Requirements Engineering & Management

Scenarios II – Use Cases

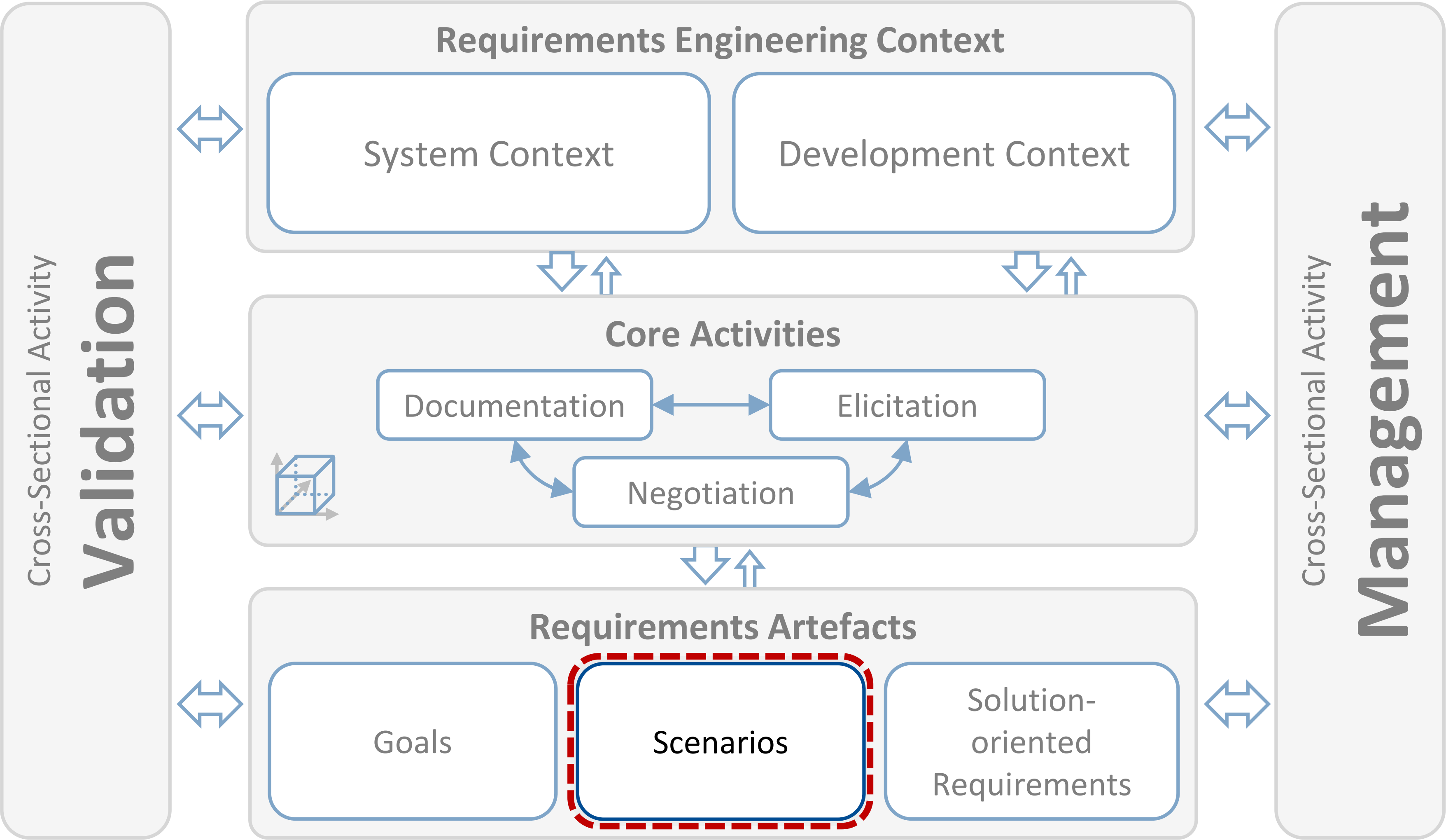
Prof. Dr. Klaus Pohl

The Ruhr Institute for Software Technology

Agenda

1. Introduction to Use Cases
2. Modelling Use Cases with Use Case Diagrams
3. Specifying Use Cases with Use Case Templates

Framework for Requirements Engineering

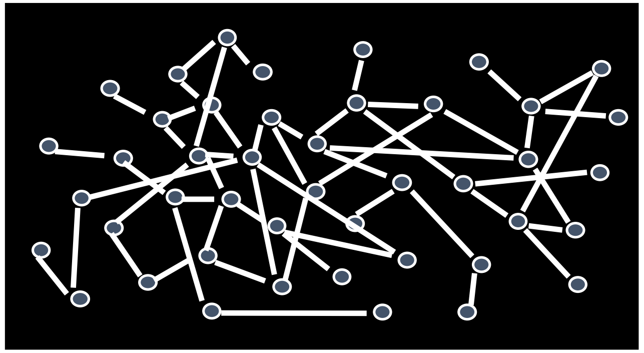


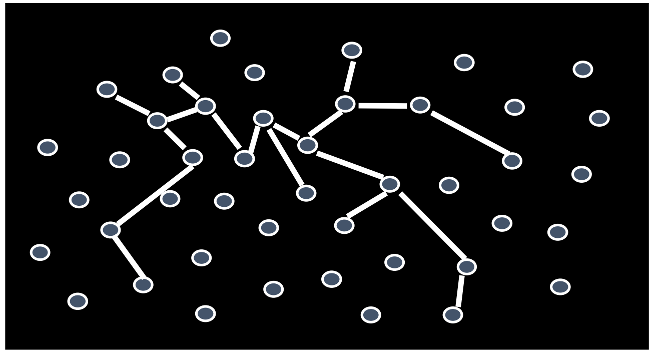
© SSE, Prof. Dr. Klaus Pohl

1. Introduction to Use Cases

Why Use Cases?

- Use cases **structure complex processes** with multitudes of possible **sequences of actions**.
- One use case shows a single sequence of actions to **reduce complexity**.
- Hence, use cases support the **understandability** of requirements.





