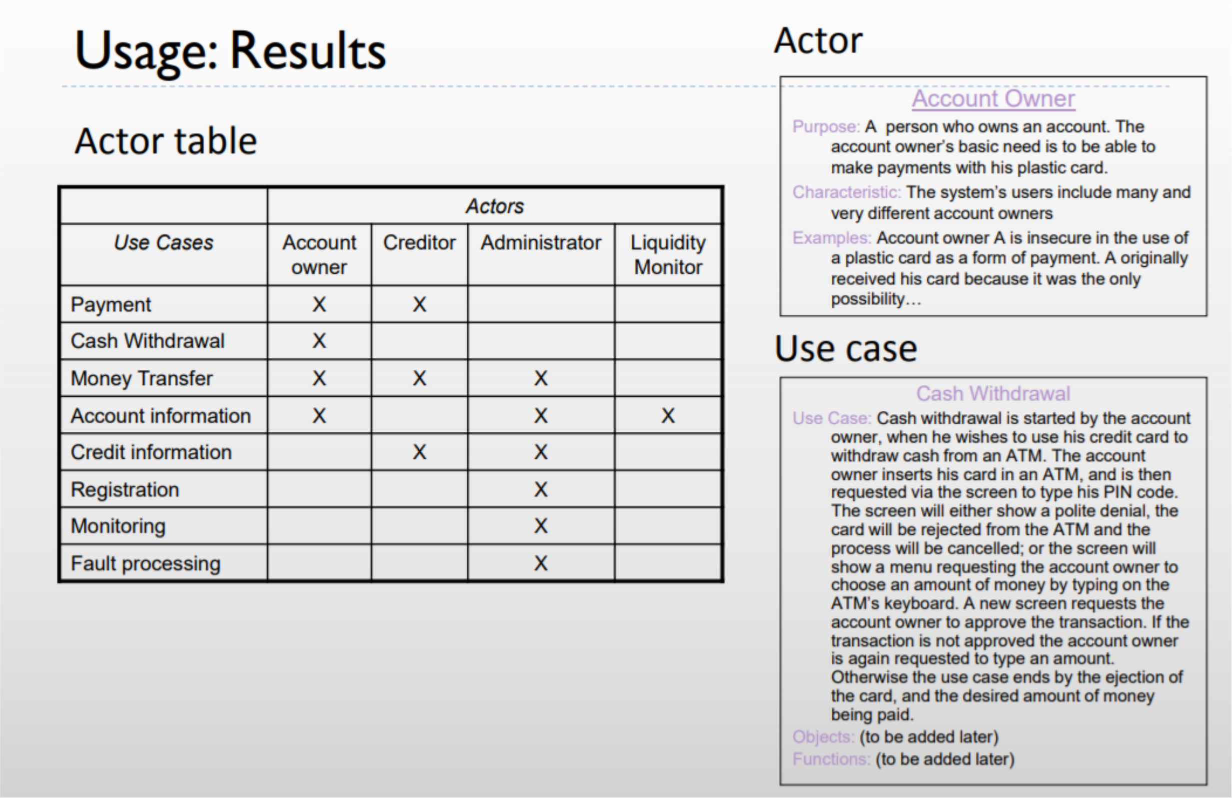
III: 6: Usage: Actors, Functions, Work Tasks, Use Case, Patterns (Procedual, Material).

Event tables: * = 0 or more, + = 0 or 1 time.

Application Domain Analysis: Results

- Actors (users and other systems) and Use case



Usage: Results

Actor table

	Actors			
Use Cases	Account owner	Creditor	Administrator	Liquidity Monitor
Payment	X	X		
Cash Withdrawal	X			
Money Transfer	X	X	X	
Account information	X		X	X
Credit information		X	X	
Registration			X	
Monitoring			X	
Fault processing			X	

Actor

Account Owner

Purpose: A person who owns an account. The account owner's basic need is to be able to make payments with his plastic card.

