

Software Engineering Research Group: Processes and Measurement
TU Kaiserslautern
Fachbereich Informatik

Grundlagen des Software Engineering

Fundamentals of Software Engineering

Winter Term 2010/11

Prof. Dr. Dr. h. c. Dieter Rombach
Dipl.-Inf. Jörg Dörr

Chapter 4: Software Application Engineering - Requirements
Engineering for Information Systems

4 Software Application Engineering

4.1 Motivation

4.2 Introduction and Principles

4.3 Problem Description

4.4 System Requirements (Focus: Customer/User Requirements)

4.5 Requirements Engineering for Embedded Systems

4.6 Requirements Engineering for Information Systems

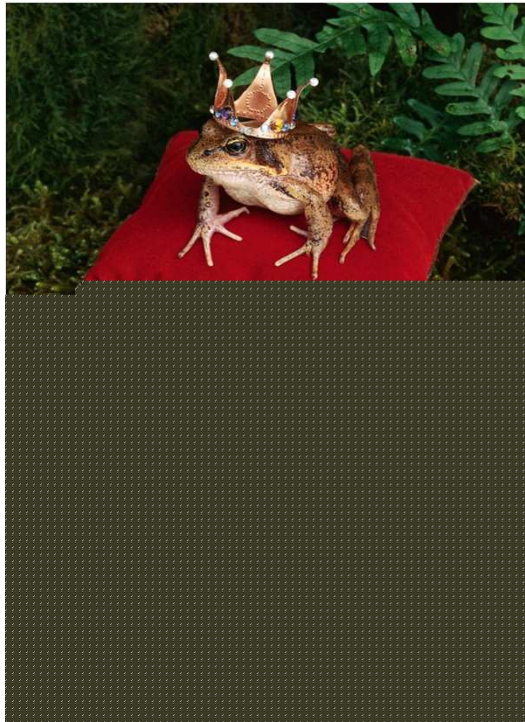
What makes an information system different?

- Which types of systems do we talk about?
- Who participates in such systems?
- Which qualities matter?



Usability

What is Usability? – Some fairy tales



- “Usability is about graphics and colors!”
- “Usability can be fixed at the end!”

What is it then?

Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. (ISO 9241-11)

+ Effectiveness

Which part of the work is supported by software?
Functionality, less errors,....

+ Efficiency

How fast can the task be performed?
Performance,...

+ Satisfaction

Does the user like the system?
Attractiveness, trust, ...

= Usability

How to Achieve Usability? [Lauesen 2005]

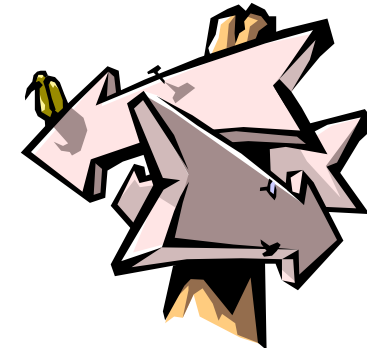
- Task-oriented: Developers have to know users and their tasks
- Develop iteratively
- Evaluate early: Make Prototypes as early as possible and review them with users

But:

- Not systematic way to design screens (usually trial and error)
- Hard to find test users
- Expensive?

TORE – Task and object-oriented Requirements Engineering

- Making decisions regarding the effects of the system that is going to be developed [Kovitz, 1999] Design decisions
- RE-Approaches differ in
 - Type of decision to make
 - Guidance on making decision
 - Order of decision types
 - Notations to document decisions
- RE-Methods
 - help to expand the set of possible solutions for one decision
 - support the decision process (select one solution out of this set)
 - define how to document design decisions



Design Decisions and their Models

- Different approaches for requirement specification are distinguished by procedure and used models
- What similarities do all those approaches have?
- What kind of models are there?

Armour [Armour & Miller, 01]

- Use Case Diagram
- Domain Object Modell
- Initial what-is System Use Case
- Initial what will be System Use Case
- Base System Use Case
- Internal Use Case
- Elaborated System Use Case
- Transaction Information Model
- Transaction Trees
- Analysis Object Model

Many different models and tasks, but basic design decisions are in common

Holtzblatt

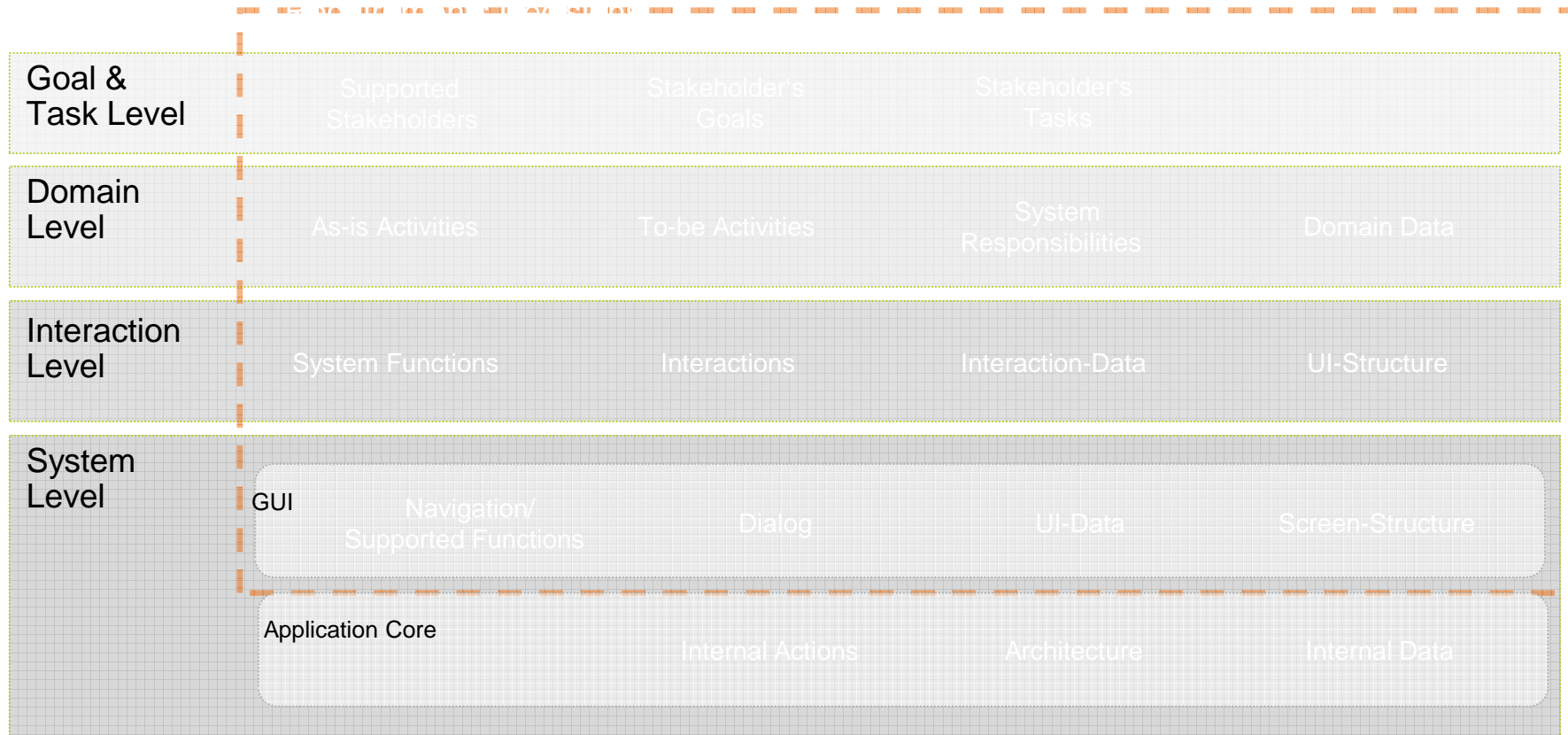
[Beyer & Holtzblatt, 98]

