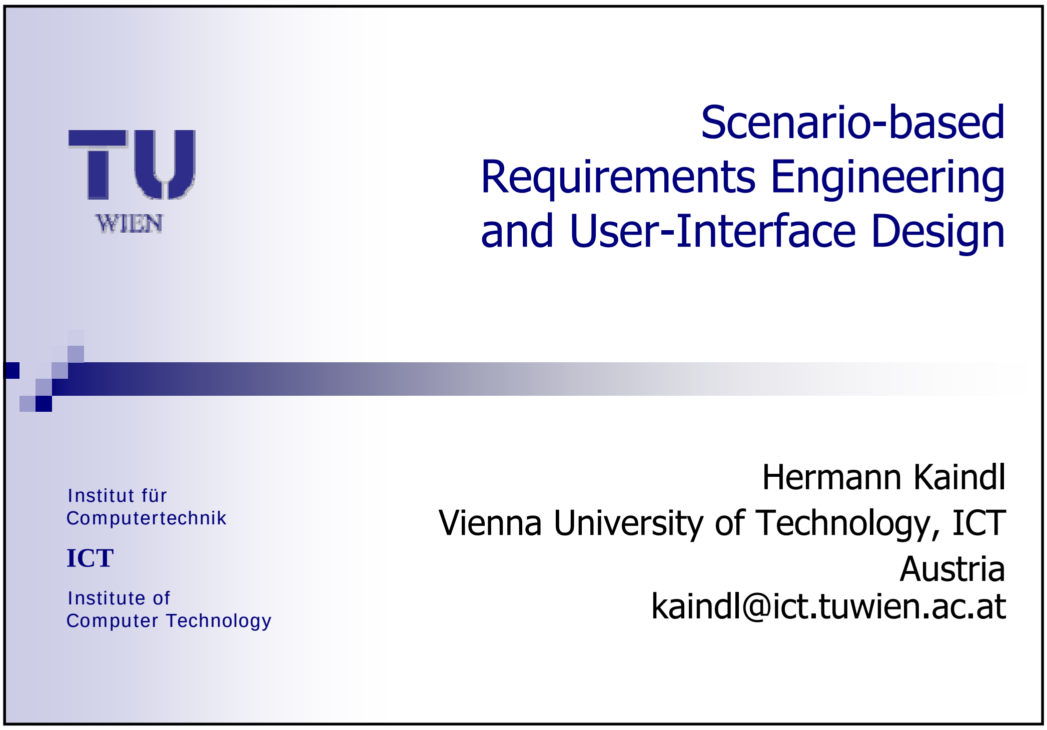


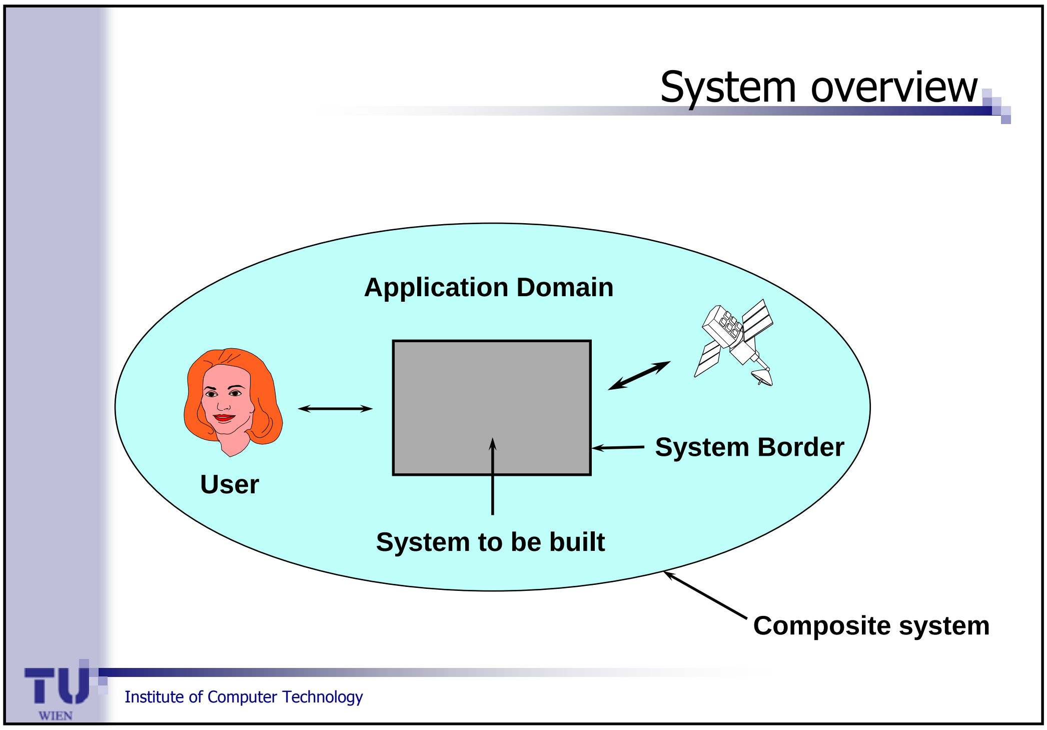
Scenario-based Requirements Engineering and User-Interface Design

Hermann Kaindl
Vienna University of Technology, ICT
Austria
kaindl@ict.tuwien.ac.at

Institut für Computertechnik

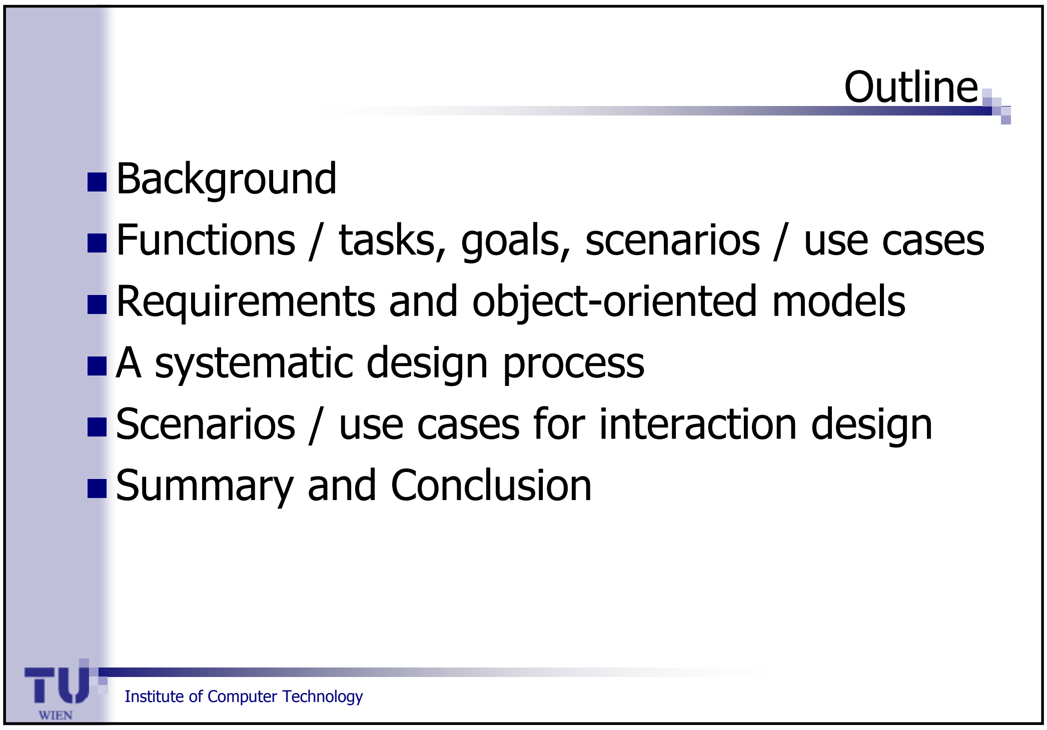
ICT

Institute of Computer Technology



System overview

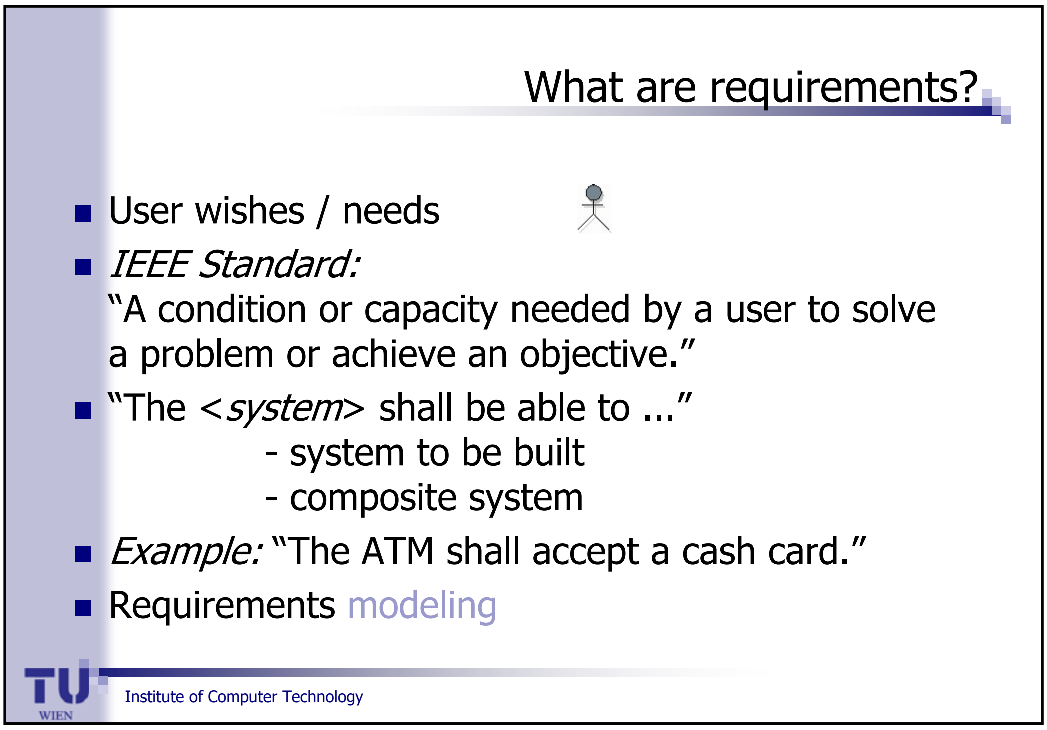
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- Requirements and object-oriented models
- A systematic design process
- Scenarios / use cases for interaction design
- Summary and Conclusion

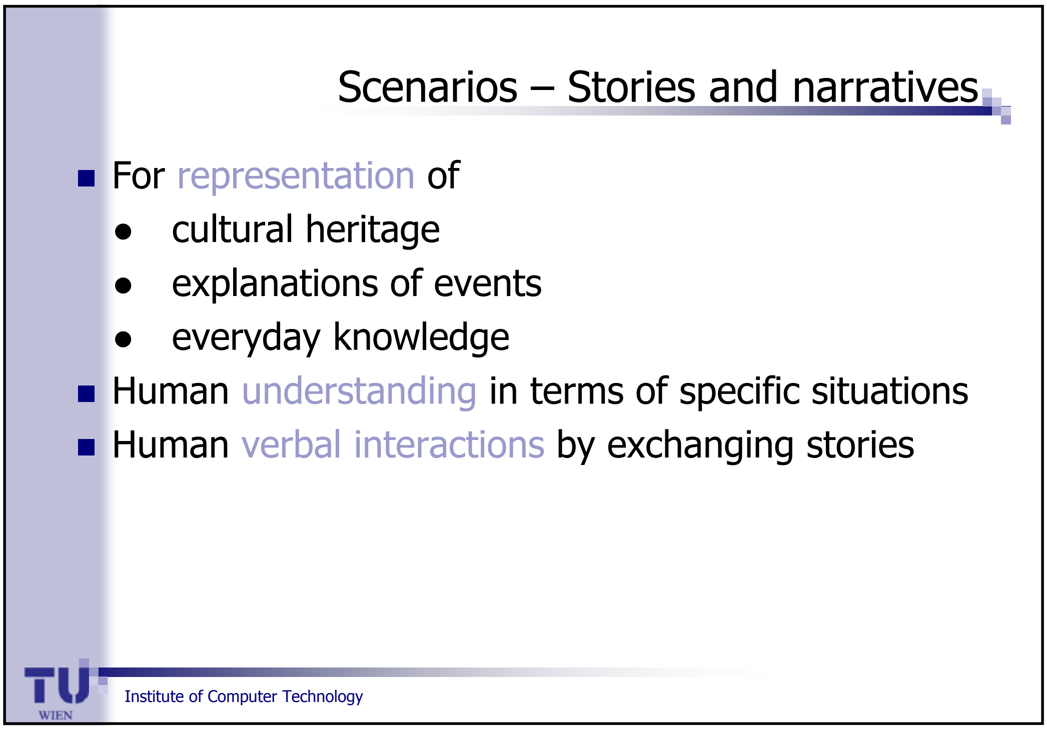
TU WIEN Institute of Computer Technology



What are requirements?

- User wishes / needs
- *IEEE Standard:*
  "A condition or capacity needed by a user to solve a problem or achieve an objective."
- "The <*system*> shall be able to ..."
  - system to be built
  - composite system
- *Example:* "The ATM shall accept a cash card."
- Requirements modeling

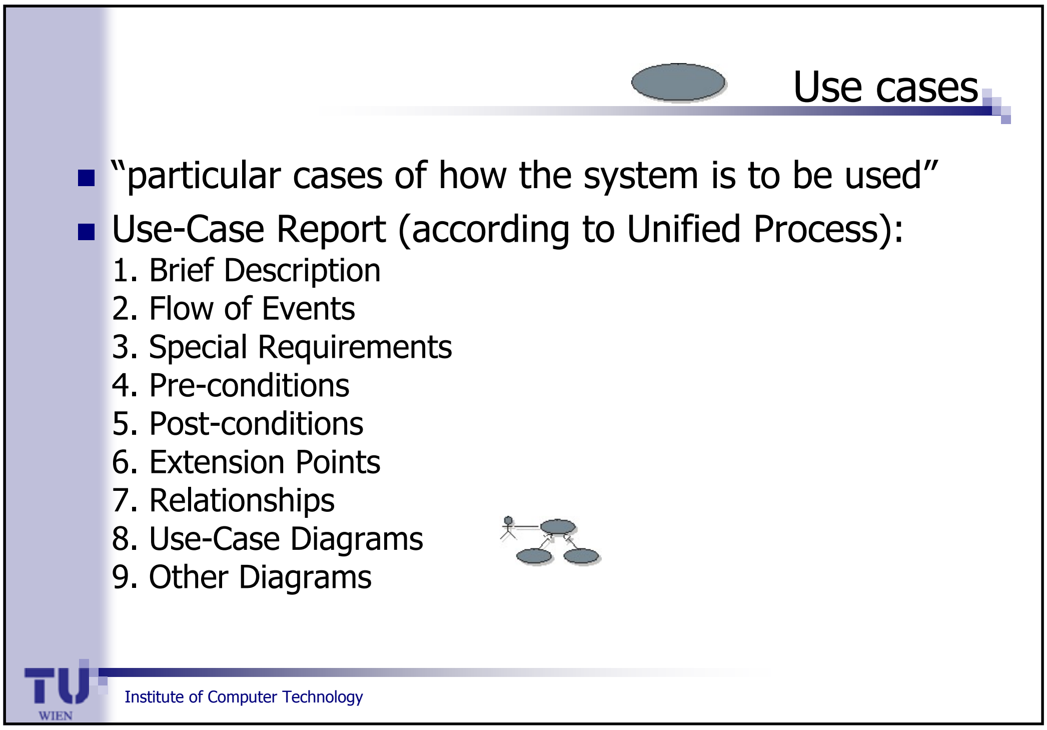
TU WIEN Institute of Computer Technology



Scenarios – Stories and narratives

- For representation of
  - cultural heritage
  - explanations of events
  - everyday knowledge
- Human understanding in terms of specific situations
- Human verbal interactions by exchanging stories

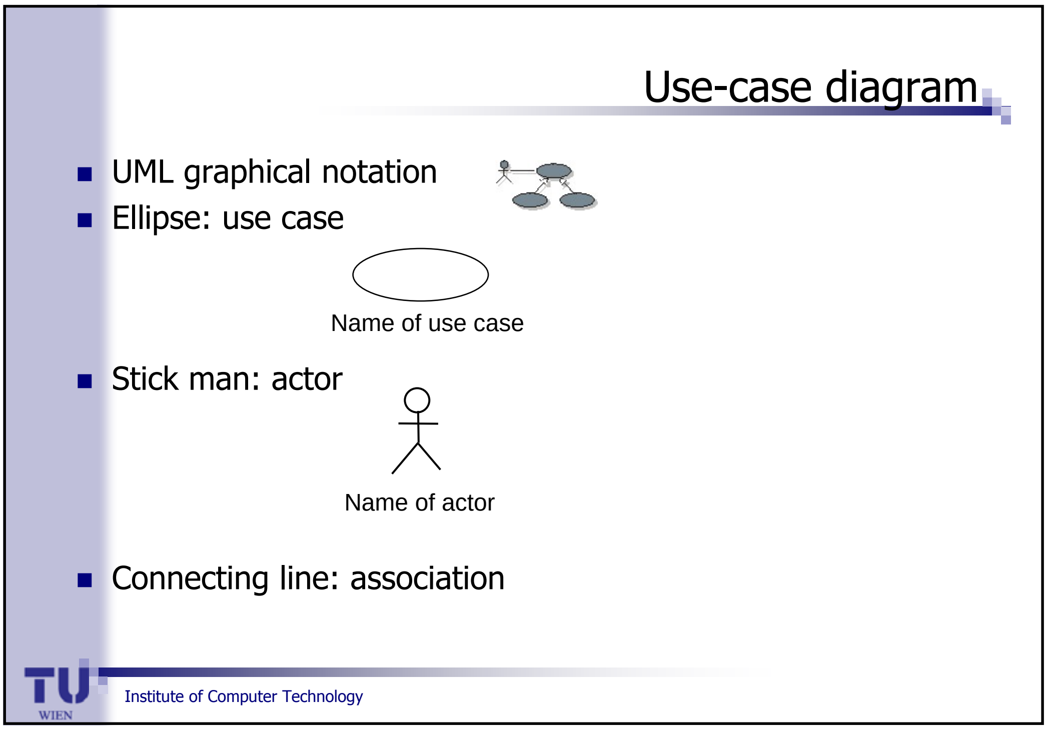
Institute of Computer Technology



Use cases

- “particular cases of how the system is to be used”
- Use-Case Report (according to Unified Process):
  1. Brief Description
  2. Flow of Events
  3. Special Requirements
  4. Pre-conditions
  5. Post-conditions
  6. Extension Points
  7. Relationships
  8. Use-Case Diagrams
  9. Other Diagrams