Use Case: Definition

D A use case is a **specific way of using the system** by performing some part of the functionality. Each use case **constitutes a complete course of events** initiated by an actor and it specifies the **interaction that takes place between an actor and the system**. A use case is thus a special sequence of related transactions performed by an actor and the system in dialogue. The **collected use cases specify all the existing ways of using the system**.

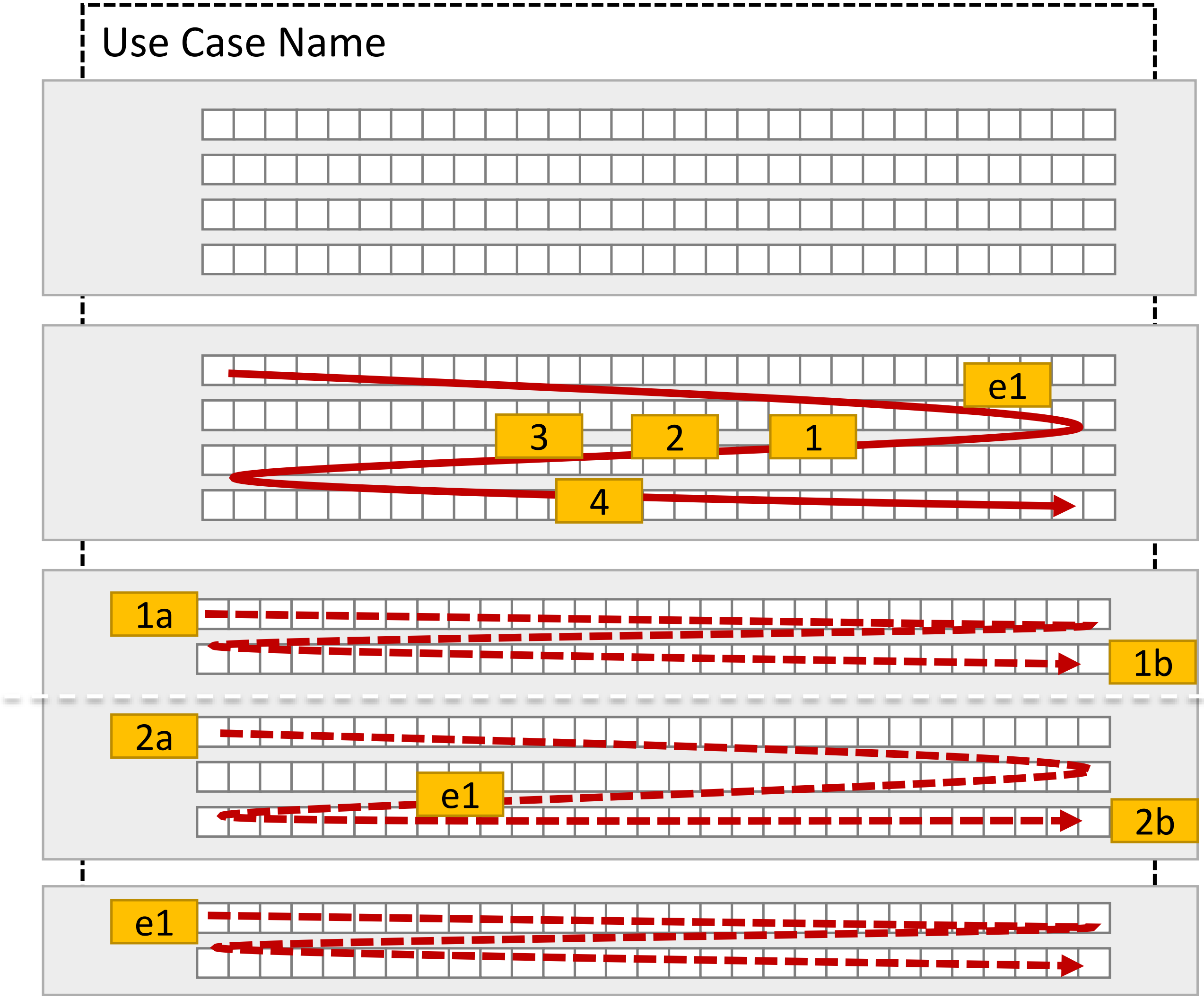
[Jacobson et al. 1992]

Scenarios and Use Cases

- Use cases can be used to **group** scenarios:
  - Related to a specific set of goals.

- Use cases can **integrate** related scenarios:
  - **Main** scenario
  - **Alternative** scenarios
  - **Exception** scenarios

Integration of Scenarios with Use Cases



Use case context
(e.g., goals, pre-conditions, post-conditions)

Main scenario
(exactly one scenario)

Alternative scenarios
(zero to multiple scenarios)

Exception scenarios
(zero to multiple scenarios)

2. Modelling Use Cases with Use Case Diagrams

Use Case Diagrams – Overview

- Use case diagrams represent the
  - relevant **relations** between external actors (such as users) and the use case
  - and **relations** between use cases.

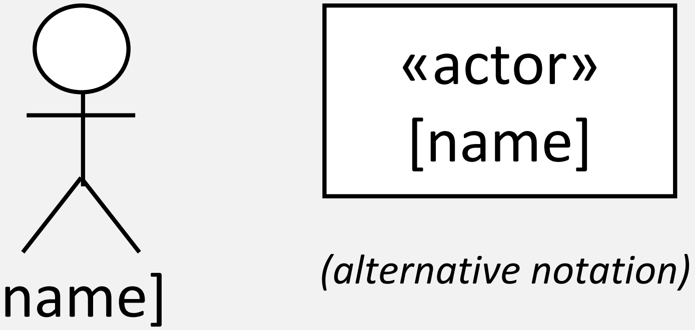
- **A single use case diagram does not represent the involved use cases at a sufficient level of detail!**
  - A more detailed specification of the involved use cases is required.

More details later in this lecture!

