2IW80 Software specification and architecture

Use cases

Alexander Serebrenik

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Before we start…

Requirement for the art-gallery website:

Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.

http://www.ics.uci.edu/~taylor/ICS_52_FQ02/ics52_Fall02_Req_Ver3.doc

This is an example of

a. Functional requirement
b. Non-functional requirement
c. What was this all about?

Before we start…

Requirement for the art-gallery website:

Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.

http://www.ics.uci.edu/~taylor/ICS_52_FQ02/ics52_Fall02_Req_Ver3.doc

This is an example of

a. Functional requirement ✓
b. Non-functional requirement
c. What was this all about?

SMART requirements

Requirements used as a specification technique

- To be useful as a specification technique, requirements should be
  - **S**pecific
  - **M**easurable
  - **A**ttainable
  - **R**ealisable
  - **T**raceable

SMART

Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.

Mike Mannion and Barry Keepence. 1995. SMART requirements. SIGSOFT Softw. Eng. Notes 20, 2 (April 1995), 42-47.

Is this requirement SMART?

a. Yes

b. No

SMART requirements

Requirements used as a specification technique

- To be useful as a specification technique, requirements should be
  - Specific
  - Measurable
  - Attainable
  - Realisable
  - Traceable

SMART

Every painting always belongs to only one "Theme" for example such as modern art or nature, etc.

Mike Mannion and Barry Keepence. 1995. SMART requirements. SIGSOFT Softw. Eng. Notes 20, 2 (April 1995), 42-47.

Is this requirement SMART?

a. Yes

b. No ✓

Requirements documents in practice

SEVENTH FRAMEWORK PROGRAMME

INFORMATION AND COMMUNICATION TECHNOLOGIES

Project:
Accessibility Assessment Simulation Environment for New Applications Design and Development
(ACCESSIBLE, Grant Agreement No. 224145)

accessible
Applications Design and Development

Deliverable number and title:

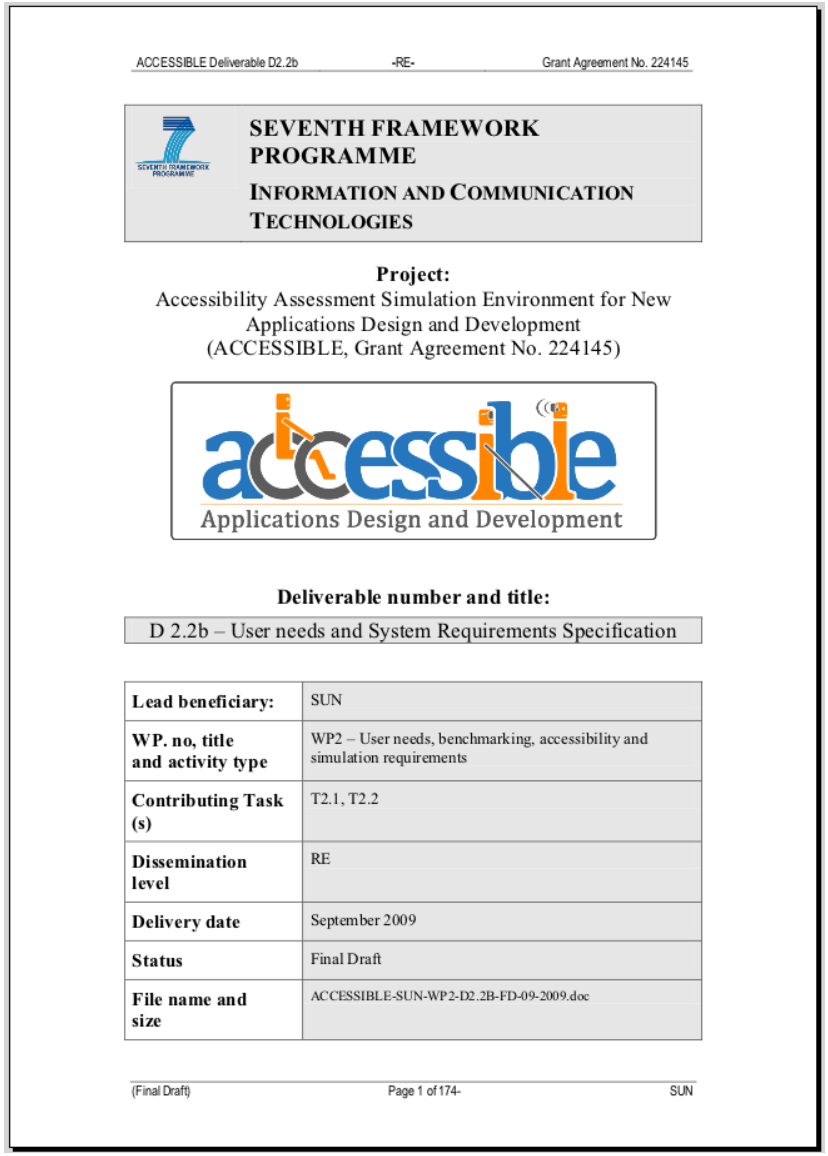
D 2.2b – User needs and System Requirements Specification

Lead beneficiary:	SUN
WP. no, title and activity type	WP2 – User needs, benchmarking, accessibility and simulation requirements
Contributing Task (s)	T2.1, T2.2
Dissemination level	RE
Delivery date	September 2009
Status	Final Draft
File name and size	ACCESSIBLE-SUN-WP2-D2.2B-FD-09-2009.doc

Page 1 of 174-

Technische Universiteit Eindhoven University of Technology

Requirements documents



ACCESSIBLE Deliverable D2.2b -RE- Grant Agreement No. 224145

SEVENTH FRAMEWORK PROGRAMME

INFORMATION AND COMMUNICATION TECHNOLOGIES

Project:
Accessibility Assessment Simulation Environment for New Applications Design and Development
(ACCESSIBLE, Grant Agreement No. 224145)

accessible
Applications Design and Development

Deliverable number and title:

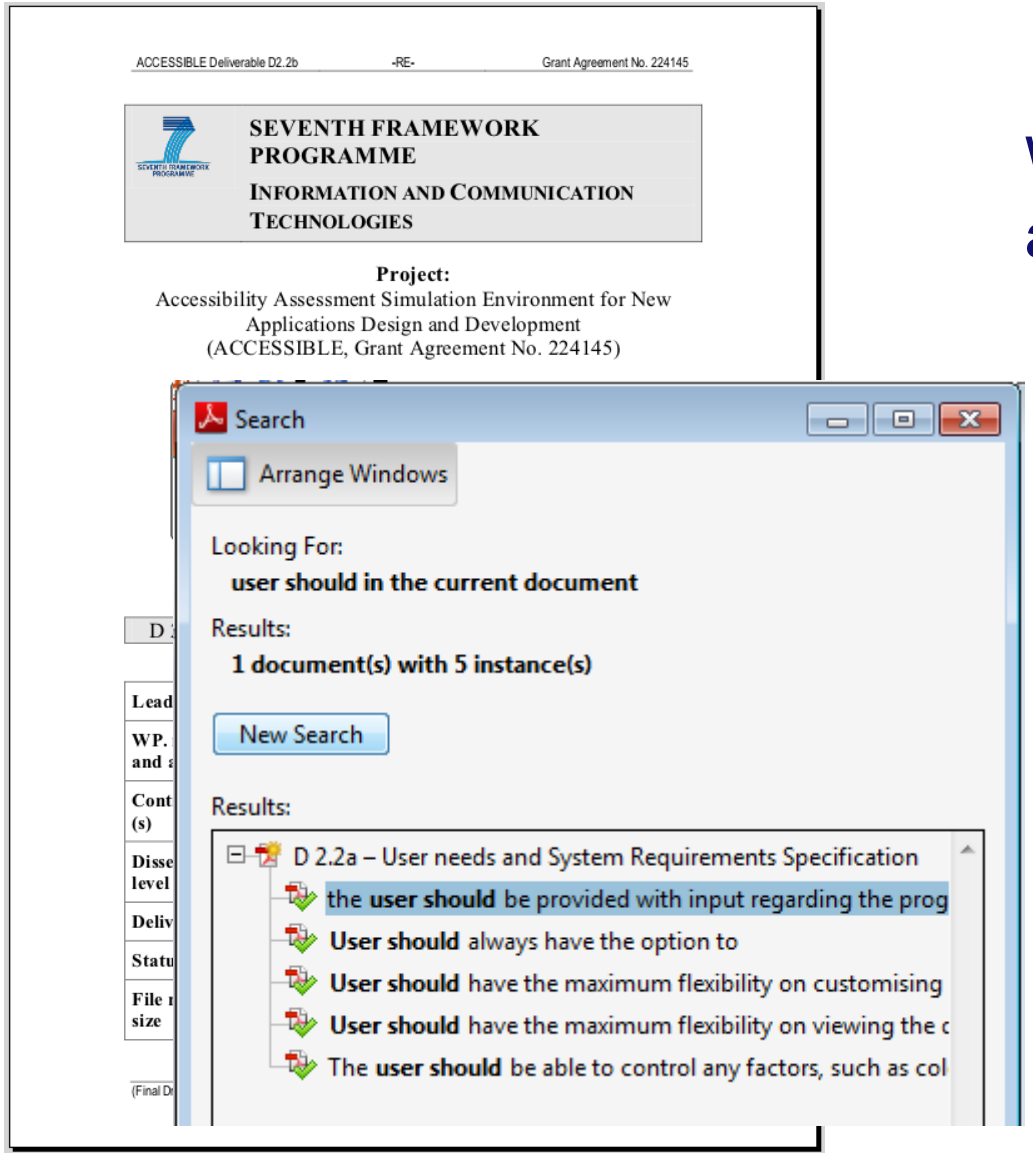
D 2.2b – User needs and System Requirements Specification

Lead beneficiary:	SUN
WP. no, title and activity type	WP2 – User needs, benchmarking, accessibility and simulation requirements
Contributing Task (s)	T2.1, T2.2
Dissemination level	RE
Delivery date	September 2009
Status	Final Draft
File name and size	ACCESSIBLE-SUN-WP2-D2.2B-FD-09-2009.doc

(Final Draft) Page 1 of 174- SUN

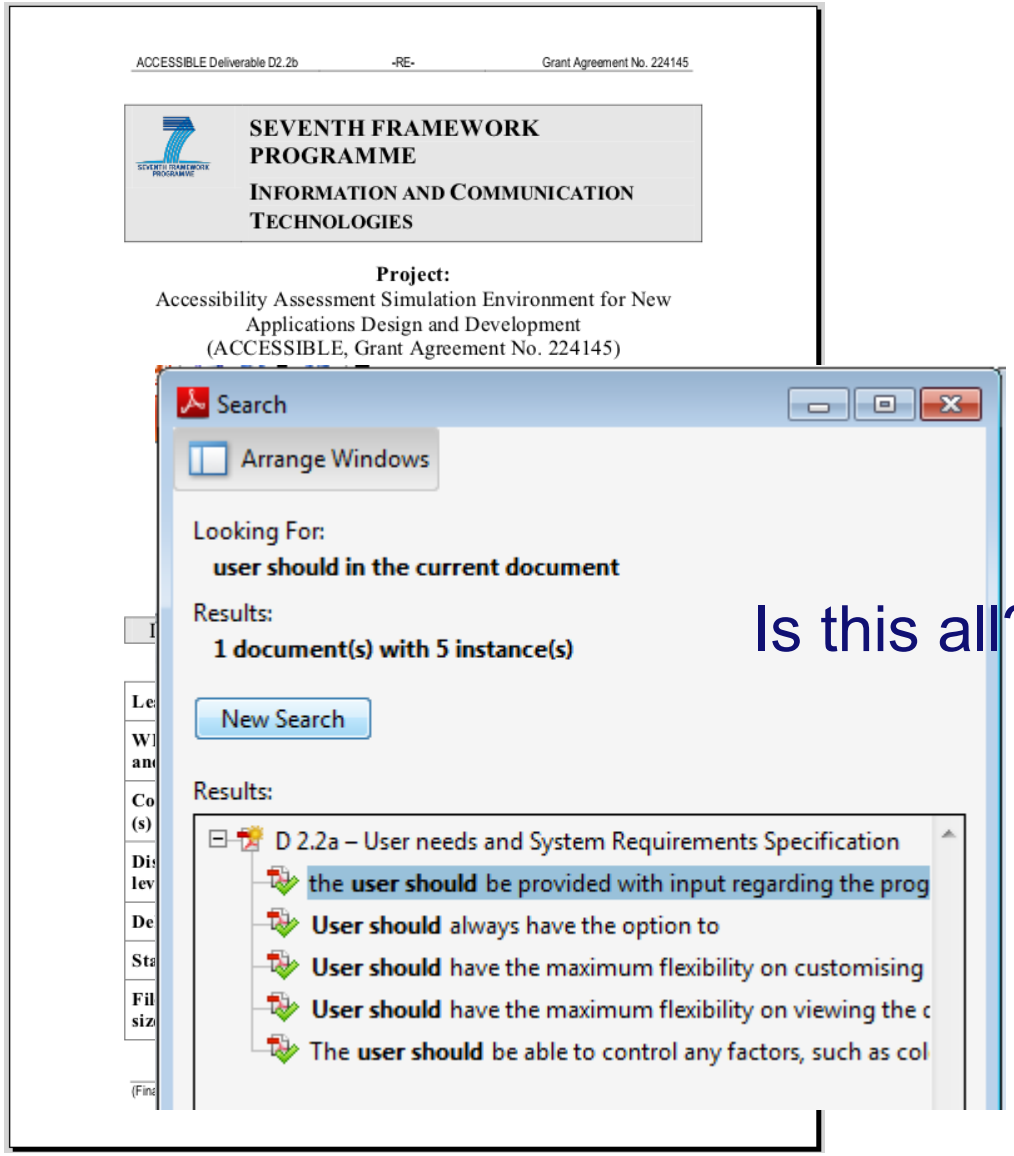
What should a user be able to do?