Institute of Computer Technology



Use-case diagram

- UML graphical notation
- Ellipse: use case

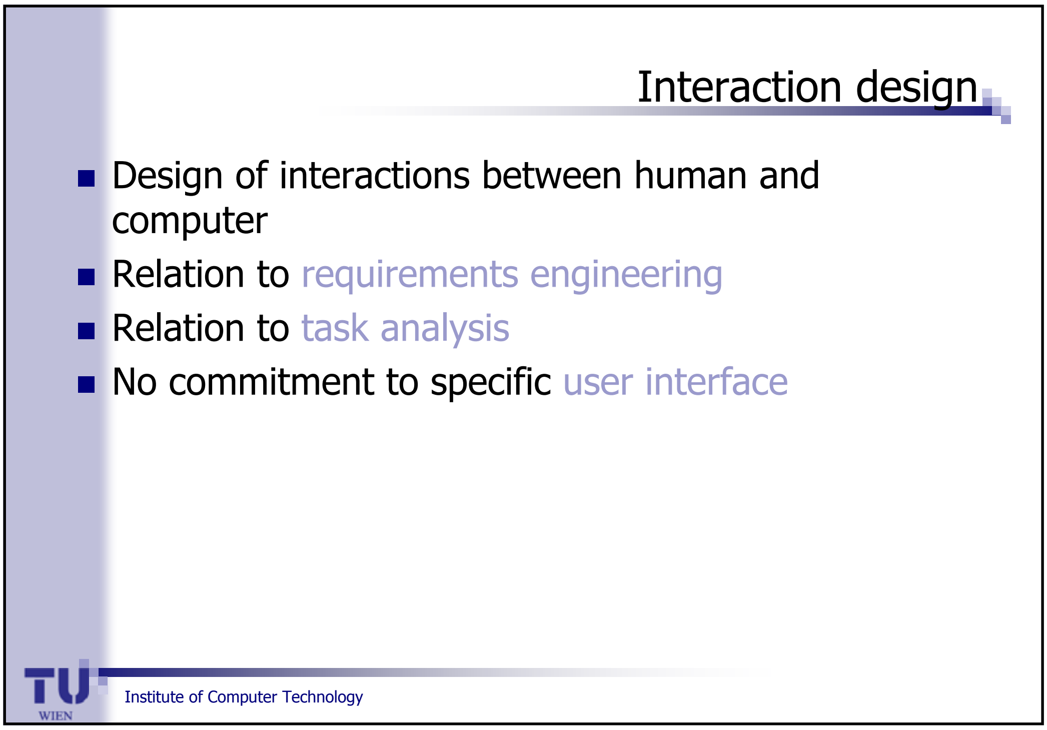
Name of use case

- Stick man: actor

Name of actor

- Connecting line: association

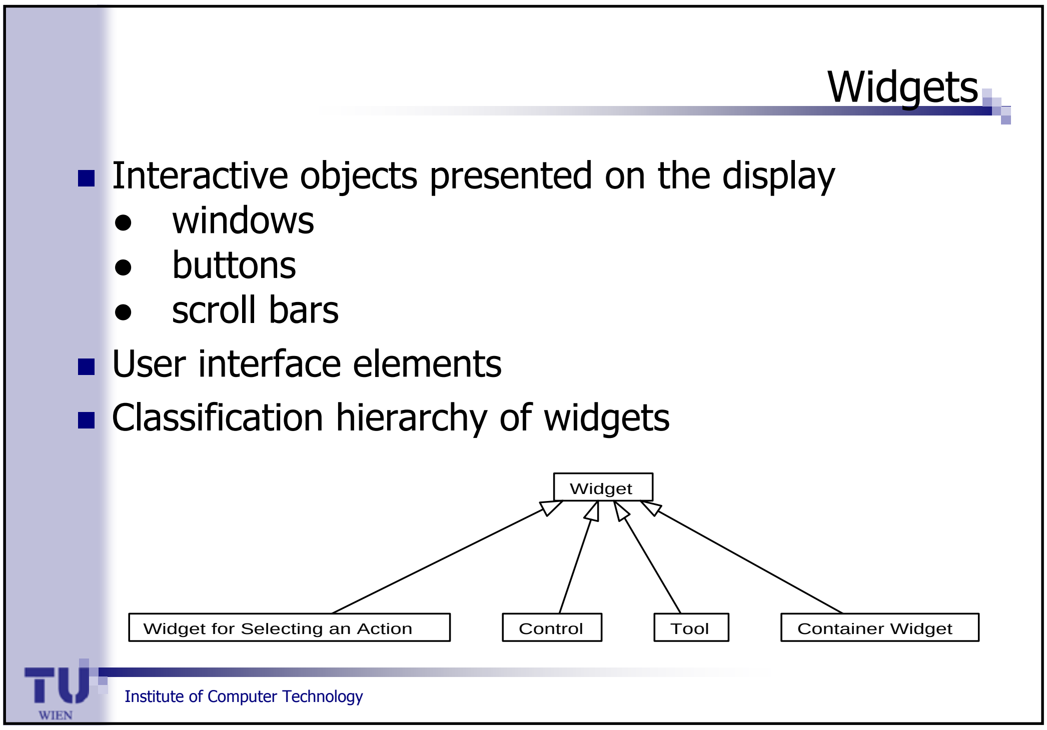
Institute of Computer Technology



Interaction design

- Design of interactions between human and computer
- Relation to requirements engineering
- Relation to task analysis
- No commitment to specific user interface

Institute of Computer Technology

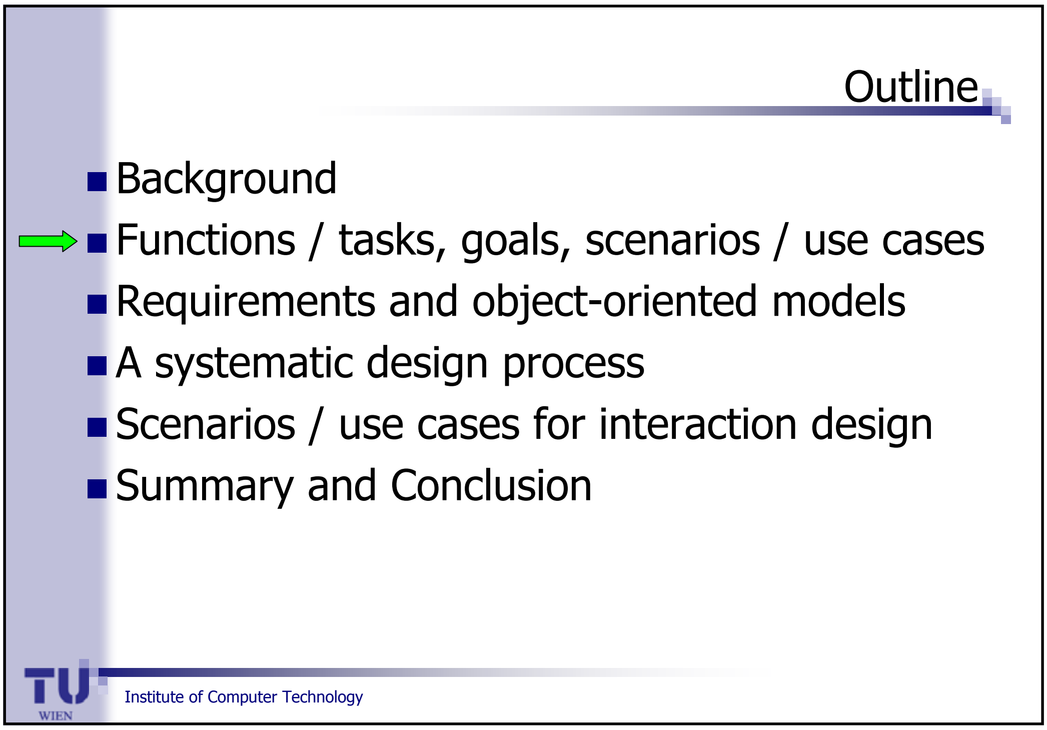


Widgets

- Interactive objects presented on the display
  - windows
  - buttons
  - scroll bars
- User interface elements
- Classification hierarchy of widgets

Widget
Widget for Selecting an Action
Control
Tool
Container Widget

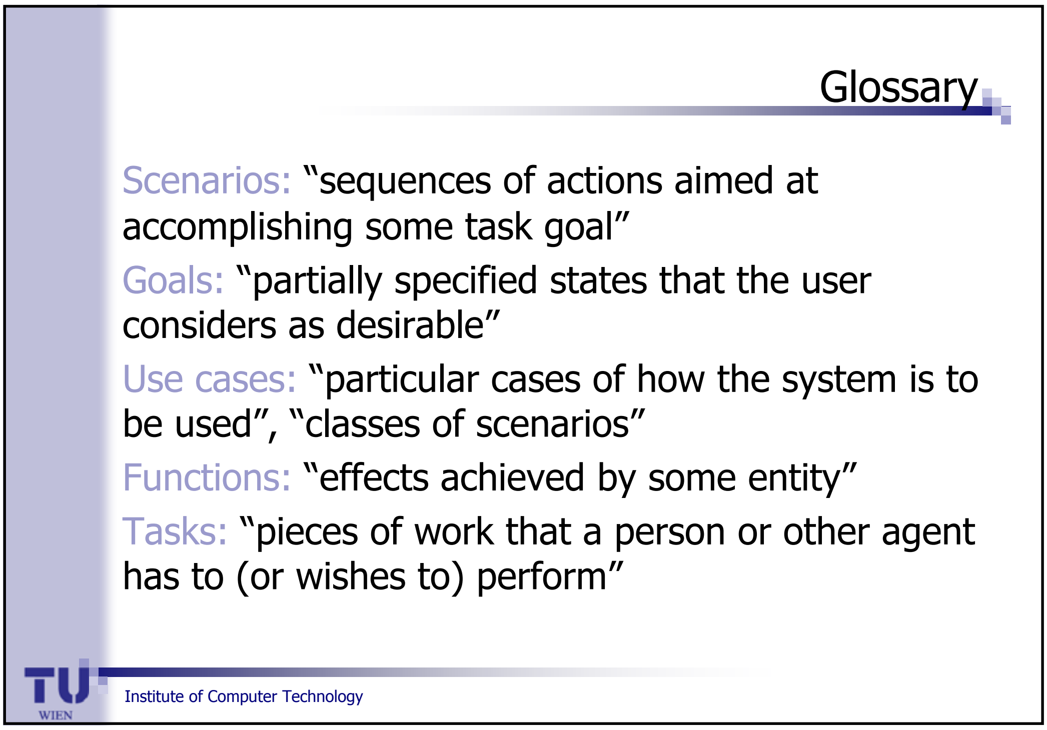
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- Requirements and object-oriented models
- A systematic design process
- Scenarios / use cases for interaction design
- Summary and Conclusion

Institute of Computer Technology



Glossary

Scenarios: “sequences of actions aimed at accomplishing some task goal”

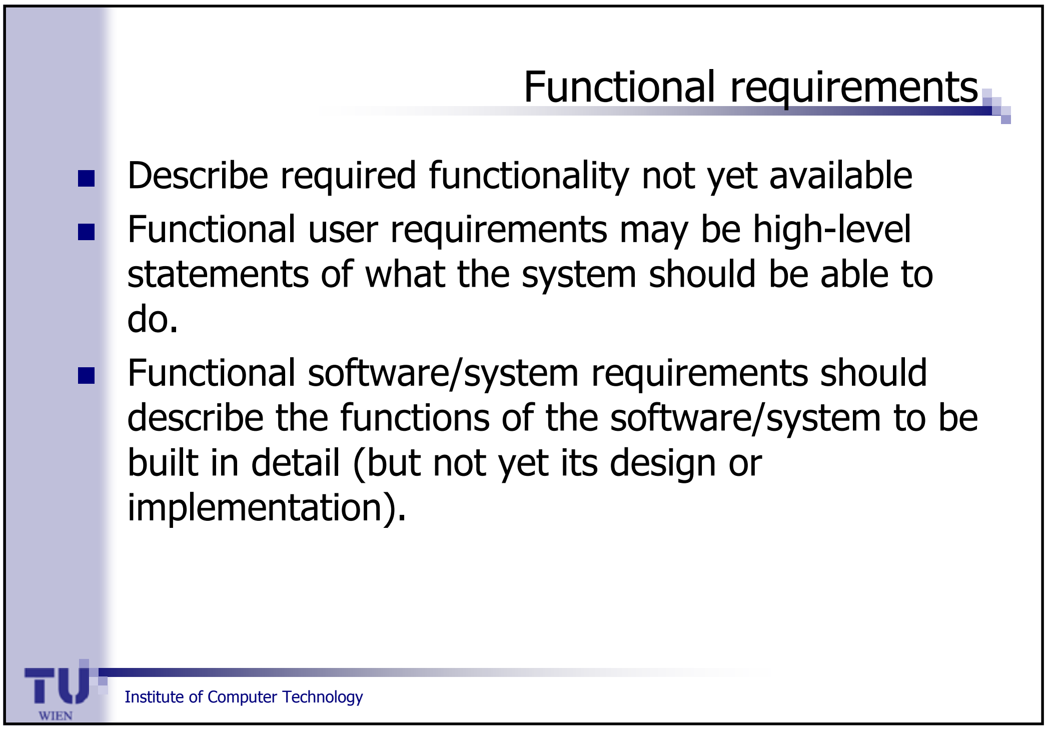
Goals: “partially specified states that the user considers as desirable”

Use cases: “particular cases of how the system is to be used”, “classes of scenarios”

Functions: “effects achieved by some entity”

Tasks: “pieces of work that a person or other agent has to (or wishes to) perform”

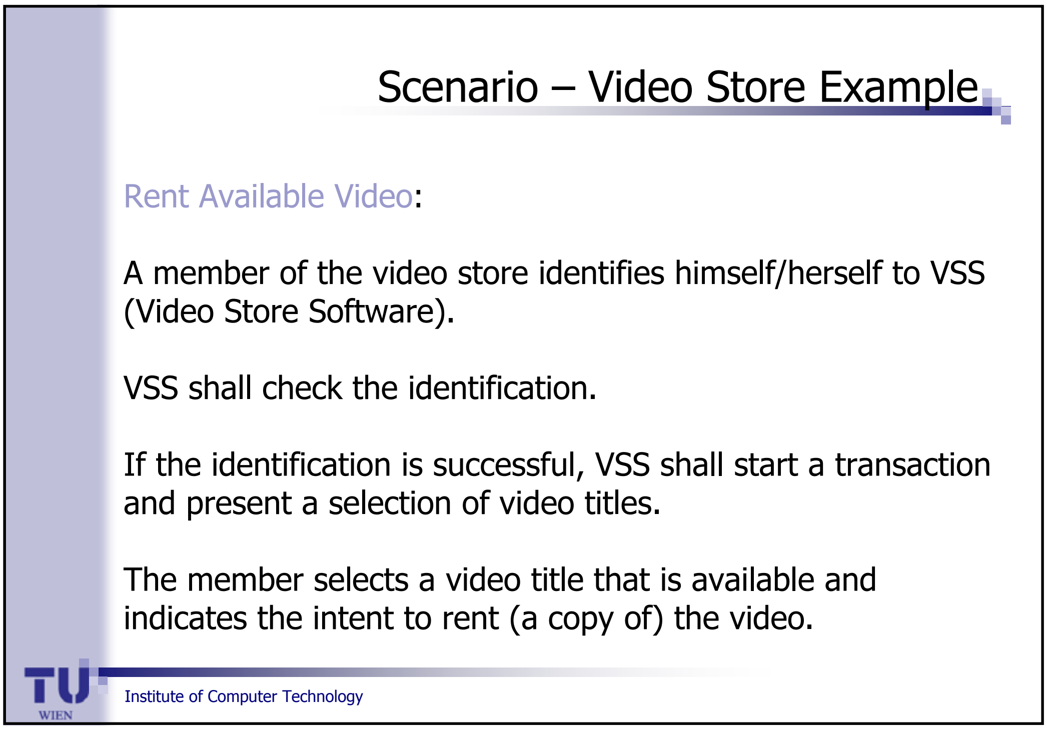
Institute of Computer Technology



Functional requirements

- Describe required functionality not yet available
- Functional user requirements may be high-level statements of what the system should be able to do.
- Functional software/system requirements should describe the functions of the software/system to be built in detail (but not yet its design or implementation).

Institute of Computer Technology



Scenario – Video Store Example

Rent Available Video:

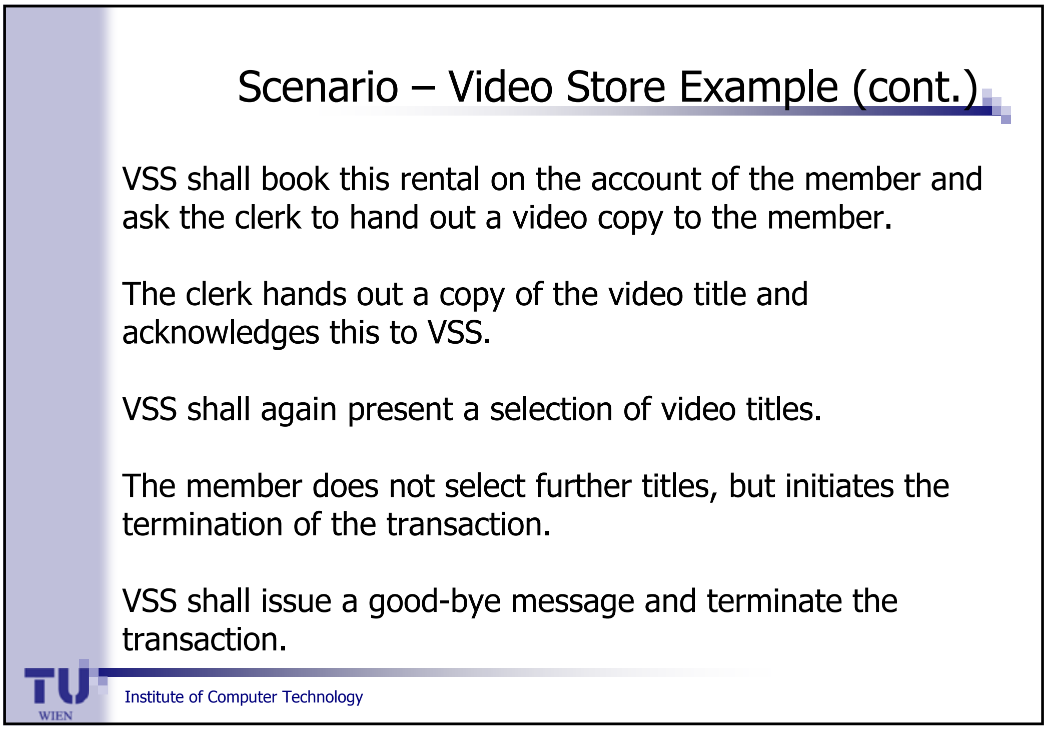
A member of the video store identifies himself/herself to VSS (Video Store Software).

VSS shall check the identification.

If the identification is successful, VSS shall start a transaction and present a selection of video titles.

The member selects a video title that is available and indicates the intent to rent (a copy of) the video.

Institute of Computer Technology



Scenario – Video Store Example (cont.)

VSS shall book this rental on the account of the member and ask the clerk to hand out a video copy to the member.

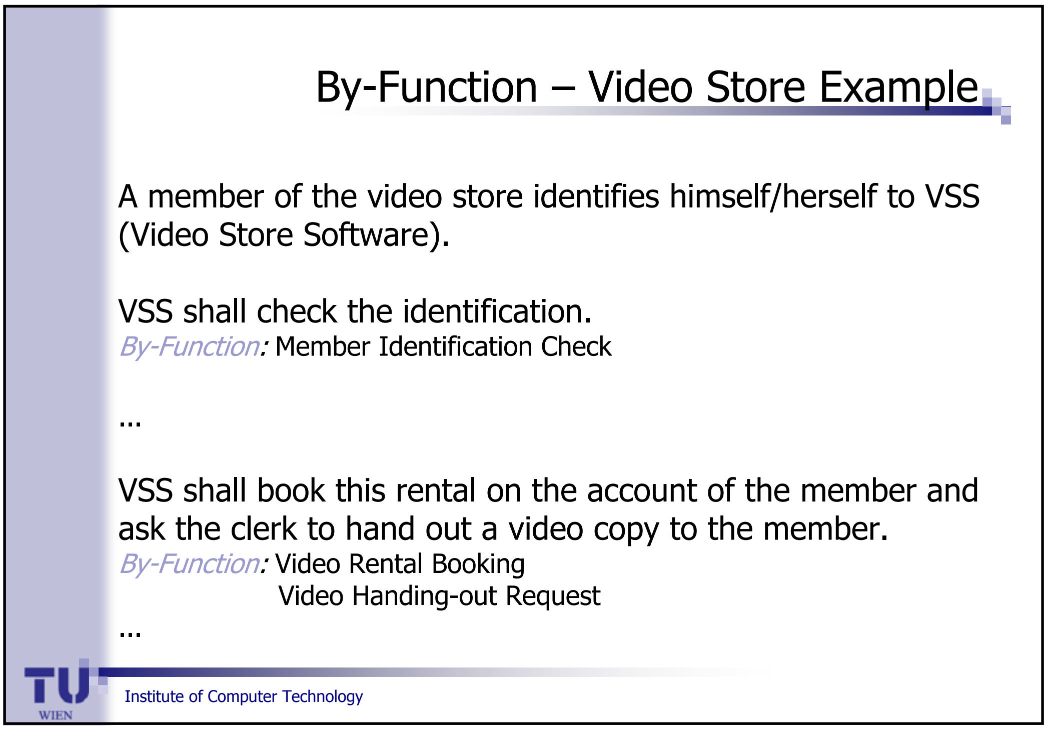
The clerk hands out a copy of the video title and acknowledges this to VSS.

VSS shall again present a selection of video titles.

The member does not select further titles, but initiates the termination of the transaction.

VSS shall issue a good-bye message and terminate the transaction.

Institute of Computer Technology



By-Function – Video Store Example

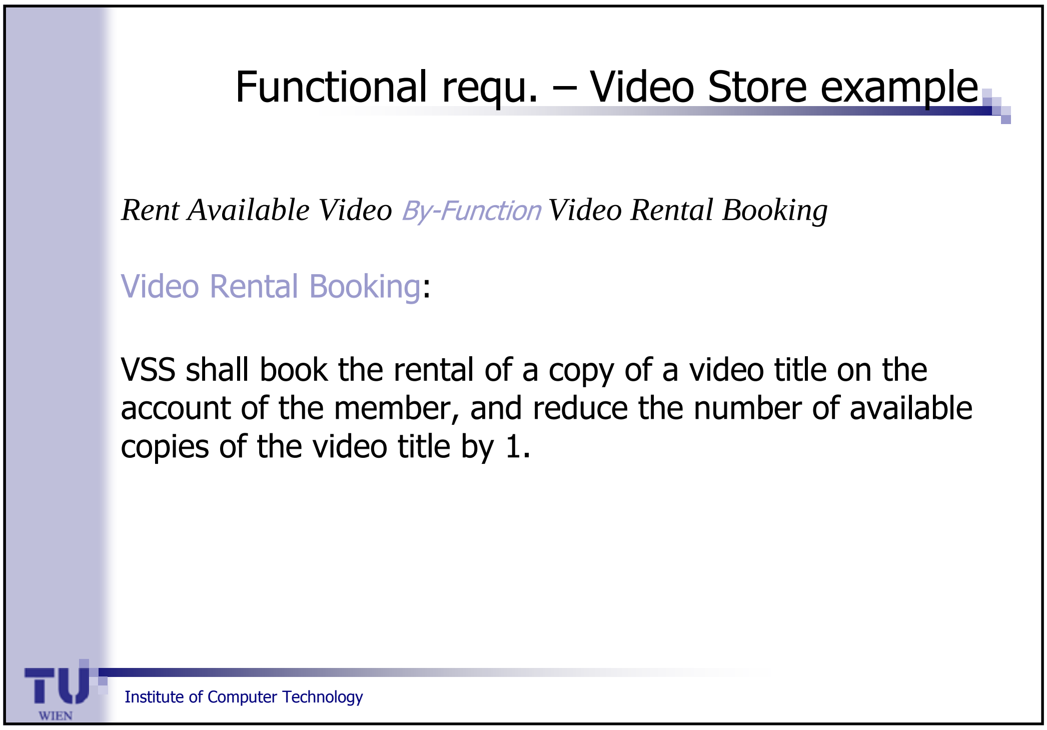
A member of the video store identifies himself/herself to VSS (Video Store Software).

