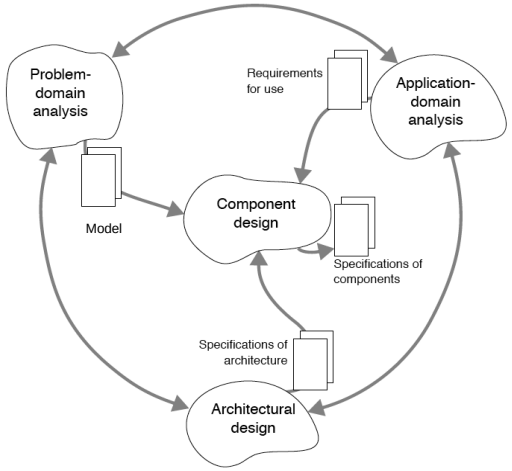
APPLICATION DOMAIN ANALYSIS: FUNCTIONS

SU:E16:L8

AKTIVITIES IN APPLICATION DOMAIN ANALYSIS



Usage
- How does the system interact with people and systems in the context?
- Use case and actor

Functions
- What are the system's information processing capabilities?
- Function

Interfaces
- What are the target system's interface requirements?
- Interface, user interface, system interface

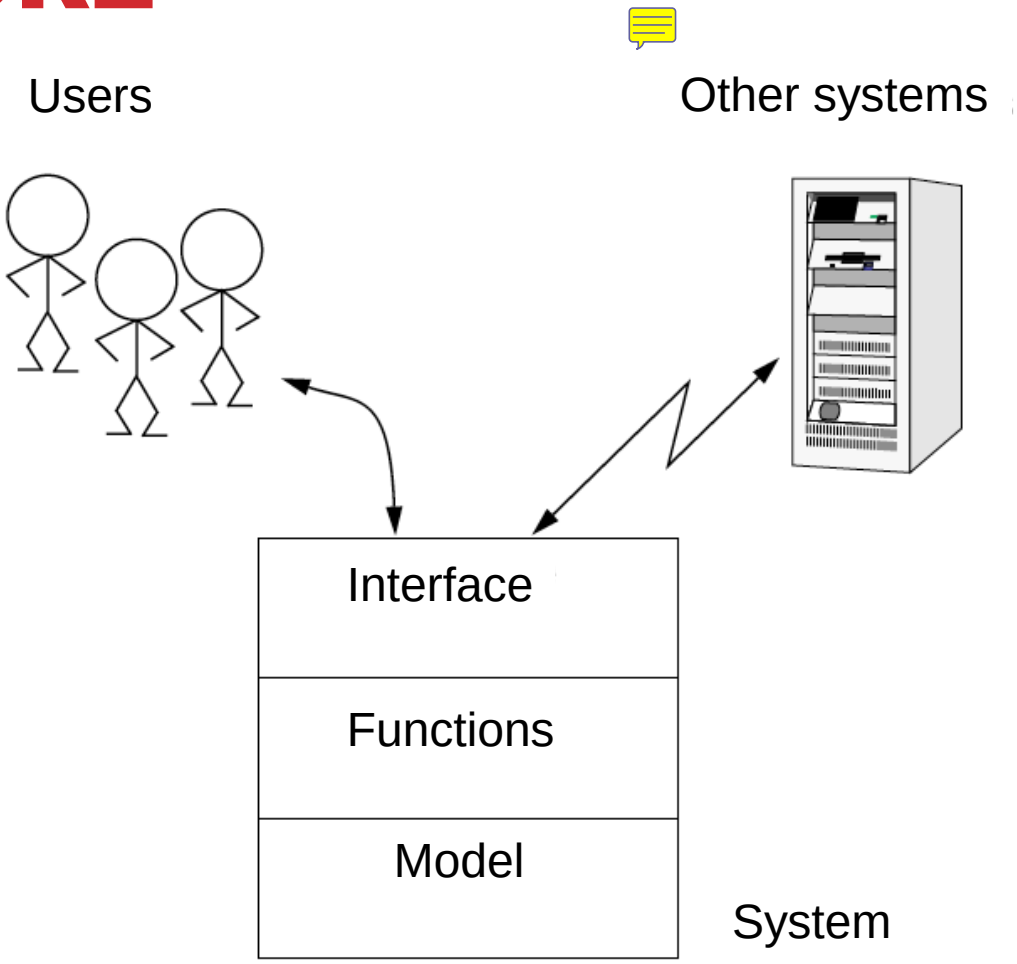
Stable properties ⟷ Transient properties

Model — Functions — Interfaces

EMPHASIZE THE ARCHITECTURE

Function: A facility for making a model useful for actors.