- Work Model
- Focus Area
- User Environment Design (UED)
- Storyboard
- Use Case
- Object Model

Constantine

[Constantine & Lockwood, 99]

- Task model
- Domain model
- Content model
- Context Navigation Map
- Essential Use Case
- Use Case Maps





- **Decisions:**
 - What roles have to be supported?
 - What are the goals of the users?
 - What tasks do these roles perform as part of their work?
- **Notations:**
 - Personas
 - Role descriptions
 - Goal Modeling Notations (i*)
 - Natural language

How to describe a user/role (1/2)?

- A **user role** is an abstract summary of **needs, interests, expectations, behavior, and responsibilities** that are characteristic for a set of future system users [according to Constantine/Lockwood99].
- A **user profile** describes **the knowledge and the skills** of typical users.
- Can be elicited by
 - asking the users
 - asking surrogate users (marketing, sales, hotline, trainer)
 - examining documents in the business process

How to describe the user (2/2)?

Role Description

- Responsibilities
- Success criteria
- Tasks
- Communication partners
- Degree of innovation

User Profile:

Knowledge/experience/skills

- regarding tasks
- regarding software system

Example: counter employee in university library (1/2)

Role Description

- **Responsibility:** taking care of readers, issuing books
- **Success criteria:** reader satisfaction, book inventory up to date
- **Tasks:** advice, issue, return, registration, cancellation
- **Communication partners:** readers, librarians
- **Degree of innovation:** low

User Profile

- **Prior knowledge of library tasks:**
 - Books: must be sufficient for advice
 - Library workflows: often low, since student assistant
- **Prior knowledge of software:**
 - Often low, since usually humanities student

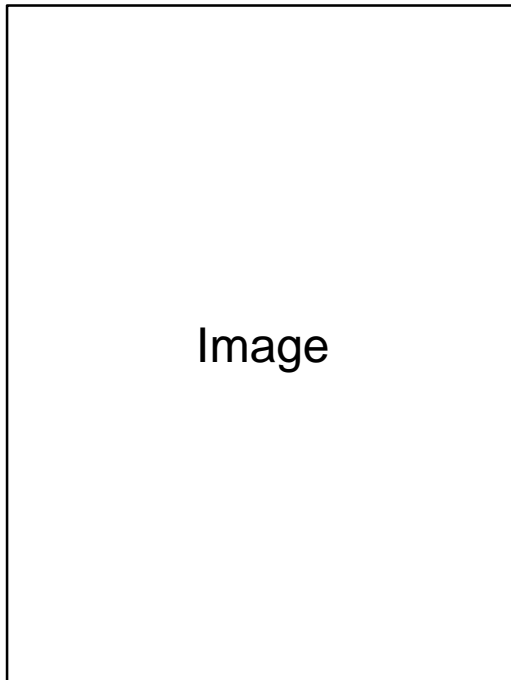
Further possibility: description of “personas”



- **Personas describe stereotypical users**
- **Personas are very concrete**



Name:



Motto:

Profile

Age:

Work environment:

Product knowledge:

Most frequent activities (with the product):

Most important activities:

Rarest activities:

Typical obstacles:

Unique features:

Family issues:

Other:

Core Characteristics:

-
-
-

Core Goals:

-
-
-

Name: Prof. Dr. med. Ziak

Profile



Motto:

“Work and make money”

Age: 50

Work environment: Own ENT office, in-patient beds in hospital, thinking about opening second office. Not too big, simple, sparsely furnished examination room. Occasionally goes to university hospital to perform surgery. Prefers to take along mobile equipment.

Product knowledge: Only what is essential. Has attended continuing education events.

Patients: Many singers and actors. Choking stimulus in case of contact with Mediastrobe. Prefers private patients.

Most frequent activities (with the product): Vocal cord diagnosis, examination of the larynx, with video archiving.

Most important activities: Early detection of larynx carcinomas; restoration of voice function.

Rarer activities: Voice analysis;

Typical obstacles: Little to no PC skills. Lack of practice in using stroboscope. Tangled cables and foot switches. Profitability.

Family issues: Wife is managing office; 2 children, daughter going to college and son supposed to take over the office one day, but has totally different plans...

Other: Strives for expansion and influence. Has a good tax advisor. Lives in Saarbrücken.