Modelling Construct: Actor

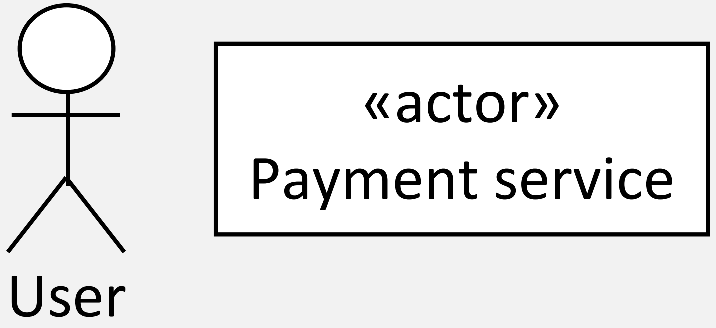
- Represents **external entities** interacting with the system
- **Abstracts** from specific actor instances by describing roles
- **Interacts** with the system by participating in use cases

Notation



(alternative notation)

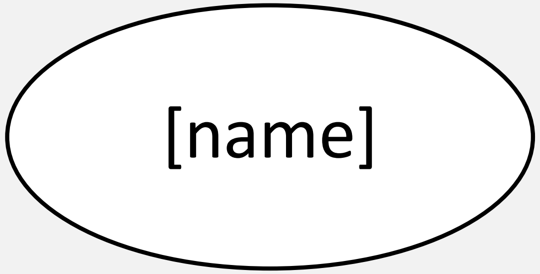
E



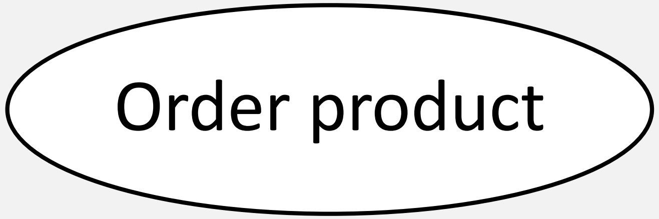
Modelling Construct: Use Case

- A **specific way of using the system** using the system "functionality"
- Constitutes (abstracts from) a **complete course of events** initiated by an actor and the **interaction** between an actor and the system
- Collected use cases specify all the existing **ways of using the system**

Notation



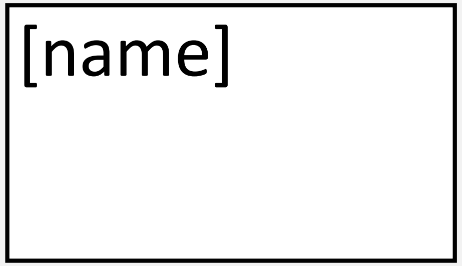
E



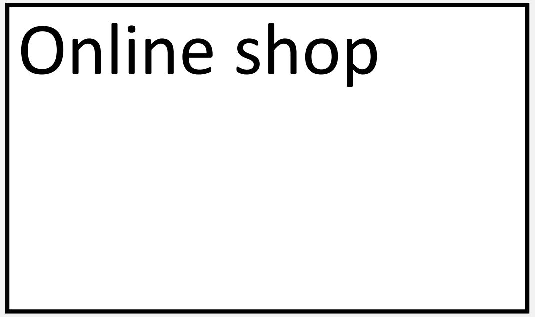
Modelling Construct: System Boundary

- **Separates** the system from its operational **context**
- **Use cases** are placed **inside** the system boundary while **external actors** are placed **outside**.
- The system boundary defines the **scope** for which the requirements are specified.

Notation



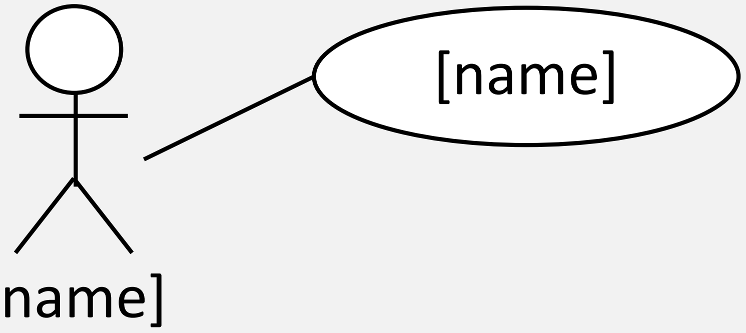
E



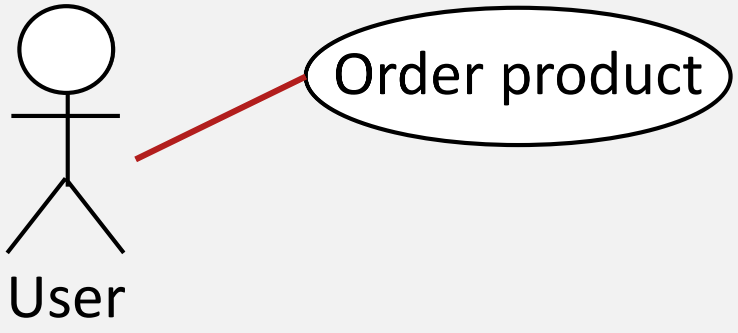
Modelling Construct: Association

- Represents the **participation** of an actor in a use case
- Instances of **actors** (i.e. instances of external entities having the respective role) can **communicate with** instances of **use cases**.
- Association relations are **bidirectional**.