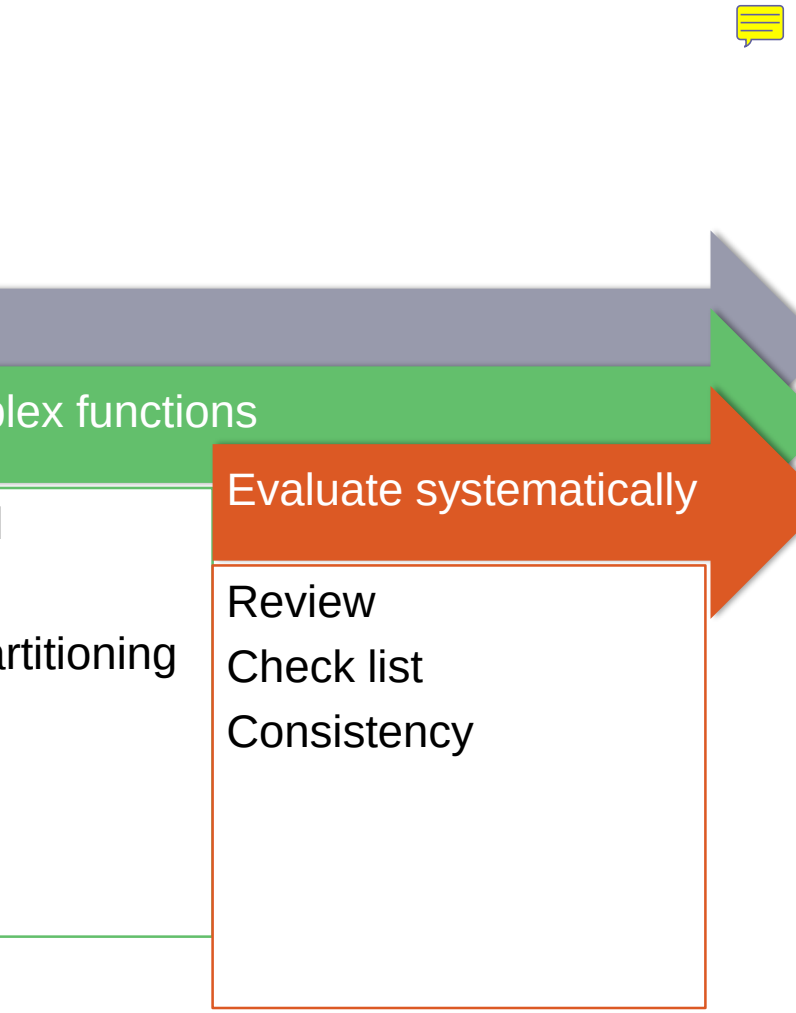
IT-system: A collection of components that implements modeling requirements, functions and interfaces.



OVERVIEW OF 'FUNCTIONS'

Purpose	• To determine the system's information processing capabilities.
Concepts	• Function: A facility for making a model useful for actors.
Principles	• Identify all functions. • Specify only complex functions. • Check consistency with use cases and the model.
Result	• A complete list of functions with specification of complex functions.