Core Characteristics:

- Dominance, influence
- Undecided, skeptical
- Professional, cool, and competent
- Bargain hunter

Core Goals:

- Reputation
- Private insurance customers
- Make money

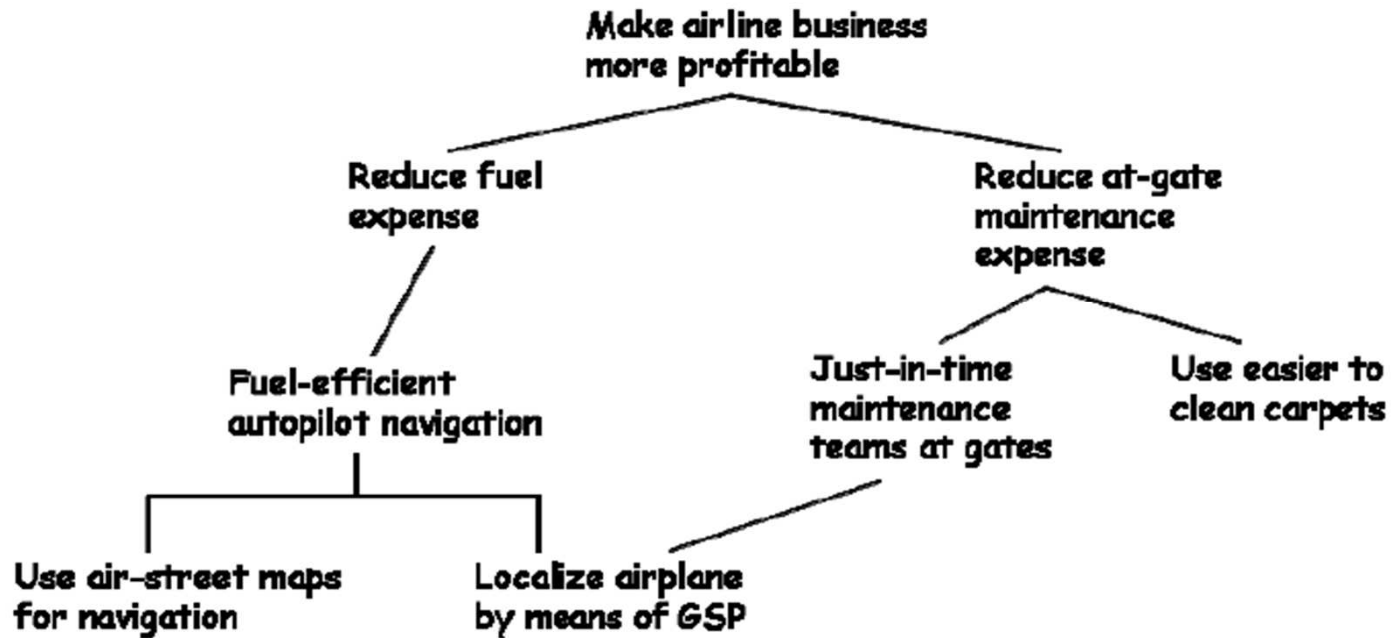
Description of Goals

- Definition **goal**: “A goal is a desirable state lying in the future, which is not reached automatically but by specific actions.”
- Goals and their dependencies are often described in conceptual models that are based on modelling languages.
- Definition **goal model**: “A goal model is a conceptual model. Its goals and decompositions are documented in sub-goals and as necessary further dependencies between (sub)-goals.”

Various notations exist for goal modeling

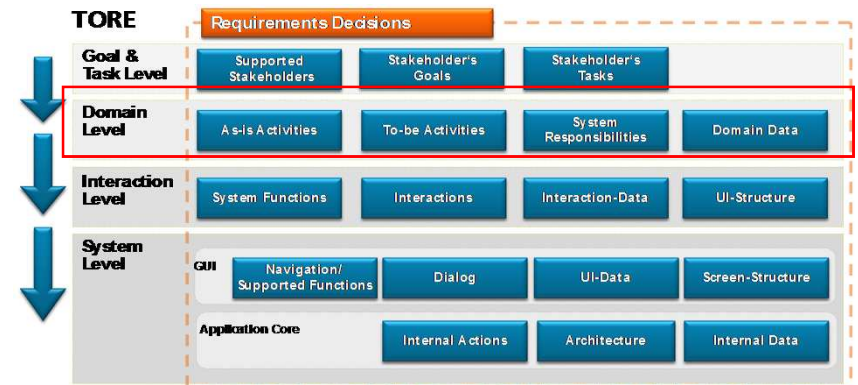
- i-star (i^*)
- GRL (\rightarrow talk on GRL)
- SIG (Softgoal interdependency graphs)
- And / OR – Trees
- Just text
- ...

Example



Example taken from Master Software Engineering for Embedded Systems, TU Kaiserslautern, Textbook E-M.2

TORE: Domain Level



Decisions:

- **As-Is:** What activities are relevant for the system?
- **To-Be:** How does the work process change by using the system?
- **System Responsibility:** What is the key contribution of the system?
- **Domain Data:** What data is relevant in the domain?

Example: As-Is Activities

Notation:

- Activities in task descriptions (natural language)
- Activity diagram
- BPMN / EPCs

Example:

Task: Book order

Activities:

- The customer selects books from the bookstore
- The bookstore gets the money from the customer.
- The customer receives the selected books from the bookstore

Task evaluation

- Objectives
- Possibilities of engagement
- Causes

Task performance

- Initial situation (precondition, priority, occurrence, rate of iterations)
- Info-In
- Info-Out
- Resources (means for work, actors, partners)

Description of Task „book return“

- Objective: **book is back in library**
- Possibilities of engagement: **check book quality**
- Causes: **Loan period expired or voluntary return**
- Initial situation: **book dispensed; high priority; frequent**
- Info-In: **book**
- Info-Out: **confirmation of return**
- Resources: **processor, book file, user file**

Event-driven Process Chain

- Semi-formal graphical notation used primarily to represent business processes
- Developed by the Institut für Wirtschaftsinformatik (IWI) at the University of Saarland
- Used in industrial (e.g SAP R/3) and academic practices, especially in the SAP environment (e.g ARIS – Architecture of Integrated Information Systems framework)

