The Unified Process

The Unified Process is a traditional incremental design driven by constructing views of a system architecture.

It is component based,
used to coordinate object oriented programming projects.

It uses UML - a diagrammatic notation for object oriented design - for all blueprints.

The design process driven by, use-cases which help keep sight of the anticipated behaviors of the system.

Design is iterative and incremental -uses sequence of design phases within a cyclic process.

Phases of Design Cycles

Design in the Unified Process proceeds through a series of cycles

Inception : produces a commitment to go ahead. By the end of this phase a concept model is agreed with system bountries.

Elaboration : takes us to a working specification of the system. By the end of this phase a basic architecture should have been produced; a plan of construction agreed; all significant risks identified;

Construction : produces beta-release system. By the end of this phase a working system should be available, sufficient for preliminary testing under realistic conditions.

Transition : introduces the system to its intended users.

Within these phases we may go through a number of iterations, each involv-ing the normal forms of workflow activity (requirements specification, analysis, design, implementation and testing). Includes series of nine iterations covering the four phases described above.

A principal product of the Unified Process is a series of models, each appropriate to a key stage in system design. Since many different models are produced, each for a different design purpose but all related to the same system.

business case should have been made; feasibility of the project assessed; and the scope of the design should be known.

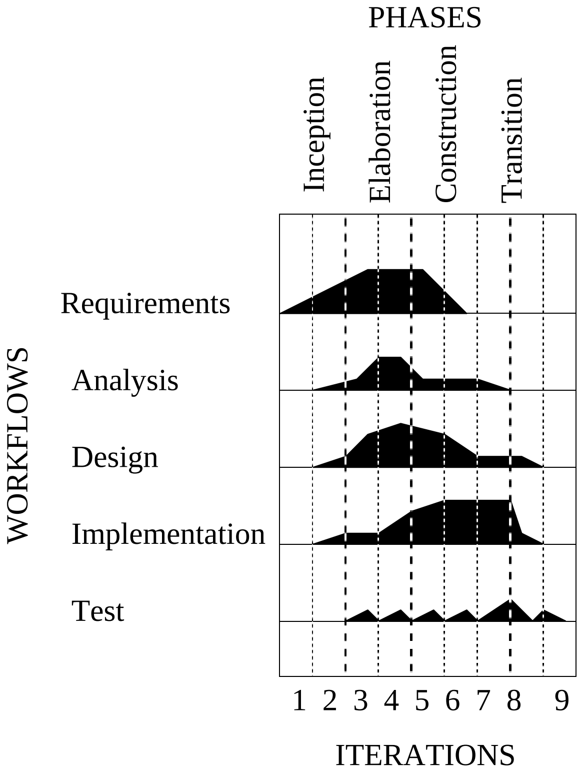
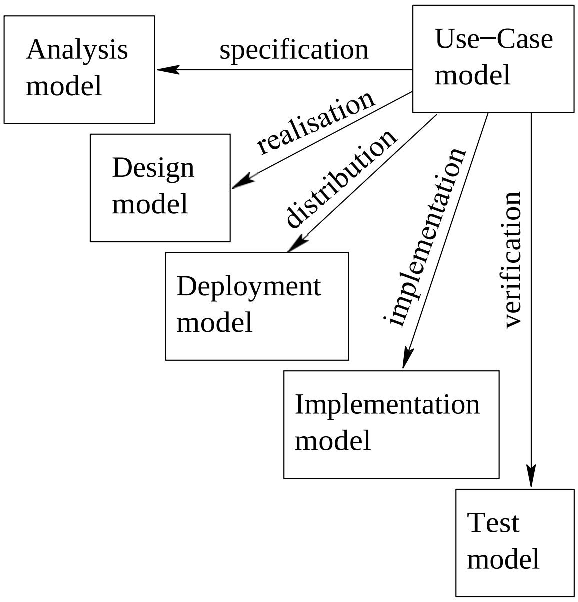


Figure 3.1: Phases in a cycle of the Unified Process



The purpose of a use case is to describe the functionality required of the system from the point of view of the client

Requirements capture.

Analysis and design of how system realises use cases.

Acceptance/system testing.

Planning of development tasks.

Traceability of design decisions back to use cases.

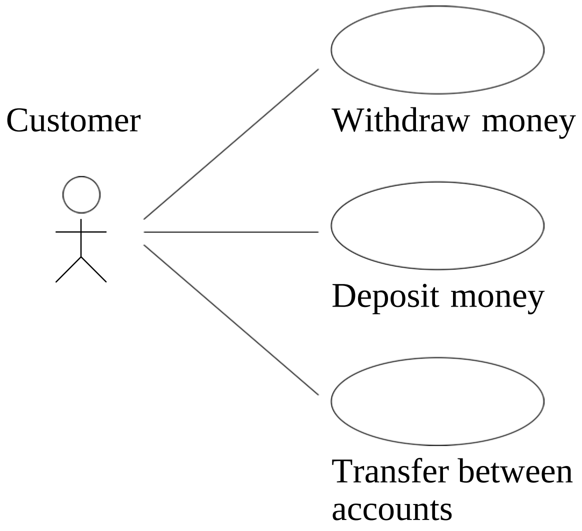
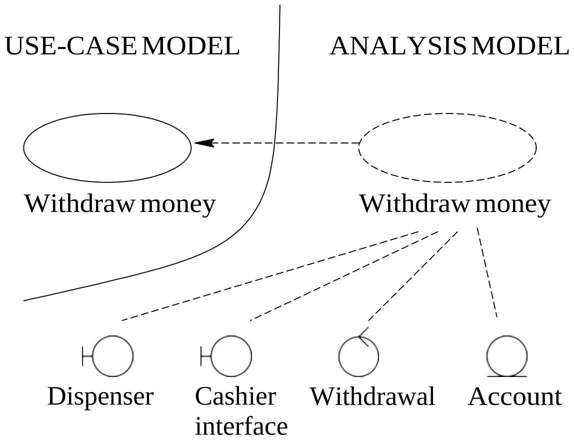