VSS shall check the identification.
By-Function: Member Identification Check

…

VSS shall book this rental on the account of the member and ask the clerk to hand out a video copy to the member.
By-Function: Video Rental Booking
Video Handing-out Request
…

Institute of Computer Technology



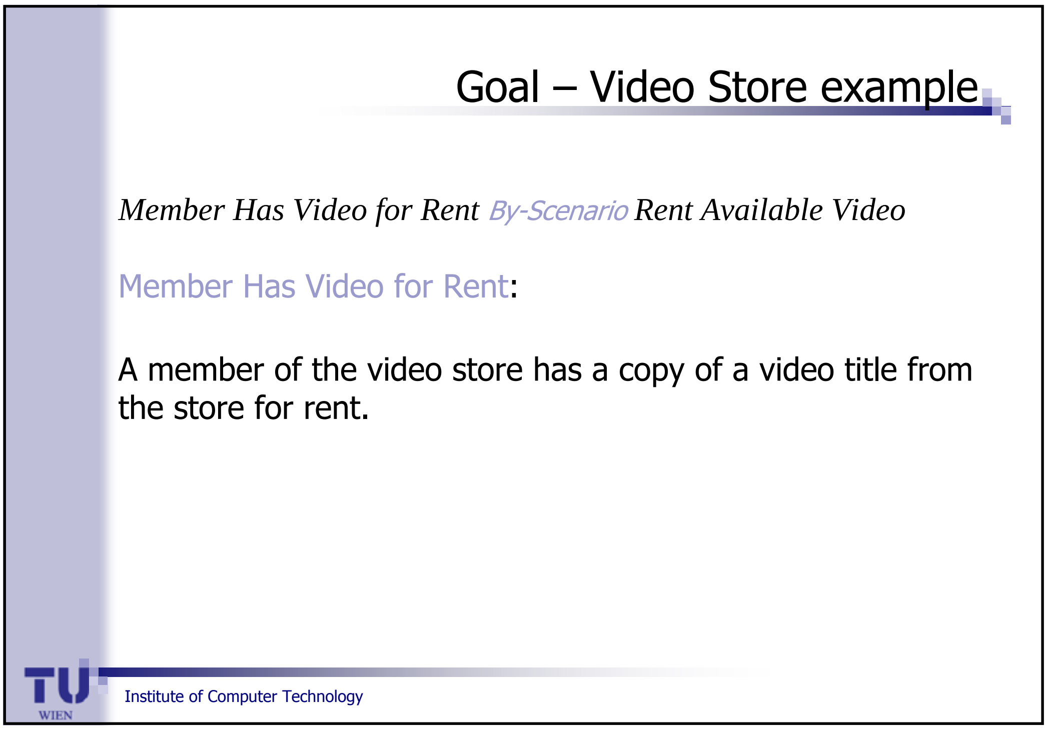
Functional requ. – Video Store example

Rent Available Video By-Function Video Rental Booking

Video Rental Booking:

VSS shall book the rental of a copy of a video title on the account of the member, and reduce the number of available copies of the video title by 1.

Institute of Computer Technology



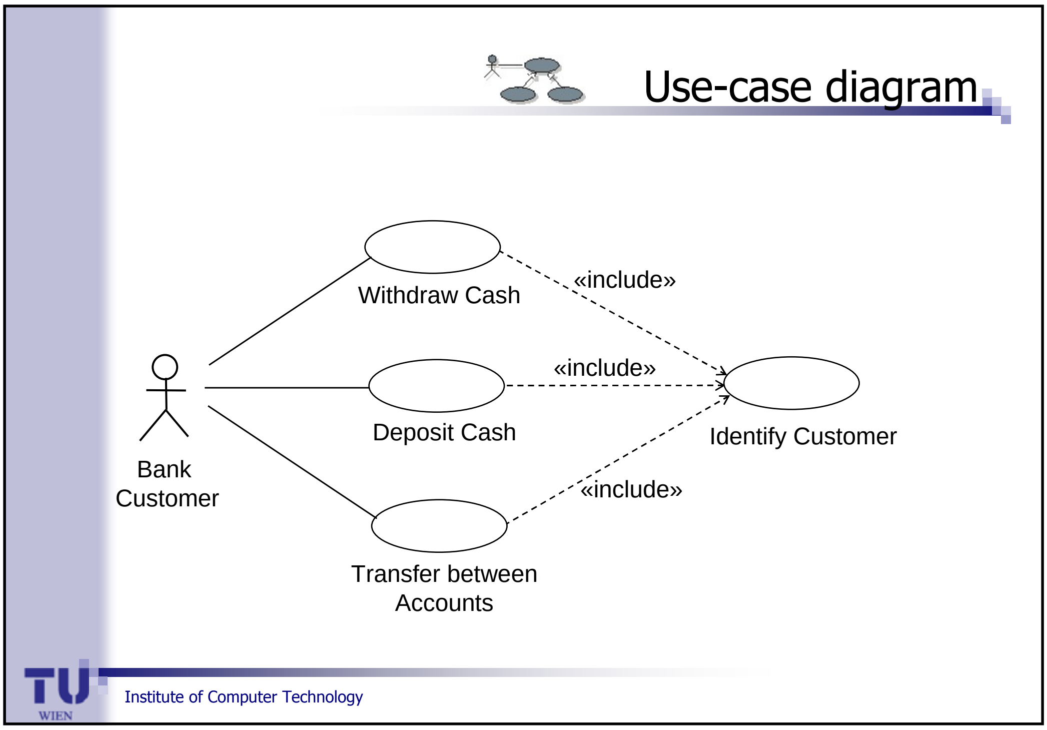
Goal – Video Store example

Member Has Video for Rent By-Scenario Rent Available Video

Member Has Video for Rent:

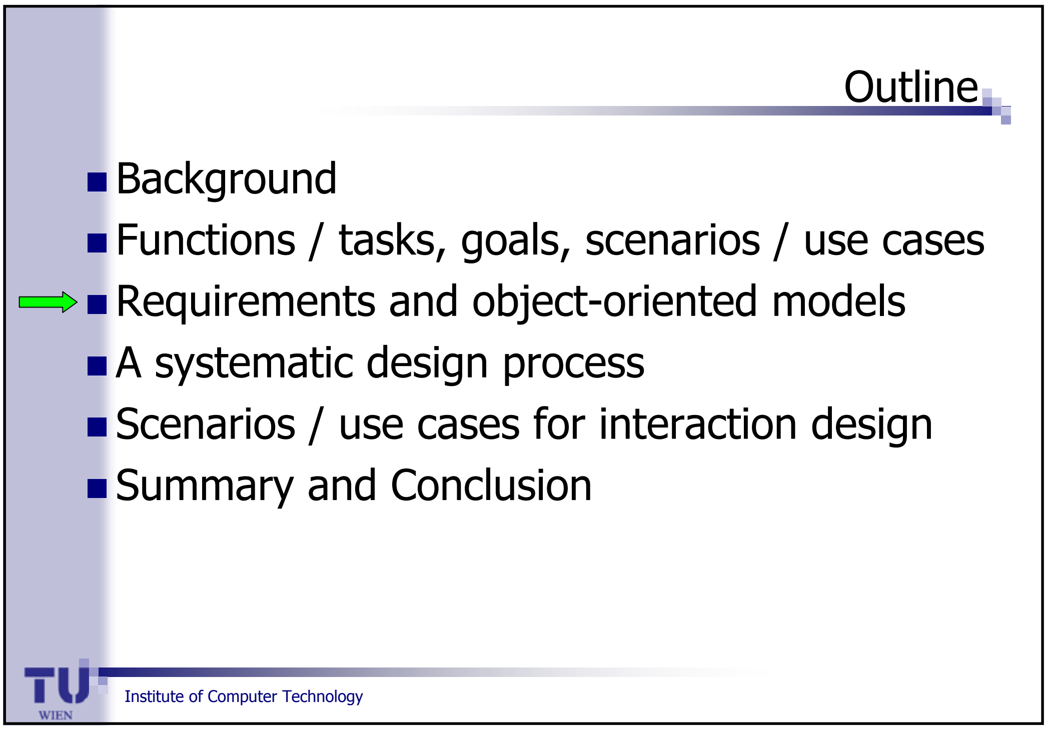
A member of the video store has a copy of a video title from the store for rent.

Institute of Computer Technology



Use-case diagram

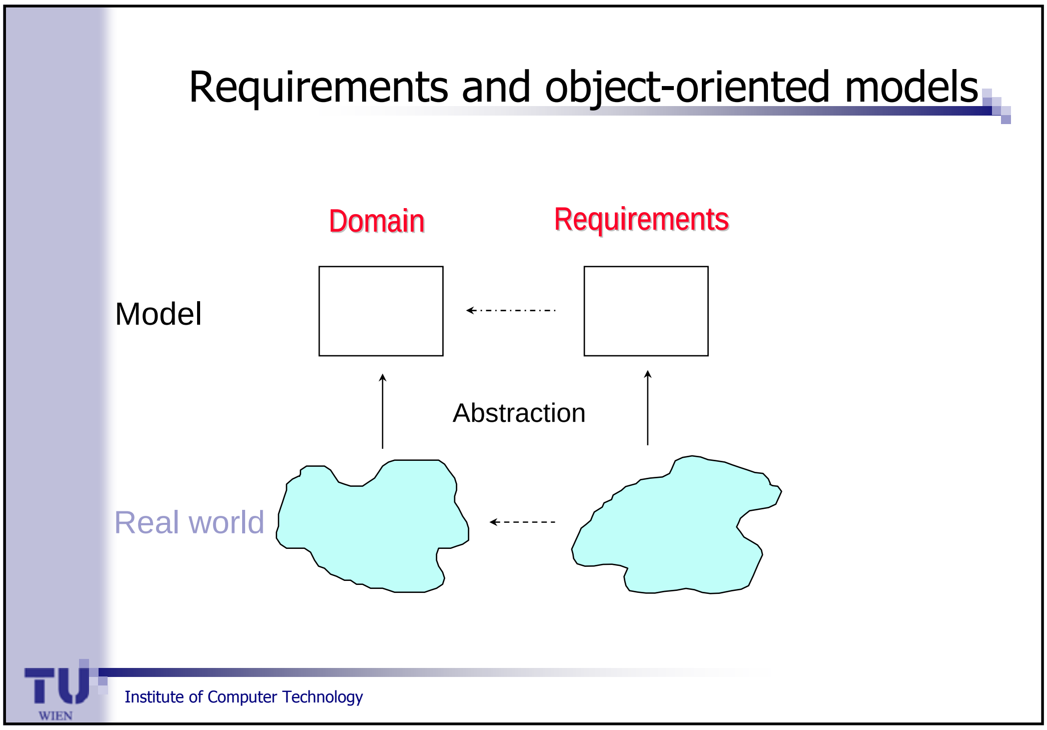
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- ⇨ Requirements and object-oriented models
- A systematic design process
- Scenarios / use cases for interaction design
- Summary and Conclusion

Institute of Computer Technology



Requirements and object-oriented models

Domain

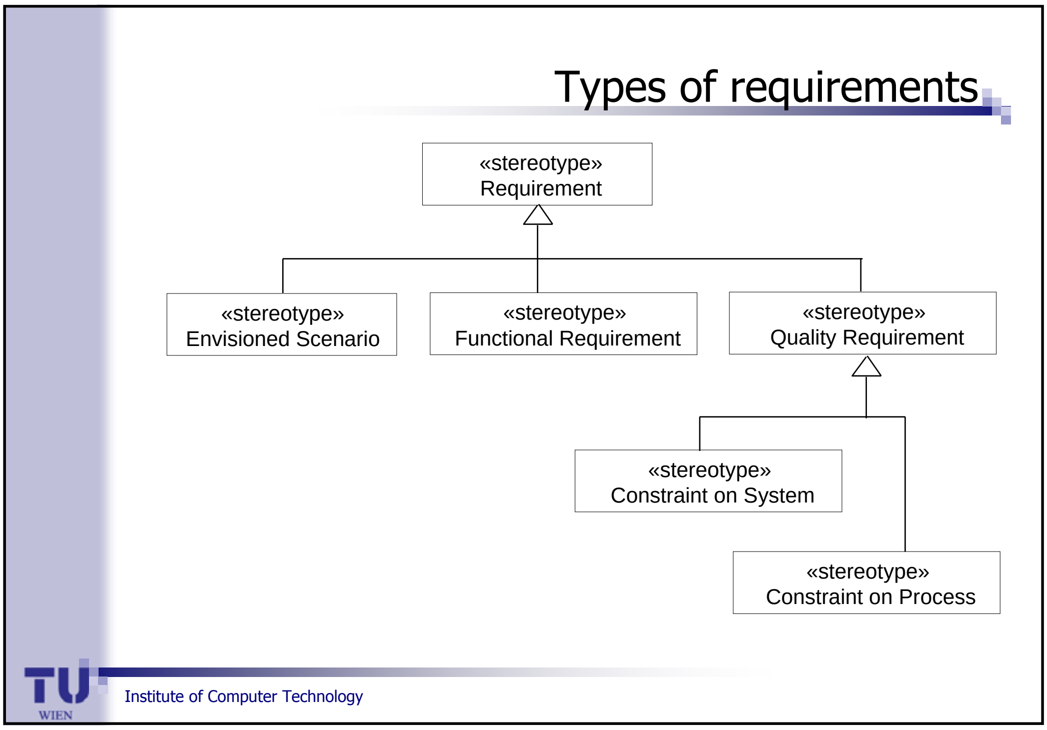
Requirements

Model

Abstraction

Real world

Institute of Computer Technology



Types of requirements

«stereotype»
Requirement

«stereotype»
Envisioned Scenario

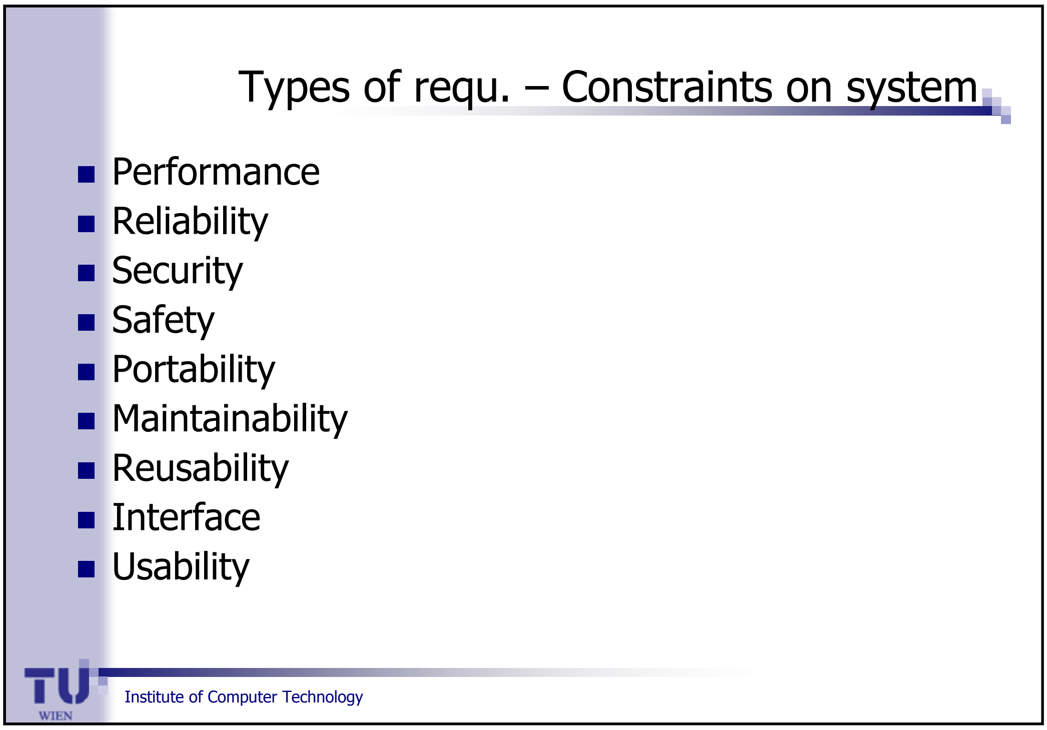
«stereotype»
Functional Requirement

«stereotype»
Quality Requirement

«stereotype»
Constraint on System

«stereotype»
Constraint on Process

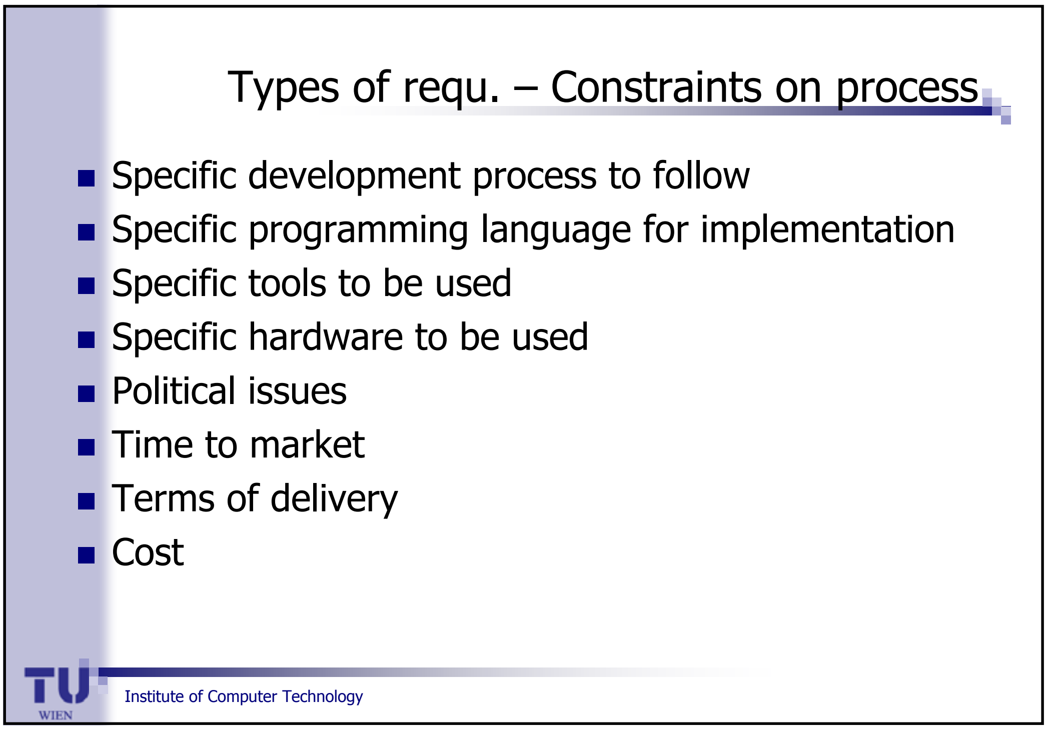
Institute of Computer Technology



Types of requ. – Constraints on system

- Performance
- Reliability
- Security
- Safety
- Portability
- Maintainability
- Reusability
- Interface
- Usability