Notation – Function

- Label : Verb- object
- **Activity or task** (detailed or abstract)
- Example : Choose Recipient



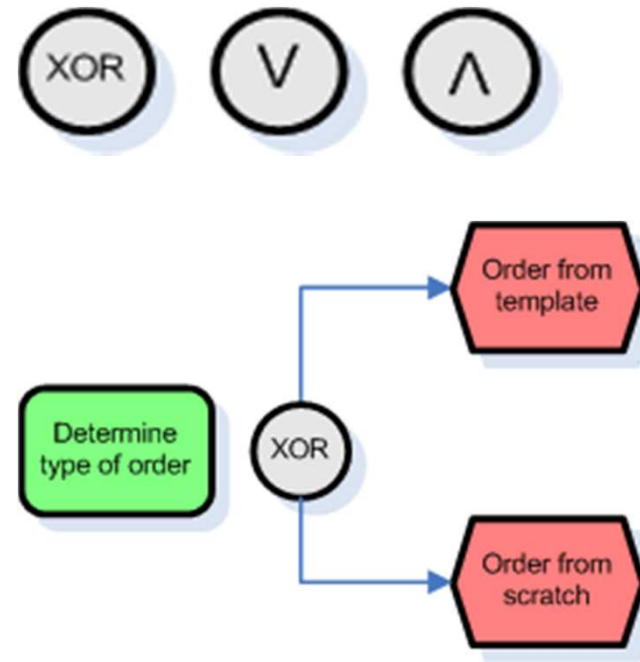
Notation - Events

- Label: object - verb in perfect tense describing the state reached
- An event can be :
 - A state **before** executing a function
 - A state reached **after** an activity has been executed
- Examples: Recipient chosen



Notation – Logical Operator

- used to connect functions and events
 - **Decisions** or choices (XOR, OR)
 - **Parallel execution** of functions (AND)
- Example: An order may be executed based on a template or completely from scratch.



Notation – Organizational Unit

- Organizational units represent roles or persons that are responsible for a certain function.
- Example: Technical staff is responsible for ordering IT equipment



Notation – Information Object

- seen as **input or output to functions**
- Examples: Order basket



Notation – Process Path

- Label : Verb – object
- Similar to the element “function”
- Used to hierarchically describe (Abstract -> Details)
- Several activities can be grouped and represented by the process path element