Requirements documents



What should a user be able to do?

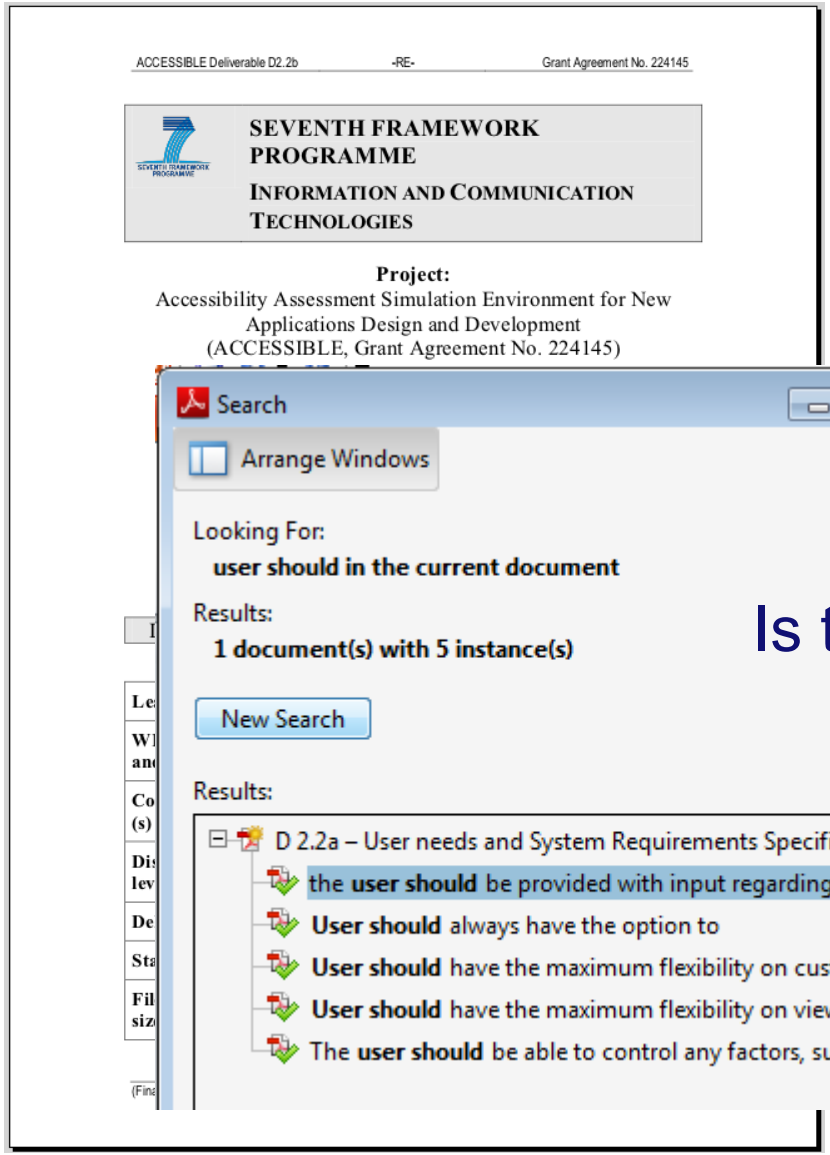
Requirements documents



What should a user be able to do?

Is this all?

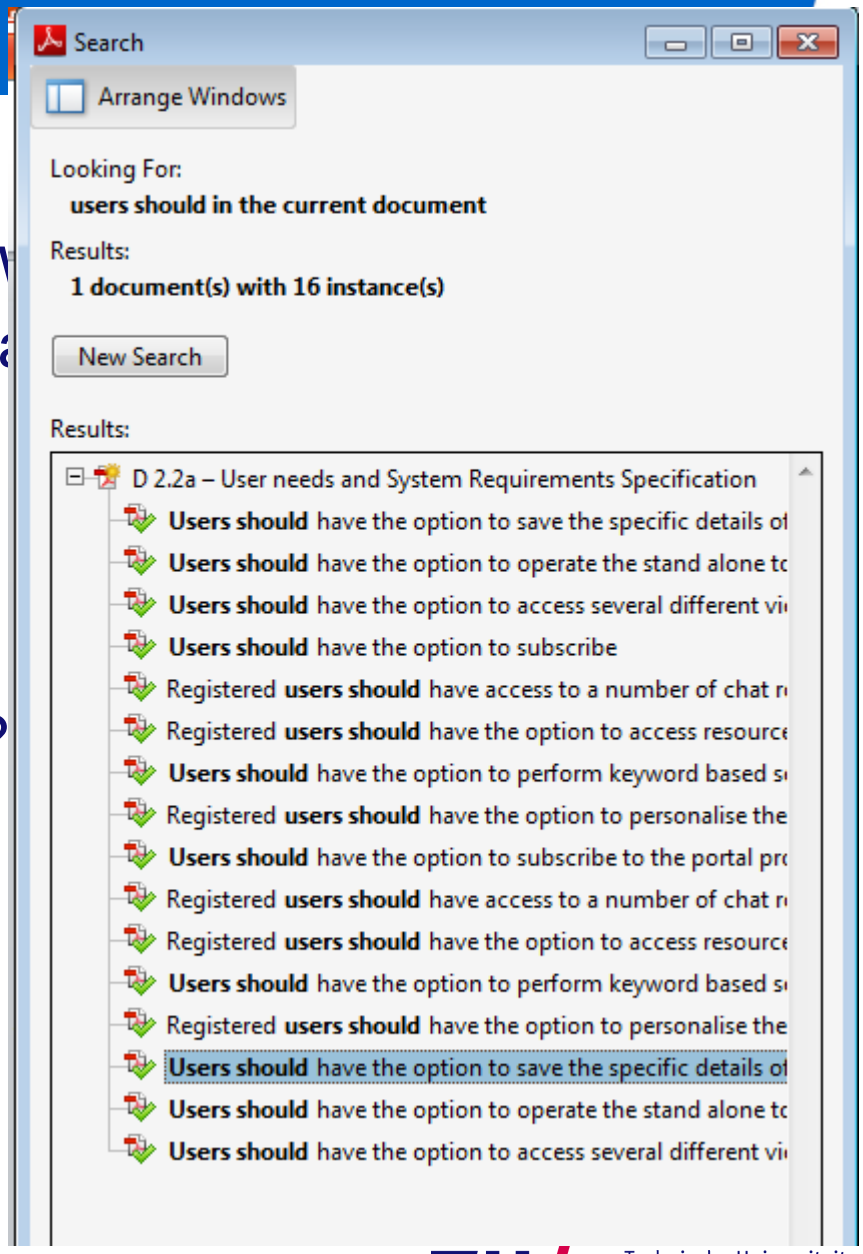
Requirements documents


ACCESSIBLE Deliverable D2.2b -RE- Grant Agreement No. 224145

SEVENTH FRAMEWORK PROGRAMME
INFORMATION AND COMMUNICATION TECHNOLOGIES

Project:
Accessibility Assessment Simulation Environment for New Applications Design and Development
(ACCESSIBLE, Grant Agreement No. 224145)

Is this all?



Technische Universiteit Eindhoven University of Technology

Requirements documents

ACCESSIBLE Deliverable D2.2b -RE- Grant Agreement No. 224145

SEVENTH FRAMEWORK PROGRAMME
INFORMATION AND COMMUNICATION TECHNOLOGIES

Project:
Accessibility Assessment Simulation Environment for New Applications Design and Development
(ACCESSIBLE, Grant Agreement No. 224145)

Search

Arrange Windows

Looking For:
user should in the current document

Results:
1 document(s) with 5 instance(s)

New Search

Results:

D 2.2a – User needs and System Requirements Specification
- the **user should** be provided with input regarding the prog
- **User should** always have the option to
- **User should** have the maximum flexibility on customising
- **User should** have the maximum flexibility on viewing the c
- The **user should** be able to control any factors, such as col

Is this all?

Search

Arrange Windows

Looking For:
users should in the current document

Results:
1 document(s) with 16 instance(s)

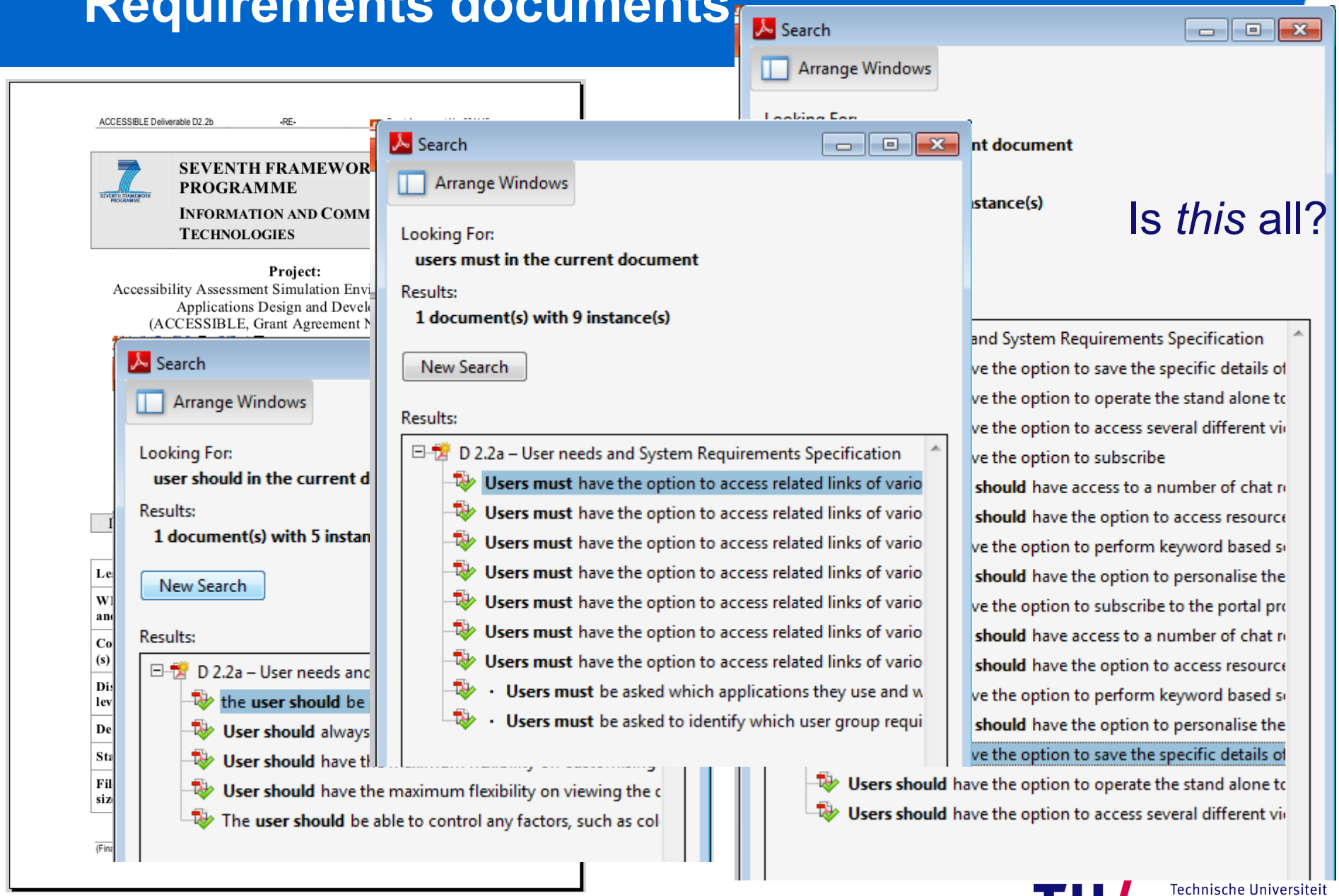
New Search

Results:

D 2.2a – User needs and System Requirements Specification
- **Users should** have the option to save the specific details of
- **Users should** have the option to operate the stand alone to
- **Users should** have the option to access several different vi
- **Users should** have the option to subscribe
- Registered **users should** have access to a number of chat r
- Registered **users should** have the option to access resource
- **Users should** have the option to perform keyword based s
- Registered **users should** have the option to personalise the
- **Users should** have the option to subscribe to the portal pr
- Registered **users should** have access to a number of chat r
- Registered **users should** have the option to access resource
- **Users should** have the option to perform keyword based s
- Registered **users should** have the option to personalise the
- **Users should** have the option to save the specific details of
- **Users should** have the option to operate the stand alone to
- **Users should** have the option to access several different vi

Is *this* all?