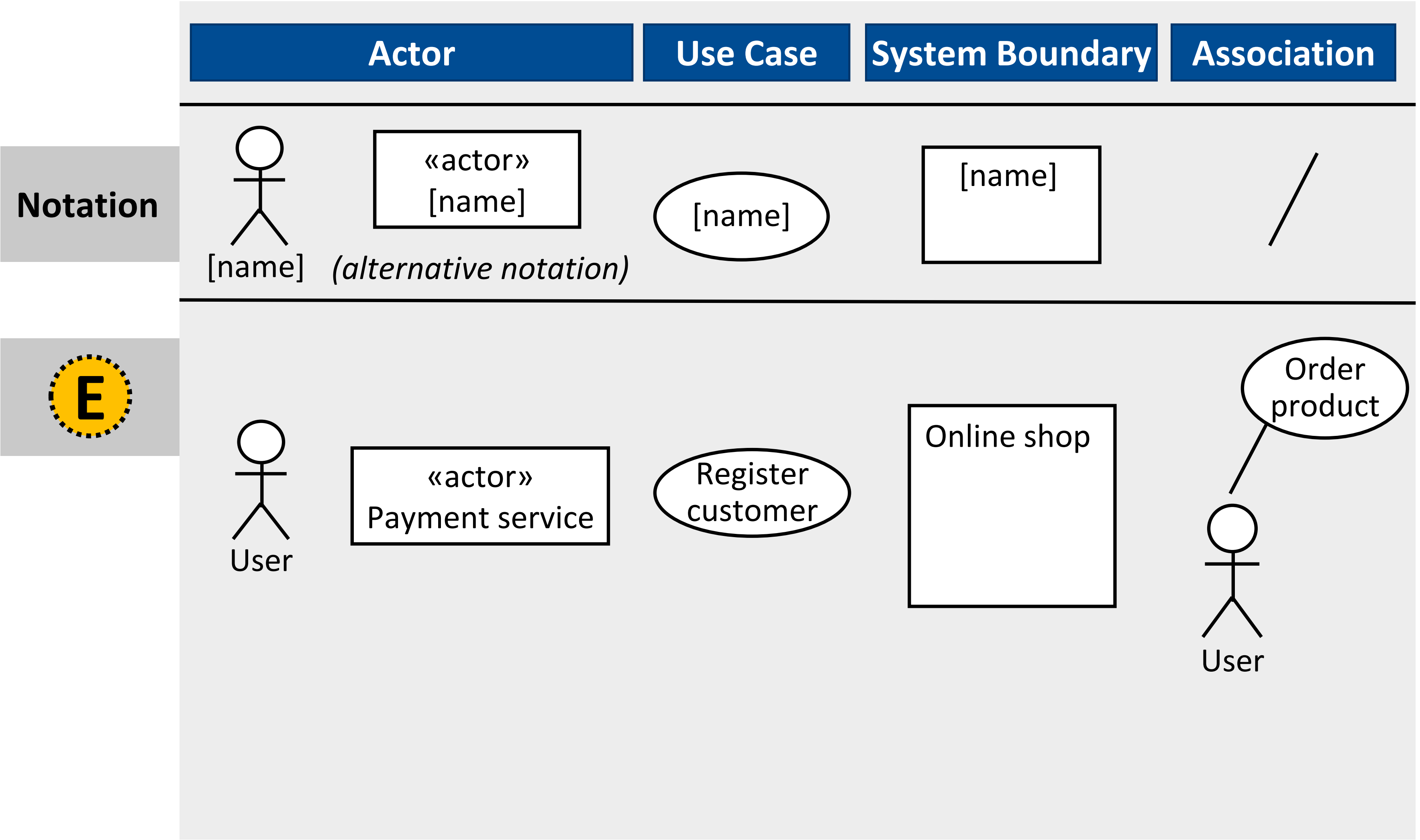
Notation



E



Overview on Introduced Modelling Constructs

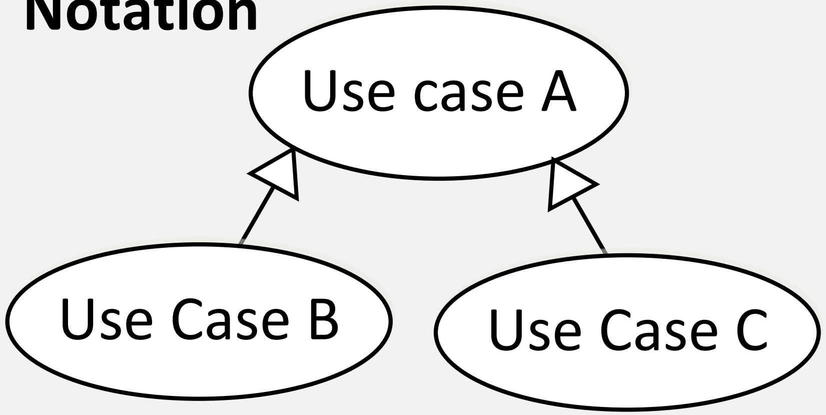


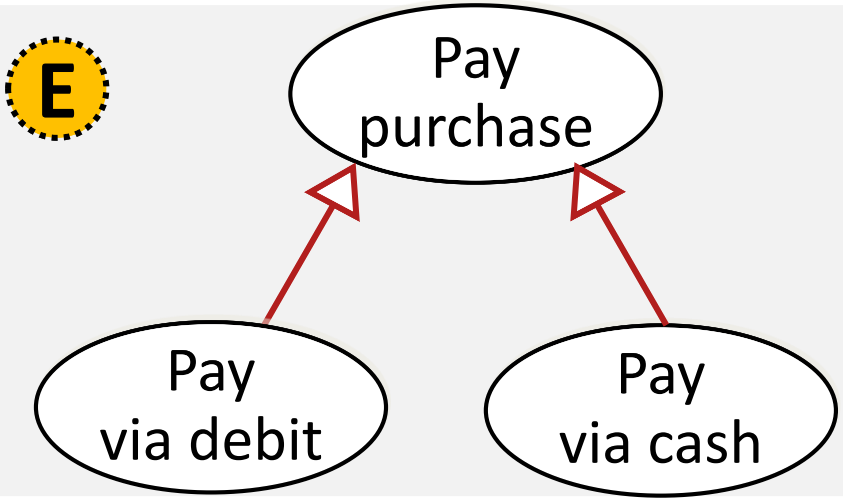
	Actor		Use Case	System Boundary	Association
Notation	[name]	«actor» [name] *(alternative notation)*	[name]	[name]	
E	User	«actor» Payment service	Register customer	Online shop	Order product User

Modelling Construct: Use Case Generalization

- Specializes a more **general use case into** one or multiple **specialized**, i.e. more specific use cases
- Specialized use cases may have **additional properties** and associations.