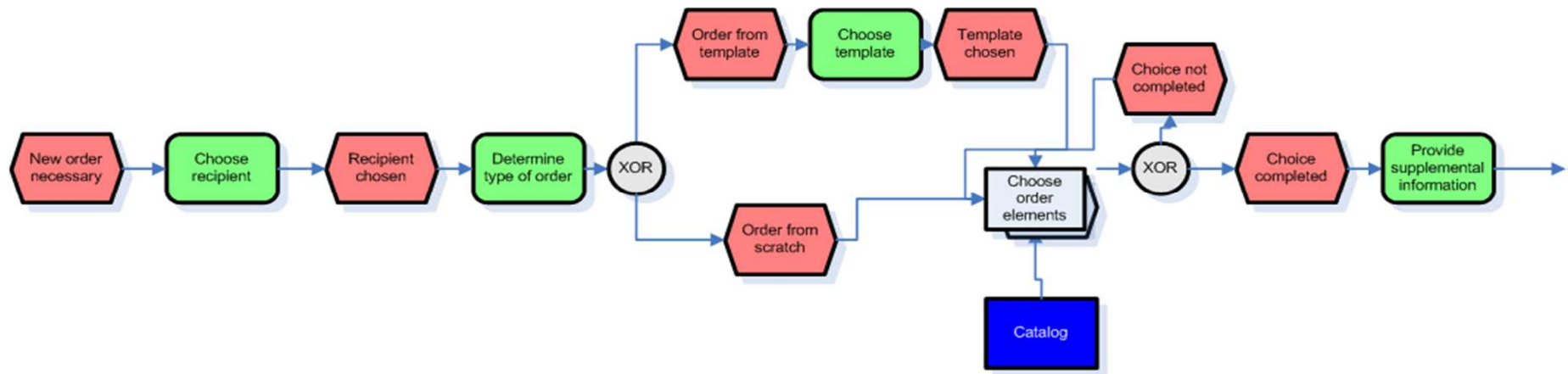


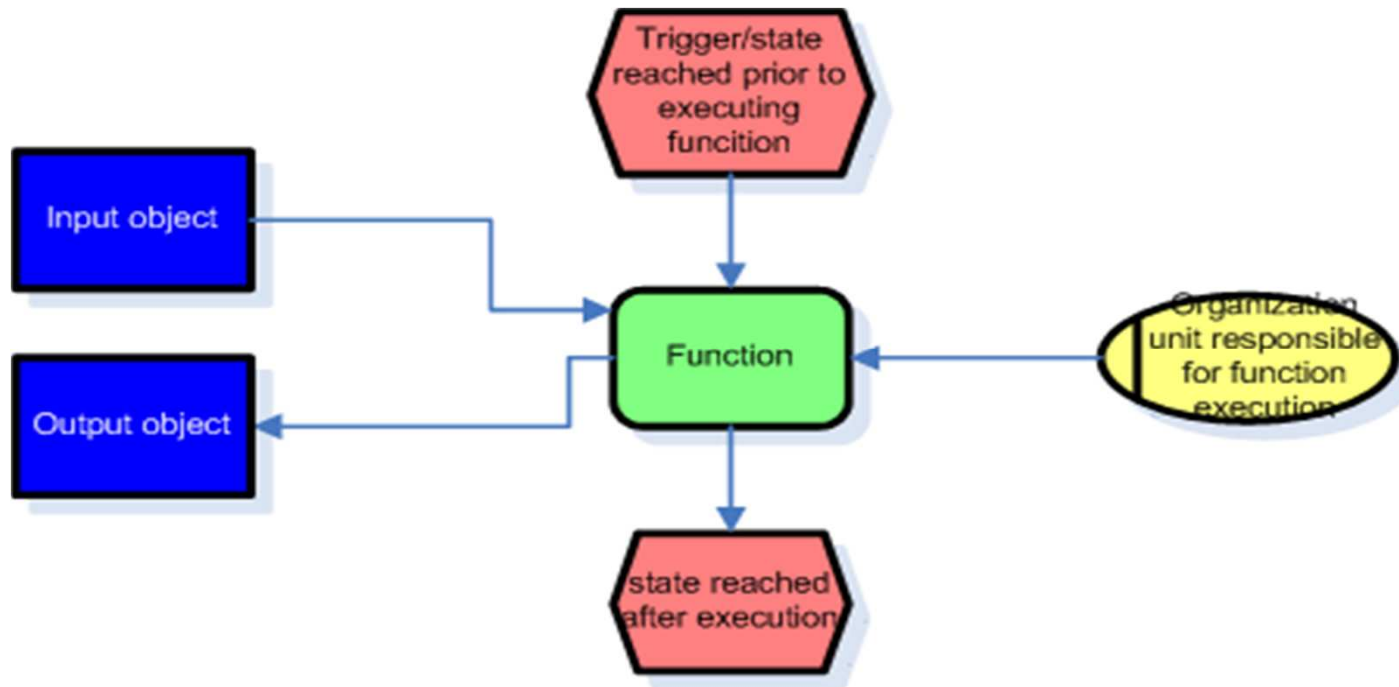
Notation – Process Path

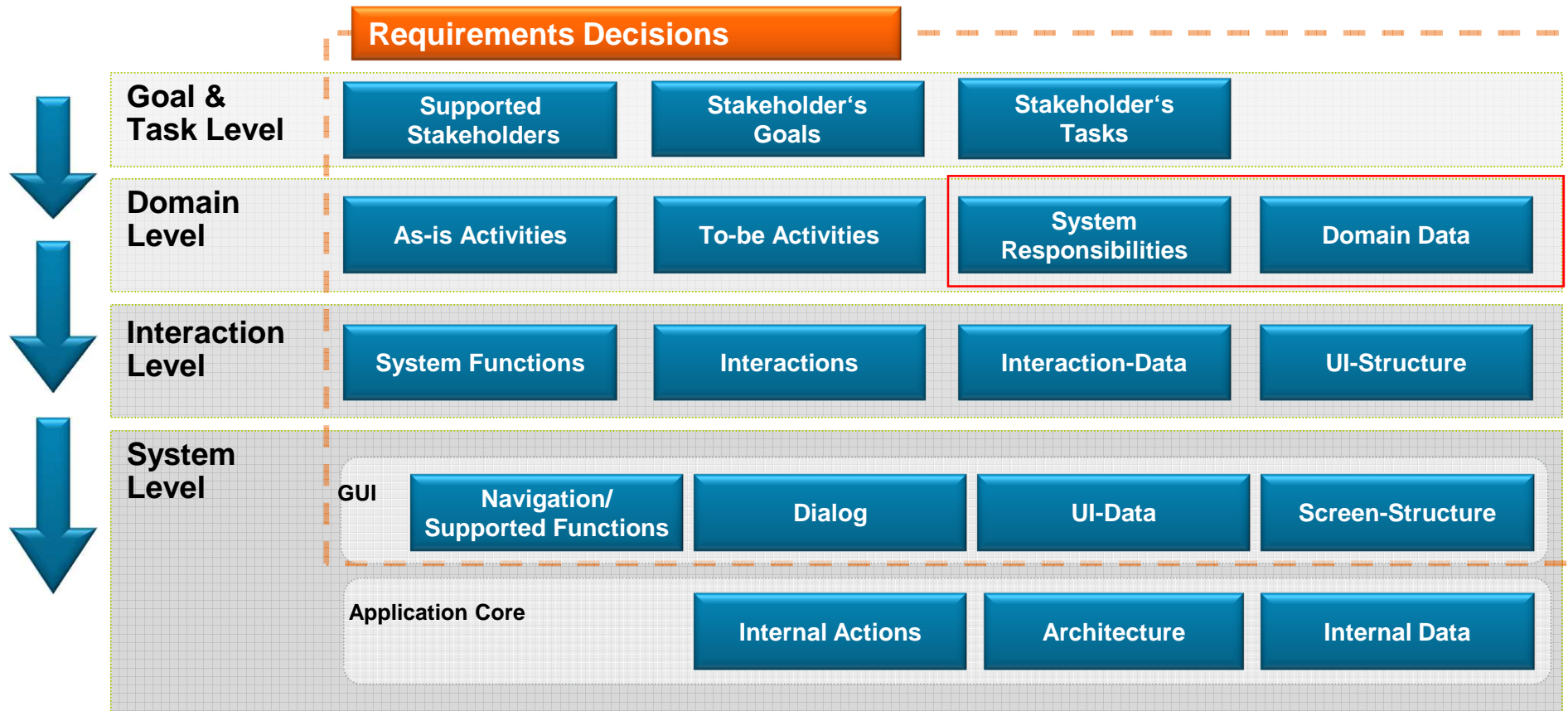
Example:



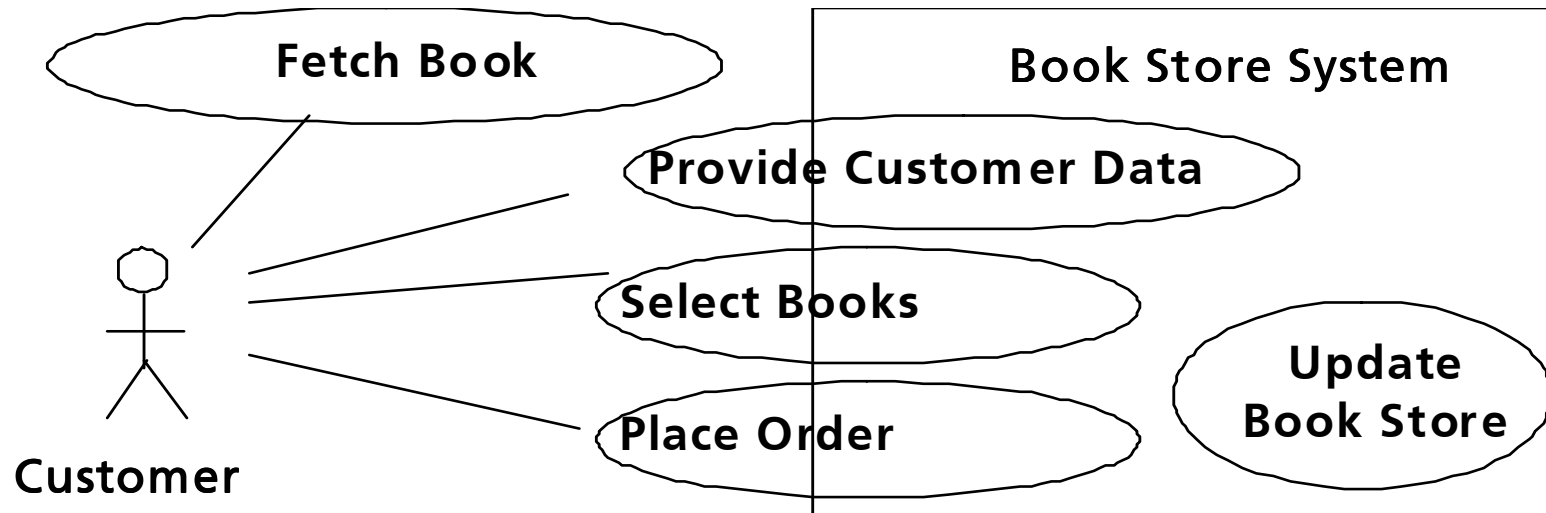
Refined in:





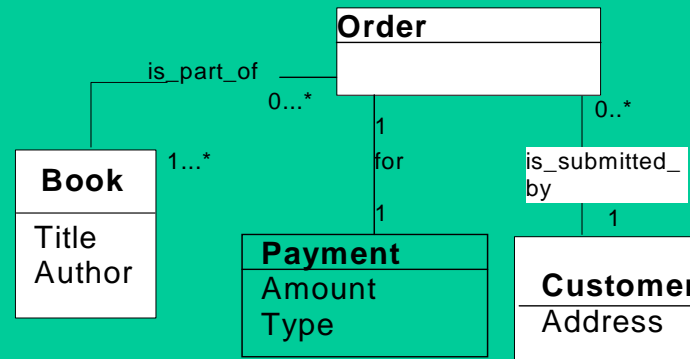


System Responsibilities - Example



Example: Domain Data

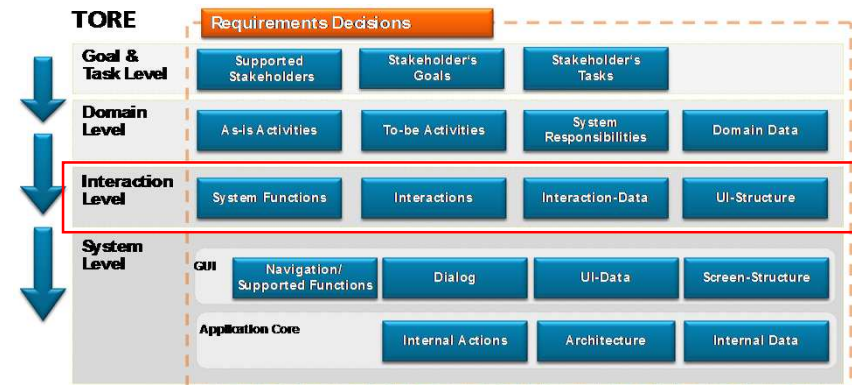
- A book has a title and an author. It can be included in zero or more orders.
- A payment transfers money from the customer to the bookstore. This can be done either by credit card or by bank transfer.



TORE: Interaction Level

Decisions:

- **System Functions:** How is the work divided between user and system?
- **Interactions:** How can the user interact with the system?
- **UI-Structure:** How to group data and functions in the UI?
- **Interaction Data:** What data is exchanged between system and user?



Example – System Functions

Name: Complete-Order function

Informal Description:

The user inputs shopping bag, payment method and address.

The system checks the payment method and stores this information.

Constraints: Shopping bag may not be empty for an order.

Receives (Inputs): Shopping bag, payment method, address

Returns (Outputs): „Order can be confirmed“

Assumes: Nothing

Result: Shopping bag, payment method, address and order is stored in the system

Example: Interaction

Name: Place Order **Initiating Actor:** Customer

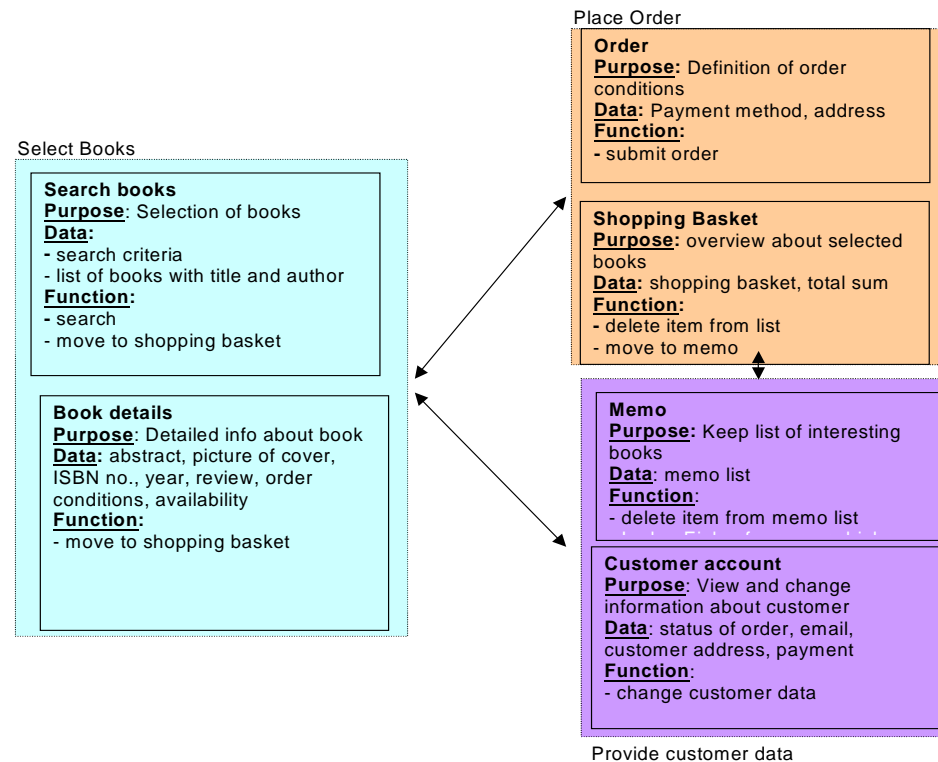
Realized User Task: Book Order

Flow of events:

1. The System displays the shopping basket with the selected book.
2. The Actor selects the “Complete Order”-function. [No Customer Data]
3. The System shows order and supports the Actor in determining the payment method and the address and submitting the order. [New selection] [New customer data] [No order]
4. The Actor selects the „Submit Order“-function.
5. The System acknowledges the order to the Actor, stores the order and supports the Clerk with the “Order Delivery”-responsibility.
6. The Actor receives the selected books

...

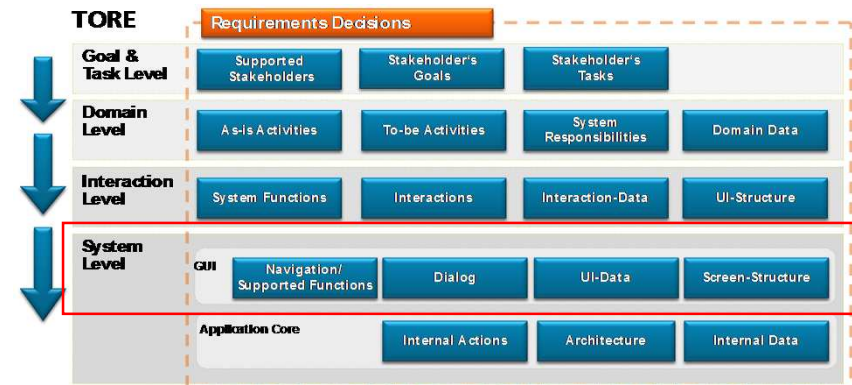
Example: UI-Structure



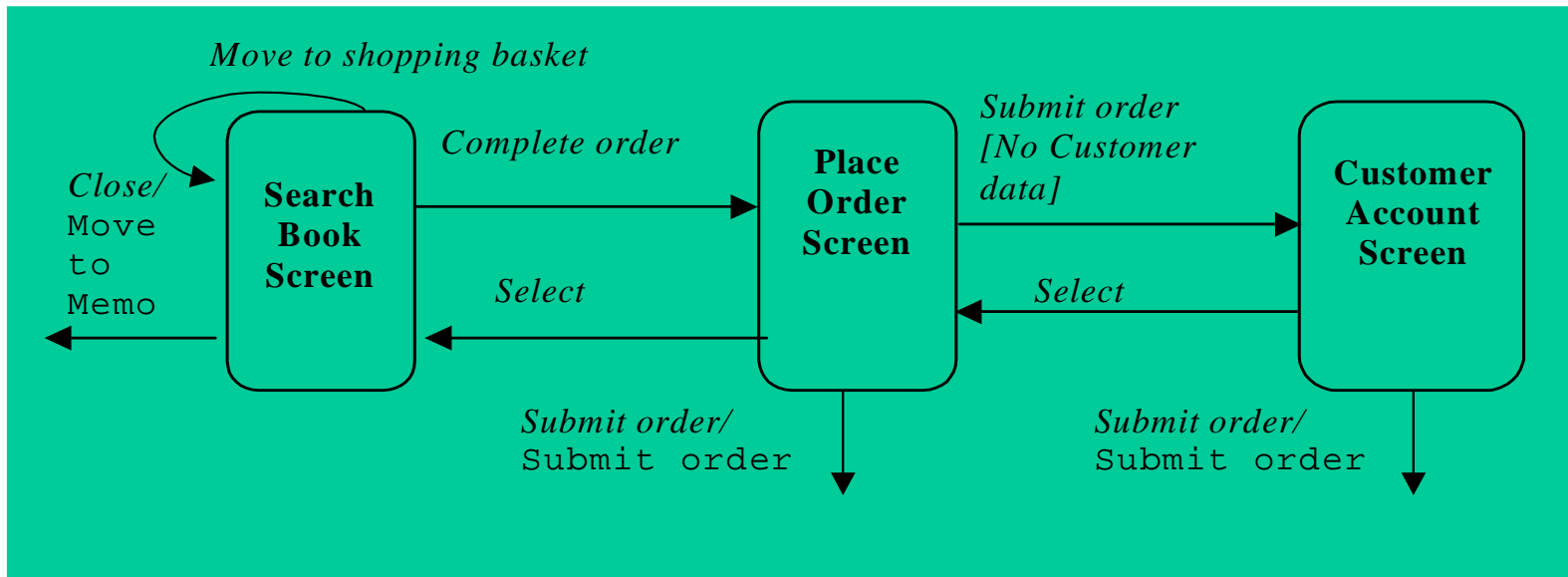
TORE: System Level

GUI Decisions:

- **UI-Data:** What input does the user has to provide? What output does the user get?
- **Support/Navigation Functions:** How can the user move from screen to screen?
- **Screen Structure:** How are functions and data represented on the screen?
- **Dialog:** How can users control the system to proceed in their task? What is the sequence of function called by the user?