Requirements documents



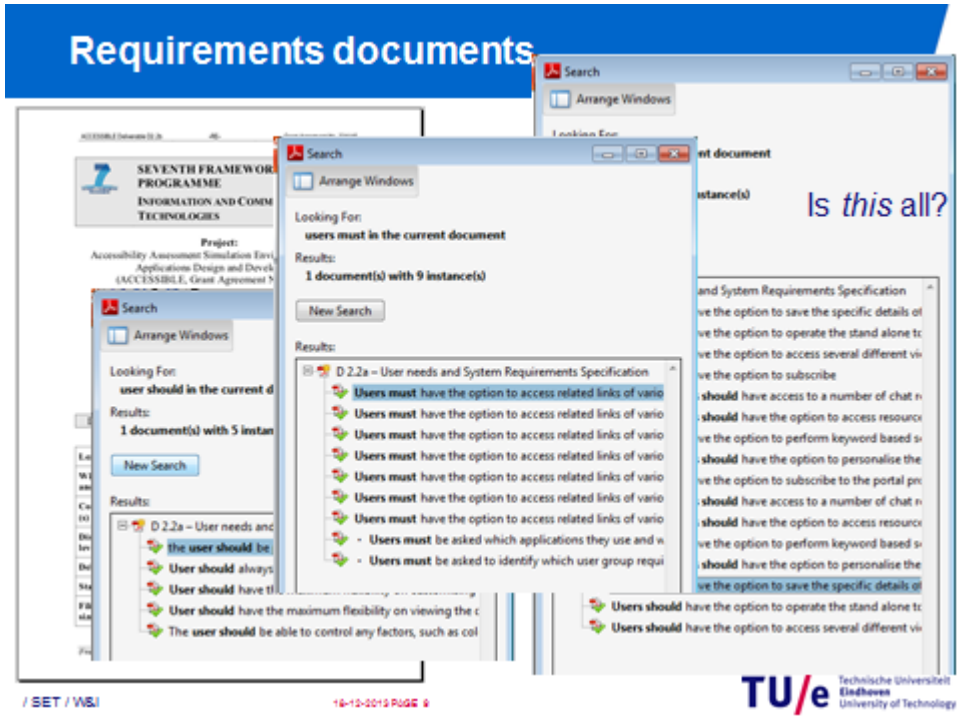
Is *this* all?

We need a better way

- **Goal:** specify how different parties (users, administrators, external systems, …) can interact with our system
  - *Recall: unambiguous, realistic, verifiable and evolvable*

- Plain-text does not work

- A more **structured** way is needed



Simple example

- What can you do with

http://globestaronline.com/panasonic-kx-ts500mx-corded-telephone/#.UrF1wPTuLy0

Simple example

- What can you do with

My answers:

Place a call

Receive a call

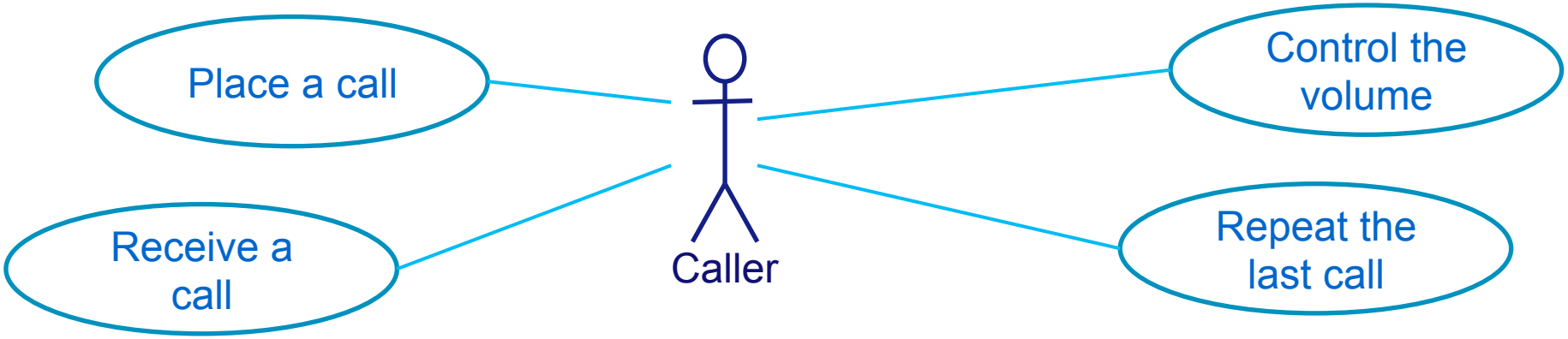
Control the volume

Repeat the last call

http://globestaronline.com/panasonic-kx-ts500mx-corded-telephone/#.UrF1wPTuLy0

Simple example

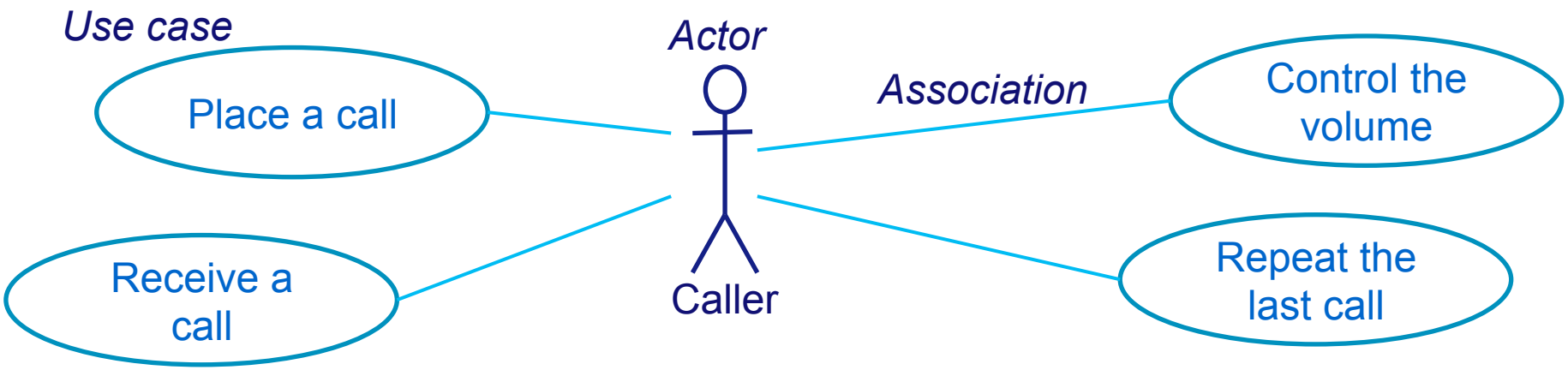
- What can you do with a telephone?



Basic use case diagram

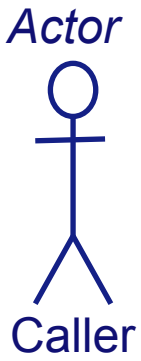
Simple example

- What can you do with a telephone?



Basic use case diagram

What is an *actor*?



- A **class of entities** (human or computer), falling **beyond the system boundaries**, **interacting** with the system

- NB:
  - Actor can be an **external software system**
  - One might have **multiple actors in a use case:** caller/callee
  - **One user** might be represented by **multiple actors**
    - doctor can be a patient

What is a *use case*?

Place a call

- **Use case:** a contract between the stakeholders of a system about its behavior *[Cockburn 2001]*
  - **Stakeholder**: a person, group or organization with an interest in a project

- Usually, a large **end-to-end** process comprising **several (inter)actions**
  - Internal operations are NOT use cases
  - Individual methods are NOT use cases
  - BUT it is one process: specification usually involves multiple use cases

Is placing a call large?

- How do you make a phone call?

  a) Pick up the phone
  b) Dial
  c) Talk

- Correct?

Is placing a call large?

- How do you make a phone call?

  a) Pick up the phone
  b) Dial
  c) Talk

- Correct?

http://cdn-media-att.vtp-media.com/ecp/documents/product_Product/1662/LargeImage/6098/CL83201%20+%20TWO%20CL80111_LargeImage.jpg

http://www.bbb.org/blog/wp-content/uploads/2011/08/phone.jpg

Is placing a call large?

- How do you make a phone call?
  a) The user picks up the telephone hook connected to the telephone line “A”.
  b) If the line is free, the user receives a dial tone sent by the line .
  c) The user dials number “B”.
  d) The call request is forwarded to the switch center.
  e) If line “B” is not busy, the call request is forwarded to “B” and a tone is sent to “A”.
  f) “B”’s telephone rings.
  g) If somebody at “B” picks up the hook, the ringing tone at “A” is stopped and a telephone connection will commence.

Is placing a call large?

- How do you make a phone call?
  a) The user picks up the telephone hook connected to the telephone line “A”.
  b) If the line is free, the user receives a dial tone sent by the line.
  c) The user dials number “B”.
  d) The call request is forwarded to the switch center.
  e) If line “B” is not busy, the call request is forwarded to “B” and a tone is sent to “A”.
  f) “B”’s telephone rings.
  g) If somebody at “B” picks up the hook, the ringing tone at “A” is stopped and a telephone connection will commence.

Main scenario

What if the “if” conditions do not hold?

b) If the line is free, the user receives a dial tone sent by the line.

b-1) If the telephone line is engaged in a conversation, the user will be connected to the same conversation.

b-2) If the user does not dial a number for a certain amount of time, a permanent tone is emitted by the switch center, no further call will be accepted and the user has to replace the hook.

What if the “if” conditions do not hold?

b) If the line is free, the user receives a dial tone sent by the line.

b-1) If the telephone line is engaged in a conversation, the user will be connected to the same conversation.

b-2) If the user does not dial a number for a certain amount of time, a permanent tone is emitted by the switch center, no further call will be accepted and the user has to replace the hook.

Alternatives

What if the “if” conditions do not hold?

e) If line “B” is not busy, the call request is forwarded to “B” and a tone is sent to “A”.

Propose an alternative for this step and discuss it with your neighbor.

What if the “if” conditions do not hold?

e) If line “B” is not busy, the call request is forwarded to “B” and a tone is sent to “A”.

My solutions

(e-1).1 If line “B” is busy, and “B” does not have call waiting the user at “A” will receive a busy tone.

(e-2).1 If line “B” is busy, and “B” has call waiting the user at “A” will receive a call-waiting tone from the switch center. When line “B” becomes free, sub-scenario (e-g) follows.

