

Unified Process

Architecture centric

Anders Kalhauge - Rolf-Helge Pfeiffer - Jens Egholm Pedersen



Fall 2017

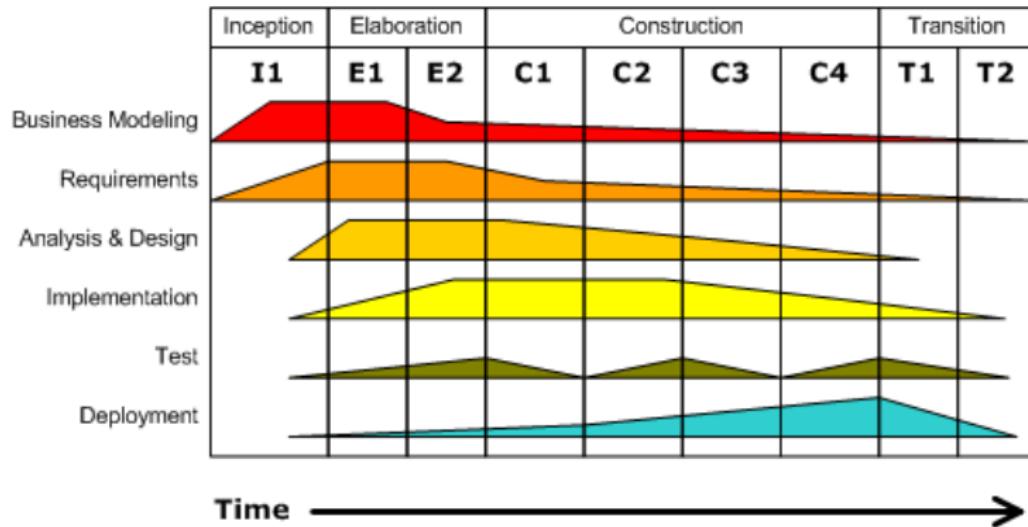
- Unified Process review
- Architecture
- Unified Modeling Language

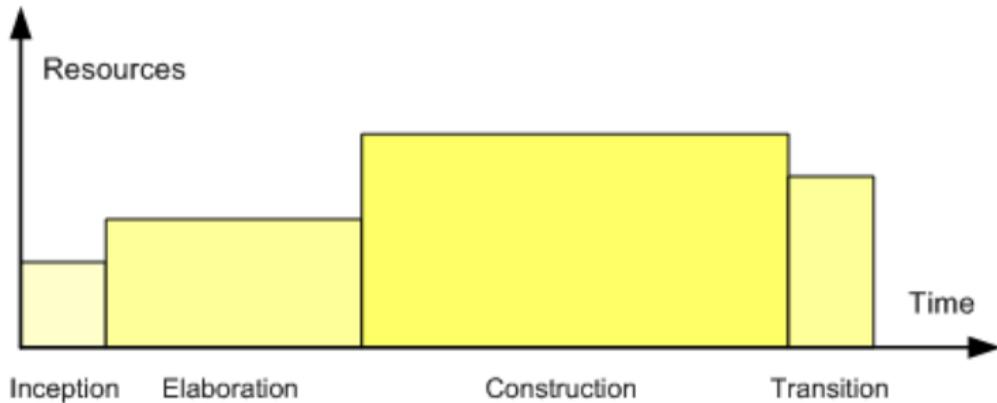
The Unified Process is

- Use-Case Driven
- **Architecture-Centric**
- Iterative and Incremental

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.





The goal in the inception phase is to make the business case to the extend necessary to justify launching the project.

- Cost/benefit

Will the gains accruing from the use or sale of the software product more than offset the cost of developing it?

- Time to market

Will it reach the market (or internal application) in time to obtain these gains?

- Resolve system scope

- Resolve ambiguities in the requirements needed in this phase

- **Establish a candidate architecture**

- Mitigate the critical risks

- A feature list.
- A first version of a business (or domain) model that describes the context of the system.
- A first cut of the models representing a first version of the use-case model, the analysis model, the design model. Of the implementation model and test model, there may be something rudimentary. There is also a first version of the supplementary requirements.
- A first draft of a **candidate architecture** description with outlines of views of the use case, analysis, design, and implementation models.
- Possibly a proof-of-concept exploratory prototype, demonstrating the use of the new system.
- An initial risk list and a use-case ranking list.
- The beginnings of a plan for the entire project, including a general plan for the phases.
- A first draft of the business case, which includes: business context and success criteria (revenue projection, market recognition, project estimate).

The goal in the elaboration phase is to capture most of the remaining requirements, formulating the functional requirements as use cases. Also a sound architectural foundation - the **architectural baseline** - must be established.

- Monitor remaining risks
- Fill in further details of the project plan.
- Judge the Worth of the Business Case

- Preferably a complete business (or domain) model which describes the context of the system.
- A new version of all models: use cases, analysis, design, deployment, and implementation. (At the end of the elaboration phase these models will be complete to less than 10% apart from the use case and analysis model that may include more (in some cases up to 80%) use cases to ascertain that the requirements have been understood. The majority of all use cases have been understood to make sure that no architecturally important use cases are left aside and that we can estimate the costs of introducing them.)
- An executable architectural baseline.
- An architecture description, including views of the use case, analysis, design, deployment, and implementation models.
- Updated risk list.
- Project plan for the construction and transition phases.
- A preliminary user manual (optional).
- Completed business case, including business bid.