ACTIVITIES IN ‘FUNCTIONS’



RESULT OF FUNCTIONS

- **Primary result: a complete list of functions**

Planning		
Make schedule	Very complex	Update
Calculate schedule consequences	Complex	Signal
Find working hours from previous period	Medium	Read
Enter contents into schedule	Complex	Update
Erase schedule	Simple	Update
…	…	…

- **Secondary result: specification of complex functions**

Query possible reservations:
given time or date or employee-name
search objects in time period-available and select those
who belong to emplyee-name, is known
have date, if known
cover point in time, if known
result objects of time period-available that fulfill the criteria

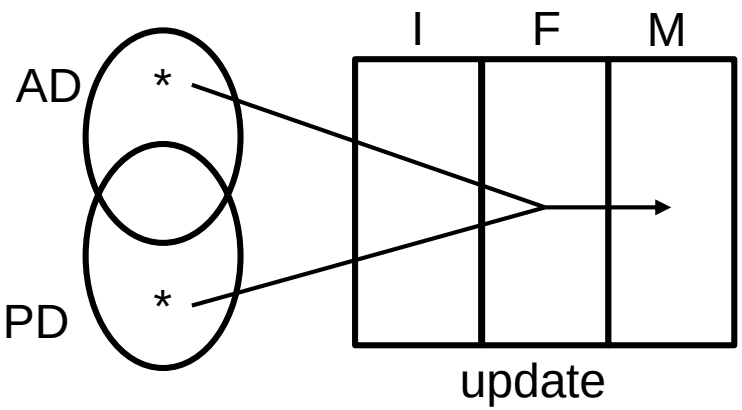
FUNCTION TYPES

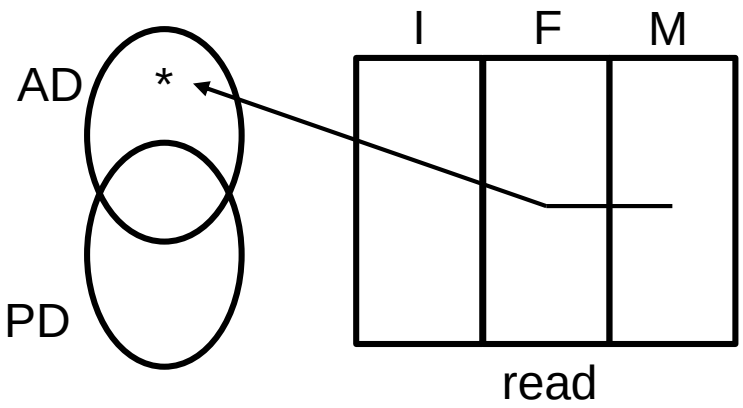
Effect of the processing

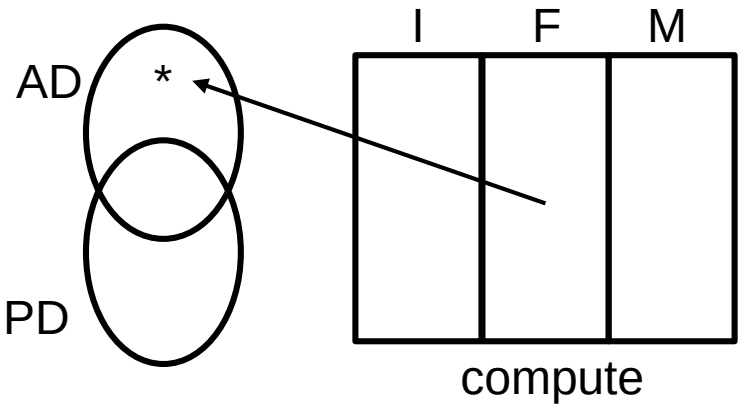
* Initiative

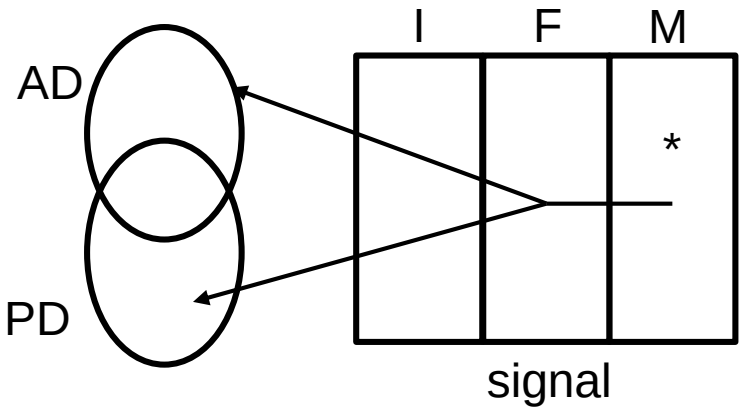
Function: A facility for making a model useful for actors.