Notation

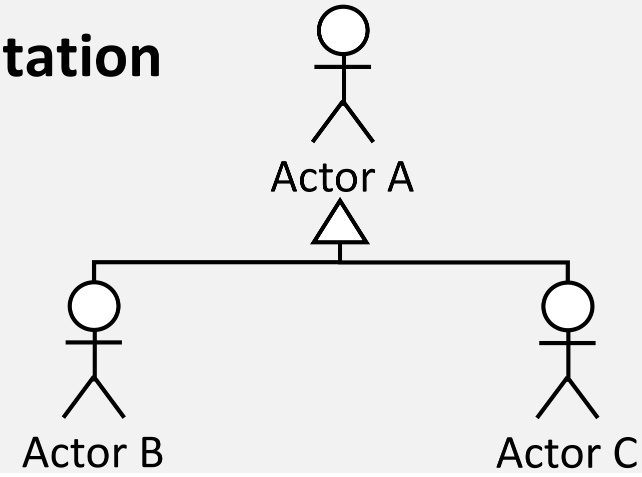


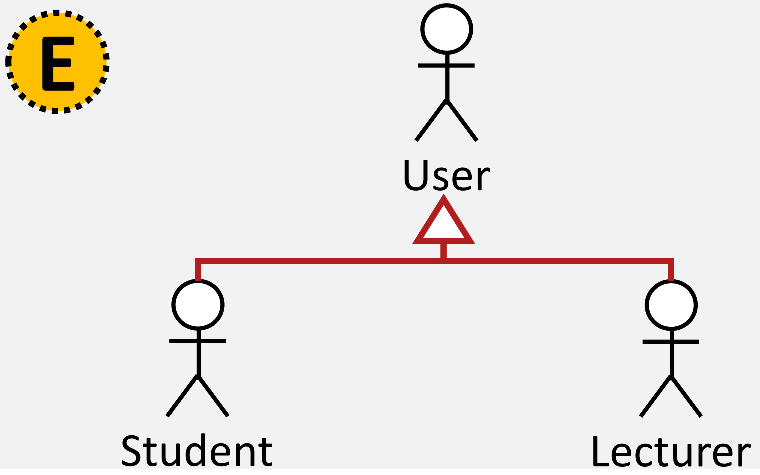


Modelling Construct: Actor Generalization

- Specializes a more **general actor** into one or multiple **specialized**, i.e. more specific actors
- Specialized actors may have **additional properties** and associations.