The team working in the construction phase, starting from an executable **architecture baseline** and working through a series of iterations and increments, develops a software product ready for initial operation in the user environment.

- Project plan for the transition phase.
- The executable software itself—the initial-operational-capability release. This is the final build from construction.
- All artifacts, including models of the system.
- Maintained and minimally updated architecture description.
- Preliminary user manual in enough detail to guide beta users.
- Business case, reflecting situation at end of phase.

This phase focuses on establishing the product in the operational environment.

- Find out whether the system really does what the business and its users request.
- Discover unanticipated risks.
- Note unresolved problems.
- Find failures.
- Fix ambiguities and gaps in the user documentation.
- Focus on areas where users appear to be deficient and in need of information or training.

- The executable software itself, including installation software.
- Legal documents such as contracts, license documents, waivers, and warranties.
- Completed and corrected product release baseline including all models of the system.
- Completed and updated architecture description.
- Final user, operator, and system administrator manuals and training materials.
- Customer support references and web references on where to find more information, to report defects, and to find information on fixes and upgrades.

Problem domain

- Functional requirements
- Use cases (verbs)
- Domain model (nouns)

Solution domain

- Non-functional requirements
- Class diagrams
- Sequence diagrams
- State machine

Architecture centric Inception?



Architecture centric Elaboration?



Architecture centric Construction?

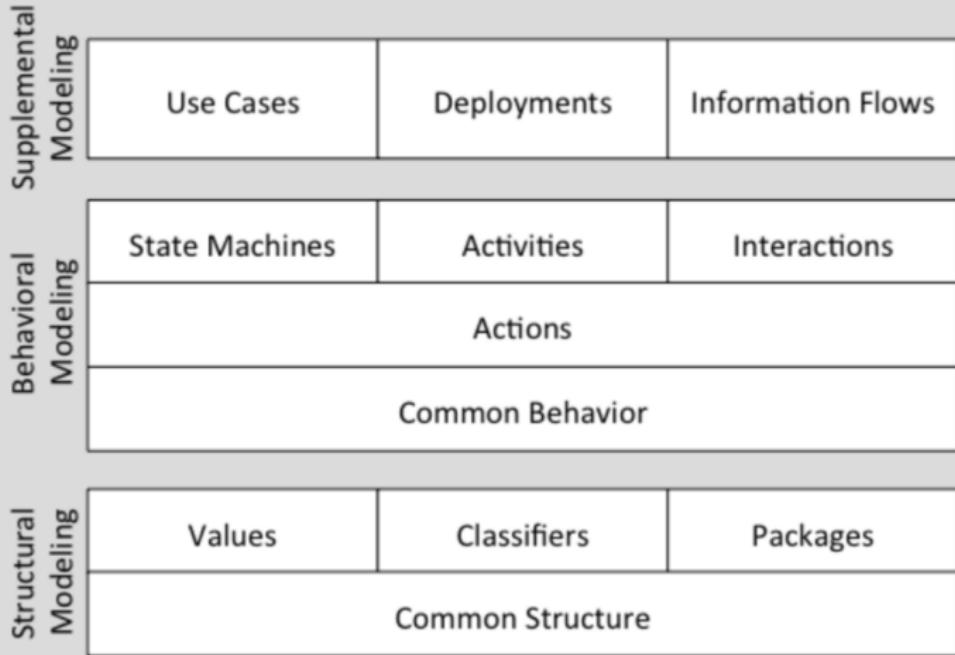


- Client/server
- Peer-to-Peer
- Repository
- Model/View/Controller
- Three (or more) tier Architecture
- Service-Oriented Architecture (SOA)
- Pipes and filters

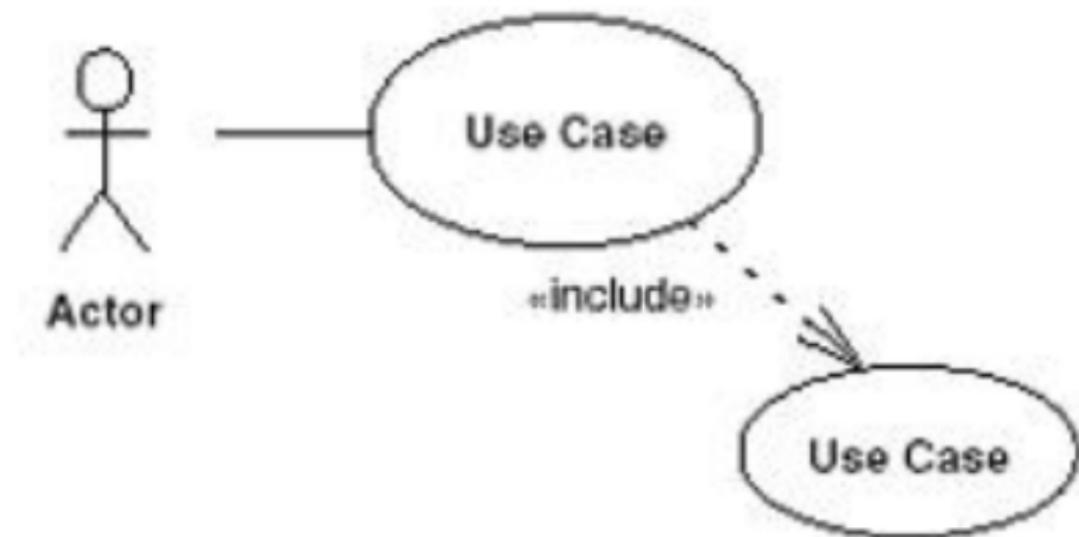
- Logical breakdown into subsystems
 - package diagrams
 - interfaces
- Dynamics of component interaction
 - system sequence diagrams
 - activity diagrams