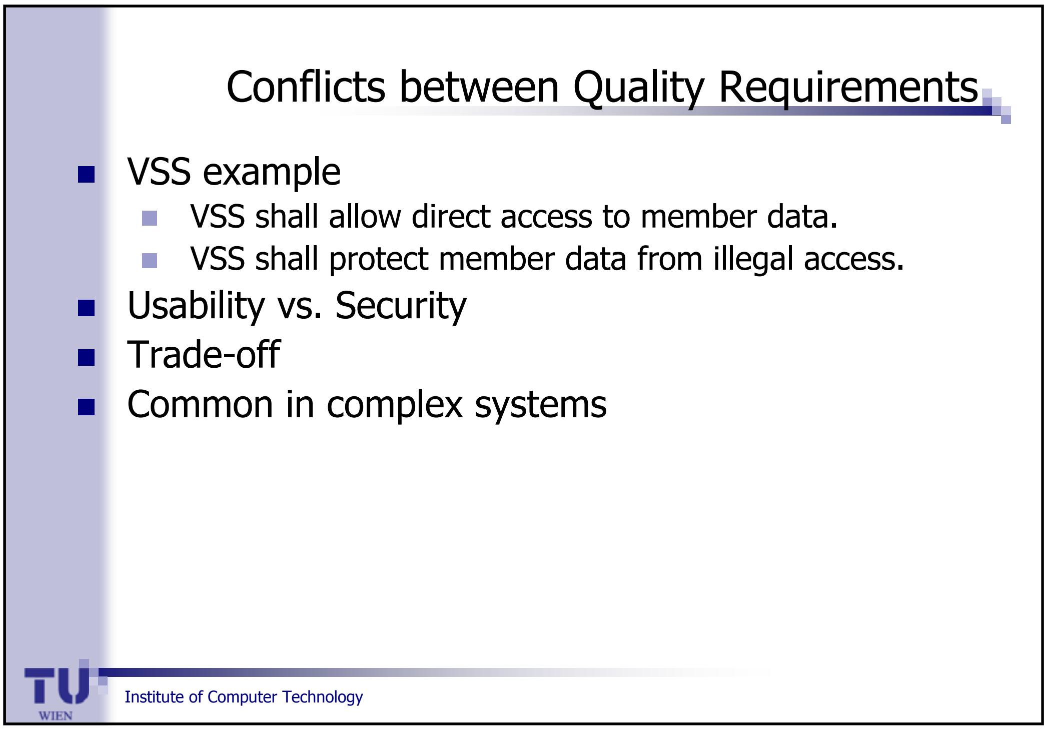
Institute of Computer Technology



Types of requ. – Constraints on process

- Specific development process to follow
- Specific programming language for implementation
- Specific tools to be used
- Specific hardware to be used
- Political issues
- Time to market
- Terms of delivery
- Cost

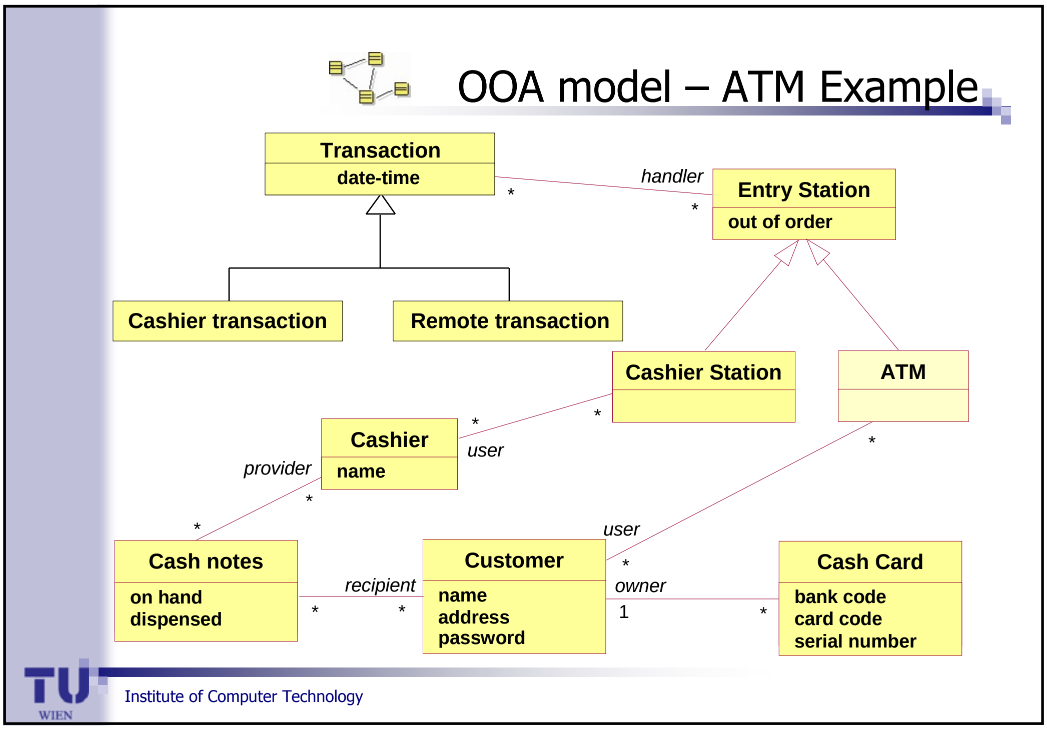
Institute of Computer Technology



Conflicts between Quality Requirements

- VSS example
  - VSS shall allow direct access to member data.
  - VSS shall protect member data from illegal access.
- Usability vs. Security
- Trade-off
- Common in complex systems

Institute of Computer Technology



OOA model – ATM Example

Transaction
date-time

handler
*
*

Entry Station
out of order

Cashier transaction

Remote transaction

Cashier Station

ATM

*
*
user

Cashier
name

provider
*
*

*
user
*

Cash notes
on hand
dispensed

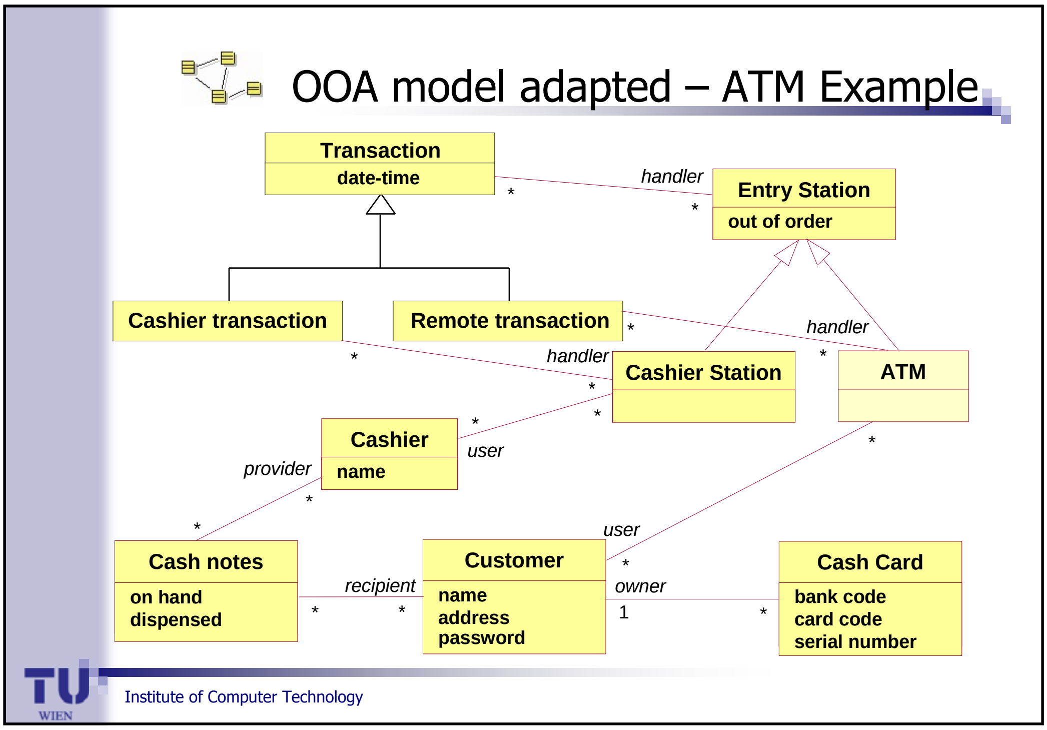
recipient
*
*

Customer
name
address
password

owner
1
*

Cash Card
bank code
card code
serial number

Institute of Computer Technology



OOA model adapted – ATM Example

Transaction
date-time

handler
*
*

Entry Station
out of order

Cashier transaction

Remote transaction

*
handler
*

*
handler
*

Cashier Station

ATM

*
*
user

Cashier
name

provider
*
*

*
user
*

Cash notes
on hand
dispensed

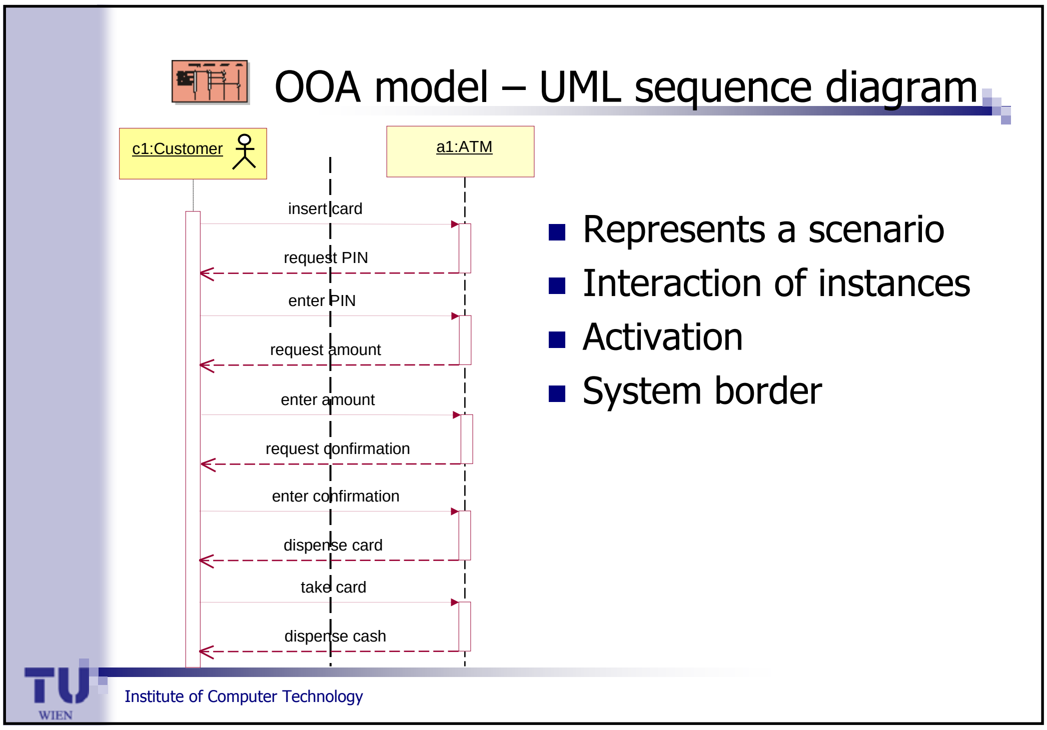
recipient
*
*

Customer
name
address
password

owner
1
*

Cash Card
bank code
card code
serial number

Institute of Computer Technology



OOA model – UML sequence diagram

c1:Customer

a1:ATM

insert card

request PIN

enter PIN

request amount

enter amount

request confirmation

enter confirmation

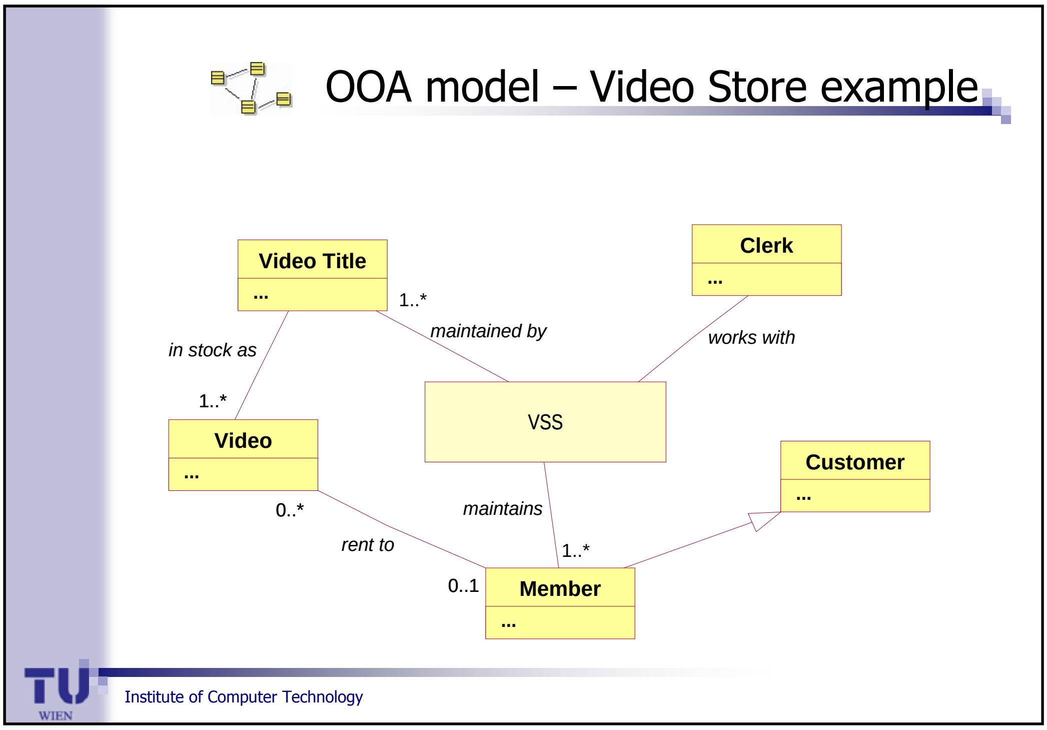
dispense card

take card

dispense cash

- Represents a scenario
- Interaction of instances
- Activation
- System border

Institute of Computer Technology



OOA model – Video Store example

Video Title

...

in stock as

1..*

Video

...

1..*

maintained by

Clerk

...

works with

VSS

0..*

rent to

0..1

maintains

1..*

Member

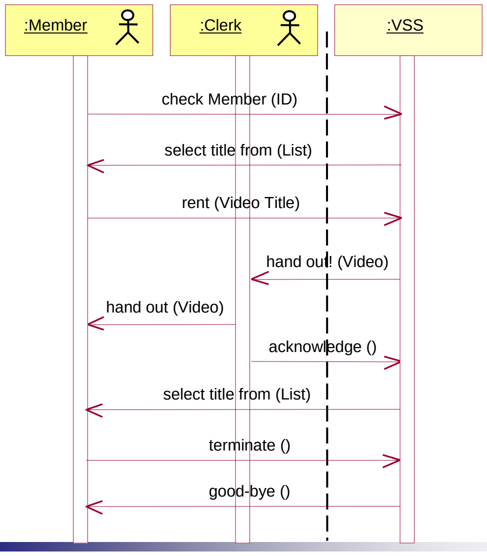
...

Customer

...

Institute of Computer Technology

OOA model – UML sequence diagram

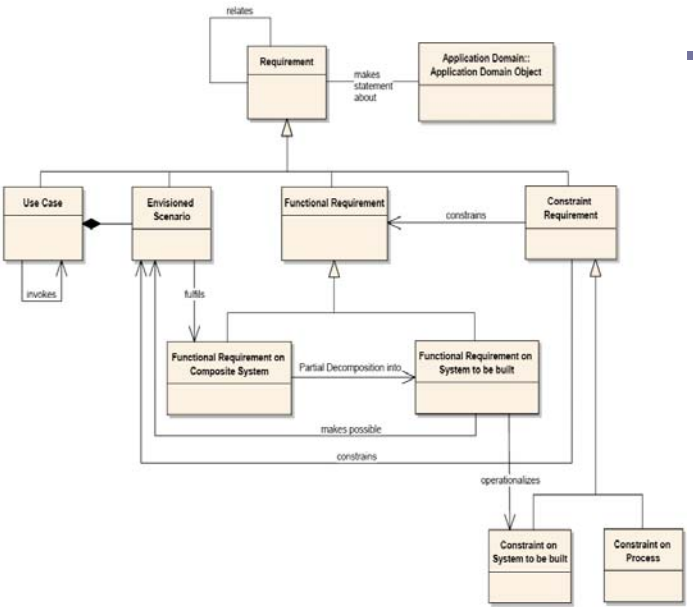


Unnamed instances

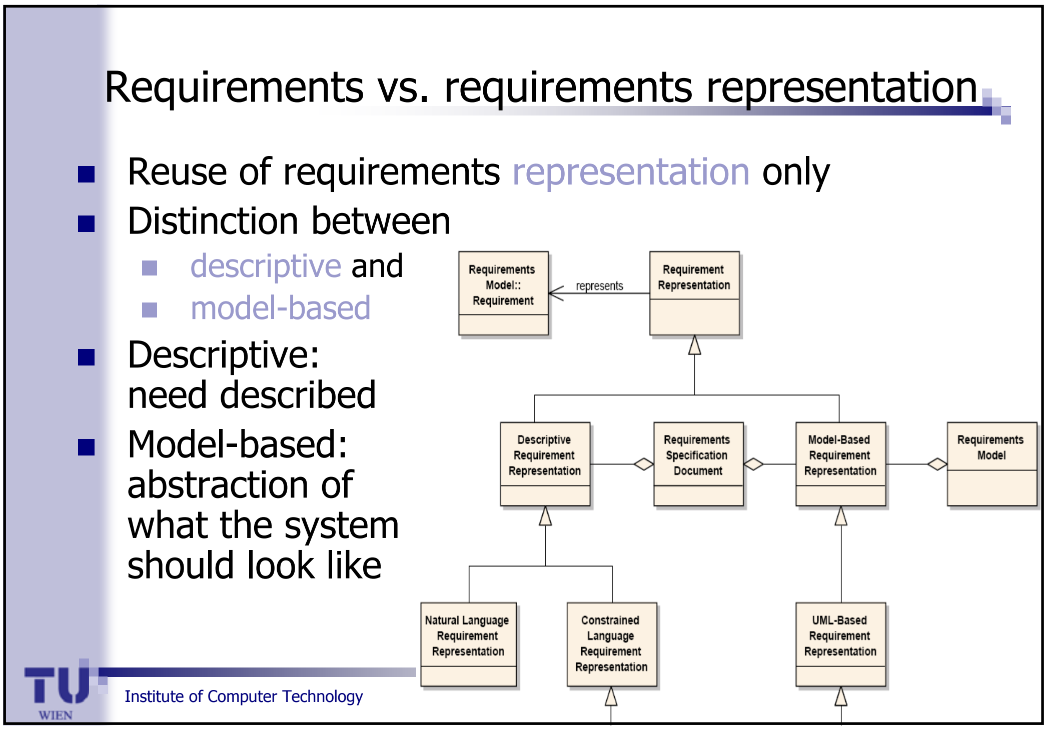
Concurrent objects

Institute of Computer Technology

Conceptual model



Institute of Computer Technology

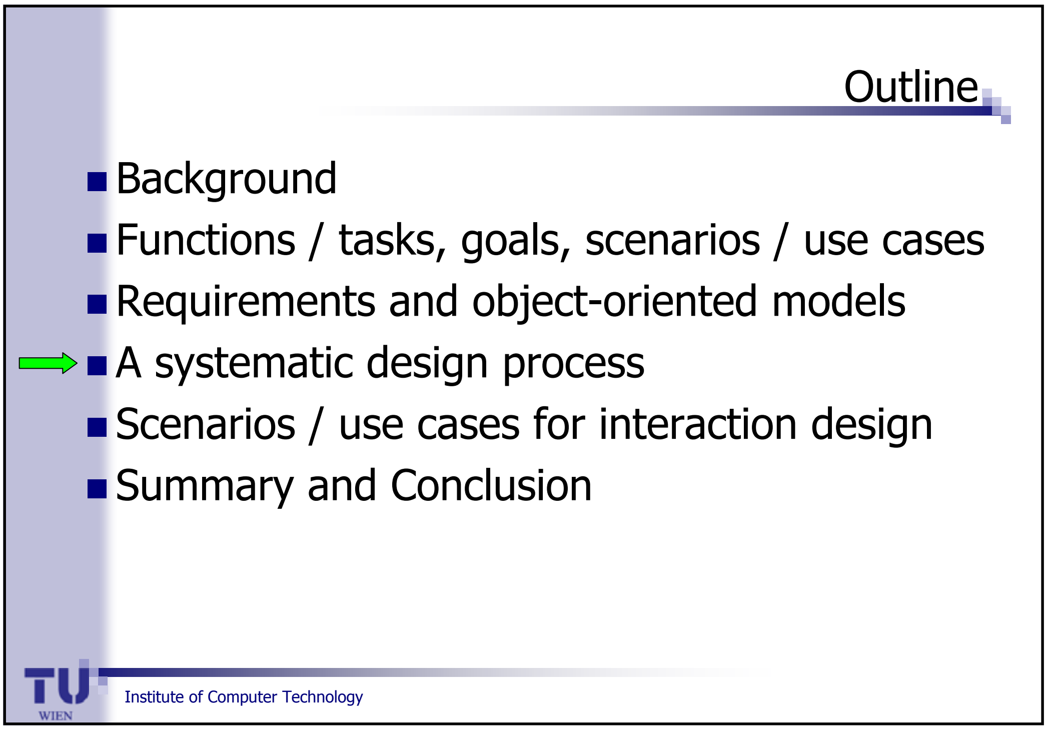


Requirements vs. requirements representation

- Reuse of requirements representation only
- Distinction between
  - descriptive and
  - model-based
- Descriptive: need described
- Model-based: abstraction of what the system should look like

Requirements Model:: Requirement
represents
Requirement Representation
Descriptive Requirement Representation
Requirements Specification Document
Model-Based Requirement Representation
Requirements Model
Natural Language Requirement Representation
Constrained Language Requirement Representation
UML-Based Requirement Representation