Characteristic: The system's users include many and very different account owners

Examples: Account owner A is insecure in the use of a plastic card as a form of payment. A originally received his card because it was the only possibility…

Use case

Cash Withdrawal

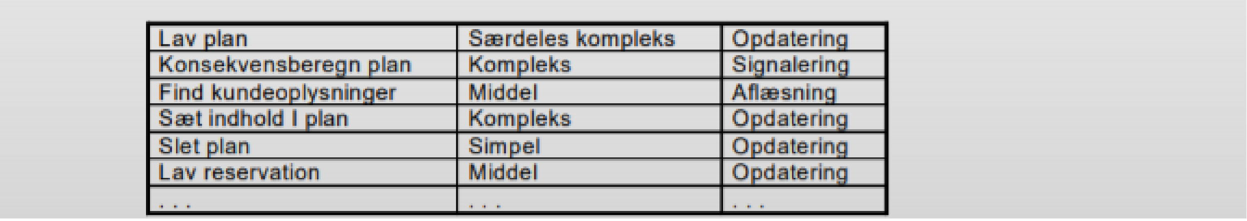
Use Case: Cash withdrawal is started by the account owner, when he wishes to use his credit card to withdraw cash from an ATM. The account owner inserts his card in an ATM, and is then requested via the screen to type his PIN code. The screen will either show a polite denial, the card will be rejected from the ATM and the process will be cancelled; or the screen will show a menu requesting the account owner to choose an amount of money by typing on the ATM's keyboard. A new screen requests the account owner to approve the transaction. If the transaction is not approved the account owner is again requested to type an amount. Otherwise the use case ends by the ejection of the card, and the desired amount of money being paid.

Objects: (to be added later)

Functions: (to be added later)

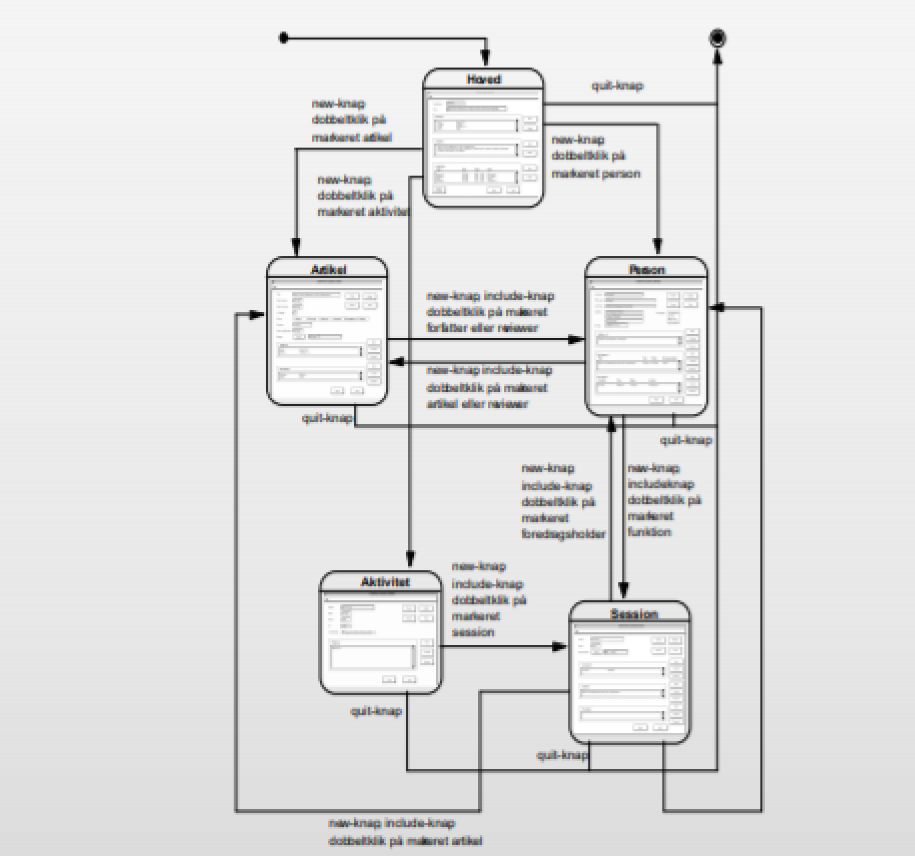
The use case: Cash Withdrawal: should mention the cancel option displayed below.