- A resource for actors.







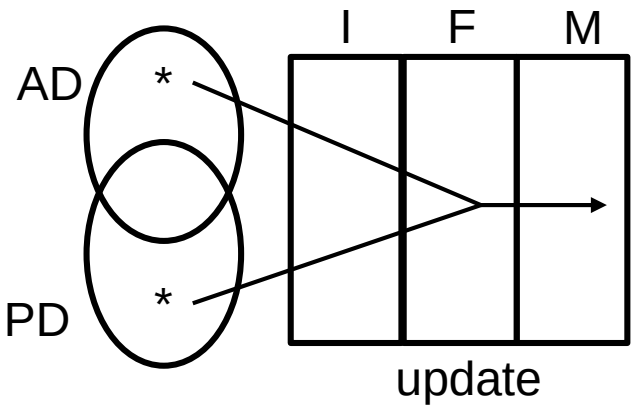


FIND FUNCTIONS

- **How do we find functions?**
- **Information sources**
    - Problem-domain descriptions
        - Classes (read, update)
        - Events (update, signal)
        - The model (signal)
    - Application-domain descriptions
        - Use cases (all types)

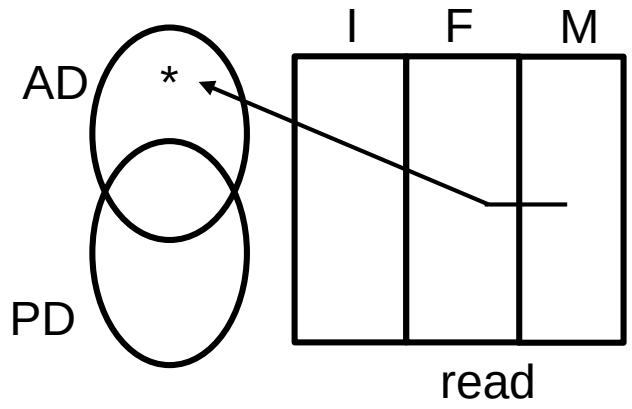
- **Ask questions related to each function type**

FIND FUNCTIONS: UPDATE



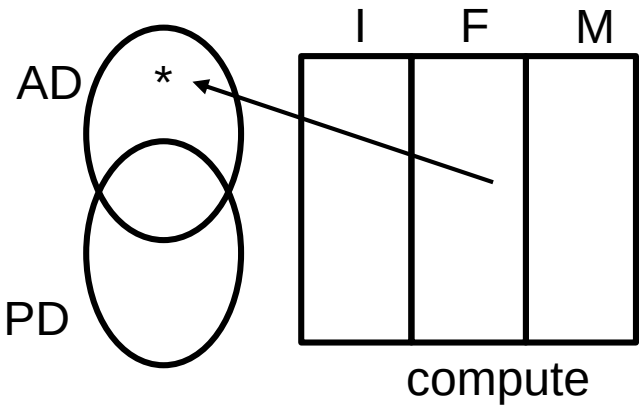
- **Update functions are activated by a problem domain event and result in a change in the model's state.**
- **Questions for each event**
  - How is the event observed, and how is it registered? In which use cases does this happen?
  - How should the use cases be supported by update functions?
  - Which objects, attributes, and object structures are affected by the event, and what requirements does this impose on the update functions?