Use case description

- **Pre-condition:** when a use case is available to its user

Use case description

- **Pre-condition:** when a use case is available to its user
  - The telephone set is connected to the telephone line “A”, it is on-hook and there is no incoming call (it is not ringing).
- **Trigger:** action that initiates the use case

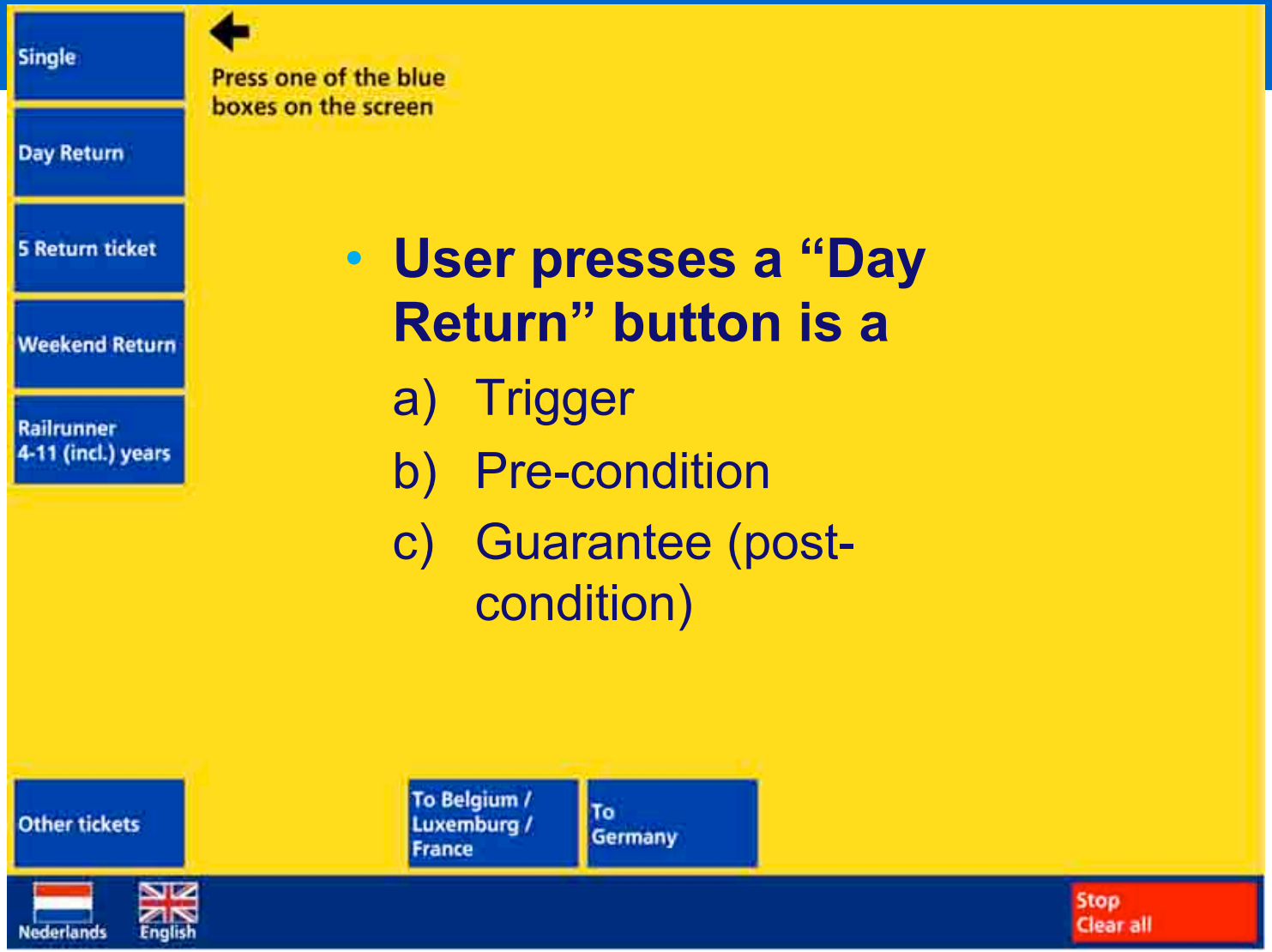
Use case description

- **Pre-condition:** when a use case is available to its user
  - The telephone set is connected to the telephone line “A”, it is on-hook and there is no incoming call (it is not ringing).
- **Trigger:** action that initiates the use case
  - The user picks up the telephone hook connected of the telephone set (connected to line “A”) and dials number “B”.
- **Guarantee (post-condition)**: what does the user achieve through the use case

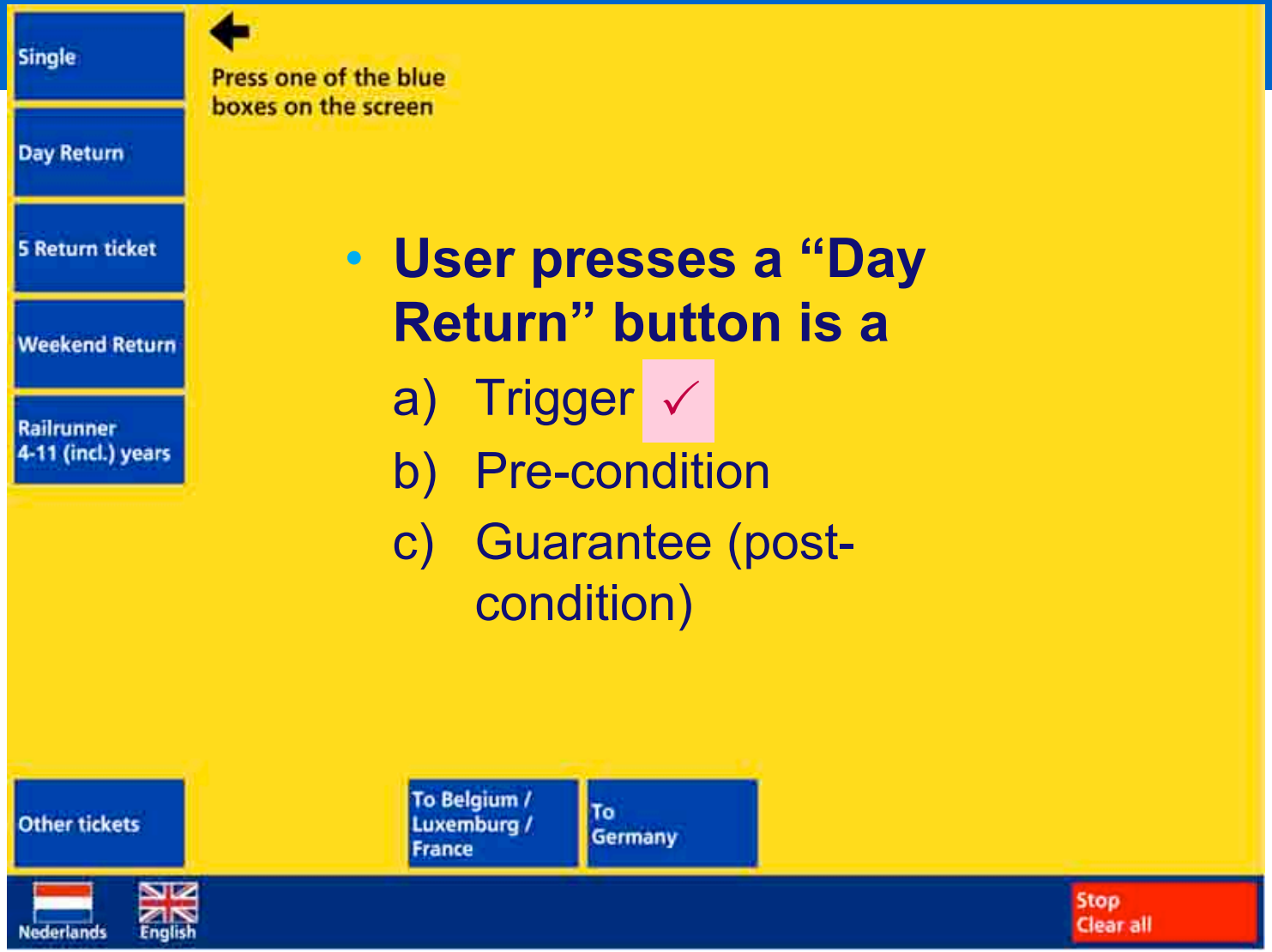
Use case description

- **Pre-condition:** when a use case is available to its user
  - The telephone set is connected to the telephone line “A”, it is on-hook and there is no incoming call (it is not ringing).
- **Trigger:** action that initiates the use case
  - The user picks up the telephone hook connected of the telephone set (connected to line “A”) and dials number “B”.
- **Guarantee (post-condition)**: what does the user achieve through the use case
  - A communication between “A” and “B” will commence
- **Main scenario**

- **Alternatives**

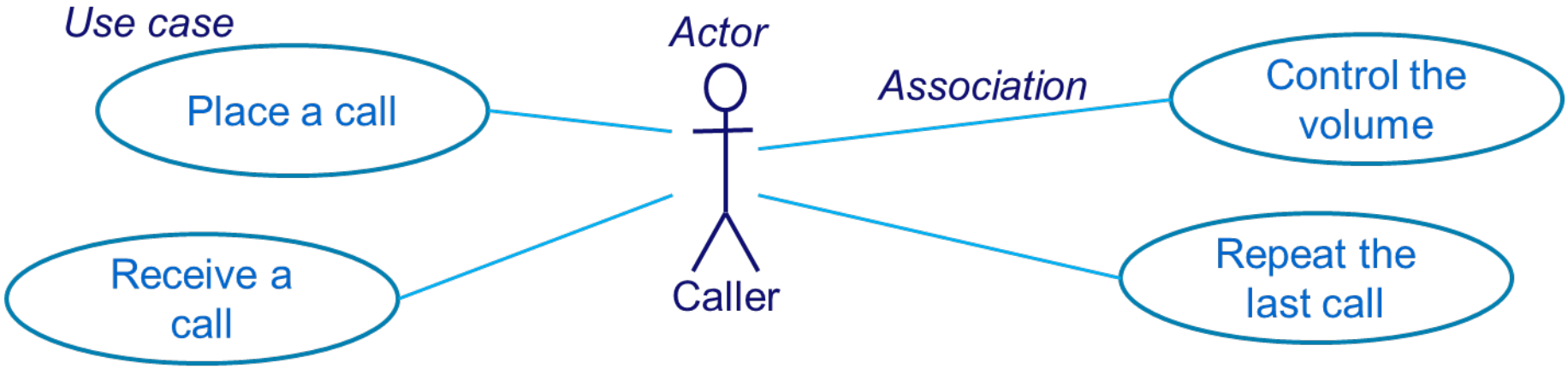


- **User presses a "Day Return" button is a**
  a) Trigger
  b) Pre-condition
  c) Guarantee (post-condition)



- **User presses a "Day Return" button is a**
  a) Trigger ✓
  b) Pre-condition
  c) Guarantee (post-condition)

Summary so far



What is an *actor*?

- A **class of entities** (human or computer), falling **beyond the system boundaries**, **interacting** with the system

- NB:
  - Actor can be an **external software system**
  - One might have **multiple actors in a use case:** caller/callee
  - **One user** might be represented by **multiple actors**
    - doctor can be a patient

Use case description

- **Pre-condition:** when a use case is available to its user
  - The telephone set is connected to the telephone line "A", it is on-hook and there is no incoming call (it is not ringing).
- **Trigger:** action that initiates the use case
  - The user picks up the telephone hook connected of the telephone set (connected to line "A") and dials number "B".
- **Guarantee (post-condition)**: what does the user achieve through the use case
  - A communication between "A" and "B" will commence
- **Main scenario**
- **Alternatives**

Practice

Make a description of the “Receive a call” use case

Discuss it with your neighbors

A more complex example

In the file system, users can create new files, execute, display (on different output devices) and delete existing files. There is a special type of delete, which removes the file permanently from the file system. The file system makes use of an access right system which specifies who the owner of each file is and what operations are allowed by which users. The owner of each file may change the access rights to the file and give or take other people's permissions to access the file. In addition to the person who creates the file, the administrator is considered the owner of all files.

Identify actors and use cases

Actors

- Recall:
  - A **class of entities** (human or computer), falling **beyond the system boundaries**, **interacting** with the system

- Hence,
  1. Identify system boundaries
  2. Identify interactions
  3. Classify the interacting entities

Actors

In the file system, **user**s can create new files, execute, display (on different output devices) and delete existing files. … The **owner of each file** may change the access rights to the file and give or take other people's permissions to access the file. In addition to the person who creates the file, the **administrator** is considered the owner of all files.

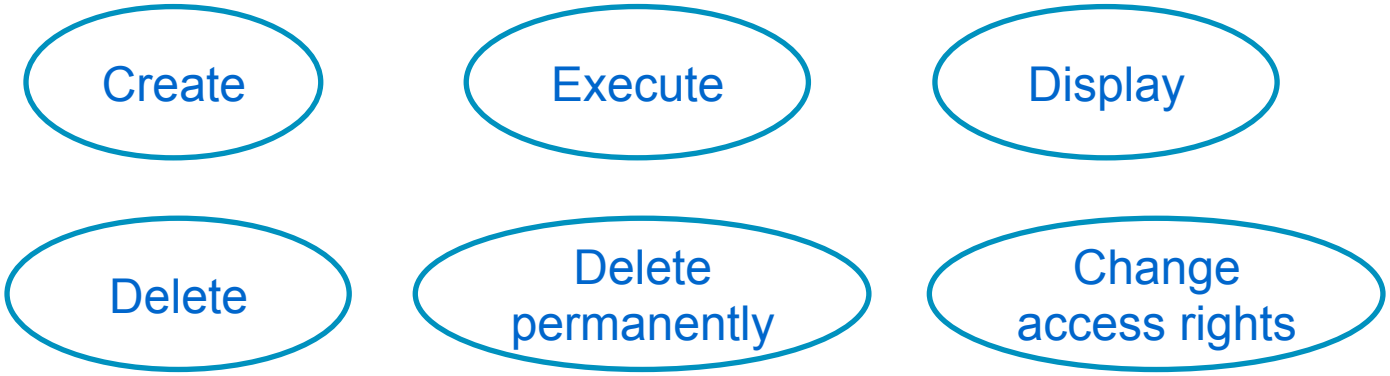
Use cases

- Recall:
  - A **contract** between the stakeholders of a system about its behavior *[Cockburn 2001]*
  - Usually, a large **end-to-end** processes comprising **several (inter)actions**