- Data shared among subsystems
 - class diagram
- Run time components
 - component diagrams
 - deployment diagrams





Use Case Diagram p. 99



Class

Class Name

p. 35

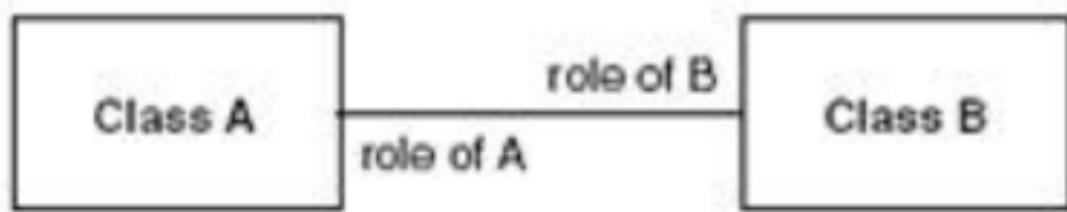
Class Name

attribute:Type[0..1] = initialValue

operation(arg list) : return type

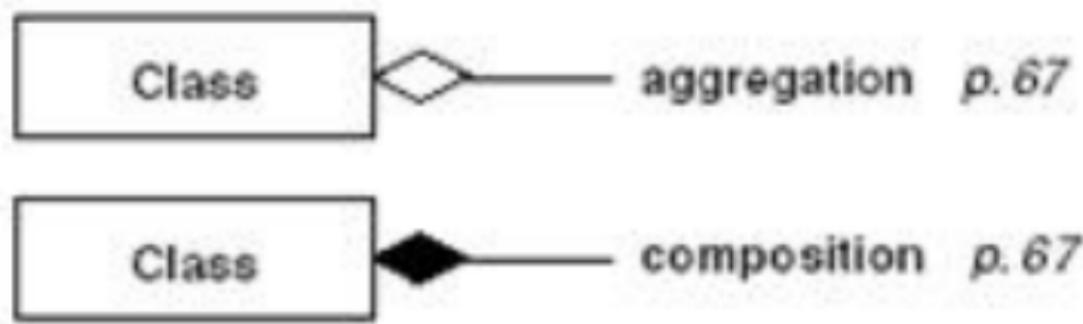
abstractOperation

Association p. 37

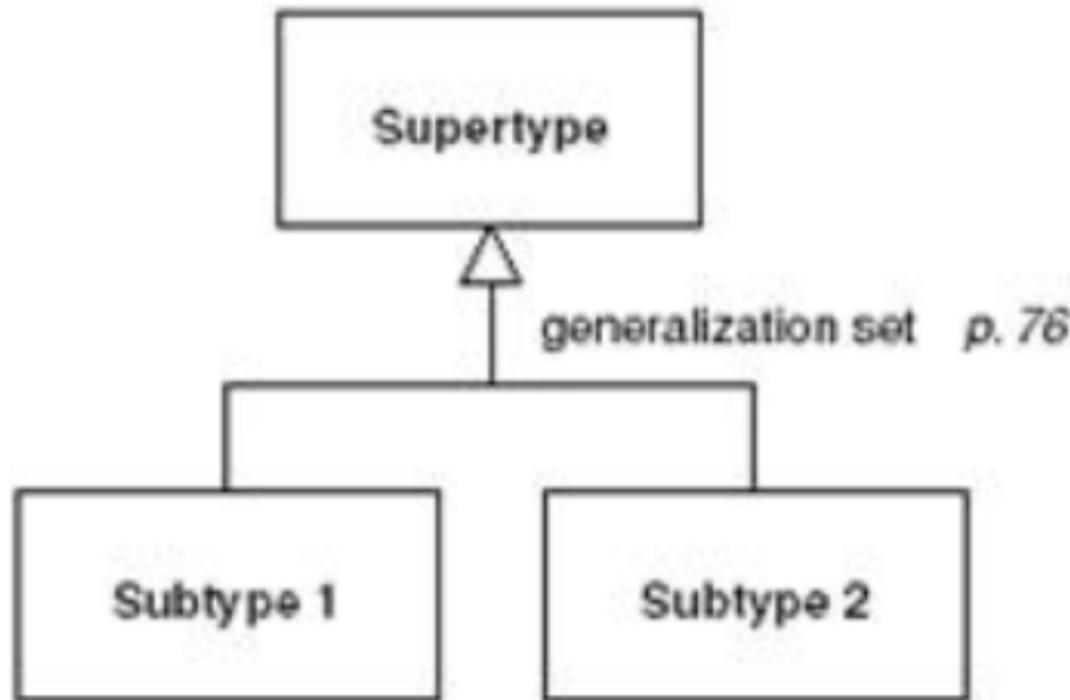


Class diagram 3

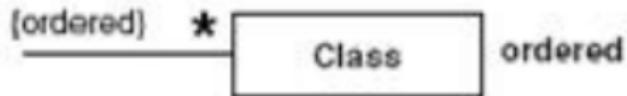
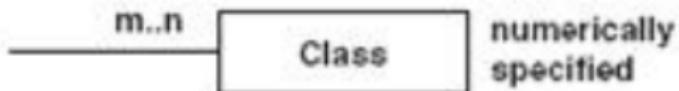
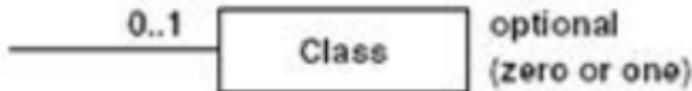