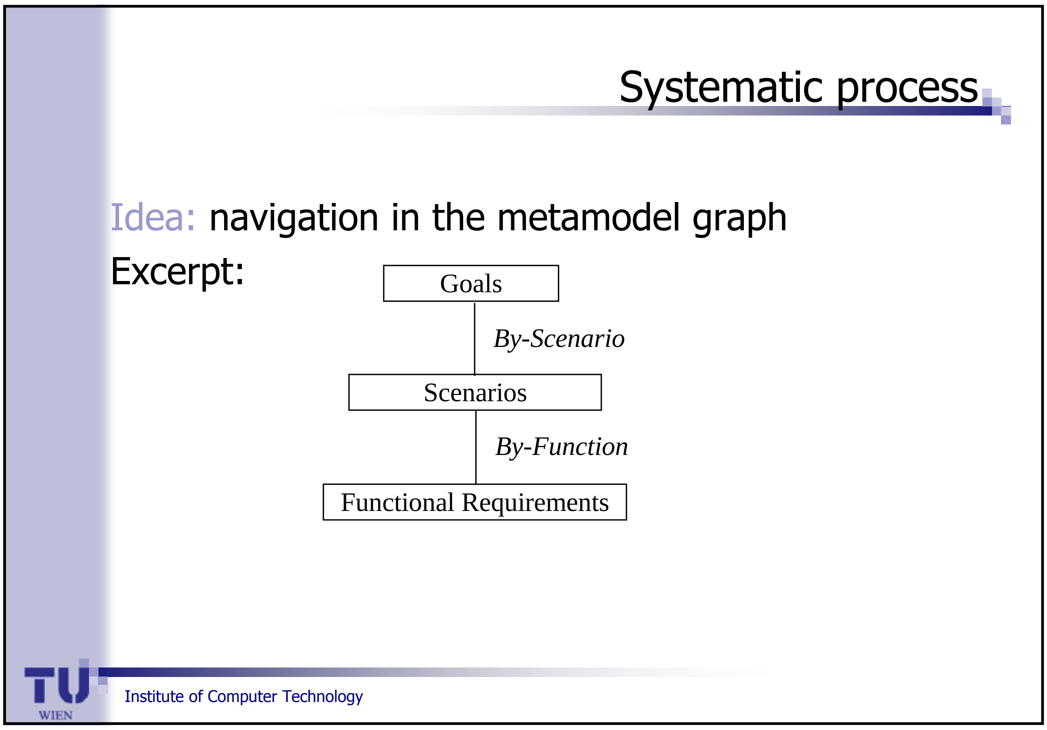
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- Requirements and object-oriented models
- A systematic design process
- Scenarios / use cases for interaction design
- Summary and Conclusion

Institute of Computer Technology



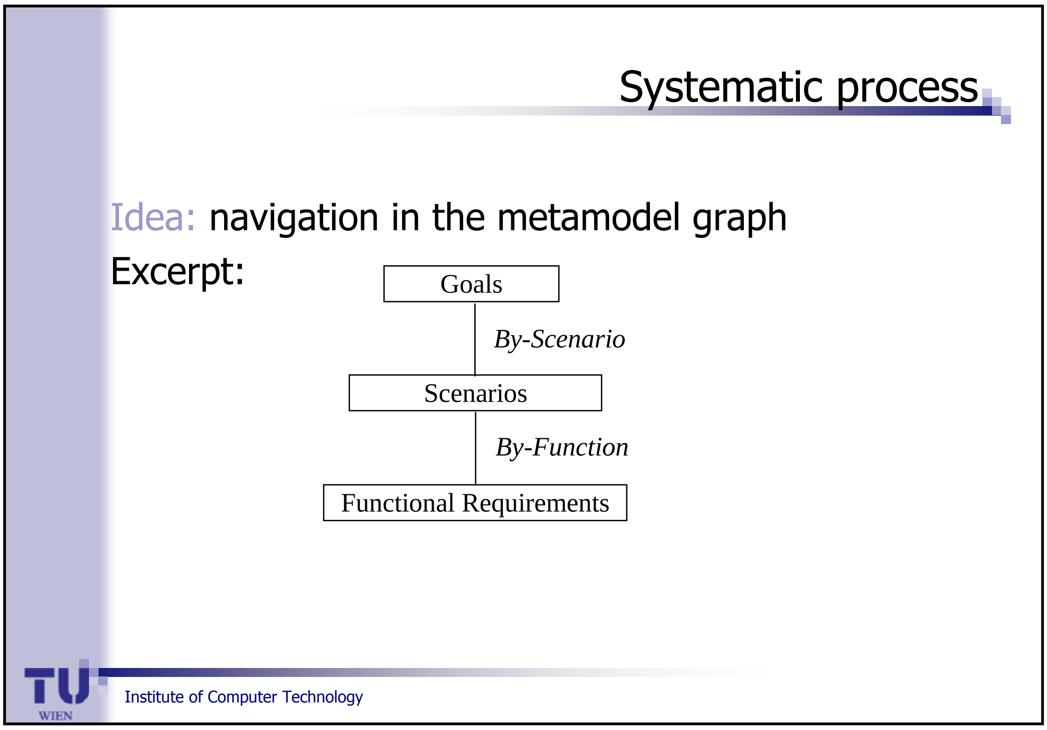
Systematic process

Idea: navigation in the metamodel graph

Excerpt:

Goals
By-Scenario
Scenarios
By-Function
Functional Requirements

Institute of Computer Technology



Systematic process

Idea: navigation in the metamodel graph

Excerpt:

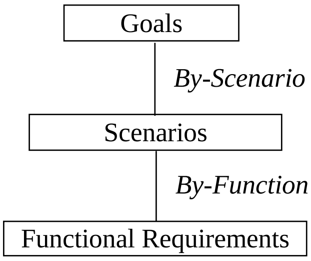
Goals
By-Scenario
Scenarios
By-Function
Functional Requirements

Institute of Computer Technology

Systematic process

Idea: navigation in the metamodel graph

Excerpt:



Goals

By-Scenario

Scenarios

By-Function

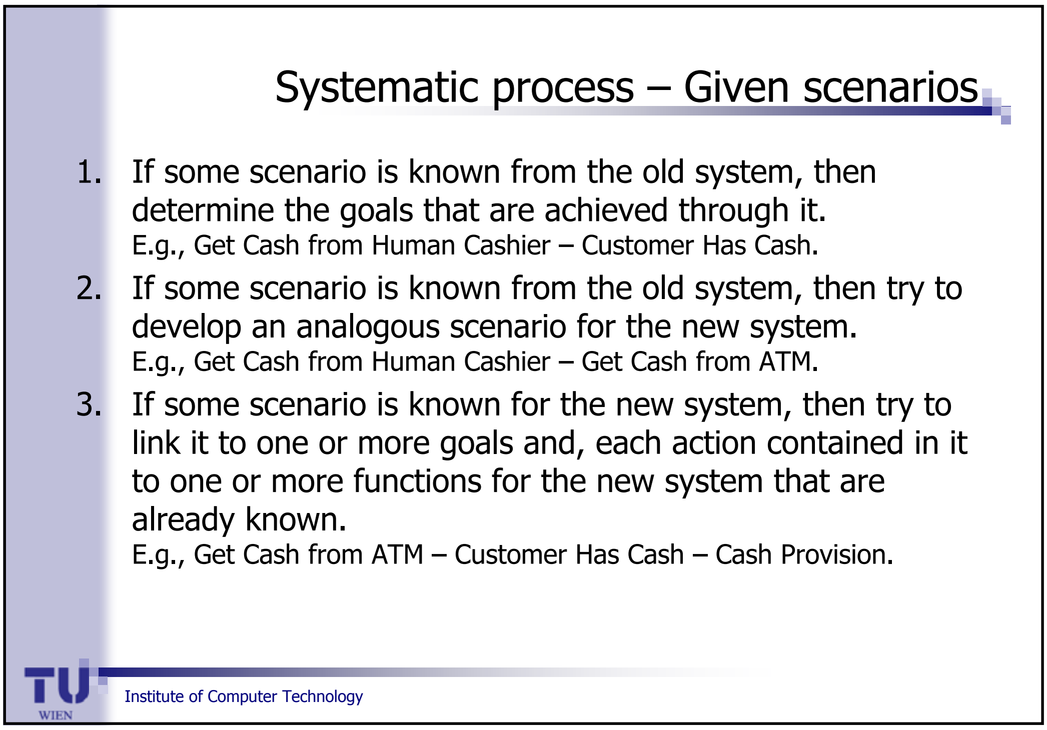
Functional Requirements

What is known already?

Old system or system to be built?

Systematic process – Given goals

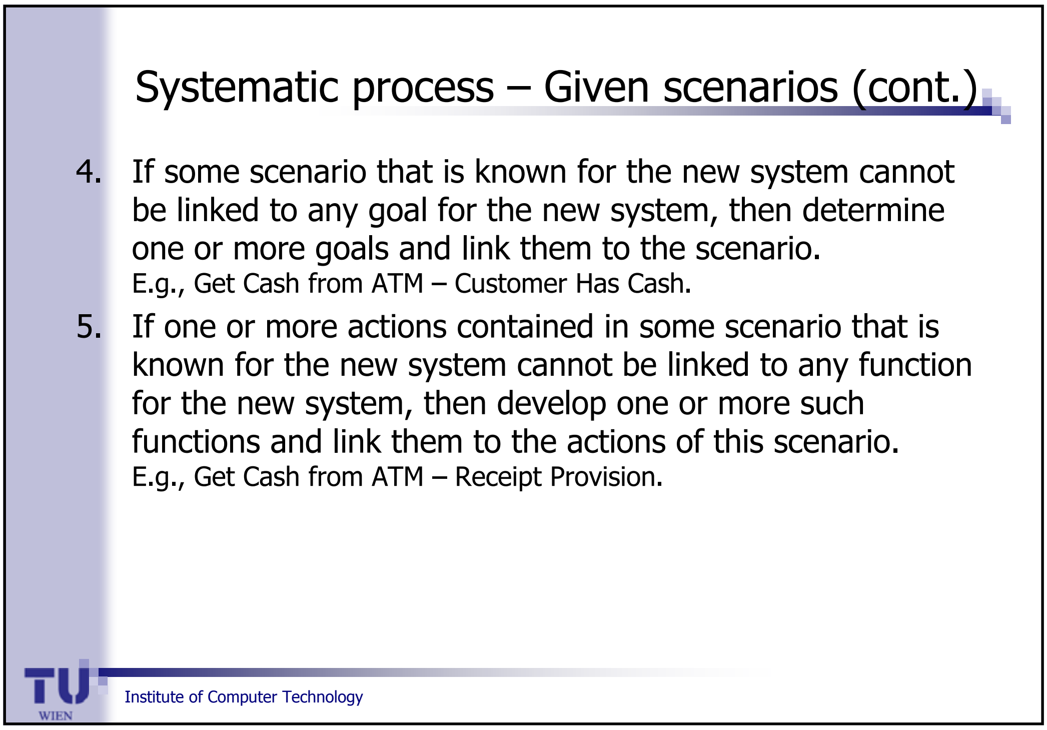
1. If some goal is known from the old system, then figure out whether this is still a goal in the new system that will include the system to be built.
   E.g., Meeting a Friendly Person, Customer Has Cash.
2. If some goal is known for the new system, then try to link it to one or more scenarios for the new system that are already known.
   E.g., Customer Has Receipt – Get Cash from ATM.
3. If some goal that is known for the new system cannot be linked to any scenario for the new system, then develop one or more such scenarios and link them to the goal.
   E.g., Customer Has Cash – Get Cash from ATM.



Systematic process – Given scenarios

1. If some scenario is known from the old system, then determine the goals that are achieved through it.
   E.g., Get Cash from Human Cashier – Customer Has Cash.
2. If some scenario is known from the old system, then try to develop an analogous scenario for the new system.
   E.g., Get Cash from Human Cashier – Get Cash from ATM.
3. If some scenario is known for the new system, then try to link it to one or more goals and, each action contained in it to one or more functions for the new system that are already known.
   E.g., Get Cash from ATM – Customer Has Cash – Cash Provision.

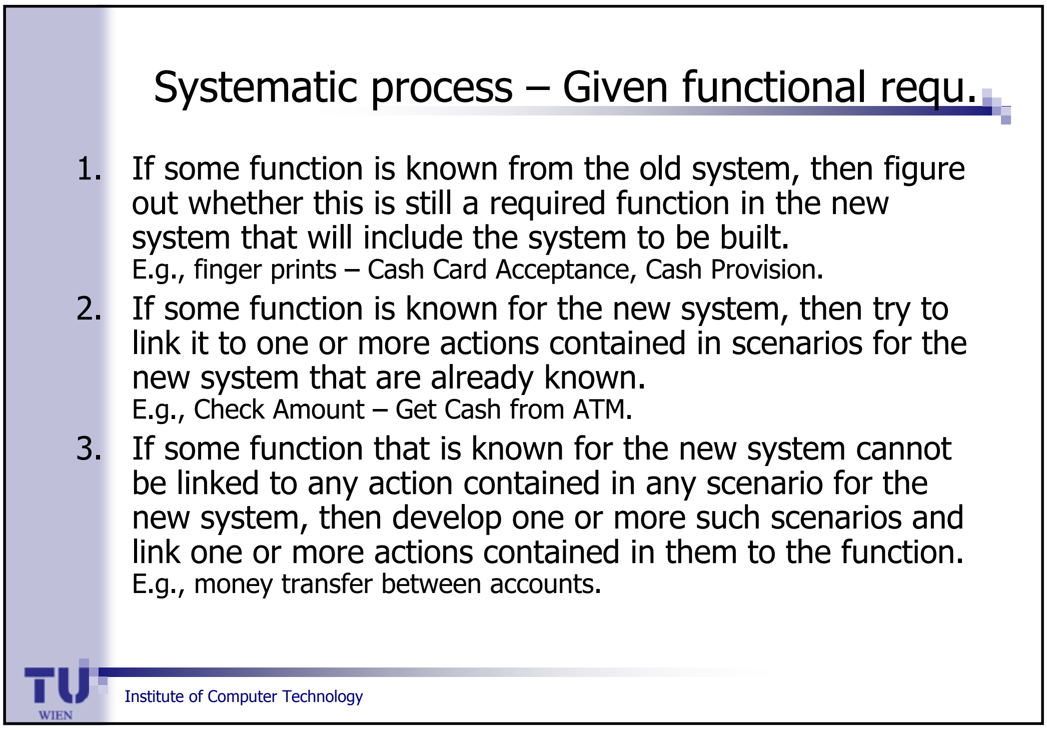
Institute of Computer Technology



Systematic process – Given scenarios (cont.)

4. If some scenario that is known for the new system cannot be linked to any goal for the new system, then determine one or more goals and link them to the scenario.
   E.g., Get Cash from ATM – Customer Has Cash.
5. If one or more actions contained in some scenario that is known for the new system cannot be linked to any function for the new system, then develop one or more such functions and link them to the actions of this scenario.
   E.g., Get Cash from ATM – Receipt Provision.

Institute of Computer Technology

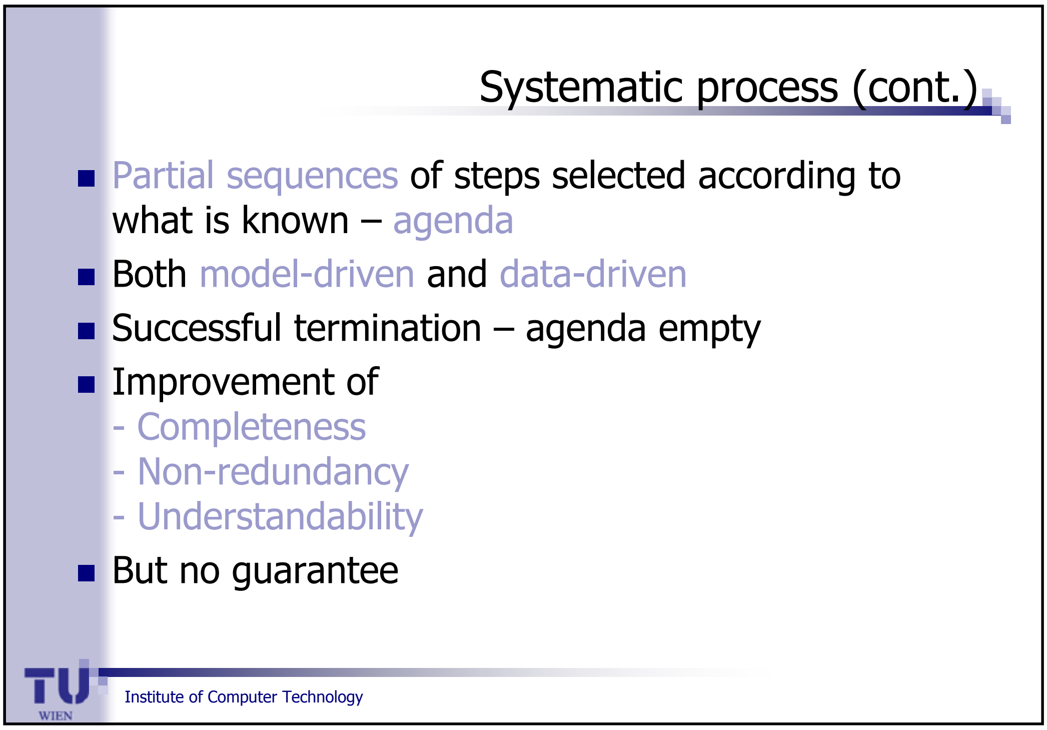


Systematic process – Given functional requ.

1. If some function is known from the old system, then figure out whether this is still a required function in the new system that will include the system to be built.
   E.g., finger prints – Cash Card Acceptance, Cash Provision.
2. If some function is known for the new system, then try to link it to one or more actions contained in scenarios for the new system that are already known.
   E.g., Check Amount – Get Cash from ATM.
3. If some function that is known for the new system cannot be linked to any action contained in any scenario for the new system, then develop one or more such scenarios and link one or more actions contained in them to the function.
   E.g., money transfer between accounts.

TU WIEN

Institute of Computer Technology

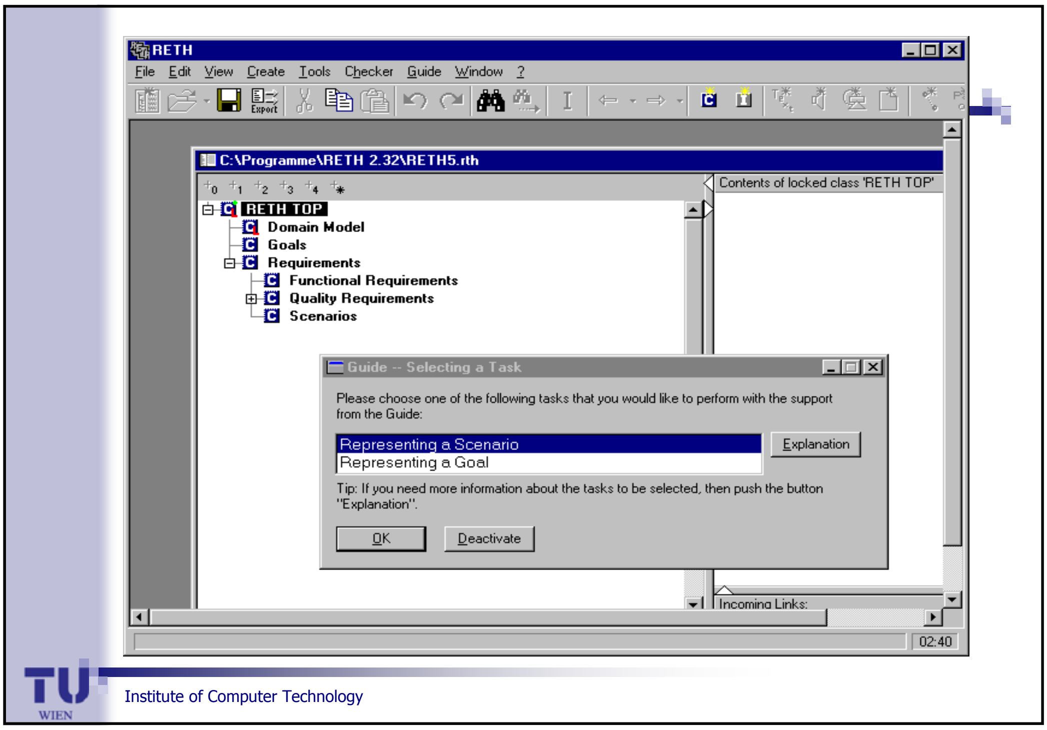


Systematic process (cont.)