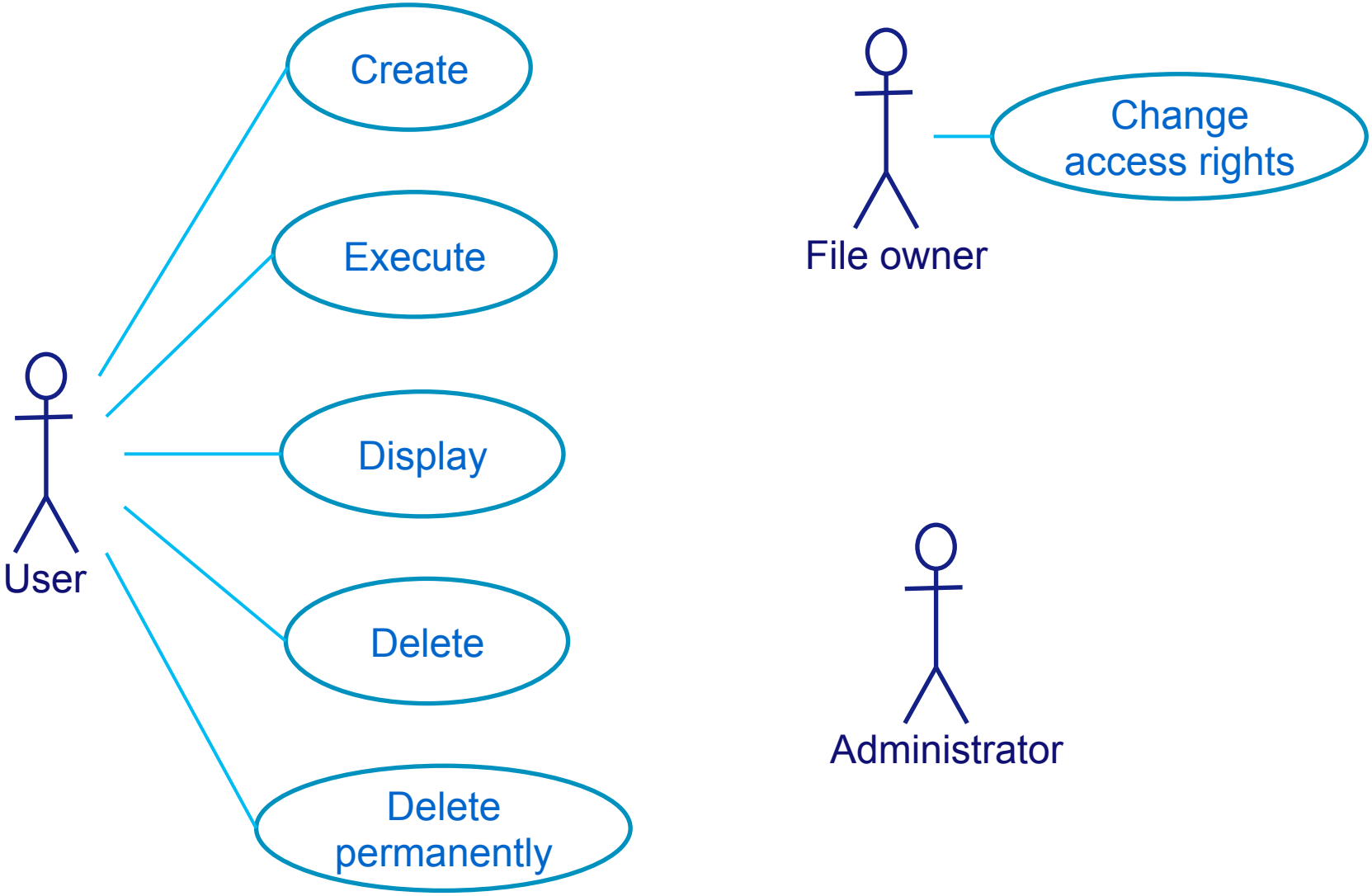
- Hence,
  1. Identify goals of the actors
  2. Check whether they are “end-to-end’’

Use cases

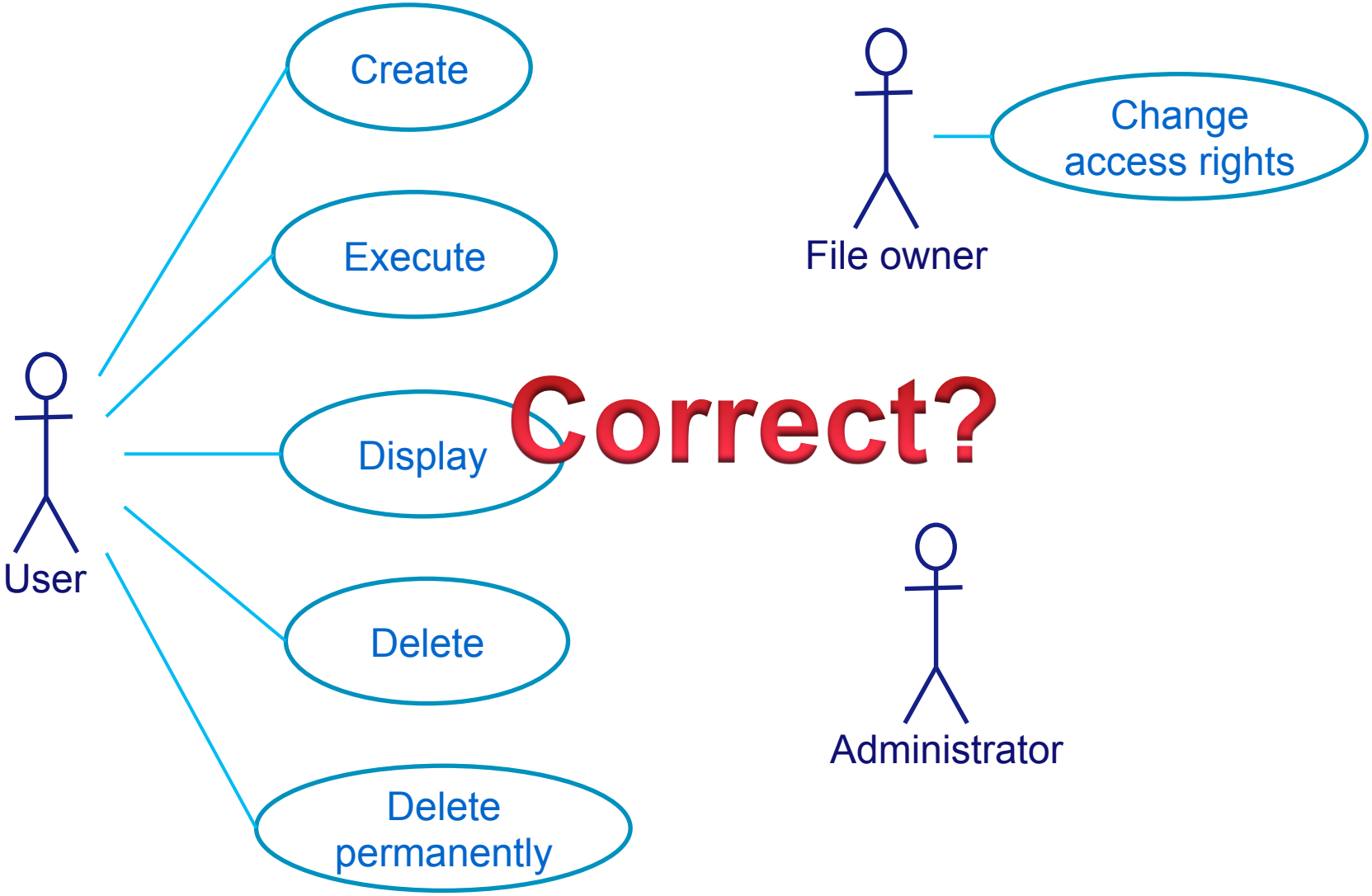
In the file system, users can **create new files**, **execute**, **display** (on different output devices) and **delete** existing files. There is a special type of **delete**, which removes the file **permanently** from the file system. The owner of each file may **change the access rights** to the file and give or take other people's permissions to access the file.



Who does what? Associations



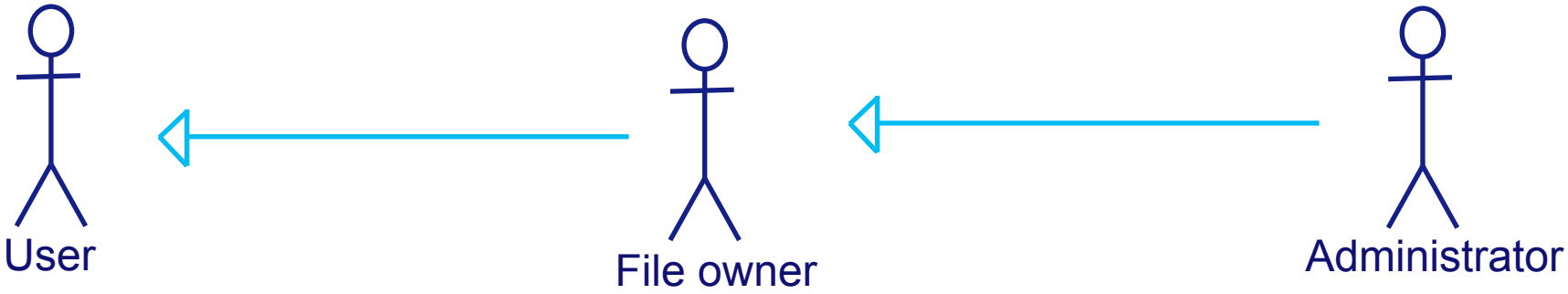
Who does what? Associations



Not all actors have been created equal

- Some users are file owners, but all file owners are users
  - File owner is a special kind of a user.

- Some file owners are administrators, but all administrators are file owners
  - Recall: *the administrator is considered the owner of all files*
  - Administrator is a special kind of a file owner.

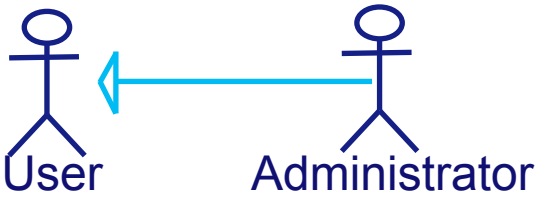


What about use cases?

- There is a special type of delete, which removes the file permanently from the file system.
  - Some deletions are permanent deletions, but all permanent deletions are deletions.
  - Permanent deletion is a special case of deletion

Generalization

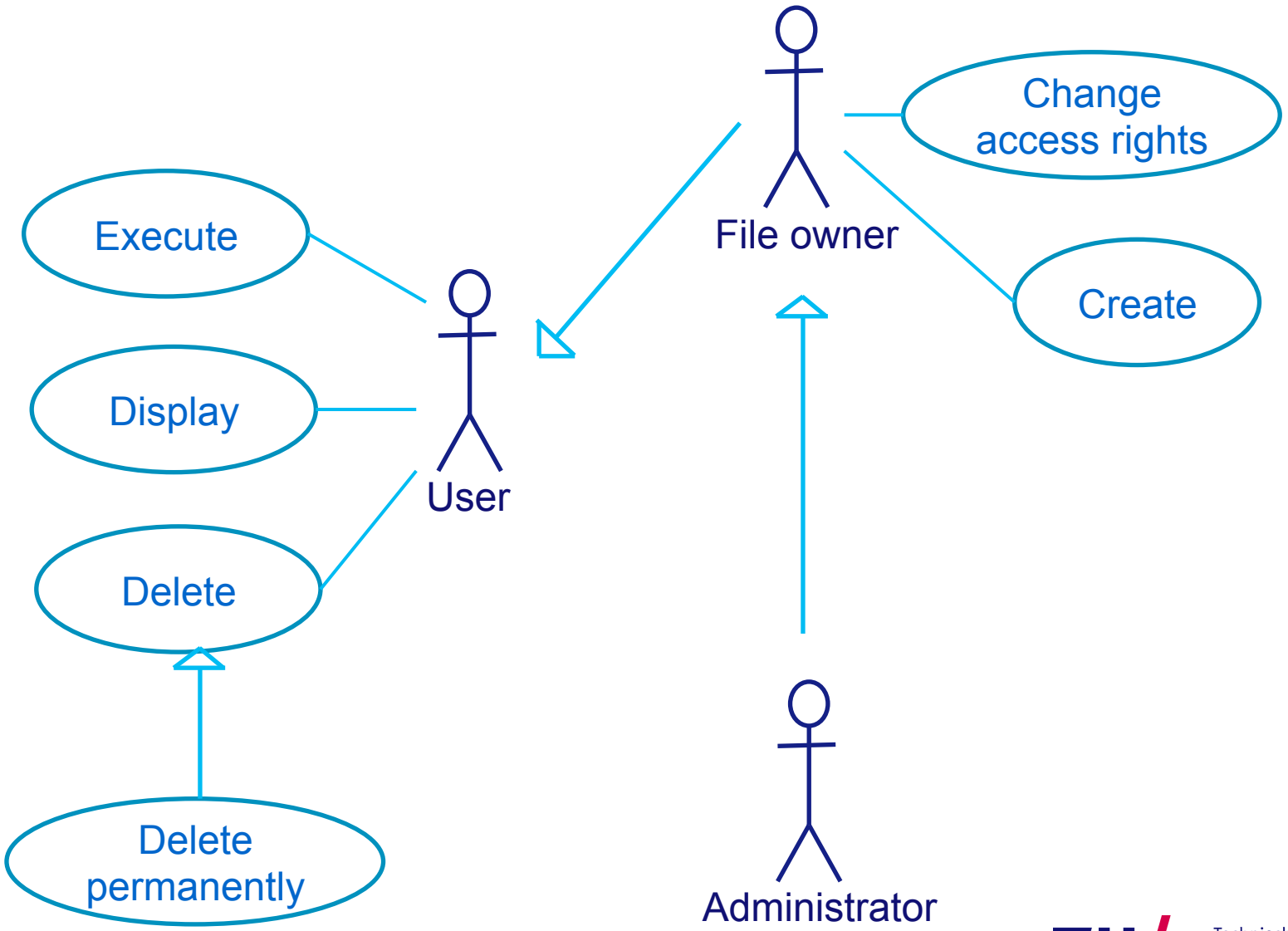
- **Generalization**: A use case / actor is a special case of another one.