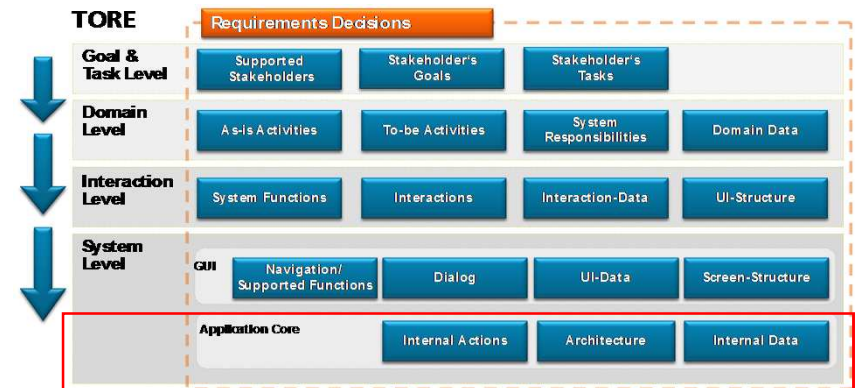
Example: Dialog

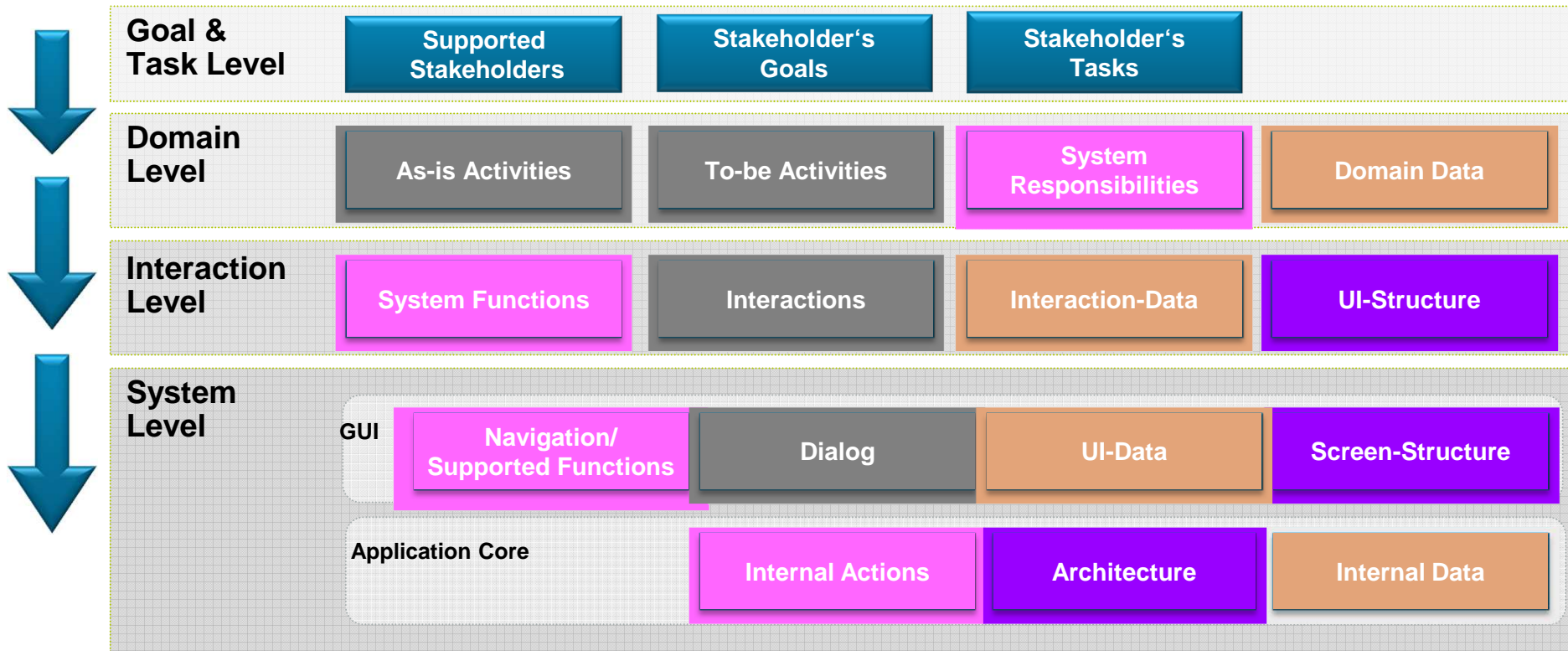


TORE: System Level

Application Core Decisions:

- **Internal System Function:** What are the internal system functions?
- **Internal System Data:** What are the internal data?
- **Architecture:** How is the system structured?





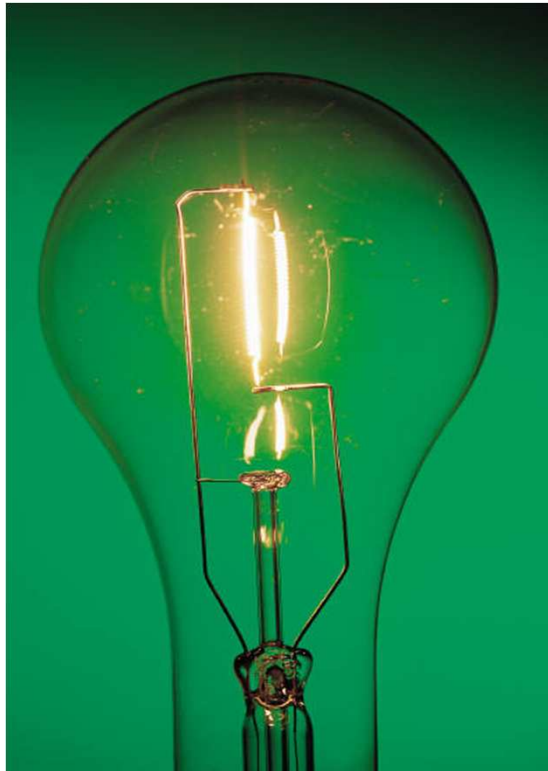
[Paech&Kohler 2002]

Advantages

Thinking about decisions helps ..

- **ensuring completeness**
 - Be aware of the decisions you make. They are made anyway, it is only a question whether this is explicit or implicit
- **setting the focus**
 - Document only decisions important in the given context

Summary – TORE: RE for Information Systems



- Integrates HCI and RE decision types: helps to achieve high usability
- Provides a task driven approach
- Is independent from notations
- Does not enforce documentation of all decision types
- Models for static and dynamic system attributes complete one another
- Is a suitable approach for RE for information systems