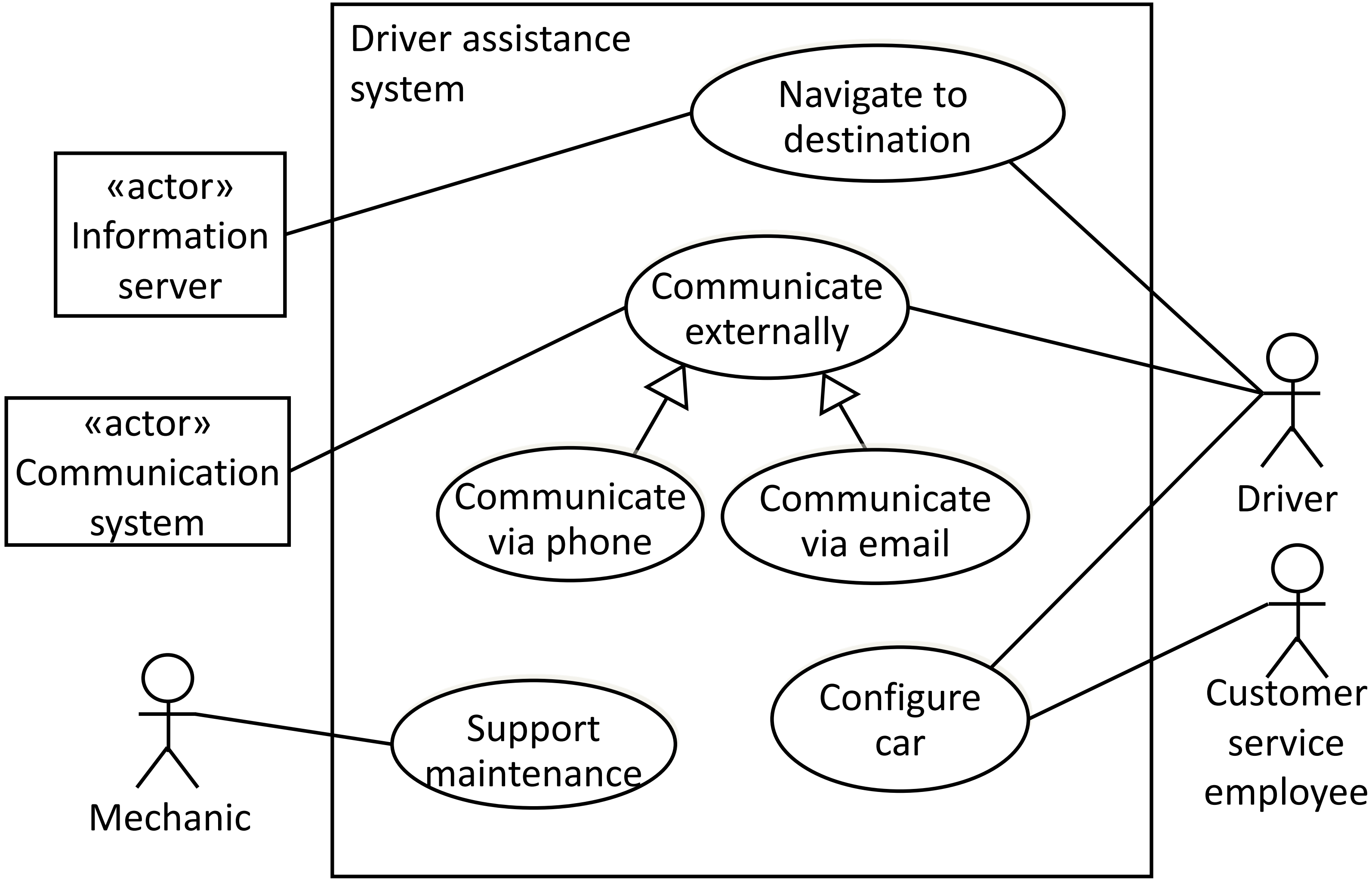
Notation





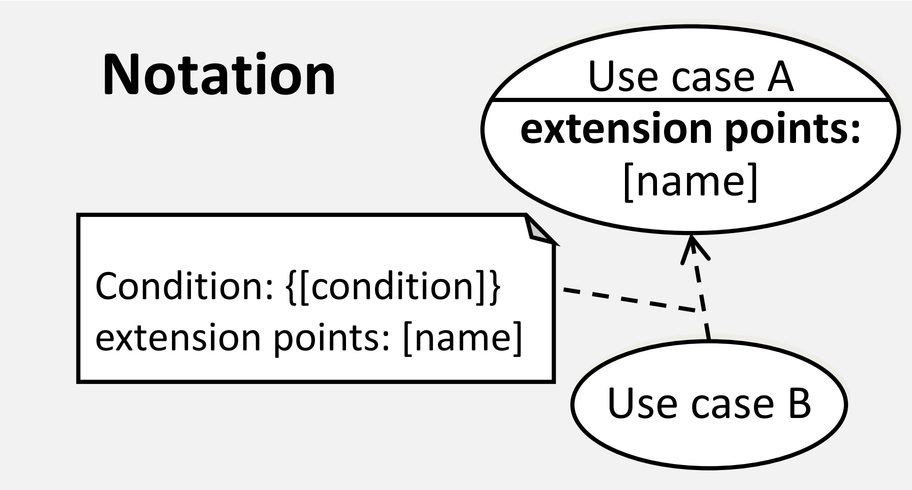
Example of a Use Case Diagram

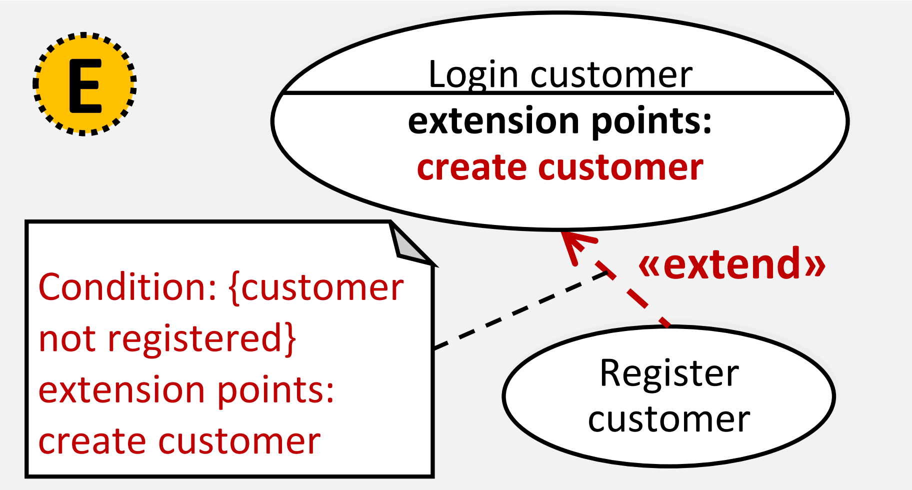
E



Modelling Construct: Extend Relationship

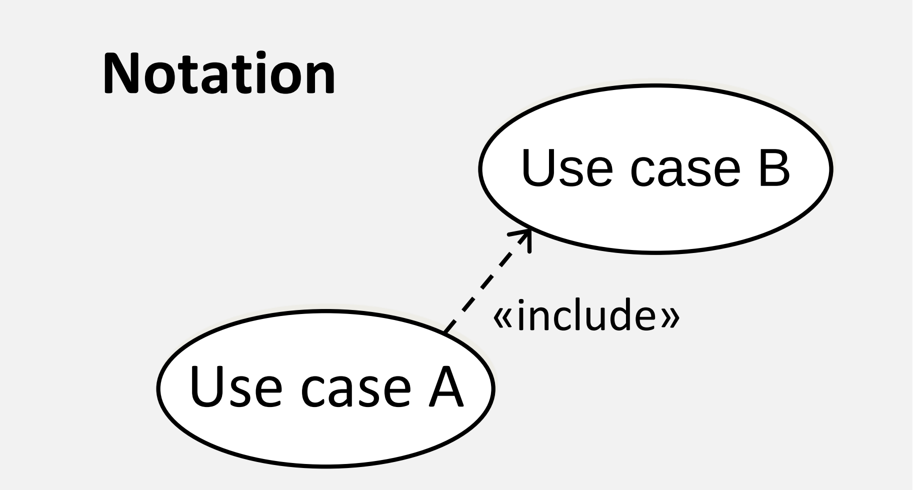
- The extend relationship allows to extend a use case with **additional functionality**, i.e. the corresponding sequence of interactions is conditionally executed.
- **Extending** use cases can be used **in different extended** use cases.
- An extend relationship consists of a **condition** and a reference to an **extension point** within the extended use case.