- **Substitutability principle**: A specialized class (use case, actor) can always replace the general one.
  - Administrator can do everything the user can (and may be more)

- Does this remind you of the Database course?

Putting it all together: so far

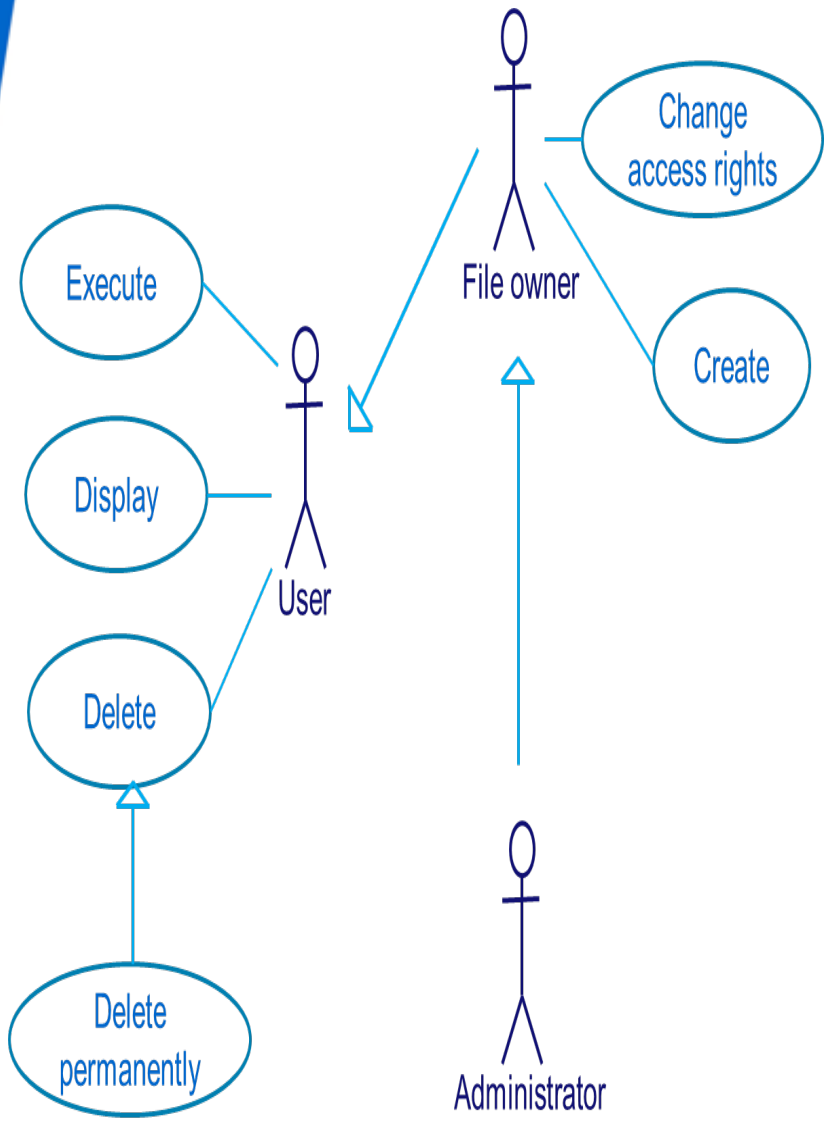


Have we specified everything we wanted?

A more complex example

In the file system, users can create new files, execute, display (on different output devices) and delete existing files. There is a special type of delete, which removes the file permanently from the file system. The file system makes use of an access right system which specifies who the owner of each file is and what operations are allowed by which users. The owner of each file may change the access rights to the file and give or take other people's permissions to access the file. In addition to the person who creates the file, the administrator is considered the owner of all files.

Identify actors and use cases



What have we missed so far

- “The file system makes use of an access right system which specifies who the owner of each file is and what operations are allowed by which users.”

- What does this mean for the Execute, Create, Display and Delete use cases?

What have we missed so far

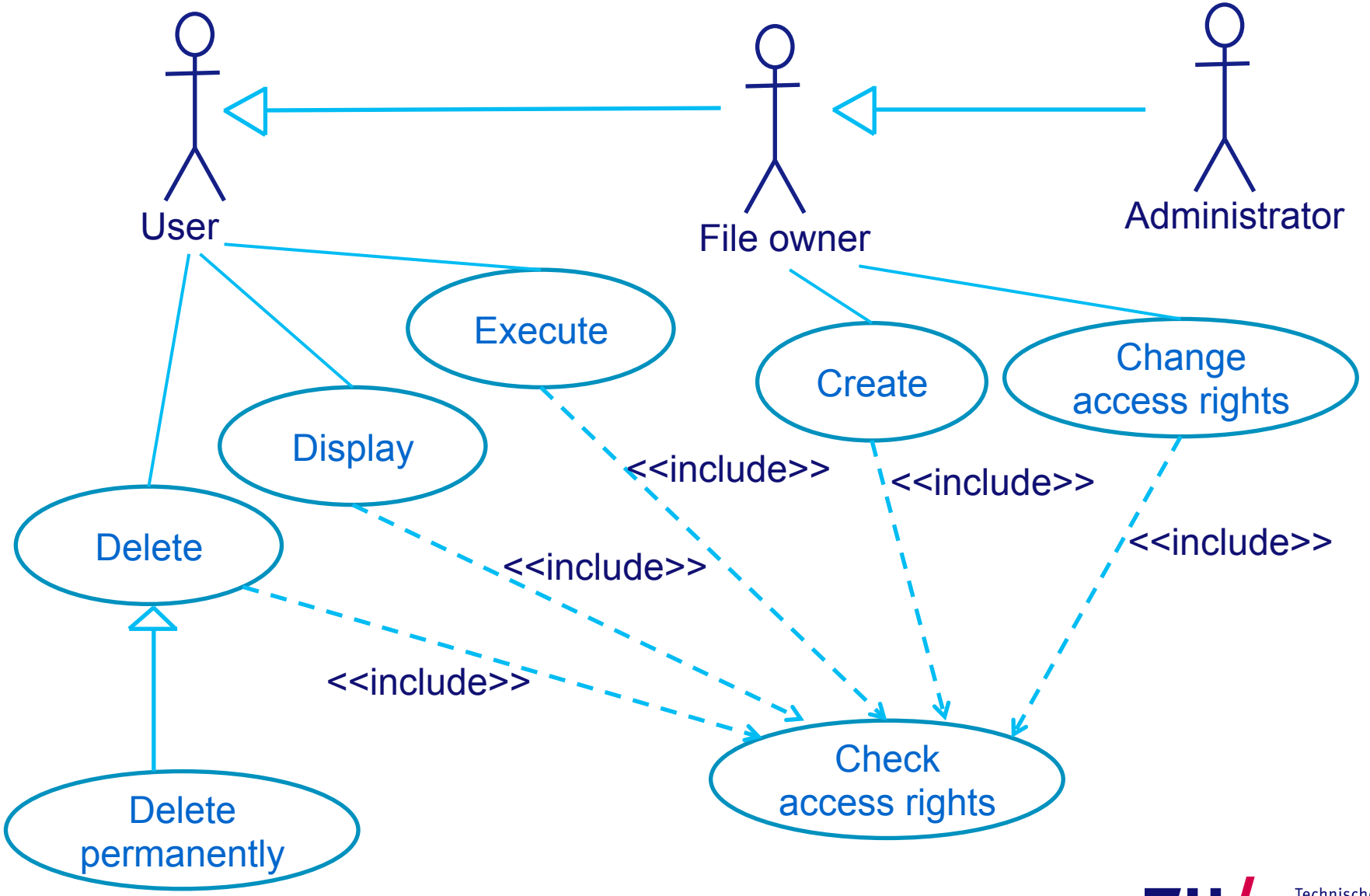
- “The file system makes use of an access right system which specifies who the owner of each file is and what operations are allowed by which users.”

- What does this mean for the Execute, Create, Display and Delete use cases?

- “Sub-use case” included in these use cases

Check
access rights

Including a use case in another use case



Formally

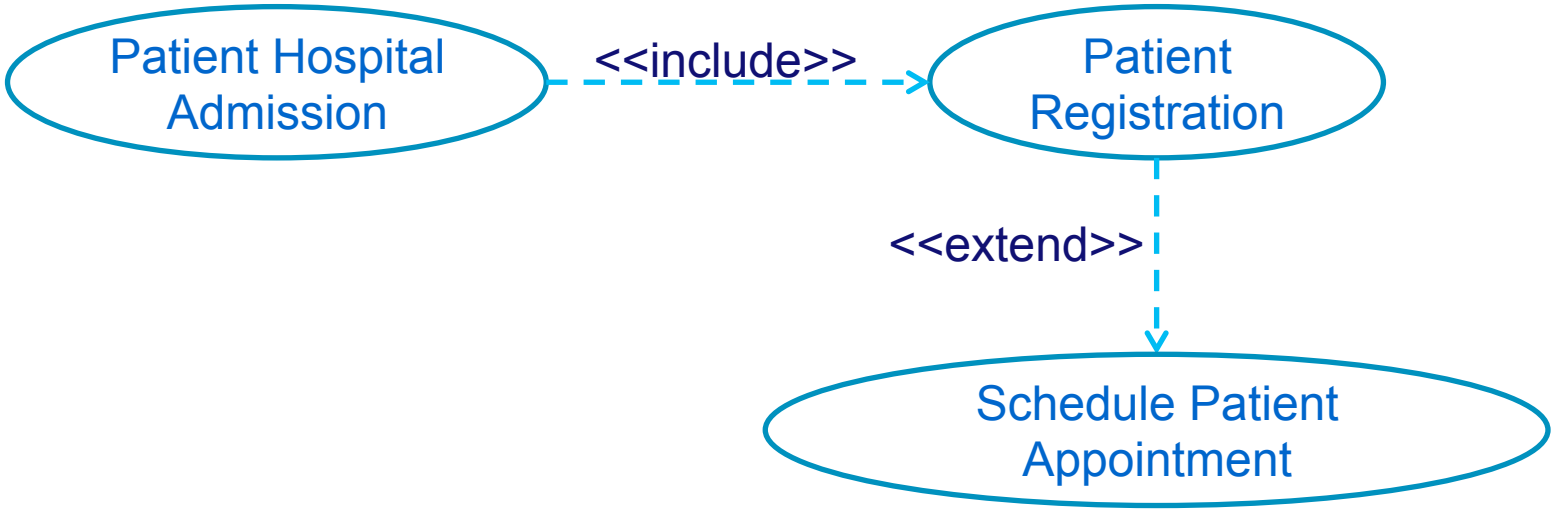
- **Dependency**: A use case depends on another use case for realizing its goal.
  - The target use case is *always* used by the source.

Formally

- **Dependency**: A use case depends on another use case for realizing its goal.
  - The target use case is *always* used by the source.

- What if the use case is used only sometimes?
  - Alternatives in the use case
  - **Extension**

Example



http://www.uml-diagrams.org/examples/hospital-management-use-case-diagram-example.html

Nota Bene:

stereotypes

	<<include>>	<<extend>>
	A ⇢ B	A ⇠ B
Read	A includes B	B extends A

Nota Bene:

	<<include>> 	<<extend>> 
Read	A includes B	B extends A
A can operate without B	no	yes
B is aware of A	no	yes

What is wrong with the following diagram?