- Partial sequences of steps selected according to what is known – agenda
- Both model-driven and data-driven
- Successful termination – agenda empty
- Improvement of
  - Completeness
  - Non-redundancy
  - Understandability
- But no guarantee

TU WIEN

Institute of Computer Technology

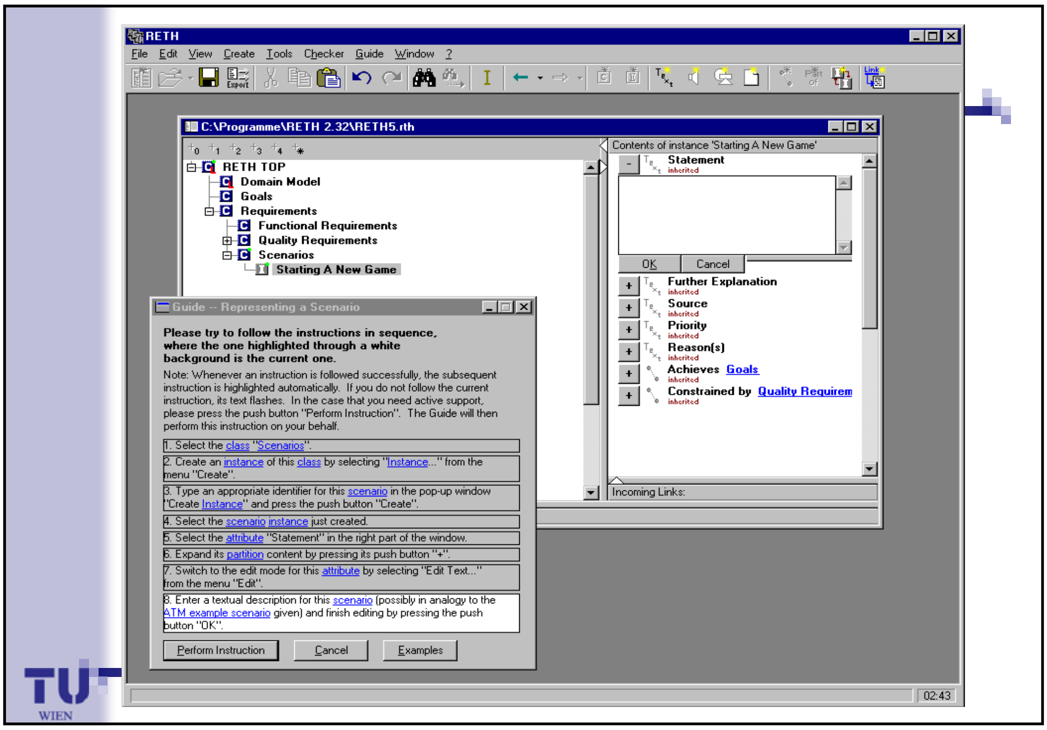

RETH
File Edit View Create Tools Checker Guide Window ?
Export
C:\Programme\RETH 2.32\RETH5.rth
RETH TOP
Domain Model
Goals
Requirements
Functional Requirements
Quality Requirements
Scenarios
Contents of locked class 'RETH TOP'
Incoming Links:
02:40
Guide -- Selecting a Task
Explanation
OK
Deactivate

Please choose one of the following tasks that you would like to perform with the support from the Guide:

Representing a Scenario
Representing a Goal

Tip: If you need more information about the tasks to be selected, then push the button "Explanation".

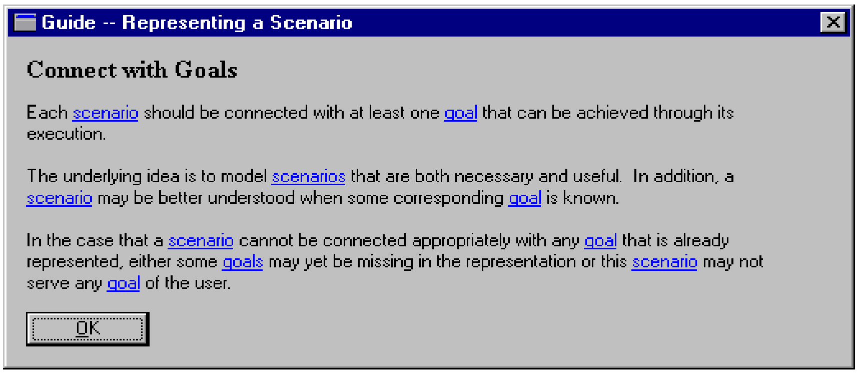
Institute of Computer Technology



Please try to follow the instructions in sequence, where the one highlighted through a white background is the current one.

Note: Whenever an instruction is followed successfully, the subsequent instruction is highlighted automatically. If you do not follow the current instruction, its text flashes. In the case that you need active support, please press the push button "Perform Instruction". The Guide will then perform this instruction on your behalf.

1. Select the class "Scenarios".
2. Create an instance of this class by selecting "Instance..." from the menu "Create".
3. Type an appropriate identifier for this scenario in the pop-up window "Create Instance" and press the push button "Create".
4. Select the scenario instance just created.
5. Select the attribute "Statement" in the right part of the window.
6. Expand its partition content by pressing its push button "+".
7. Switch to the edit mode for this attribute by selecting "Edit Text..." from the menu "Edit".
8. Enter a textual description for this scenario (possibly in analogy to the ATM example scenario given) and finish editing by pressing the push button "OK".



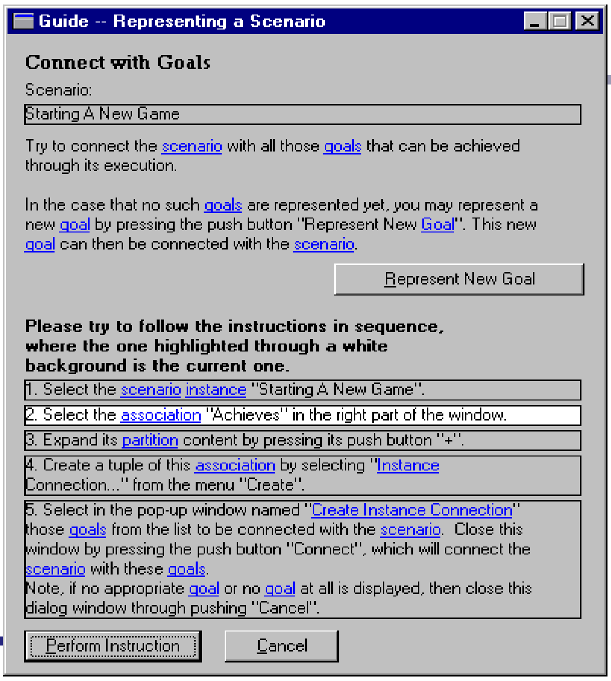
Connect with Goals

Each scenario should be connected with at least one goal that can be achieved through its execution.

The underlying idea is to model scenarios that are both necessary and useful. In addition, a scenario may be better understood when some corresponding goal is known.

In the case that a scenario cannot be connected appropriately with any goal that is already represented, either some goals may yet be missing in the representation or this scenario may not serve any goal of the user.

Institute of Computer Technology



Connect with Goals

Scenario:

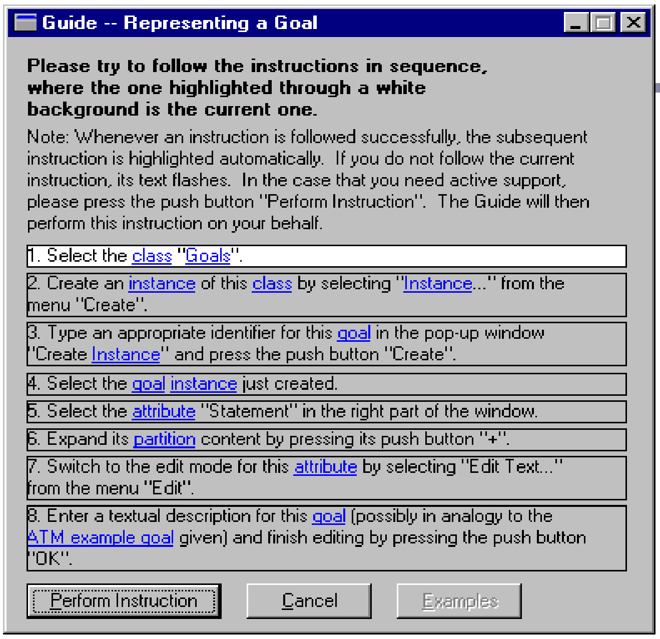
Starting A New Game

Try to connect the scenario with all those goals that can be achieved through its execution.

In the case that no such goals are represented yet, you may represent a new goal by pressing the push button "Represent New Goal". This new goal can then be connected with the scenario.

Please try to follow the instructions in sequence, where the one highlighted through a white background is the current one.

1. Select the scenario instance "Starting A New Game".
2. Select the association "Achieves" in the right part of the window.
3. Expand its partition content by pressing its push button "+".
4. Create a tuple of this association by selecting "Instance Connection..." from the menu "Create".
5. Select in the pop-up window named "Create Instance Connection" those goals from the list to be connected with the scenario. Close this window by pressing the push button "Connect", which will connect the scenario with these goals.
Note, if no appropriate goal or no goal at all is displayed, then close this dialog window through pushing "Cancel".



Guide -- Representing a Goal

Please try to follow the instructions in sequence, where the one highlighted through a white background is the current one.

Note: Whenever an instruction is followed successfully, the subsequent instruction is highlighted automatically. If you do not follow the current instruction, its text flashes. In the case that you need active support, please press the push button "Perform Instruction". The Guide will then perform this instruction on your behalf.

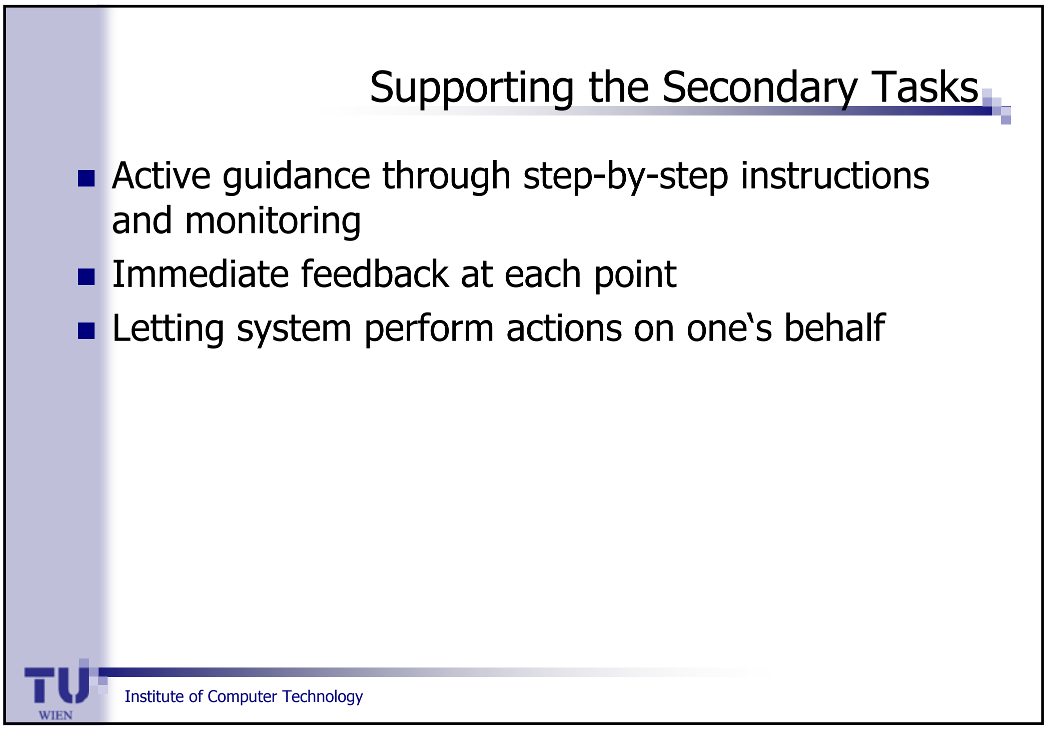
1. Select the class "Goals".
2. Create an instance of this class by selecting "Instance..." from the menu "Create".
3. Type an appropriate identifier for this goal in the pop-up window "Create Instance" and press the push button "Create".
4. Select the goal instance just created.
5. Select the attribute "Statement" in the right part of the window.
6. Expand its partition content by pressing its push button "+".
7. Switch to the edit mode for this attribute by selecting "Edit Text..." from the menu "Edit".
8. Enter a textual description for this goal (possibly in analogy to the ATM example goal given) and finish editing by pressing the push button "OK".

Institute of Computer Technology

Supporting the Primary Tasks

- Presenting choice of conceptual entities to be modeled
- Providing example descriptions of conceptual entities
- Guidance in linking entities according to the method

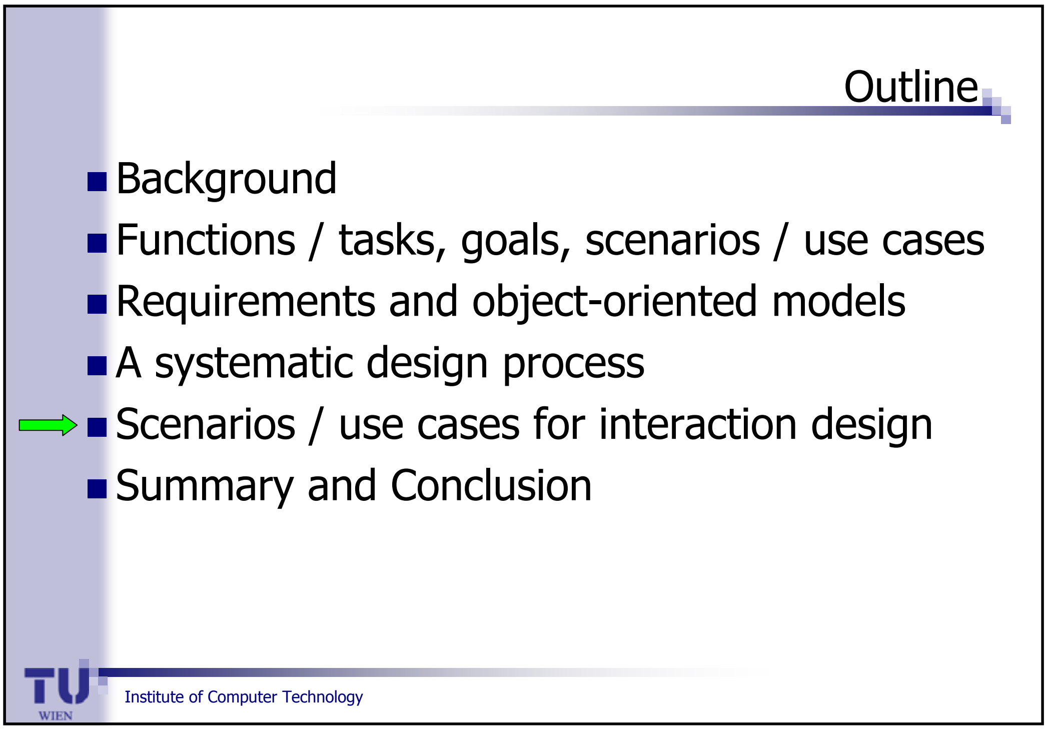
Institute of Computer Technology



Supporting the Secondary Tasks

- Active guidance through step-by-step instructions and monitoring
- Immediate feedback at each point
- Letting system perform actions on one‘s behalf

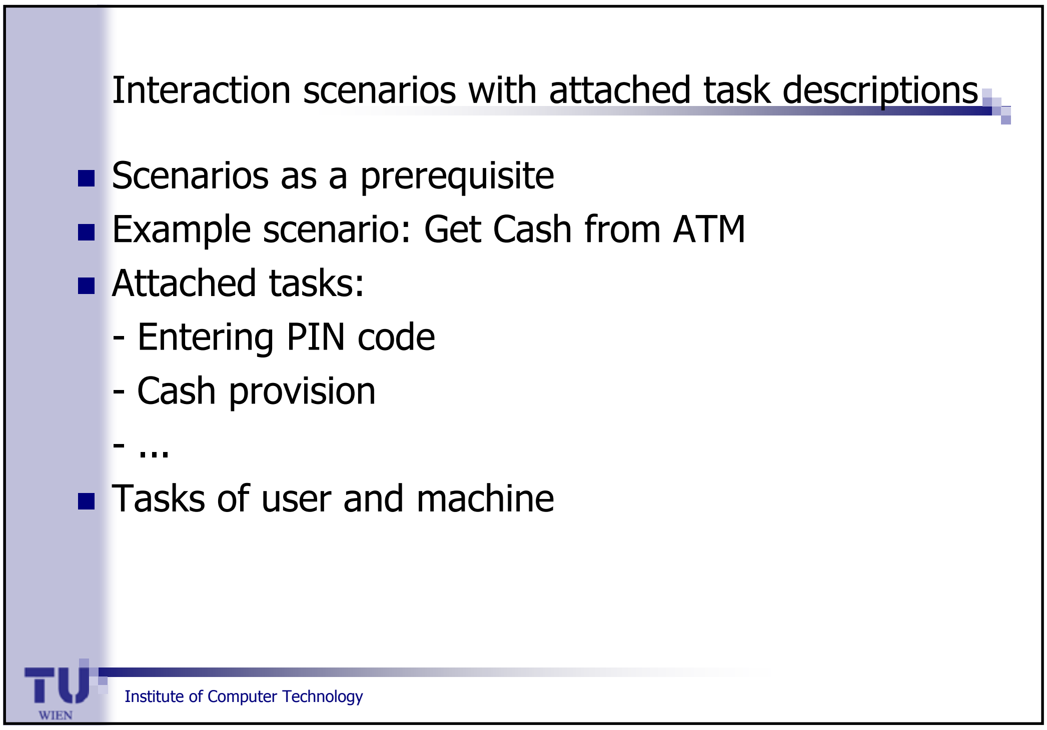
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- Requirements and object-oriented models
- A systematic design process
- ➡ Scenarios / use cases for interaction design
- Summary and Conclusion

Institute of Computer Technology



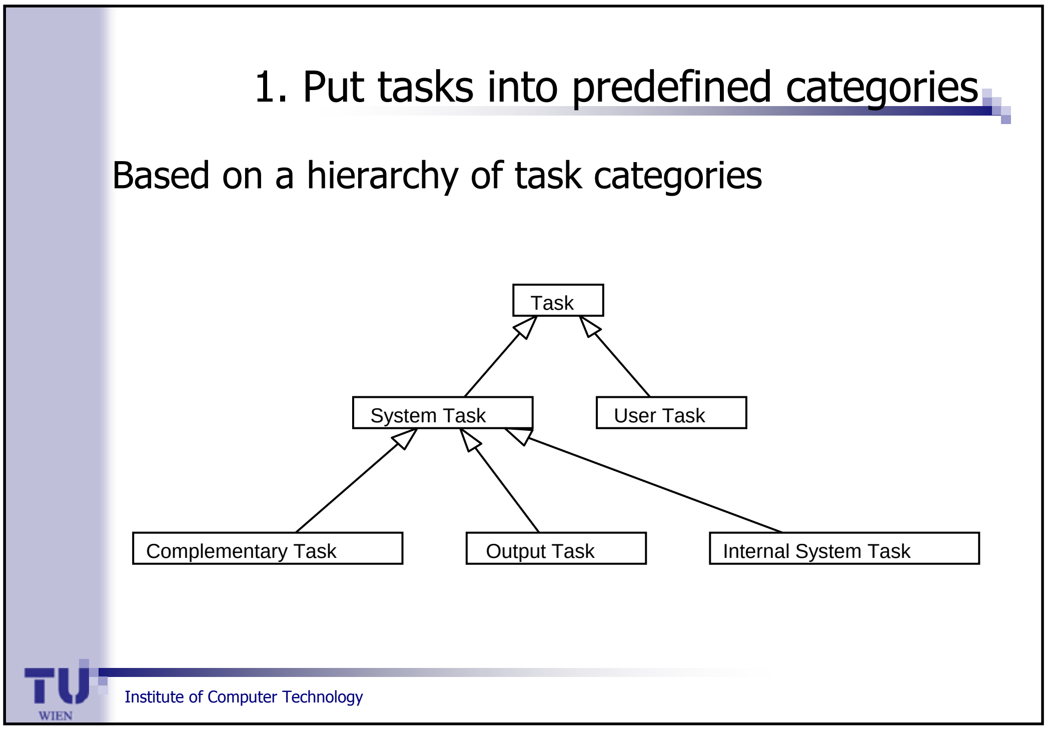
Interaction scenarios with attached task descriptions

- Scenarios as a prerequisite
- Example scenario: Get Cash from ATM
- Attached tasks:

  - Entering PIN code

  - Cash provision

  - ...
- Tasks of user and machine



1. Put tasks into predefined categories

Based on a hierarchy of task categories

Task

System Task

User Task

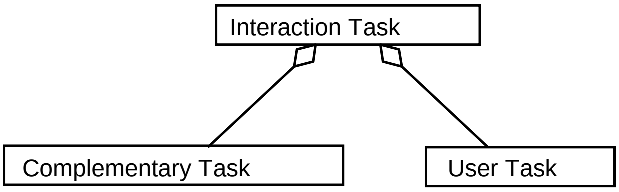
Complementary Task

Output Task

Internal System Task

2. Aggregate complementary tasks

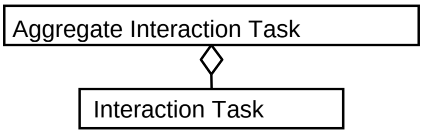
- Task of user and complementary task of machine
- Together make up an interaction through the UI