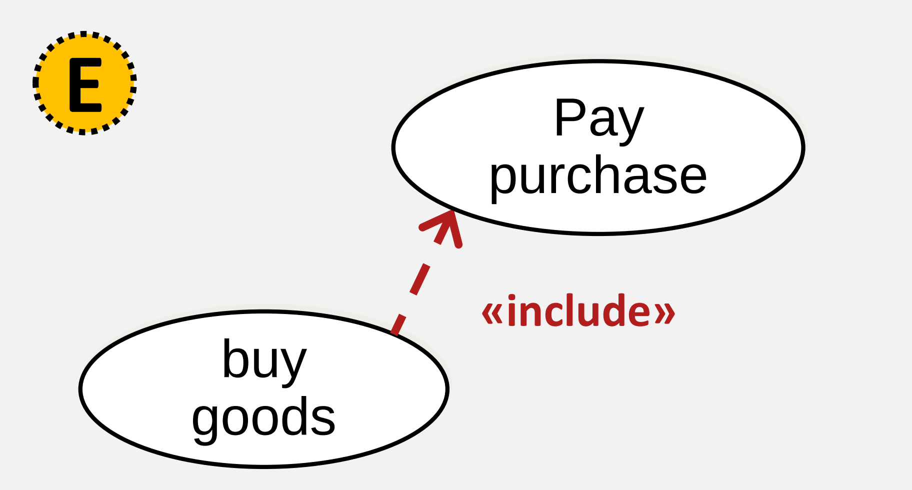




Modelling Construct: Include Relationship

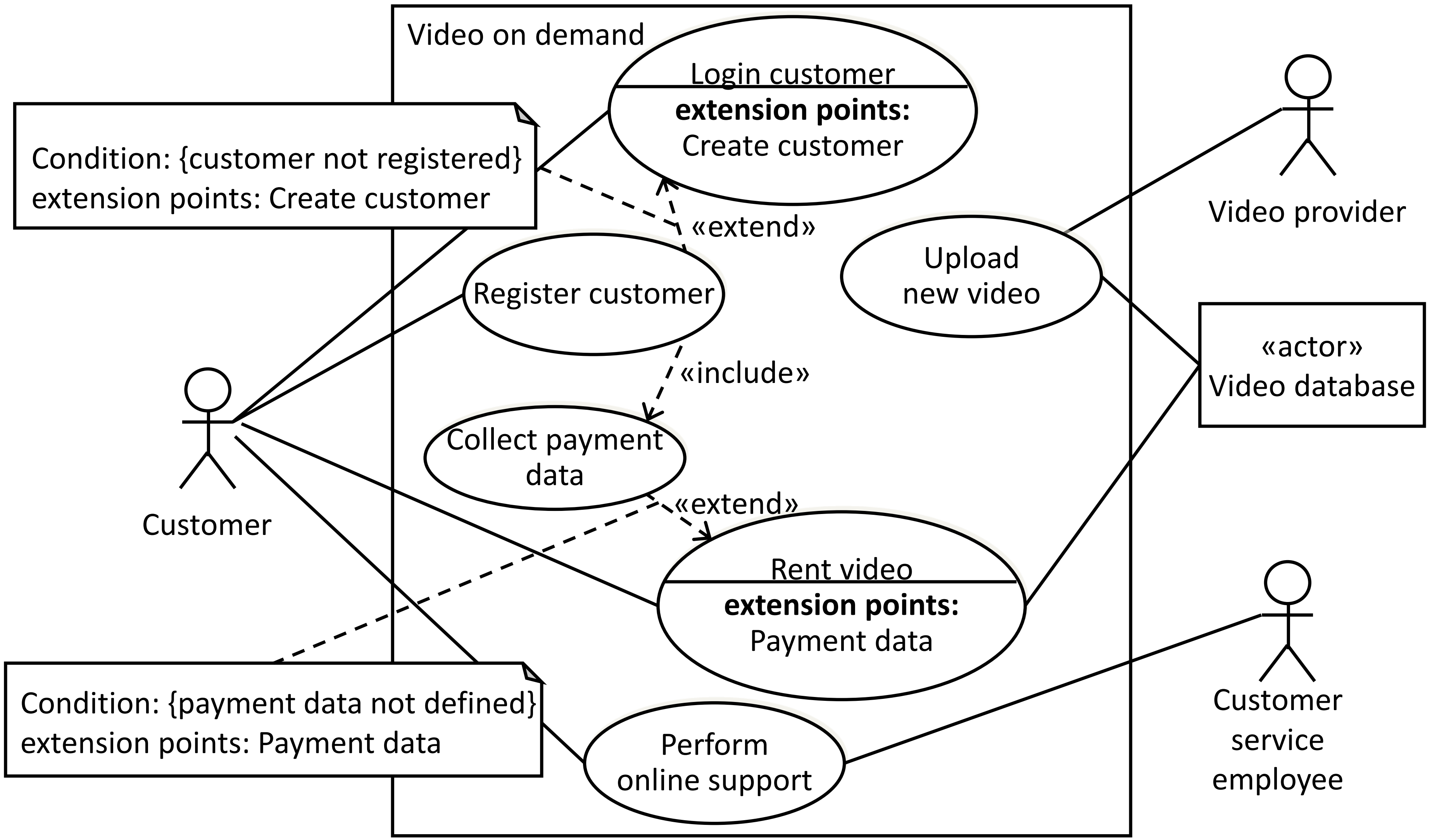
- An include relationship expresses that **one (included) use case** is **executed when another use case is executed**.
- Use case A **includes in any case** the behaviour of use case B.
- Include relationships are typically used when **a sequence of interactions is part of more than one use cases** of the system.