- Functions



Lav plan	Særdeles kompleks	Opdatering
Konsekvensberegn plan	Kompleks	Signalering
Find kundeoplysninger	Middel	Aflæsning
Sæt indhold I plan	Kompleks	Opdatering
Slet plan	Simpel	Opdatering
Lav reservation	Middel	Opdatering
. . .	. . .	. . .

- User interface (Used/learned in DEB)



Stable versus Transient Properties

Stable properties Transient properties

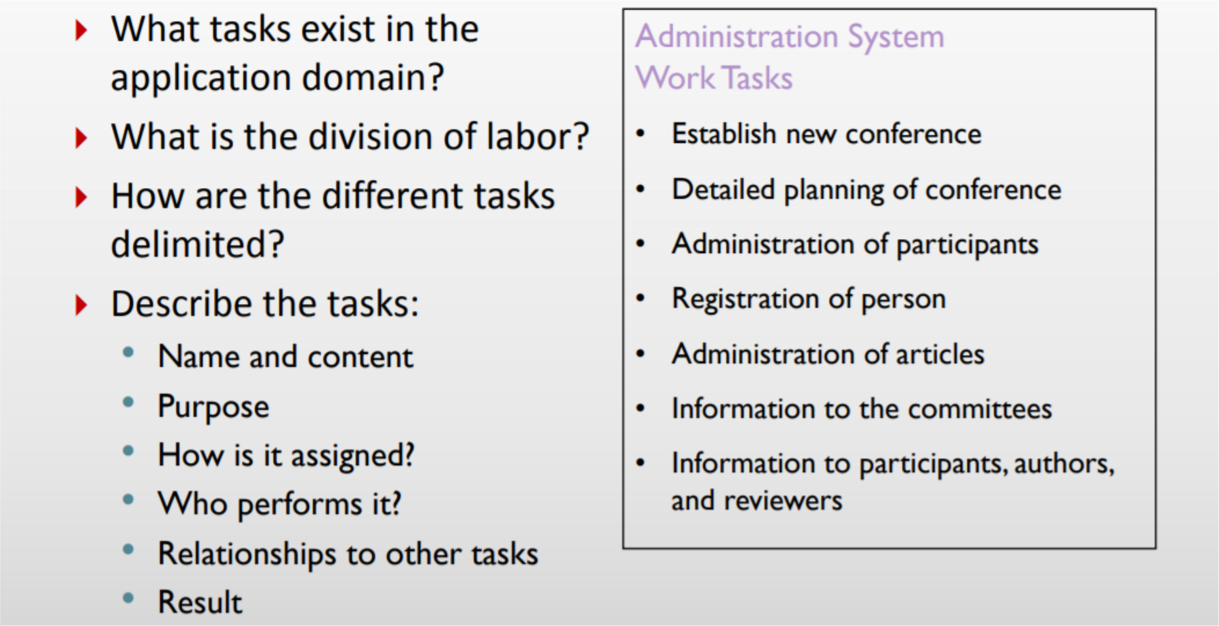
← — →

Model Functions Interfaces

- Compare the model, functions, and interfaces between a classical bank and a modern internet-based bank.

The model is pretty much the same: users, banker and such is the same. Functions is the same and more has properly been added. Interfaces has changed the most; much smaller and more options. It is totally different.

Start from Work Tasks



- What tasks exist in the application domain?
- What is the division of labor?
- How are the different tasks delimited?
- Describe the tasks:
  - Name and content
  - Purpose
  - How is it assigned?
  - Who performs it?
  - Relationships to other tasks
  - Result

Administration System Work Tasks

- Establish new conference
- Detailed planning of conference
- Administration of participants
- Registration of person
- Administration of articles
- Information to the committees
- Information to participants, authors, and reviewers

Actor

- Identify actors
  - Determine the distribution of roles of the works tasks related to the system.
  - Consider human actors.
  - Consider other systems as actors
- Describe actors
  - Make actor specifications

See top of chapter for example.