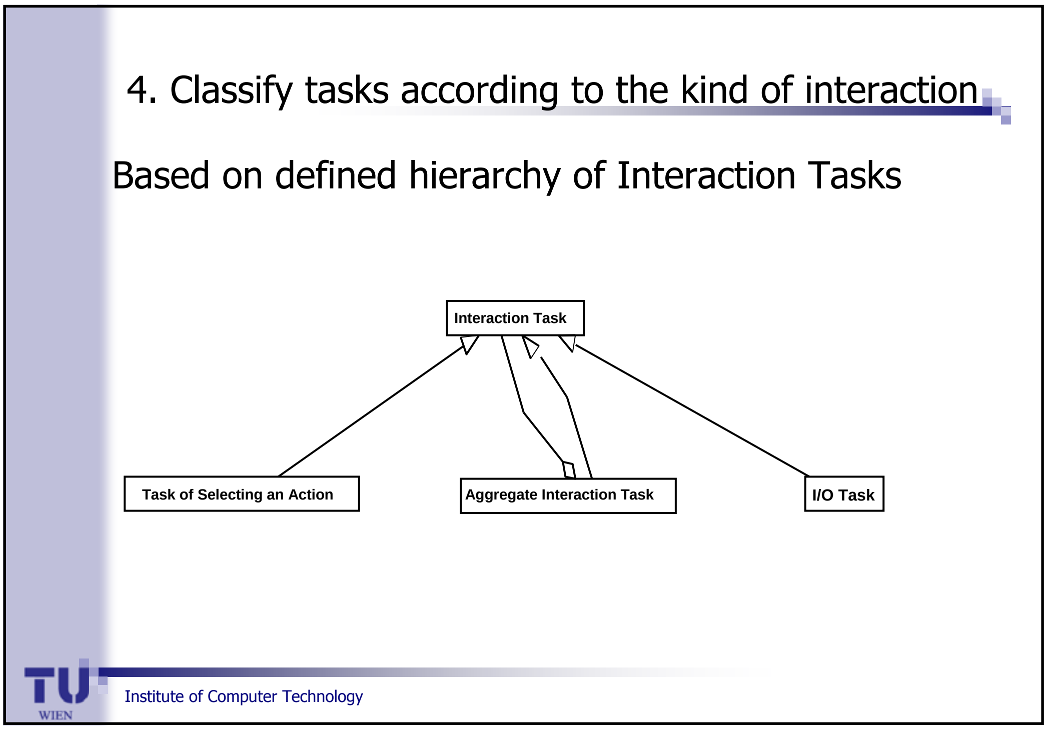


3. Compose and decompose tasks

Find right granularity

- aggregate
- decompose





4. Classify tasks according to the kind of interaction

Based on defined hierarchy of Interaction Tasks

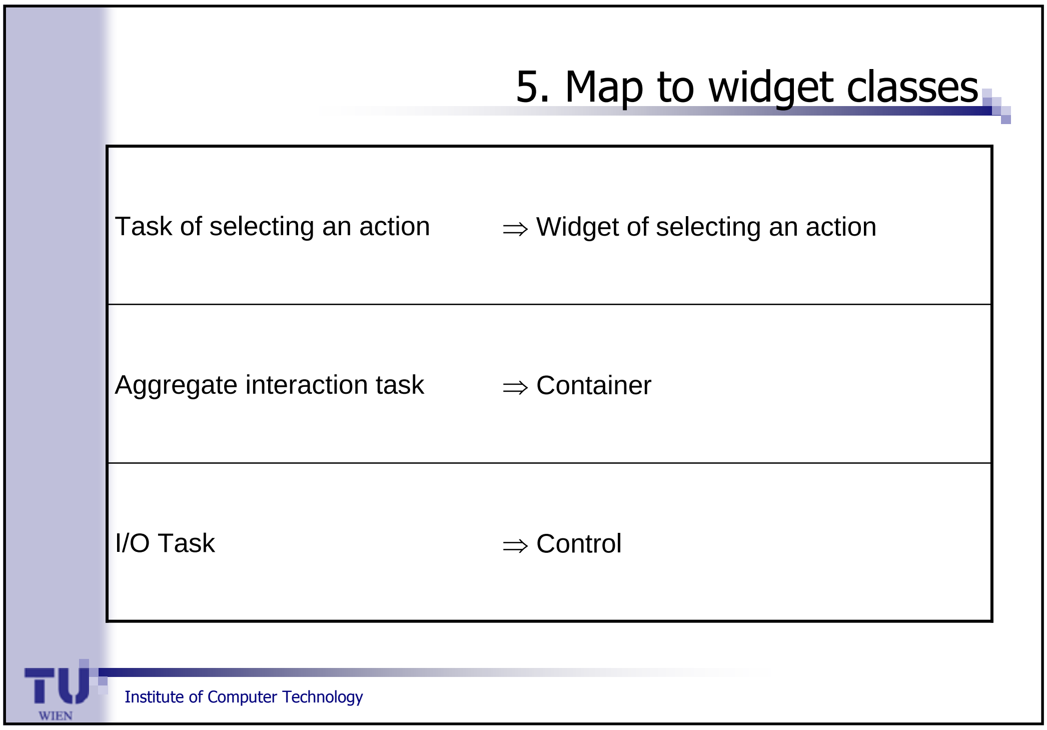
Interaction Task

Task of Selecting an Action

Aggregate Interaction Task

I/O Task

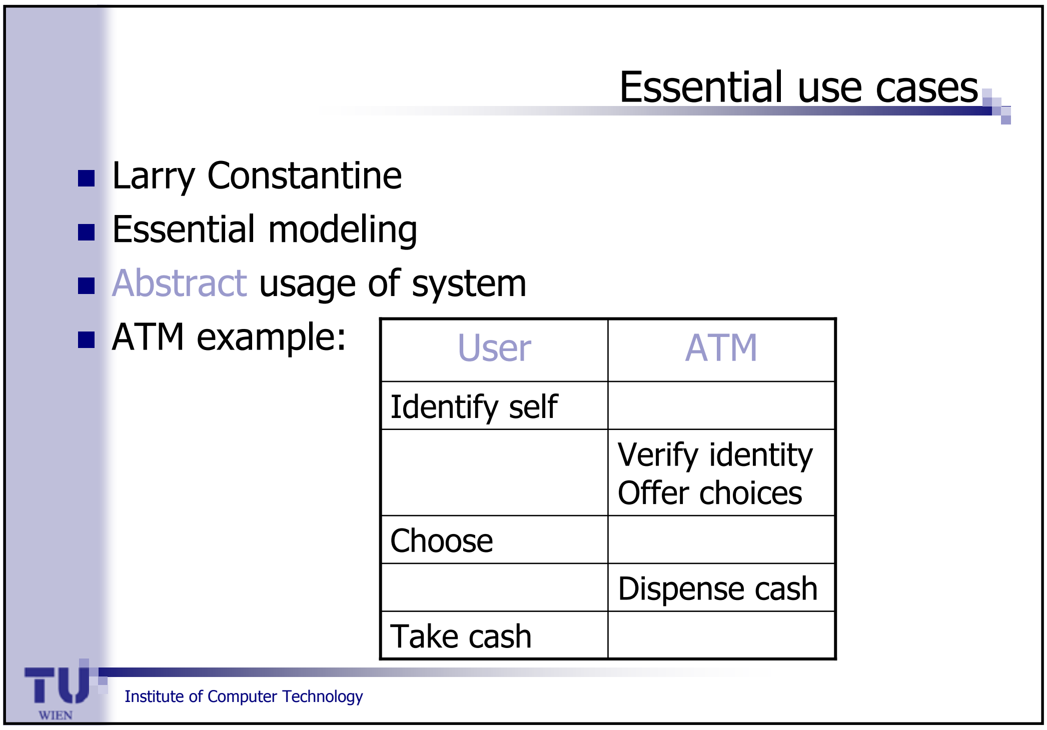
Institute of Computer Technology



5. Map to widget classes

Task of selecting an action	⇒ Widget of selecting an action
Aggregate interaction task	⇒ Container
I/O Task	⇒ Control

Institute of Computer Technology

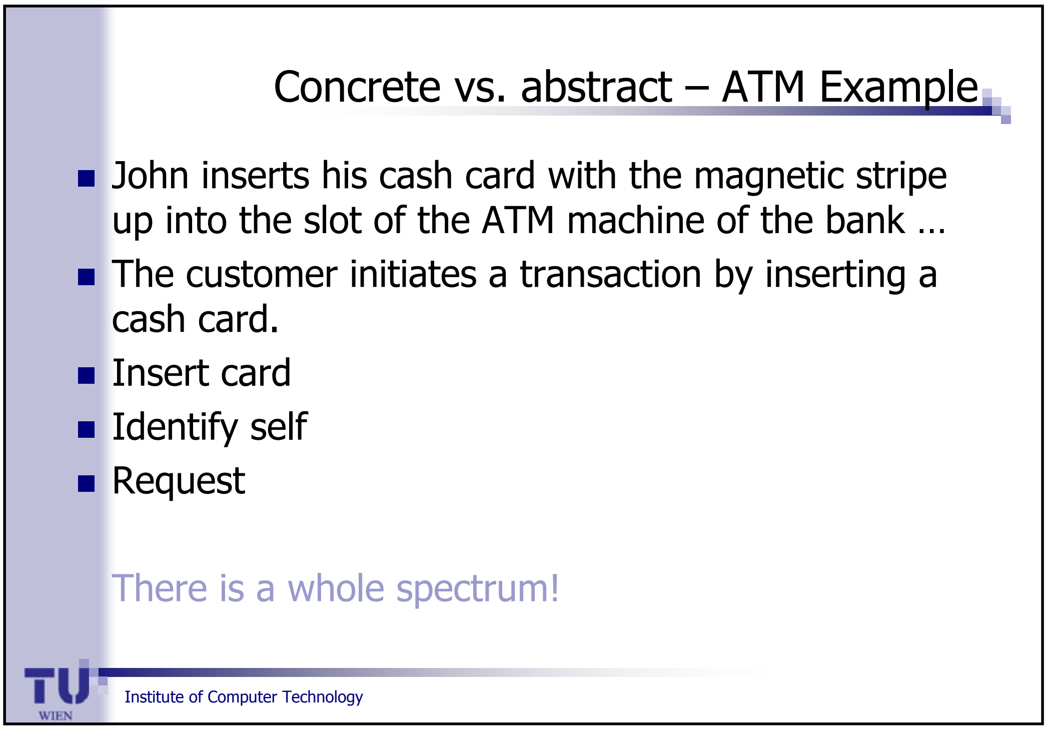


Essential use cases

- Larry Constantine
- Essential modeling
- Abstract usage of system
- ATM example:

User	ATM
Identify self	
	Verify identity Offer choices
Choose	
	Dispense cash
Take cash	

Institute of Computer Technology

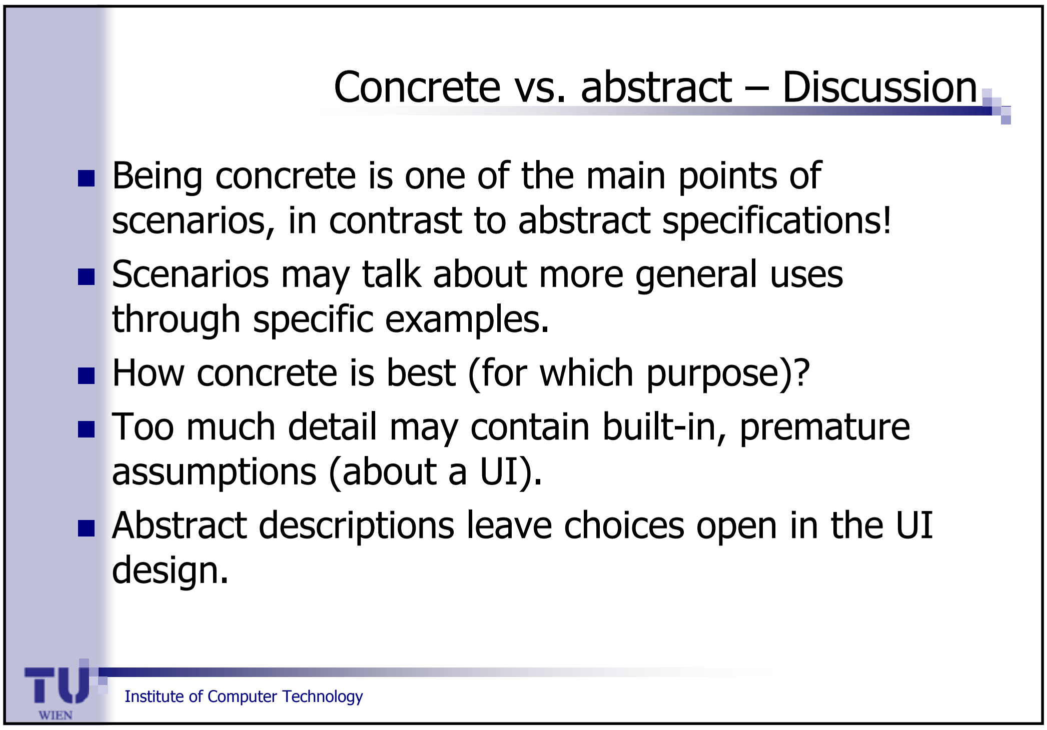


Concrete vs. abstract – ATM Example

- John inserts his cash card with the magnetic stripe up into the slot of the ATM machine of the bank …
- The customer initiates a transaction by inserting a cash card.
- Insert card
- Identify self
- Request

There is a whole spectrum!

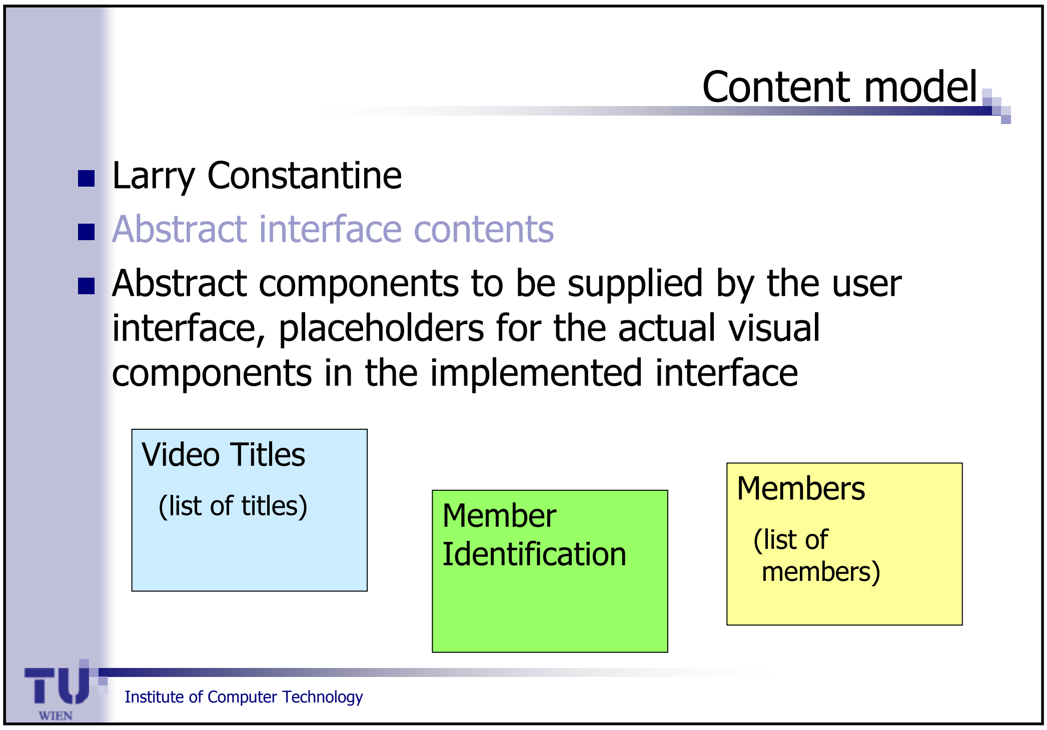
Institute of Computer Technology



Concrete vs. abstract – Discussion

- Being concrete is one of the main points of scenarios, in contrast to abstract specifications!
- Scenarios may talk about more general uses through specific examples.
- How concrete is best (for which purpose)?
- Too much detail may contain built-in, premature assumptions (about a UI).
- Abstract descriptions leave choices open in the UI design.

Institute of Computer Technology



Content model

- Larry Constantine
- Abstract interface contents
- Abstract components to be supplied by the user interface, placeholders for the actual visual components in the implemented interface

Video Titles

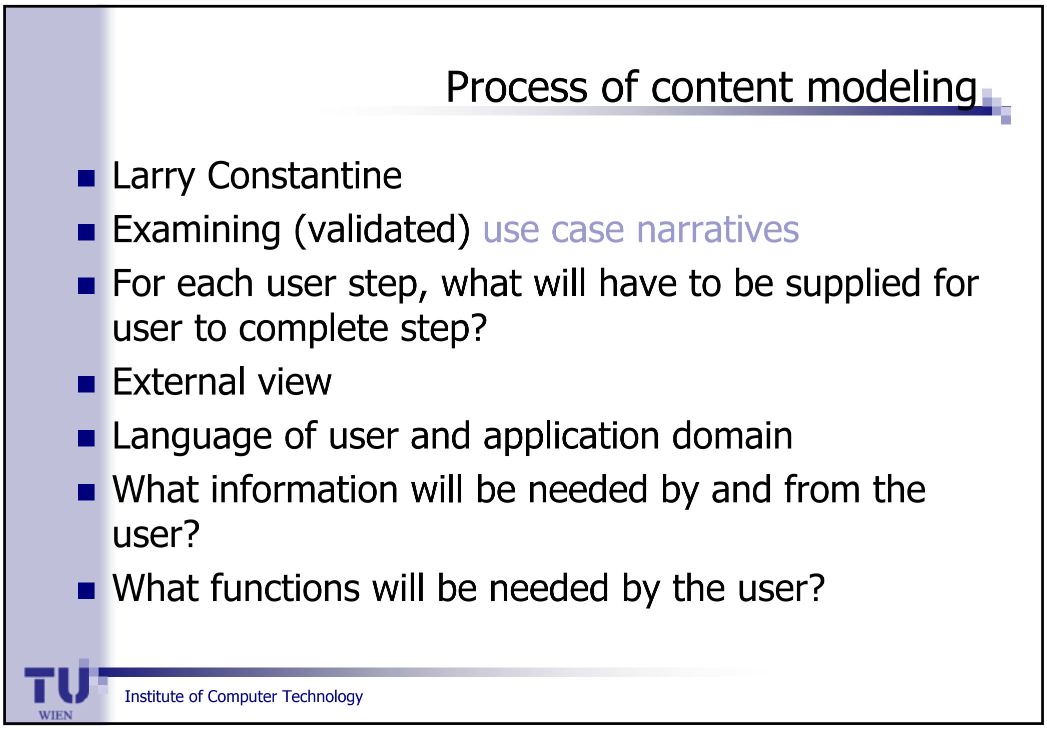
(list of titles)

Member Identification

Members

(list of members)

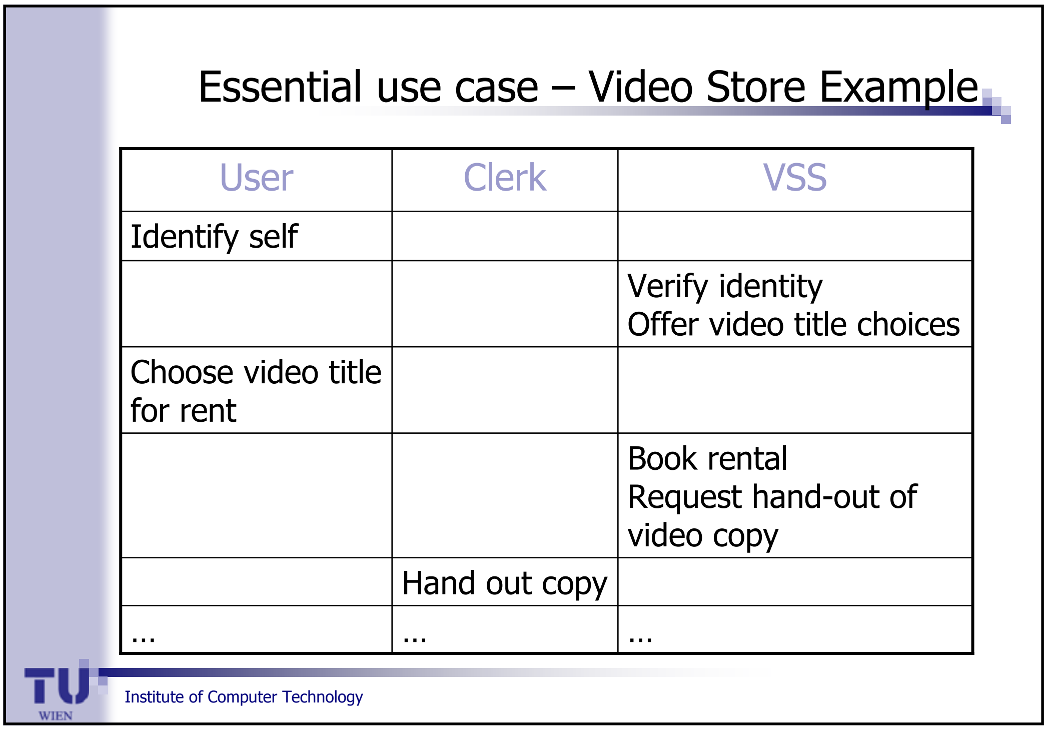
Institute of Computer Technology



Process of content modeling

- Larry Constantine
- Examining (validated) use case narratives
- For each user step, what will have to be supplied for user to complete step?
- External view
- Language of user and application domain
- What information will be needed by and from the user?
- What functions will be needed by the user?