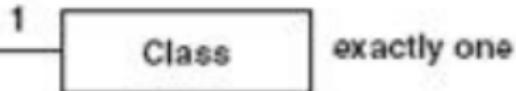
From UML distilled



Generalization p. 45



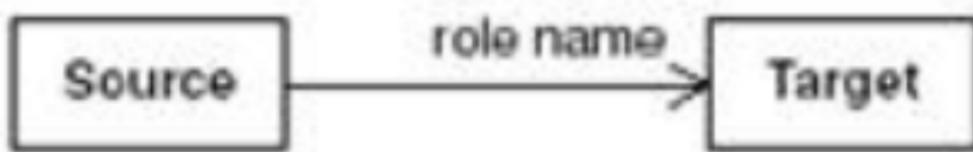
Multiplicities p 38



Qualified Association p. 74



Navigability p. 42

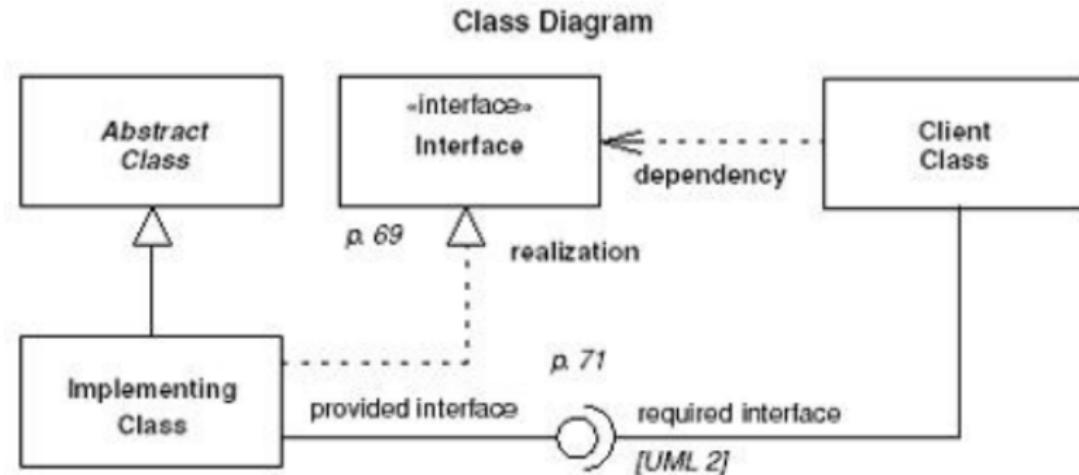


Instance Specification p. 87

object name: Class Name

Class diagram 8

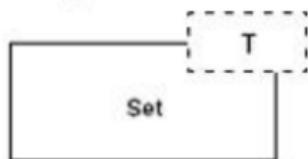
From UML distilled



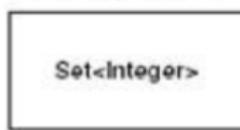
Class diagram 9

From UML distilled

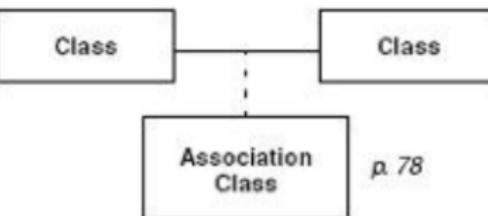
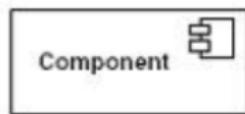
template class p. 81