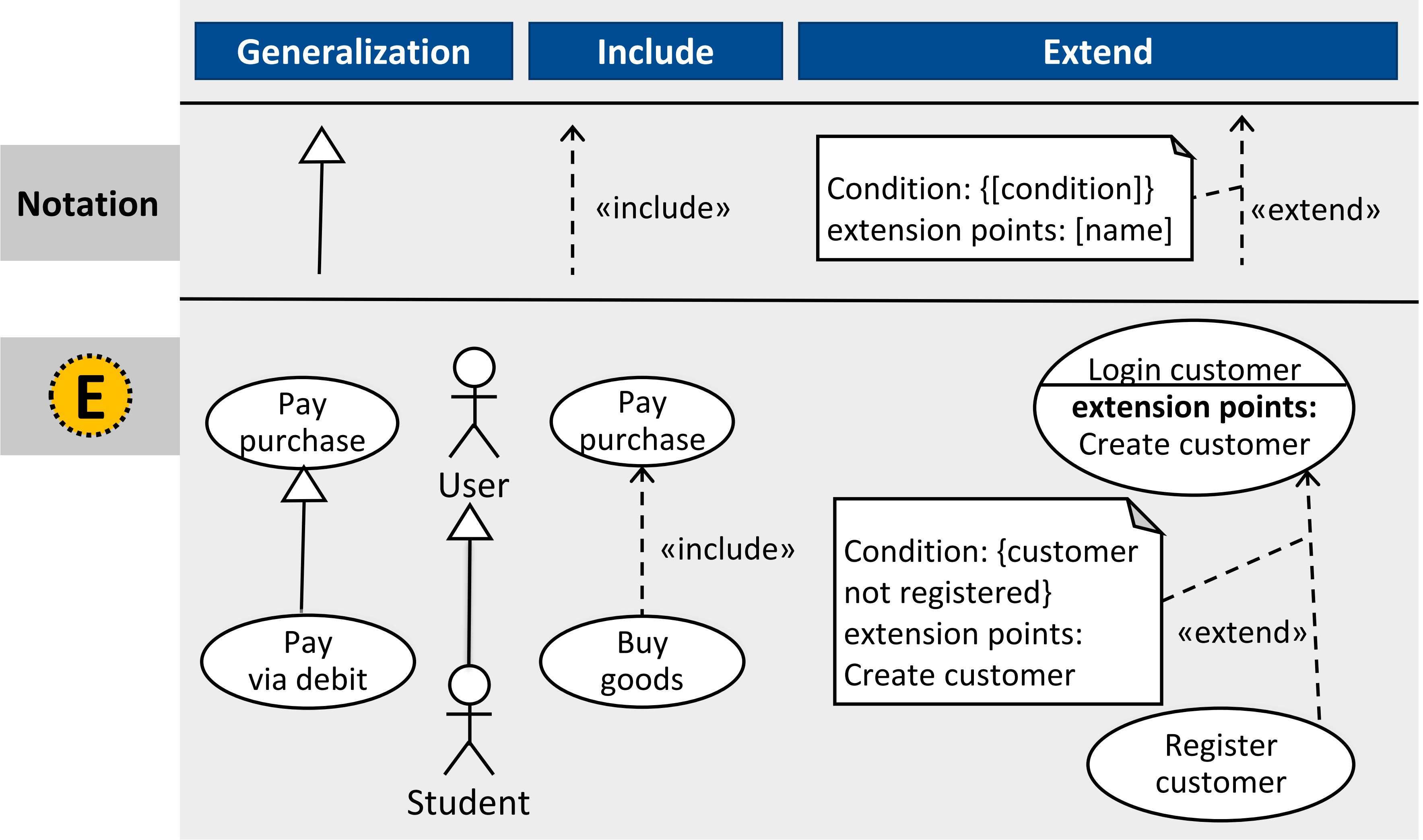


Example of a Use Case Diagram

E



Overview on Introduced Modelling Constructs (Cont.)



Generalization

Include

Extend

Notation

«include»

Condition: {[condition]}
extension points: [name]

«extend»

E

Pay purchase

Pay via debit

User

Student

Pay purchase

«include»

Buy goods

Login customer
extension points:
Create customer

Condition: {customer not registered}
extension points: Create customer

«extend»

Register customer

3. Specifying Use Cases with Use Case Templates

Use Case Templates - Overview

- A **template** in general:
  - is based on **expert knowledge**.
  - defines **relevant** types of **information**.
  - **structures** the information.
- **Use case templates**:
  - provide **detailed descriptions** for the use cases defined in a use case diagram.
  - Use case templates shall be **specifically designed** for each company’s or project specific purposes.
- Common use case **reference templates** are a good **starting point** for individual, project-specific customization.

Use Case Templates - Categories

- Use case templates provide slots to define information in different categories.
- Commonly used **categories** are:
  - Use case **management** information
  - Use case **diagram** information
  - **Contextual** information
  - **Scenario** information
  - **References** to other models and diagrams
- Use case templates can be defined on different **levels of detail**:
  - **High-level** use case templates
  - **Detailed** use case templates

Specifying Use Cases with Use Case Templates

High-Level Use Case Template

No.		Section	Content / Explanation
ID	1.2	Name	Unique name for the use case.
Management	2.1	Author(s)	Name of the authors of the use case description.
Context	3.1	Source	Source from where the use case stems.
Use Case Definition	4.2	Short description	Concise description of the use case.
	4.4	Goal(s)	Goal(s) that shall be satisfied by executing the use case.
	4.5/ 4.6	Actor(s)	Enumeration of all actors involved in the use case.
	4.7	Pre-condition(s)	Prerequisites to be fulfilled before use case execution.
	4.8	Post-condition(s)	A list of conditions that hold after execution of the use case.
Relationships	5.2	Relationship to other use cases	Short description of the relations to other use cases (apply only if these relationships are not documented by a use case diagram.

Summary

- A use case specifies a complete sequence of system–user interactions related to a part of the system’s functionality.
- A use case connects scenarios related to the same set of goals and thereby structures all possible interactions sequences to reduce complexity.
- A use case groups main, alternative and exception scenarios.
- A use case diagram represents a part of a use case and focuses on relations between use cases and use cases with actors.
- Use case templates are based on expert knowledge and define and structure relevant information.

Literature

[Caroll 2000] J.M. Caroll (Ed.): Making Use – Scenario-Based Design of Human Computer Interactions. MIT Press, Cambridge, 2000.

[Cockburn 2001] A. Cockburn: Writing effective Use Cases. Addison-Wesley, Bosten, 2001.

[Jacobson et al. 1992] I. Jacobson, M. Christerson, P. Jonsson, G. Oevergaard: Object-Oriented Software Engineering – A Use Case Driven Approach. Addison-Wesley, Reading, 1992.

[Rumbaugh et al. 2005] J. Rumbaugh, I. Jacobson, G. Booch: The Unified Language Reference Manual. 2nd edition, Addison-Wesley, Boston, 2005.

[Weidenhaupt et al. 1998] K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer: Scenario Usage in System Development – A Report on Current Practice. IEEE Software, Vol. 15, No. 2, IEEE Press, Los Alamitos, pp. 22-45.

Literature for Further Reading

[Alexander and Maiden 2004] I. Alexander, N. Maiden (Eds.): Scenarios, Stories, Use Cases – Through the system Development Life-Cycle. Wiley, Chichester, 2004.

[Haumer et al. 1998] P. Haumer, K. Pohl, K. Weidenhaupt: Requirements Elicitation and Validation with Real World Scenes. IEEE Transactions on Software Engineering, Vol. 24, No. 12, 1998, pp. 1036-1054.

[Rolland et al. 1998a] C. Rolland, C. Ben Achour, C. Cauvet, J. Raylt, A. Sutcliffe, N. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, P. Heymans: A Proposal for a Scenario Classification Framework. Requirements Engineering Journal, Vol. 3, No. 1, Springer, Berlin, Heidelberg, 1998, pp. 23-47.

[Rolland et al. 1998b] C. Rolland, C. Souveyet, C. Ben Achour: Guiding Goal Modelling Using Scenarios. IEEE Transactions on Software Engineering, Vol. 24, No 1., 1998, pp. 1055-1071.

[Sutcliffe et al. 1998] A. Sutcliffe, N. Maiden, S. Minocha, M. Darrel: Supporting scenario-Based Requirements Engineering. IEEE Transactions on Software Engineering, Vol. 24, No. 12, 1998, pp. 1072-1088.

Image References

[1] Licensed by http://www.iconshock.com/

[2] Provided by Microsoft Office

Legend

D Definition

E Example

Requirements Engineering & Management

Vielen Dank für Ihre Aufmerksamkeit