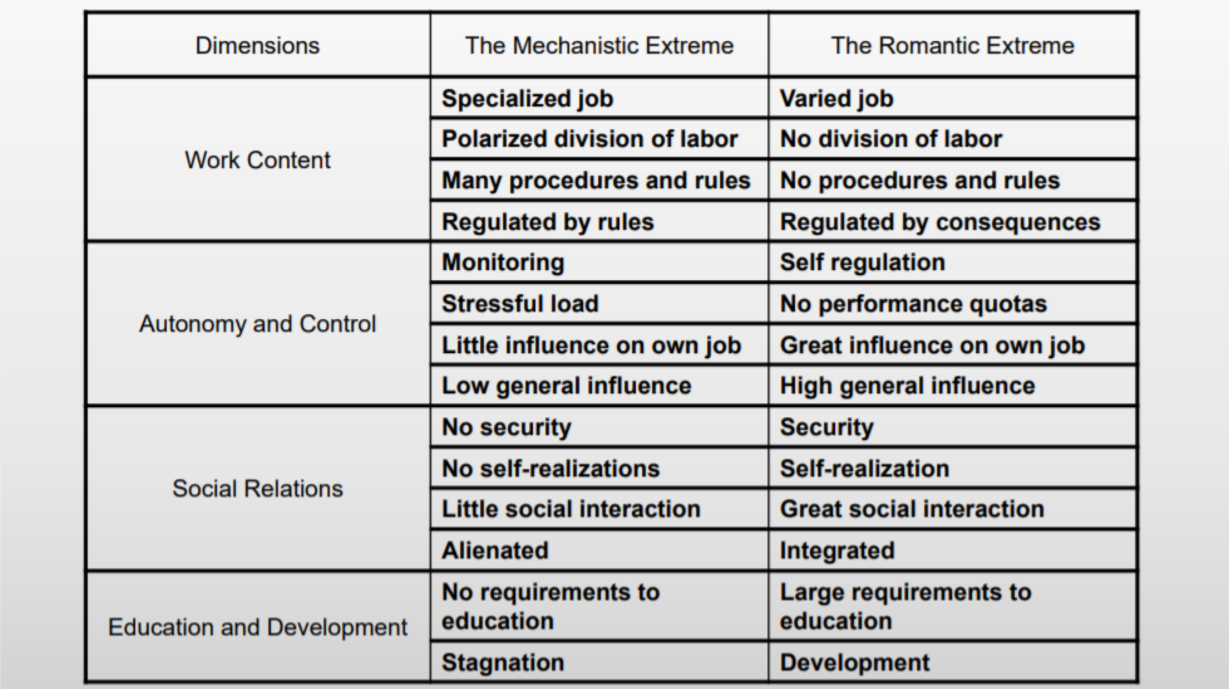
Use case

- Identify use cases where the system is used to carry out part of a work task.
- Describe use cases
  - As text.
  - As a statechart diagram.

See top of chapter for example.