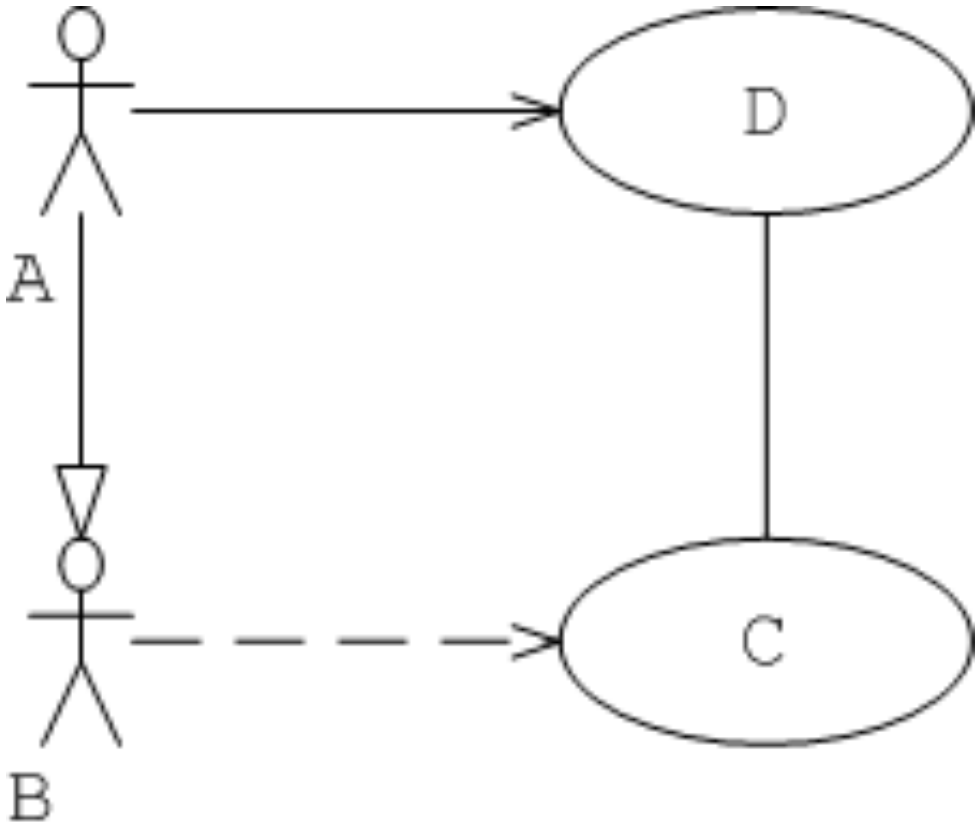
a) A-B

b) B-C

c) C-D

d) A-D

e) nothing



Example due to Ivanov and Novikov

What is wrong with the following diagram?

The only correct one

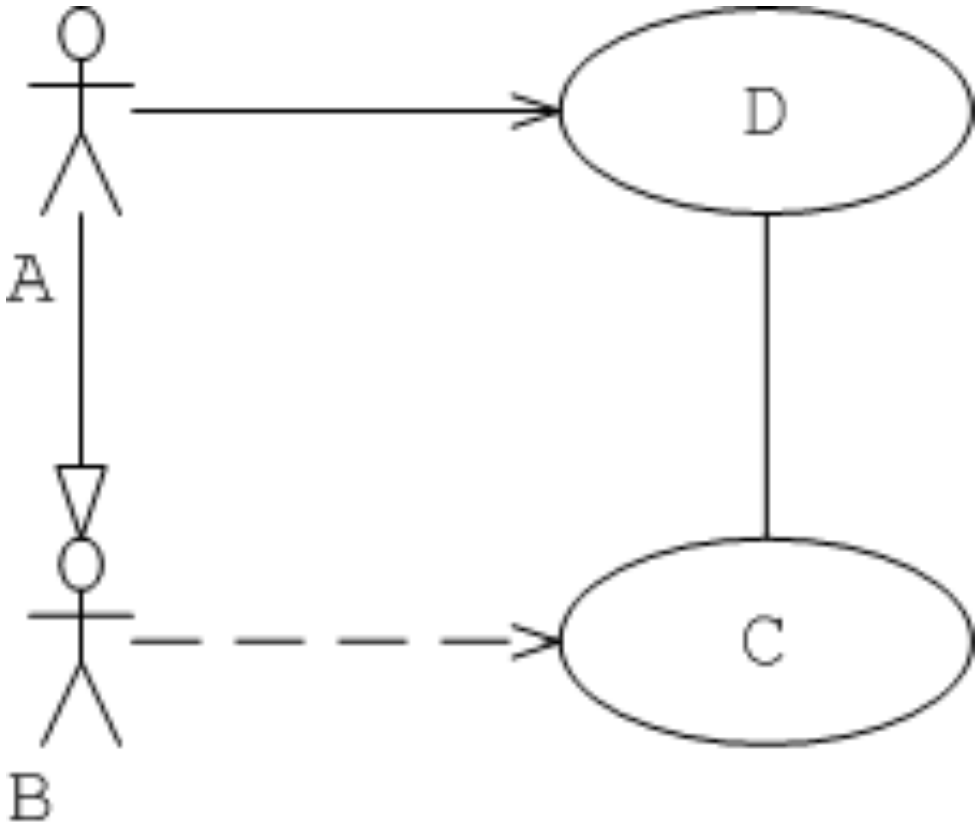
a) A-B

b) B-C ✓

c) C-D ✓

d) A-D ✓

e) nothing



Example due to Ivanov and Novikov

Actor A is associated with…

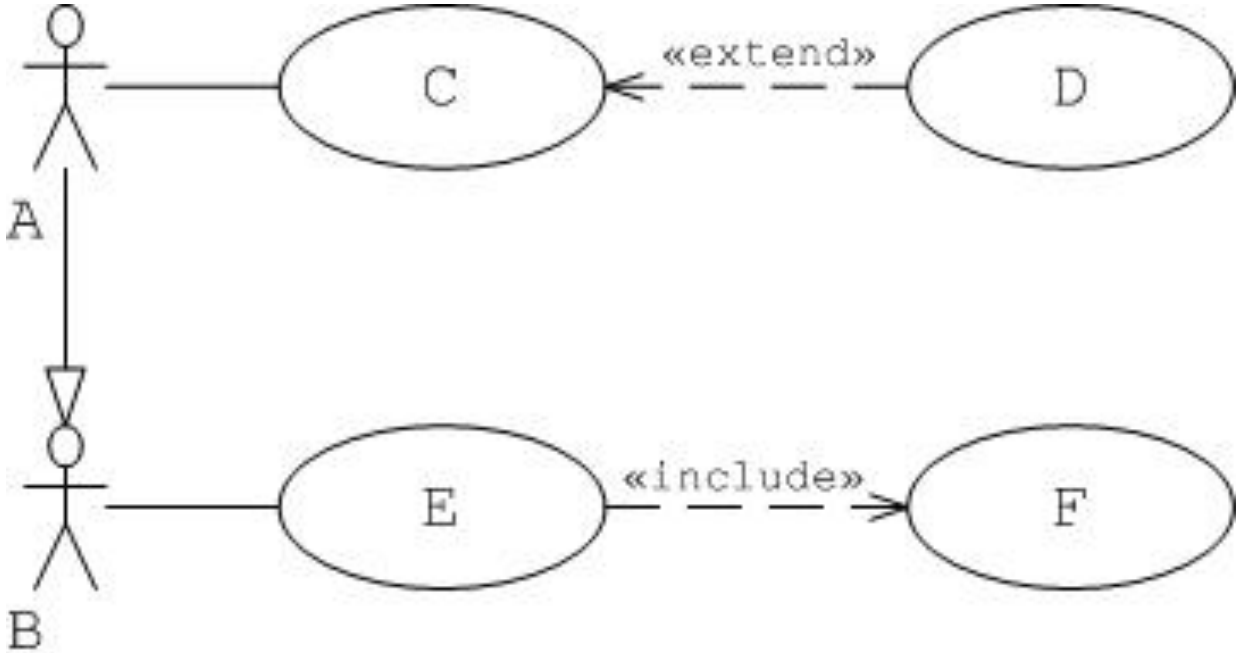
a) C

b) D

c) E

d) F

e) none



Example due to Ivanov and Novikov

Actor A is associated with…

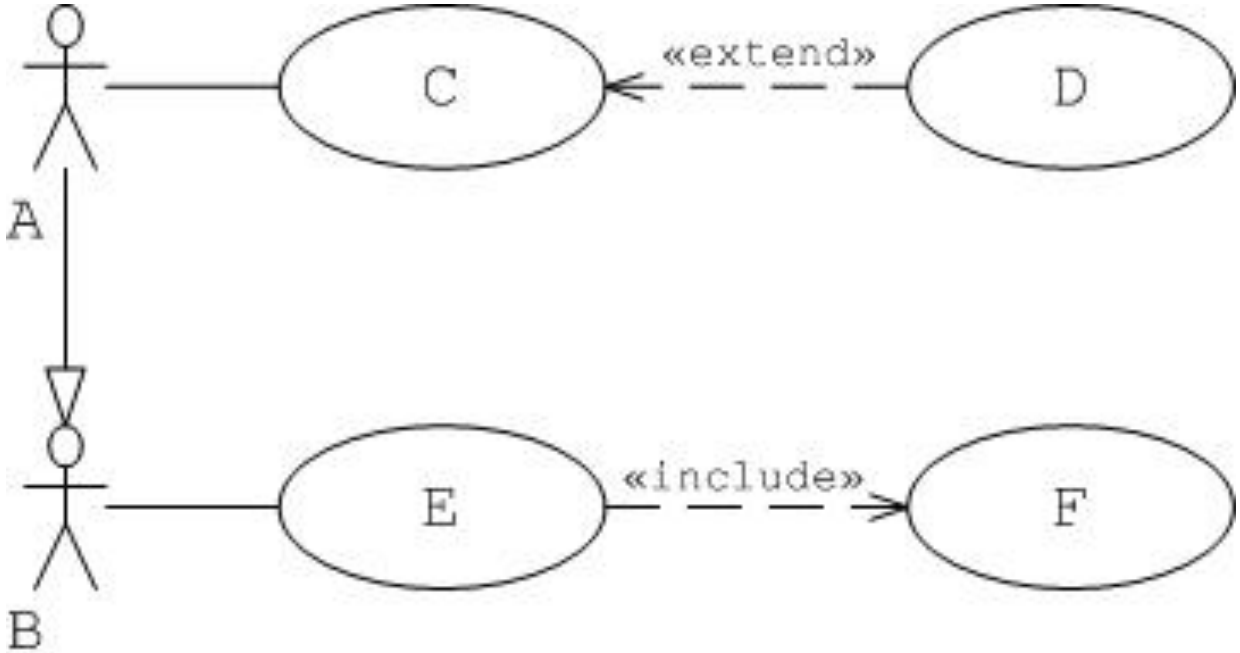
a) C ✓

b) D

c) E ✓

d) F ✓

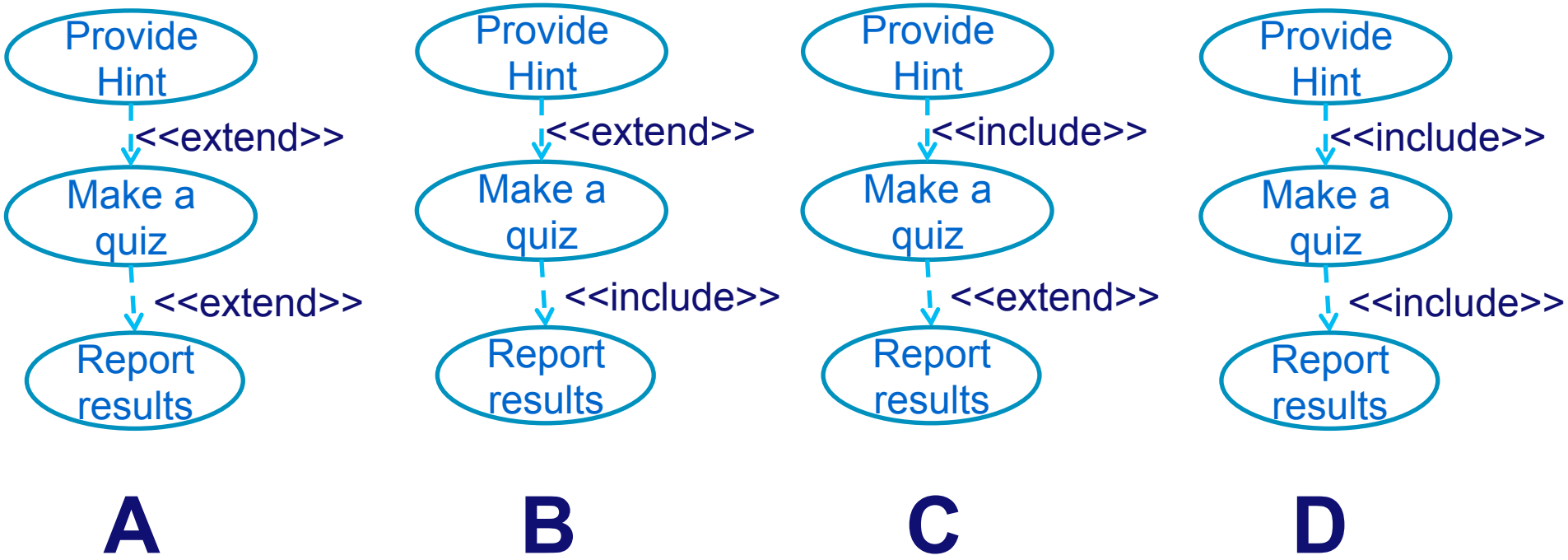
e) none



Example due to Ivanov and Novikov

Which model is correct?