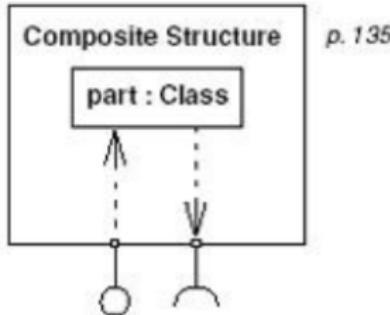
bound element



p. 139



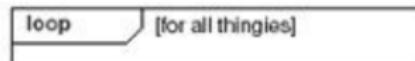
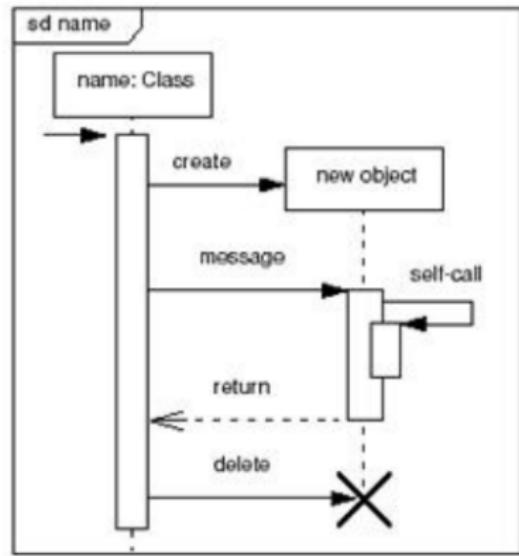
p. 78



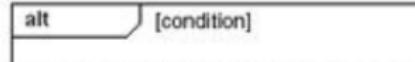
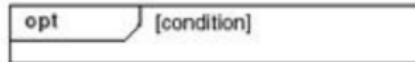
p. 135

Sequence diagram

From UML distilled

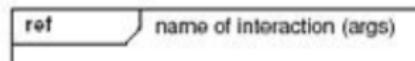


p. 57



[other condition]

[else]



synchronous p. 61

asynchronous [UML >= 1.4]

asynchronous [UML <= 1.3]

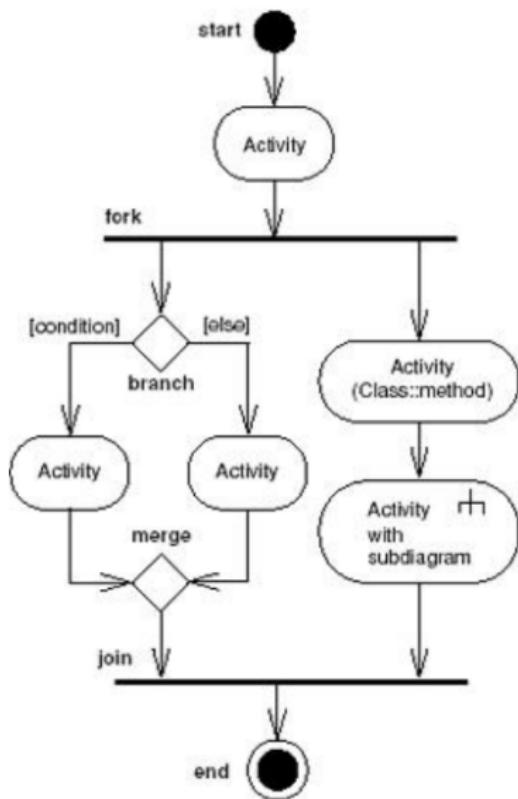
*: iteration message ()

[condition] message ()