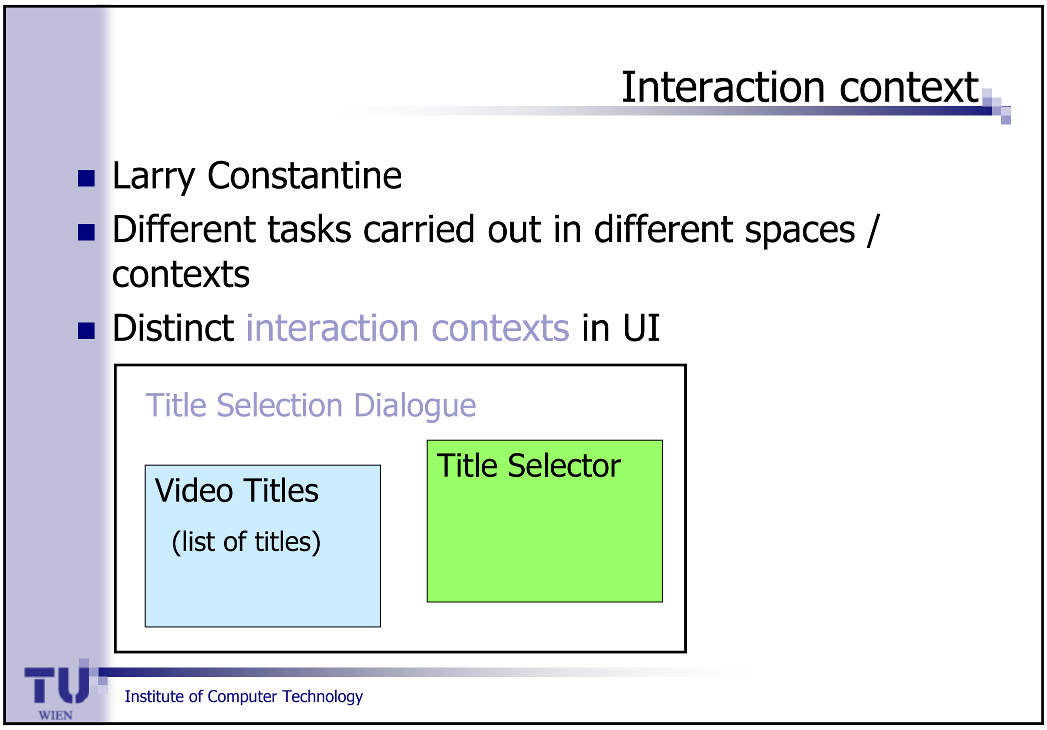
Institute of Computer Technology



Essential use case – Video Store Example

User	Clerk	VSS
Identify self		
		Verify identity Offer video title choices
Choose video title for rent		
		Book rental Request hand-out of video copy
	Hand out copy	
…	…	…

Institute of Computer Technology



Interaction context

- Larry Constantine
- Different tasks carried out in different spaces / contexts
- Distinct interaction contexts in UI

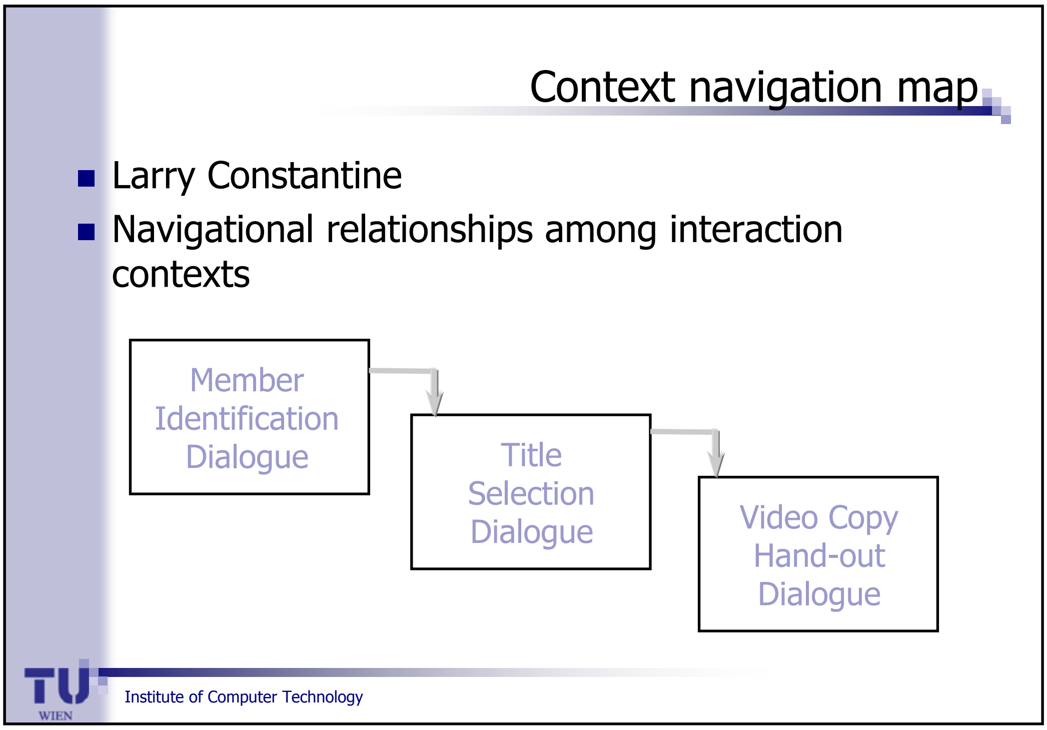
Title Selection Dialogue

Video Titles

(list of titles)

Title Selector

Institute of Computer Technology



Context navigation map

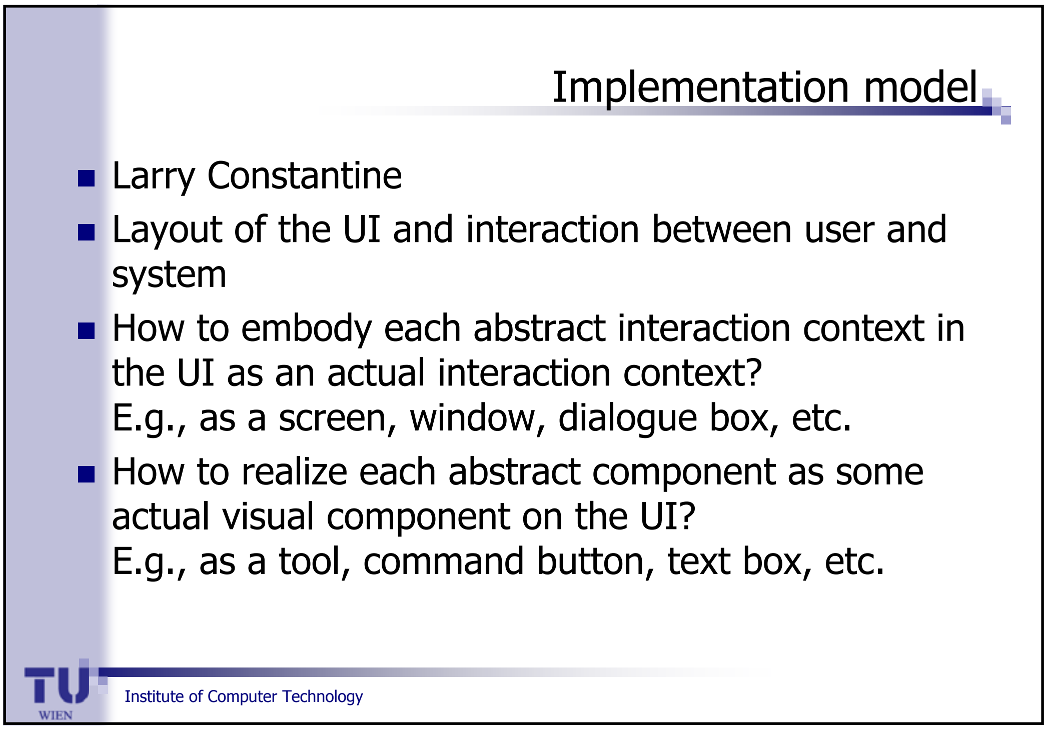
- Larry Constantine
- Navigational relationships among interaction contexts

Member Identification Dialogue

Title Selection Dialogue

Video Copy Hand-out Dialogue

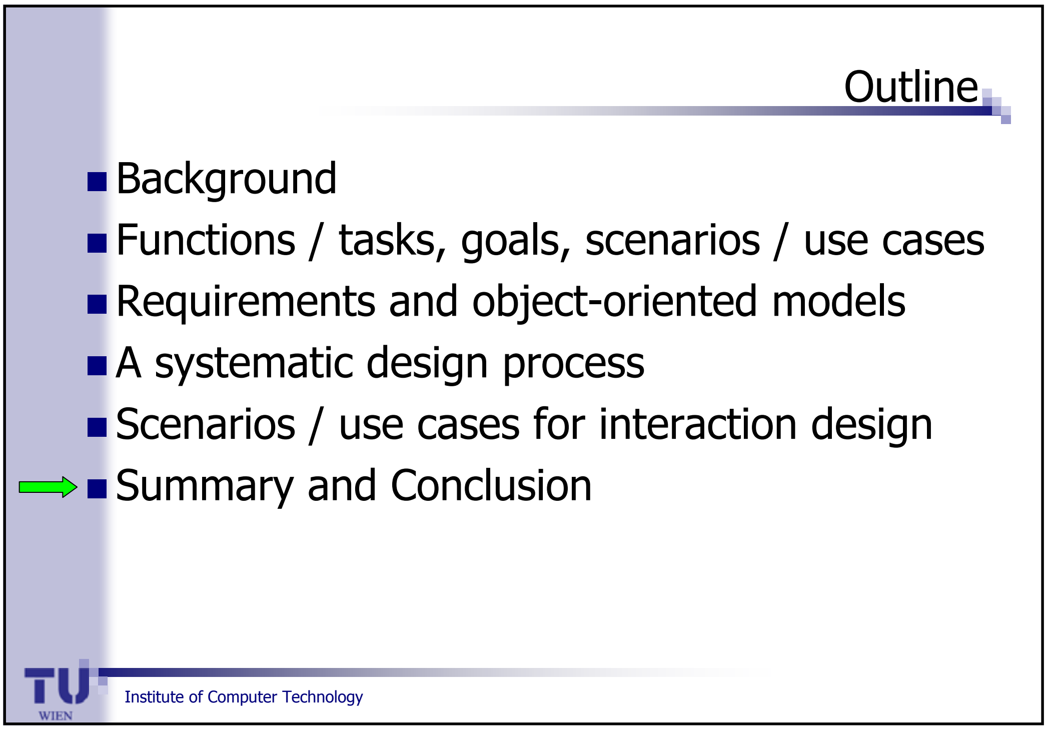
Institute of Computer Technology



Implementation model

- Larry Constantine
- Layout of the UI and interaction between user and system
- How to embody each abstract interaction context in the UI as an actual interaction context?
  E.g., as a screen, window, dialogue box, etc.
- How to realize each abstract component as some actual visual component on the UI?
  E.g., as a tool, command button, text box, etc.

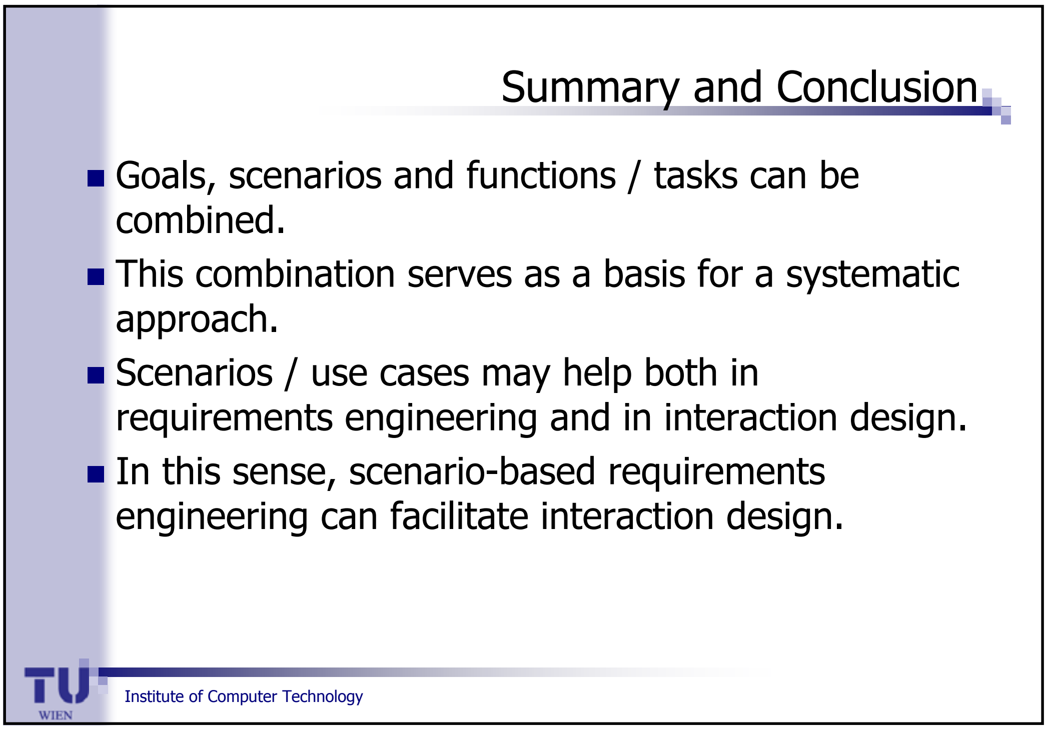
Institute of Computer Technology



Outline

- Background
- Functions / tasks, goals, scenarios / use cases
- Requirements and object-oriented models
- A systematic design process
- Scenarios / use cases for interaction design
- ➡ Summary and Conclusion

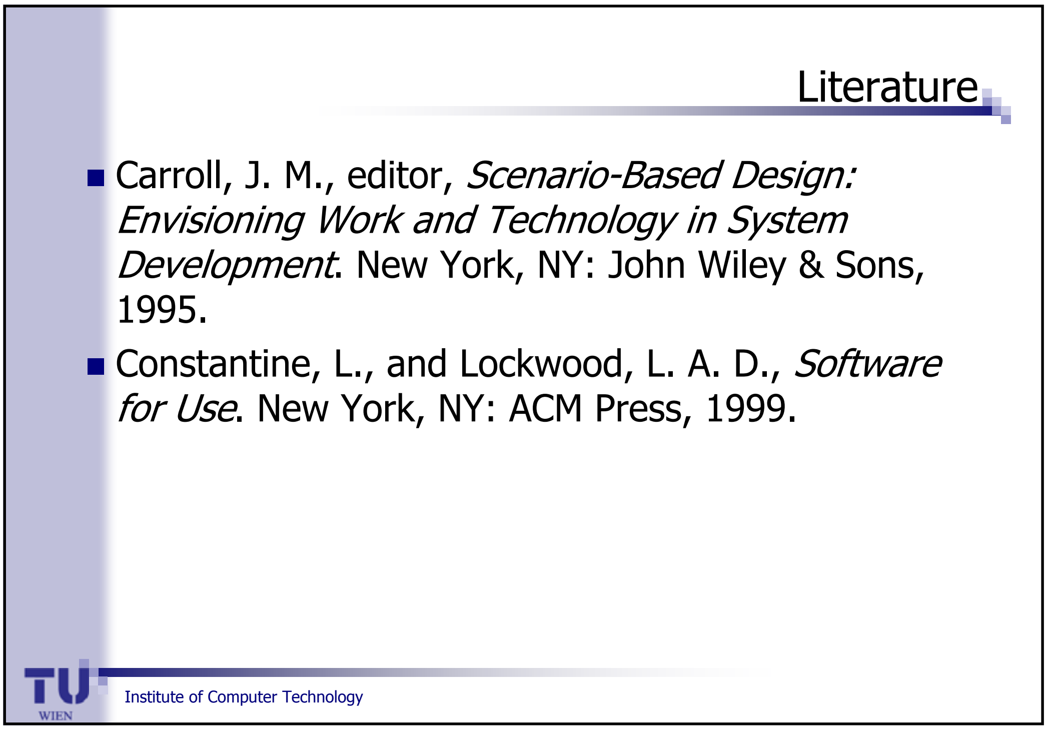
Institute of Computer Technology



Summary and Conclusion

- Goals, scenarios and functions / tasks can be combined.
- This combination serves as a basis for a systematic approach.
- Scenarios / use cases may help both in requirements engineering and in interaction design.
- In this sense, scenario-based requirements engineering can facilitate interaction design.

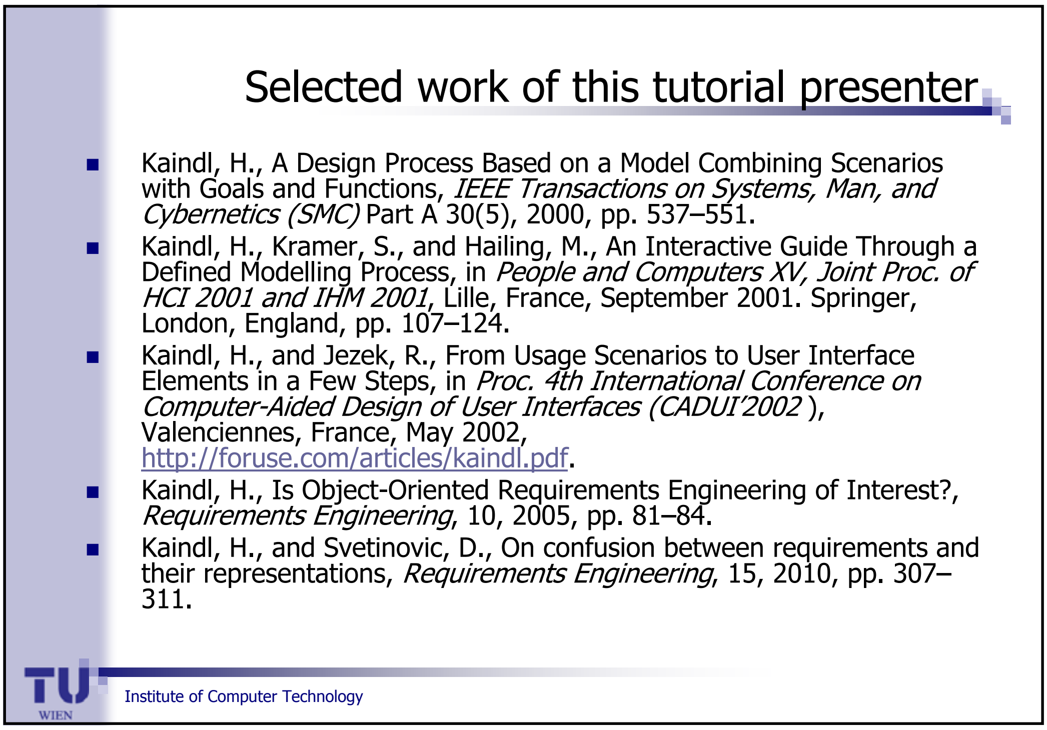
Institute of Computer Technology



Literature

- Carroll, J. M., editor, *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York, NY: John Wiley & Sons, 1995.
- Constantine, L., and Lockwood, L. A. D., *Software for Use*. New York, NY: ACM Press, 1999.

Institute of Computer Technology



Selected work of this tutorial presenter

- Kaindl, H., A Design Process Based on a Model Combining Scenarios with Goals and Functions, *IEEE Transactions on Systems, Man, and Cybernetics (SMC)* Part A 30(5), 2000, pp. 537–551.
- Kaindl, H., Kramer, S., and Hailing, M., An Interactive Guide Through a Defined Modelling Process, in *People and Computers XV, Joint Proc. of HCI 2001 and IHM 2001*, Lille, France, September 2001. Springer, London, England, pp. 107–124.
- Kaindl, H., and Jezek, R., From Usage Scenarios to User Interface Elements in a Few Steps, in *Proc. 4th International Conference on Computer-Aided Design of User Interfaces (CADUI'2002*), Valenciennes, France, May 2002, http://foruse.com/articles/kaindl.pdf.
- Kaindl, H., Is Object-Oriented Requirements Engineering of Interest?, *Requirements Engineering*, 10, 2005, pp. 81–84.
- Kaindl, H., and Svetinovic, D., On confusion between requirements and their representations, *Requirements Engineering*, 15, 2010, pp. 307–311.

Institute of Computer Technology