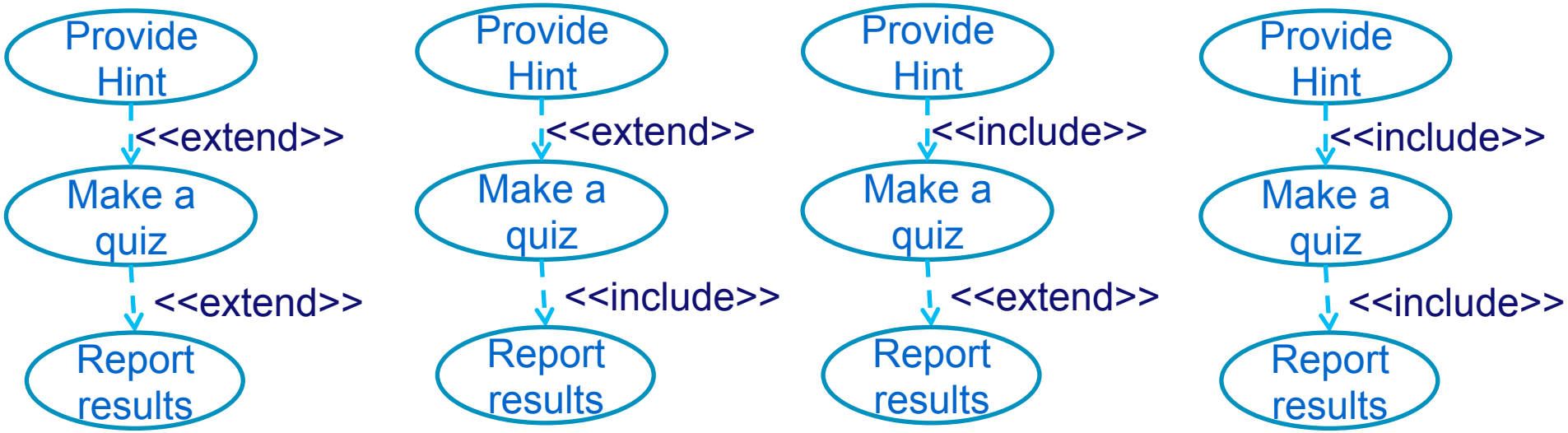
Digital learning environment supports students' learning through quizzes. Given your knowledge of this kind of systems which of the following models would be correct?



Inspired by http://www.cs.uu.nl/docs/vakken/mso/1011/asgs/usecase.sol.pdf

Which model is correct?

Digital learning environment supports students' learning through quizzes. Given your knowledge of this kind of systems which of the following models would be correct?

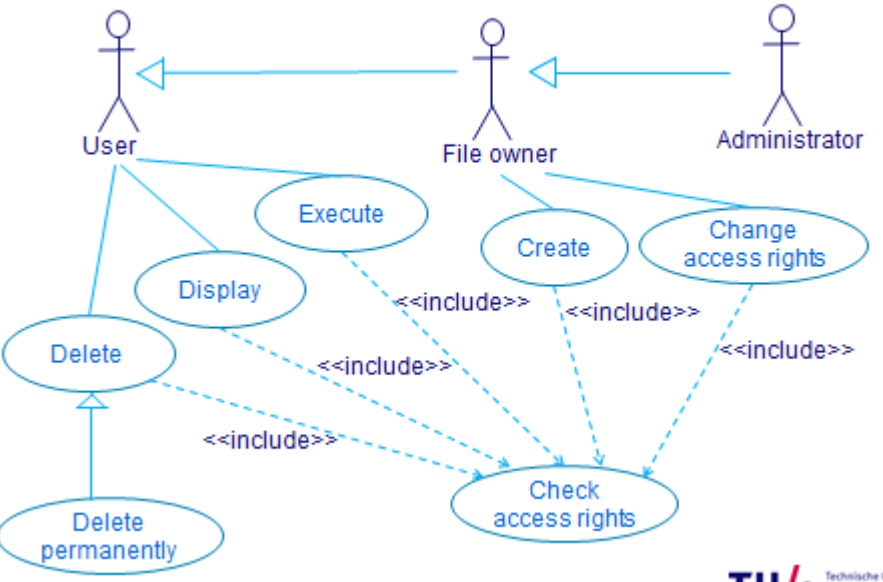


A **B** ✓ **C** **D**

Inspired by http://www.cs.uu.nl/docs/vakken/mso/1011/asgs/usecase.sol.pdf

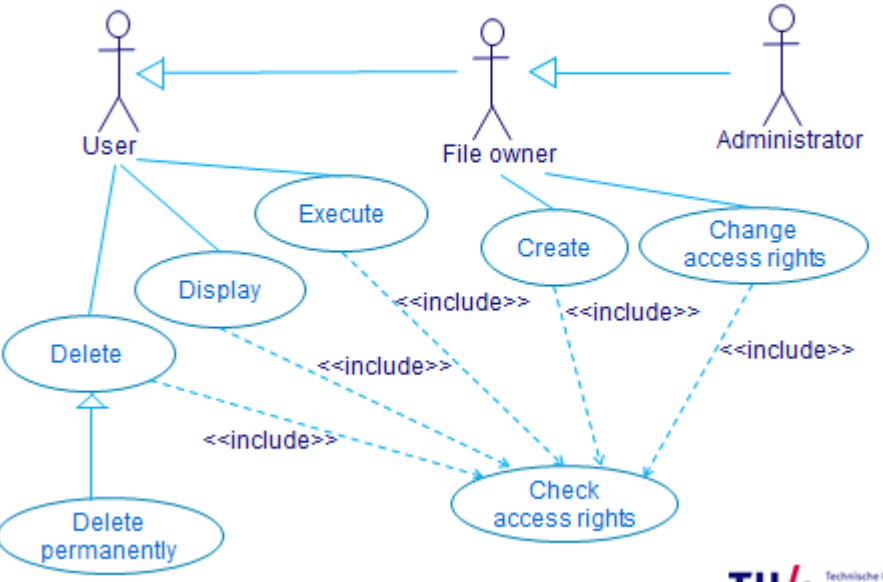
Brief recapitulation

Including a use case in another use case



Brief recapitulation

Including a use case in another use case


User
File owner
Administrator
Execute
Create
Change access rights
Display
Delete
Delete permanently
Check access rights
<<include>>

Helps to organize

Use case description

- **Pre-condition:** when a use case is available to its user
  - The telephone set is connected to the telephone line "A", it is on-hook and there is no incoming call (it is not ringing).
- **Trigger:** action that initiates the use case
  - The user picks up the telephone hook connected of the telephone set (connected to line "A") and dials number "B".
- **Guarantee (post-condition)**: what does the user achieve through the use case
  - A communication between "A" and "B" will commence
- **Main scenario**
- **Alternatives**

Use case diagrams as a specification technique?

Unambiguous?

Realistic?

Verifiable?

Evolvable?

Use case diagrams as a specification technique?

Unambiguous?

- formal rules how to compose use case diagrams, but ambiguity may be hidden in textual descriptions

Realistic?

- allows to model interaction with the external world but may be too simplistic to capture complex interactions

Verifiable?

- yes, through testing

Evolvable?

- if too many use cases are present the diagram becomes cluttered and less suited for evolution.

UML

- Use case diagram – one of the UML diagram types

- Unified Modeling Language (UML) is a standardized (ISO/IEC 19501:2005), general-purpose modeling language in the field of software engineering.
  - includes graphic notation techniques to create visual models of object-oriented software-intensive systems.
  - started in 1990s
  - UML 1.1 – 1997
  - UML 2.0 – 2005
  - UML 2.4 – 2011
  - We are using UML 2.5 (in progress), September 2013

UML 2.5

OMG Unified Modeling Language ™ (OMG UML)

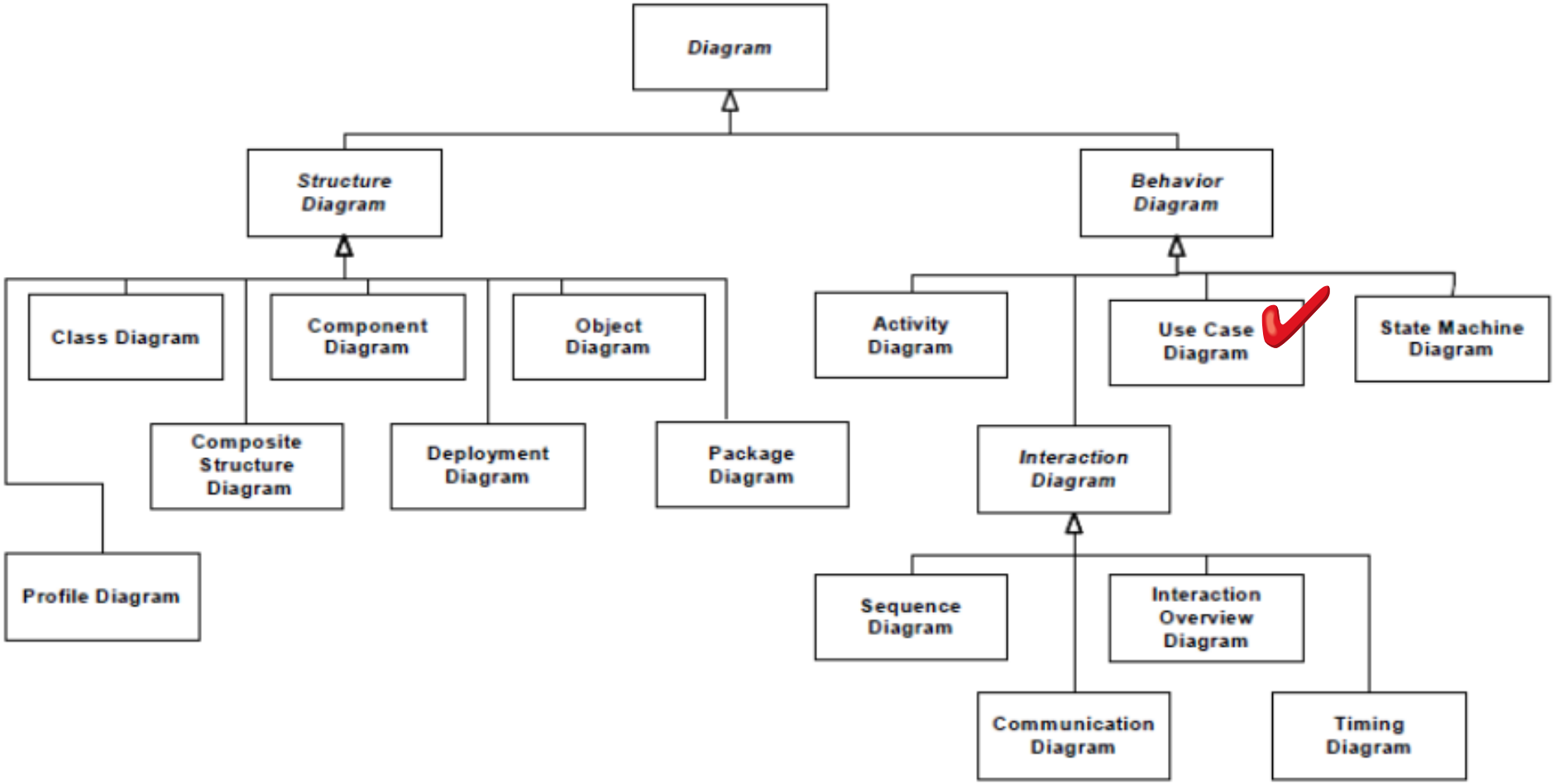
Version 2.5

http://www.omg.org/spec/UML/2.5/Beta1/

831 pages

I do not expect you to learn UML specification by heart, but please consult it in case of doubt.

UML diagram types



Reminder…

- Assignment **"Requirements"**
  - **Deadline**: February 12, 23:59
  - **Groups**:
    - software developers usually work in groups
    - learning to work together
  - **15%** of the grade of 2IW82

http://www.minibottlelibrary.com/mbl/alpha/jim-beam/fox-on-dolphin.jpg