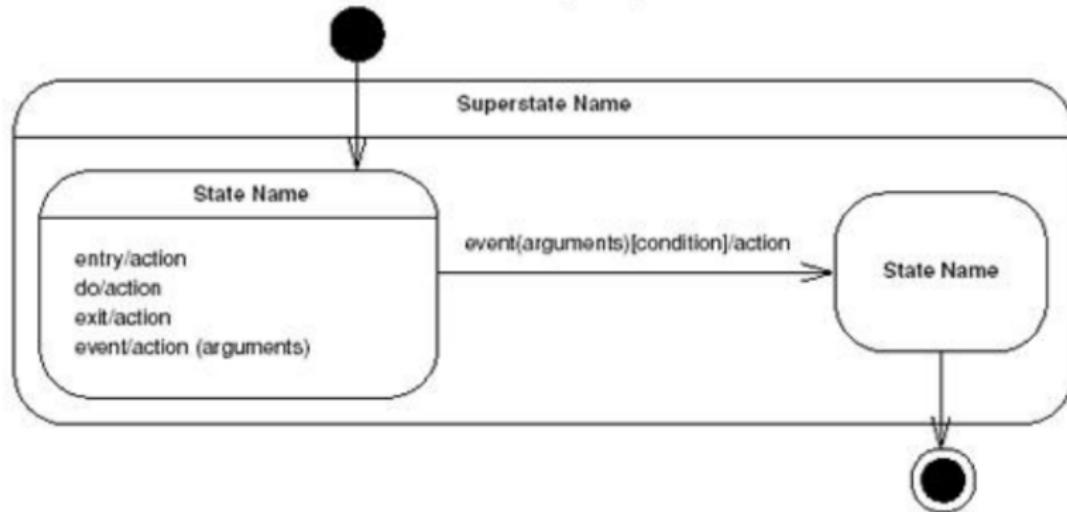
[UML 1] p. 59

Activity diagram

From UML distilled

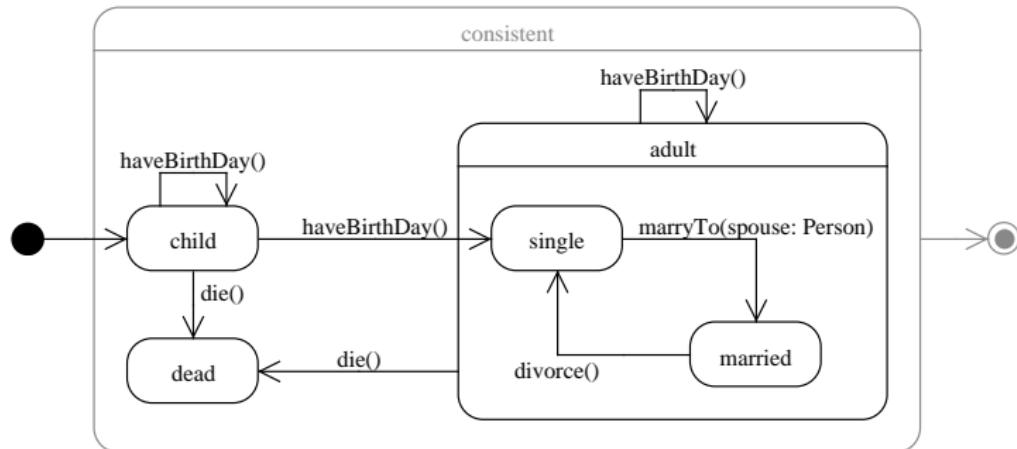


State Diagram p. 107



State diagram

An example

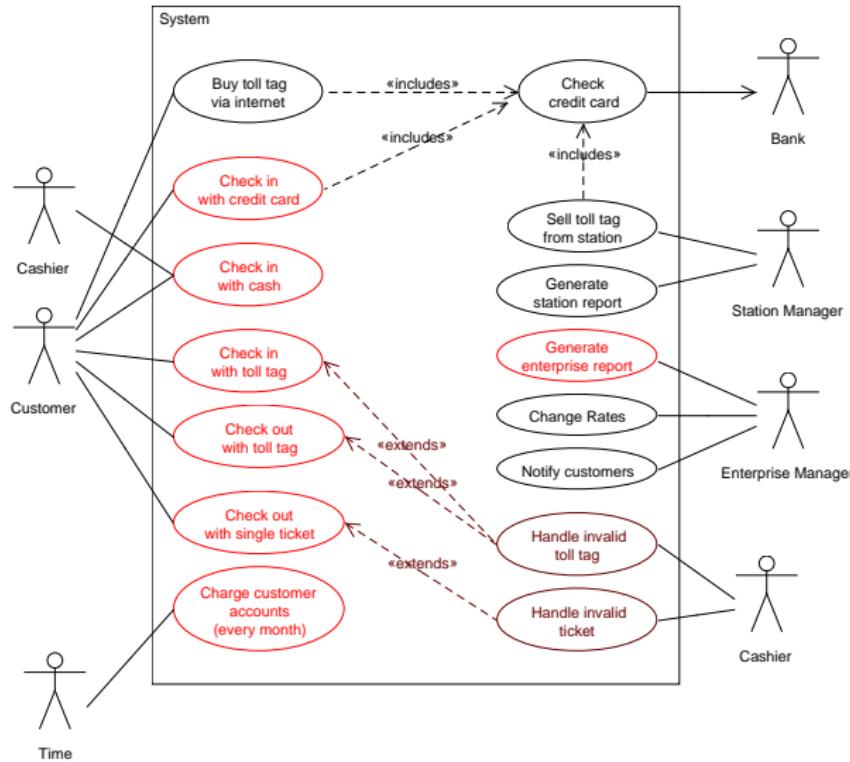


Example

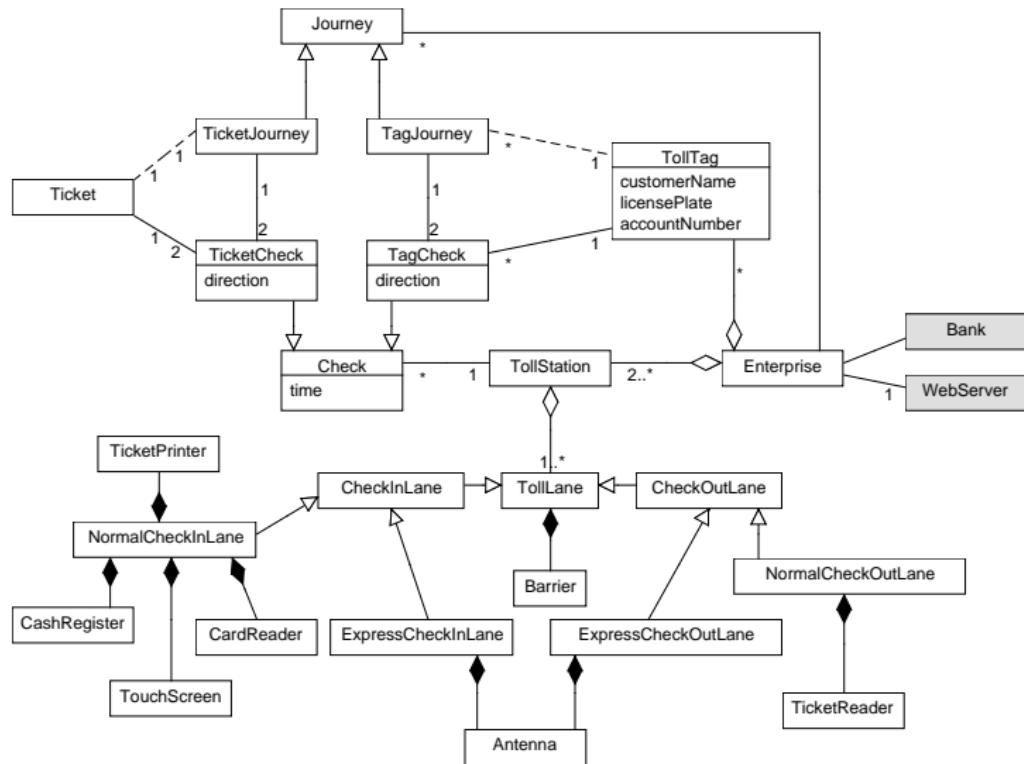
Highway toll system



Use Case Diagram



Domain model



- **Use Case Name** Check in with credit card
- **Summary** The customer wants to enter the highway. To do so, he needs to pay a toll and he wants to pay using a credit card. The customer uses the computer to set the type of vehicle and uses the card reader to pay.
- **Actors** Customer
- **Preconditions**
 - The customer is in possession of a credit card
 - The customer is at a normal check-in lane

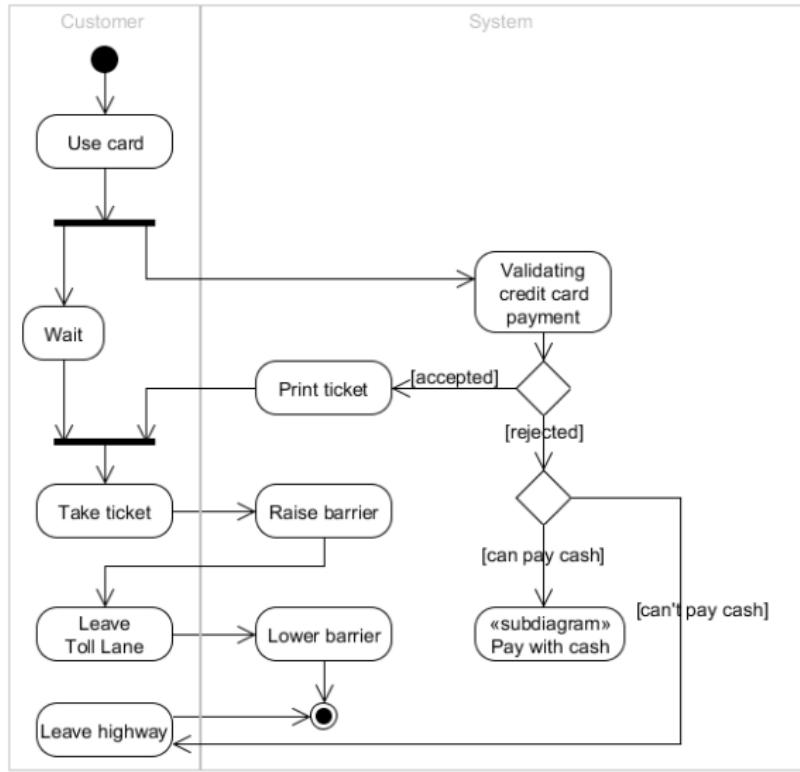
- **Main scenario** Check in with credit card

- ① The customer enters the type of vehicle into the system.
- ② The customer verifies his choice and continues, or reselects (back to step 1).
- ③ The amount of money needed to be paid is shown to the customer
- ④ The customer uses his card on the card reader
- ⑤ The device validates the card payment
- ⑥ Ticket is printed.
- ⑦ The customer picks the ticket.
- ⑧ The barrier is raised.
- ⑨ The customer leaves the toll lane and the barrier is lowered.

- **Alternative** Check in with credit card
 - 2a. The customer continues with wrong type of vehicle.
The use case continues as normal
 - 5a. The credit card is not valid for payment
 - ① A cashier is summoned
 - ② The cashier decides how to check in the customer (e.g. by receiving cash or not letting the customer on the motorway)
- **Post conditions**
 - Information about the check-in has been registered at the toll station.