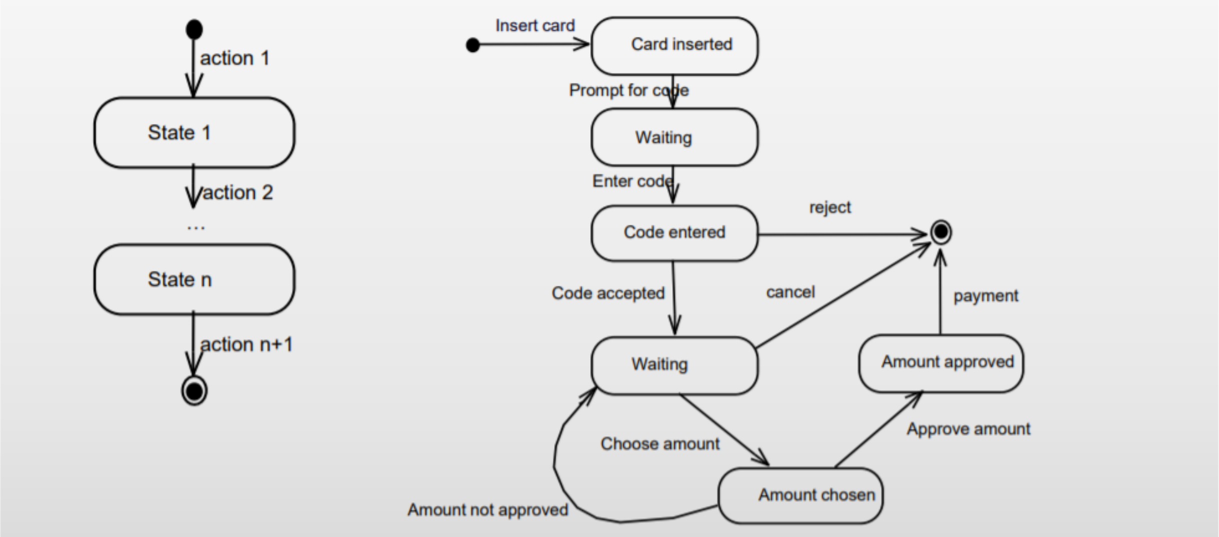
Evaluate Systematically

- Systematic review
  - Use cases should be simple and constitute a coherent whole
  - The description of actors and use case should provide understanding and overview
  - Use cases should be described in enough detail to enable identification of functions and interface elements.
- Experiment with prototypes
  - A use case is best evaluated trough planned prototype experiments.

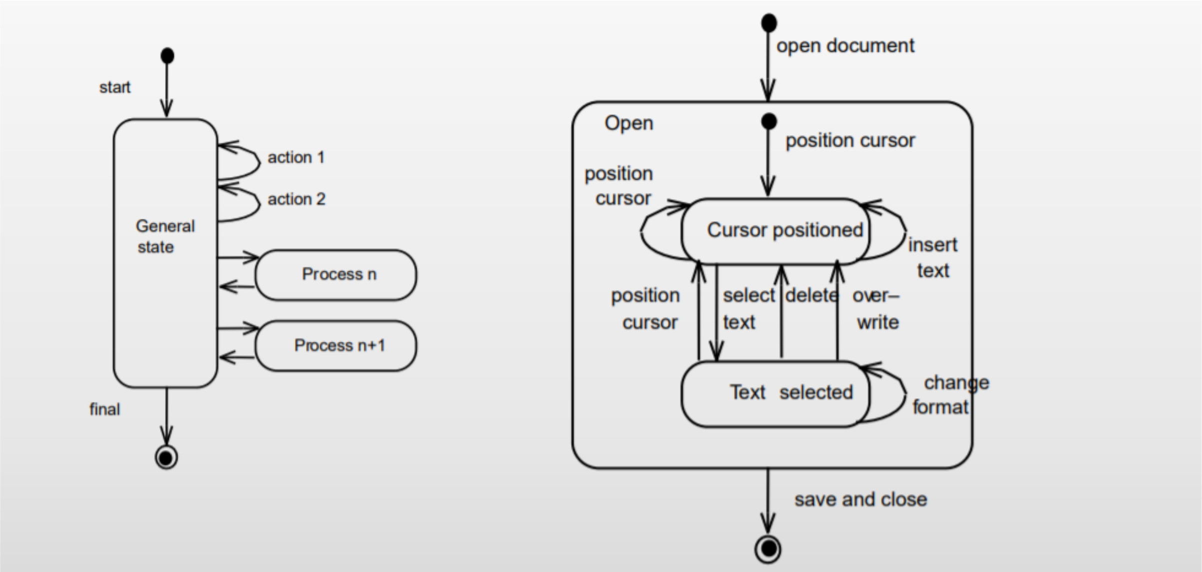


Dimensions	The Mechanistic Extreme	The Romantic Extreme
Work Content	**Specialized job**	**Varied job**
	Polarized division of labor	**No division of labor**
	Many procedures and rules	**No procedures and rules**
	Regulated by rules	**Regulated by consequences**
Autonomy and Control	**Monitoring**	**Self regulation**
	Stressful load	**No performance quotas**
	Little influence on own job	**Great influence on own job**
	Low general influence	**High general influence**
Social Relations	**No security**	**Security**
	No self-realizations	**Self-realization**
	Little social interaction	**Great social interaction**
	Alienated	**Integrated**
Education and Development	**No requirements to education**	**Large requirements to education**
	Stagnation	**Development**

Explore Pattern: Procedural



Explore Pattern: Material



Appreciate the Difference: Actors and Classes

	Application Domain	Problem Domain
Static	Actors	Classes and Structure
Dynamic	Use Cases	Behavioural Patterns