FIND FUNCTIONS: READ



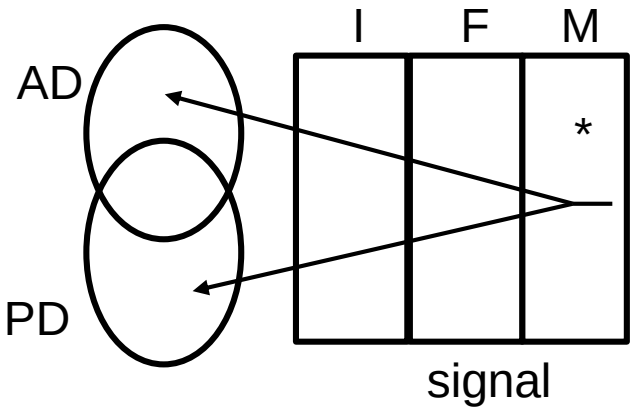
- **Read functions are activated by an actor's need for information and result in the system displaying information about the model.**
- **Questions about information needs**
  - Given the work of actors:
    - What do the actors need to know about the state of the model?
    - What read functions does this give rise to?
  - Given the model:
    - Which objects and structures will the actors need information about?
    - What read functions does this give rise to?

FIND FUNCTIONS: COMPUTE



- **Compute functions are activated by a need for information in an actor's work tasks and consists of a computation involving information provided by the actor and/or the model; the result is a display of the computation's result.**
- **Questions about needs for computation**
  - With a starting point in actors and use cases
    - Which computations (not necessarily based on the model) do the actors need to have carried out?
    - Does the computational basis come from the actors, the model, or both?
    - Which computations form complete wholes in the use cases?

FIND FUNCTIONS: SIGNAL



- **Signal functions are activated by a change in the model's state and result in a reaction in the context; this reaction can be a display to the actors in the application-domain or a direct intervention in the problem-domain**
- **Questions about critical states**
  - Examine the model of the problem-domain
    - What are the critical states for the model?
    - What is the significance of these critical states? What are the consequences when they occur?
    - How does a signal function register that the model has entered a critical state?
    - What signal does each critical state give rise to? How reliable and strong do the signals have to be?

COMPLETE LIST OF FUNCTIONS

Planning		From chapter 20
Make schedule	Very complex	Update
Calculate schedule consequences	Complex	Signal
Find working hours from previous period	Medium	Read
Enter contents into schedule	Complex	Update
Erase schedule	Simple	Update
Query earlier schedules	Medium	Read
Make appointment	Medium	Update
Cancellation	Simple	Update
Query possible appointments	Complex	Read
Register treatment	Simple	Update
Create customer	Simple	Update
Query customer information	Medium	Read
Employment	Simple	Update
Retirement	Simple	Update
Update apprentice information	Simple	Update

SPECIFY COMPLEX FUNCTIONS

- **Mathematical expression:**
  o = f(i)
- **Algorithm**
- **Functional partitioning**

Make schedule	Very complex	Update
• Create six-week period	• Simple	
• Create standard distribution for an employee	• Medium	
• Use standard distribution for en employee	• Simple	
• Adjust the distribution of employees in a week	• Medium	

EVALUATE SYSTEMATICALLY

- **Three ways to ensure that your function list is complete**
  1. Review
     - use the list in the cooperation with users
     - Support this with function prototypes
  2. Check list
     - For each function type; use the questions to ensure the possibilities are exhausted
  3. Consistency
     - Compare the function list with the system definition and the model
     - May give rise to a revision of both the system definition and the function list

EVENT, USE CASE AND FUNCTION

What is the difference between an event, a use case and a function?

- **All describe dynamic action**
- **Related**
- **But in different domains**

Event:
What happens to objects in the PD?

Use case:
How will the system be used?

Function:
What is the system going to do?

Example: Order system

- Event
  'Order received' – a customer places an order at a specific time
- Use case
  'Enter order' – a user in the application-domain creates an order for a customer by using the system.
- Function
  'Create order' – an object of the order class is created in the model of the IT-system

OVERVIEW OF 'FUNCTIONS'

Purpose	• To determine the system's information processing capabilities.
Concepts	• Function: A facility for making a model useful for actors.
Principles	• Identify all functions. • Specify only complex functions. • Check consistency with use cases and the model.
Result	• A complete list of functions with specification of complex functions.