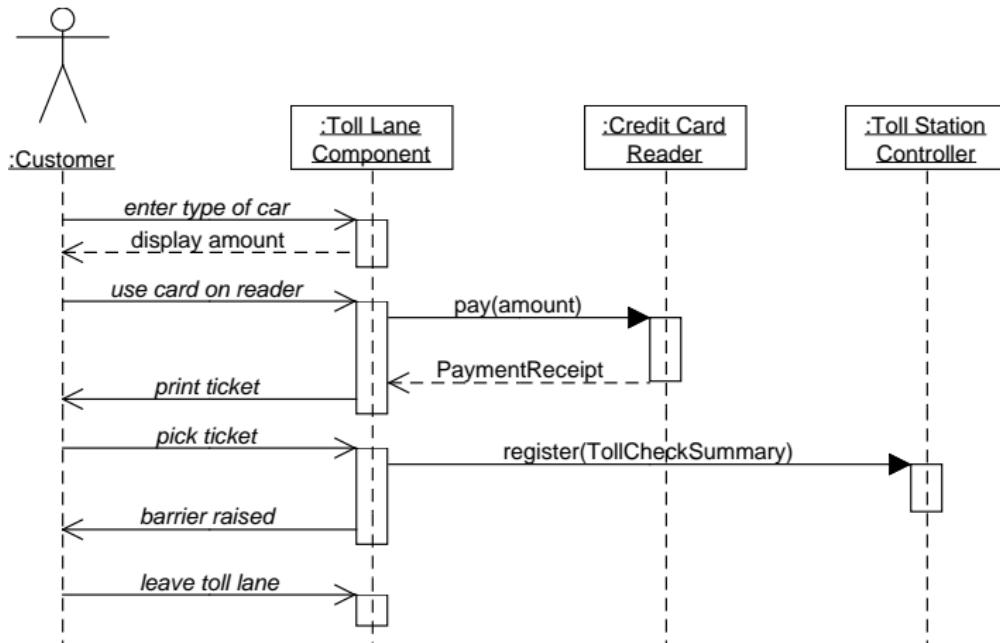
Activity Diagram

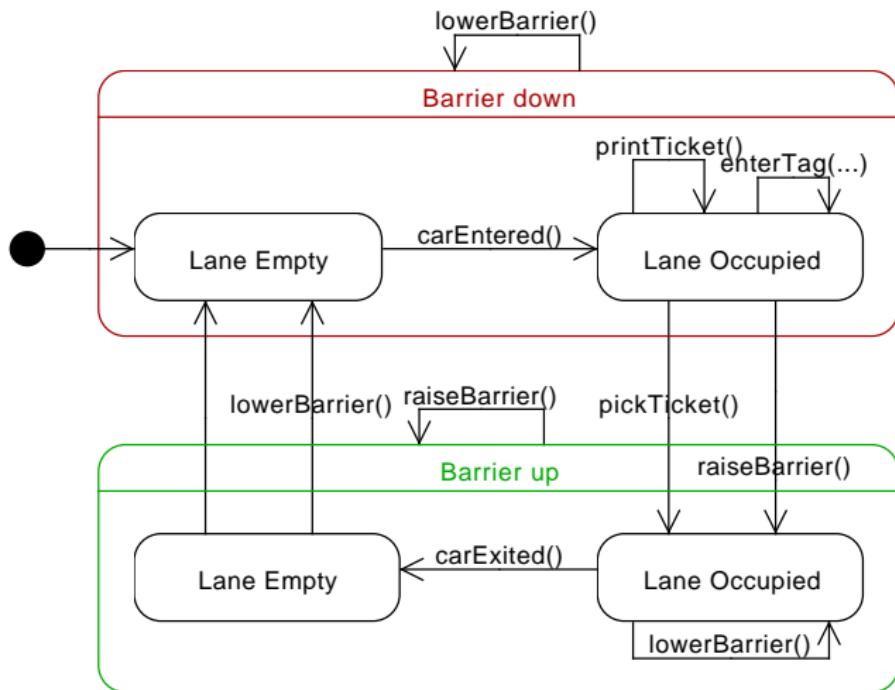
Check in with credit card



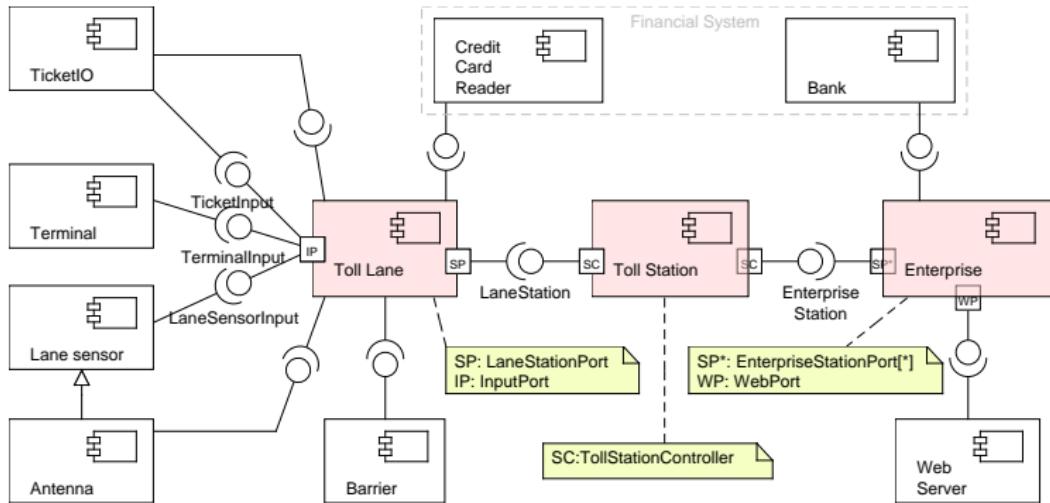
Sequence Diagram

Check in with credit card





Component Diagram



Design Class Diagram

Toll lane component

