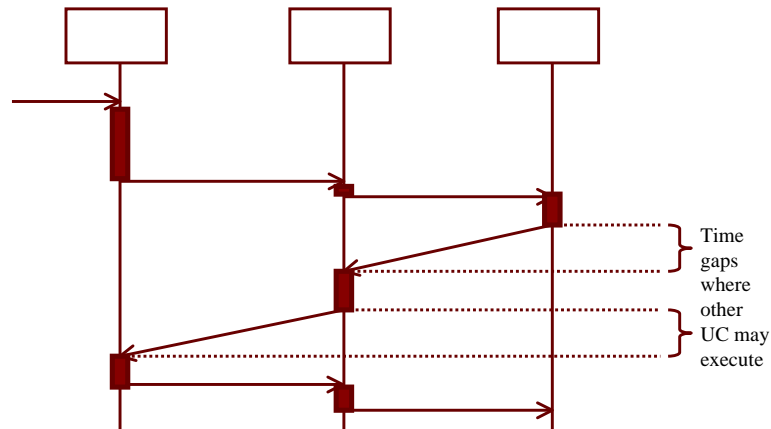


Module Design

Stimuli from Use Cases



Synchronous and Asynchronous Communication



Basic assumption:

Unless proven otherwise, module design shall assume that any stimuli may enter in any state.

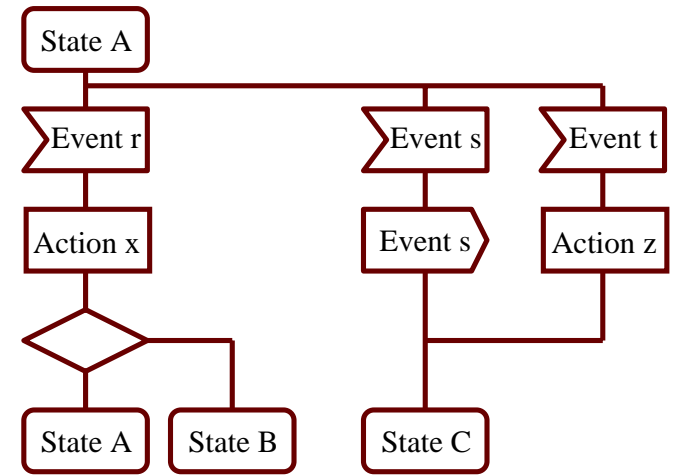


Event Driven Modelling

- ◆ Asynchronous state: Decomposition into multiple synchronous states.
- ◆ Transition time > time step: Decomposition of system transition into multiple internal transitions.
- ◆ SDL, UML (Harel State Charts)...
- ◆ Assumption: Memory is unlimited.
- ◆ Number of states in practice very high.
- ◆ Pseudo formalism!



Event-Transition (SDL notation)



The SDL Process