Figure 3.3: Initial use case diagram



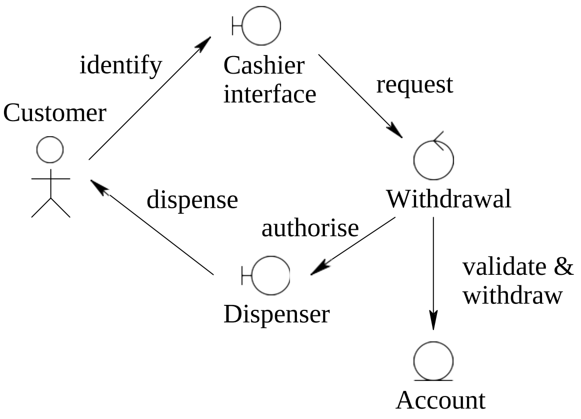
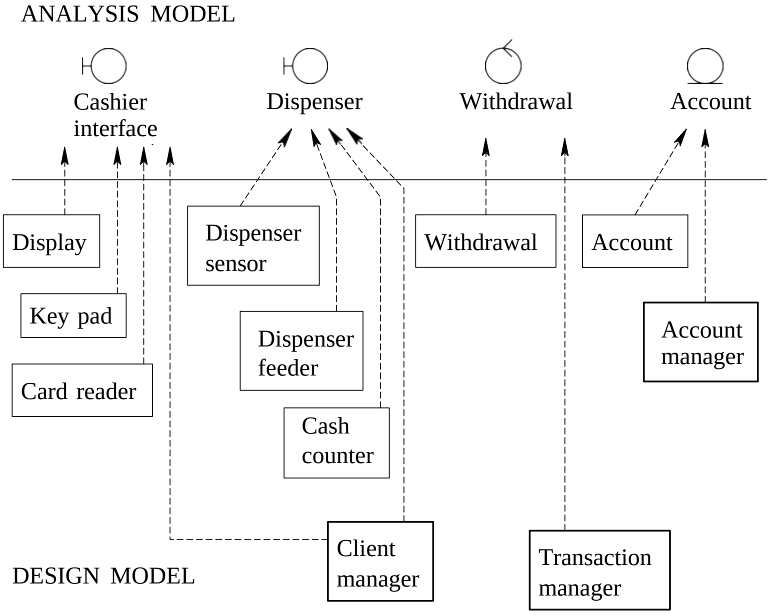


Figure 3.5: Collaboration model for withdrawing money



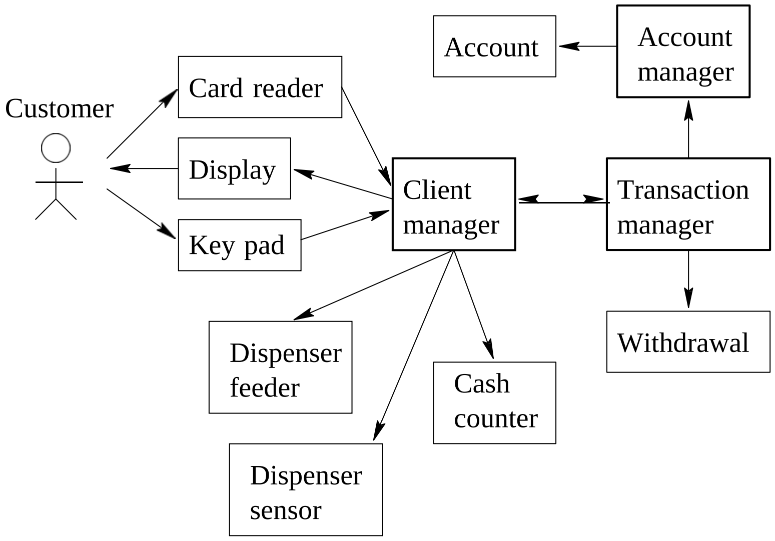
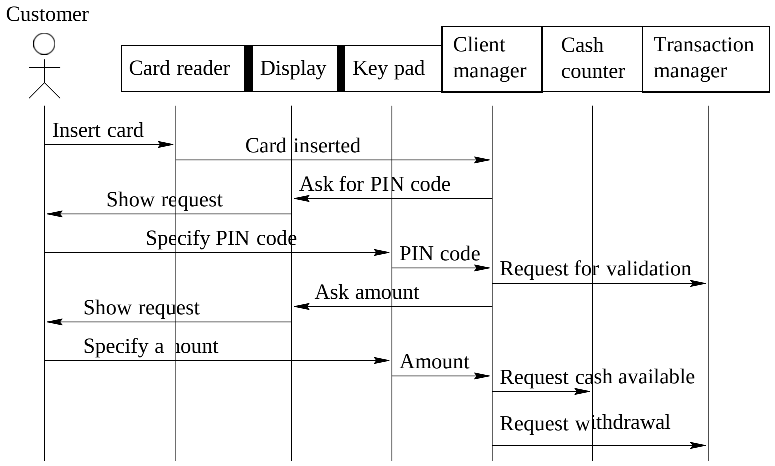


Figure 3.7: Interactions between design classes - part of design realisation



A sequence diagram for part of the realisation

Details in Jacobson, Booch & Rumbaugh *et.al.* 1998, *The Unified Software Development Process.*