Dr. Michael Eichberg
Software Engineering
Department of Computer Science
Technische Universität Darmstadt

Software Engineering

Requirements Engineering

- The following slides are primarily based on the contents of the following books:
  - Writing Effective Use Cases;
    Alistair Cockburn; Addison-Wesley, 2001
  - Applying UML and Patterns;
    Craig Larman; Markt und Technik, 2005
  - UML Distilled, Third Edition;
    Martin Fowler; Addison-Wesley, 2004

TECHNISCHE UNIVERSITÄT DARMSTADT

Requirements Engineering

- Use Cases

TECHNISCHE UNIVERSITÄT DARMSTADT

A use case is a set of scenarios tied together by a common user goal.

Examples are:

(a) “sign up for the exam”,

(b) “make a bank transfer”.

Use cases can not be used for all kinds of requirements!

Use cases are text stories, widely used to discover and record requirements.

A use case encapsulates a set of actions that are executed in a well defined order.

Use cases influence many other artifacts, e.g. analysis, design, implementation and test artifacts.

A Point of Sale (POS) System

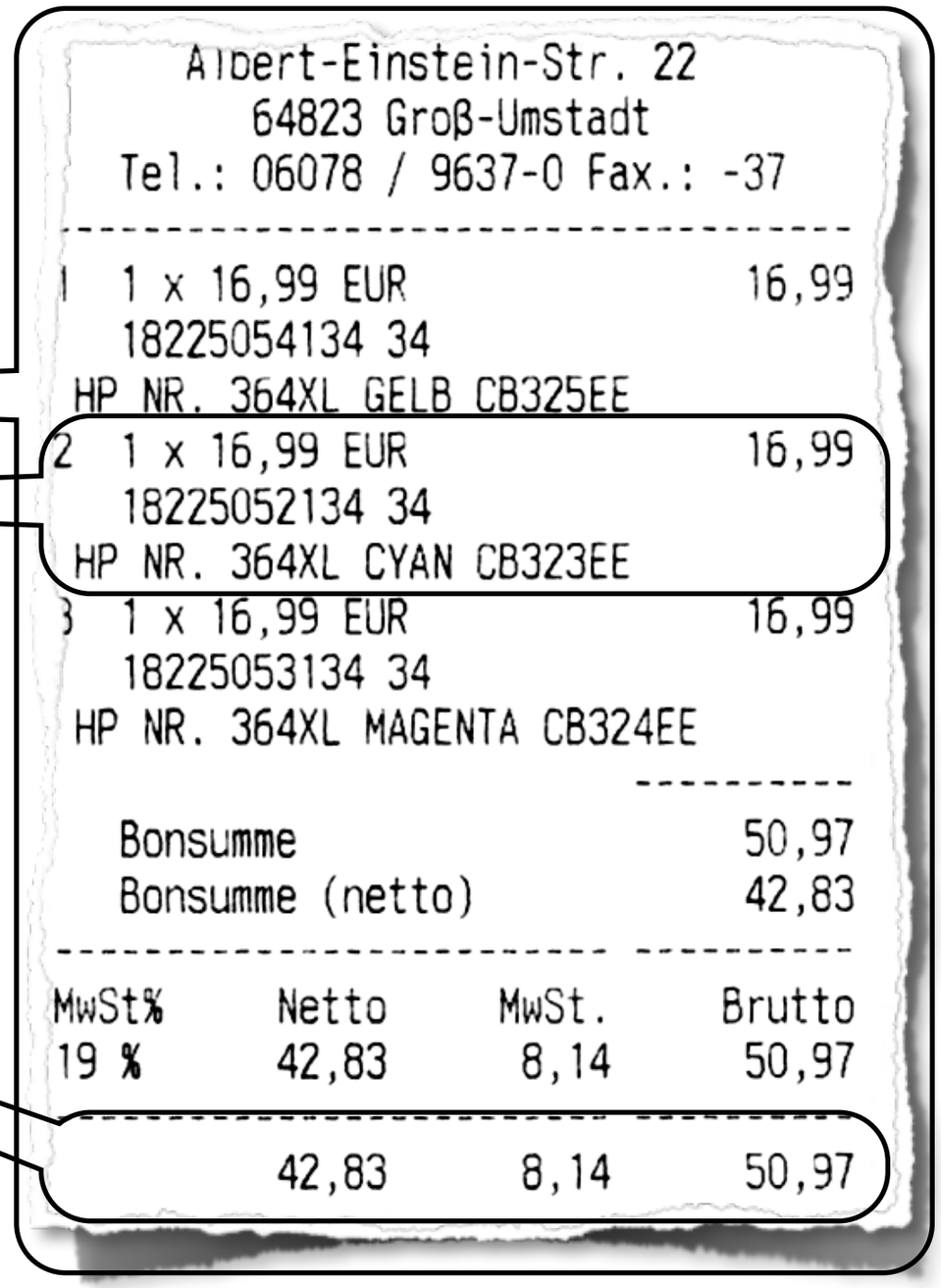
The running example scenario.

(Point of sale system (POS System) =dt. Kassensystem)

- POS systems are typically used in retail stores to record sales and to handle payments
- POS systems interface to various service applications to calculate taxes and for inventory control
- POS systems include computers, bar code scanners and software; the set of clients should vary, e.g. from thin-client web browser terminals, or rich client applications

Point of Sale Systems
Domain Terminology

- Sale - the exchange of a commodity for money
  =dt. Verkauf
- Receipt =dt. Beleg
- (Sales) Line Item =dt. Einzelposten / Belegposition
- Payment =dt. (Be)Zahlung / Vergütung
- Customer =dt. Kundin
- Cashier =dt. Kassiererin



Use cases are text stories, widely used to discover and record requirements.

Process Sale:

A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

Use Cases - some Definitions

- **Actor**

  … something with behavior; e.g. a person, computer system or organization.

- **Use case**

  … is a collection of related success and failure scenarios that describe an actor using a system to support a goal.

- **Scenario**

  [Also known as a “use case instance”.]
  … a specific sequence of actions and interactions between actors and the system. It is one particular story using a system, or one path through the use cases.

Use Case Formats

- **Brief**

  (dt. kurz)

  Terse one-paragraph summary, usually of the main success scenario (as seen above).

- **Casual**

  (dt. ungezwungen, zwanglos)

  Informal paragraph format. Multiple paragraphs that cover various scenarios.

- **Fully dressed**

  (dt. vollständig bearbeitet)

  All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.

Focus on the accuracy of use cases before you focus on the precision.

accuracy =dt. Korrektheit

precision =dt. Genauigkeit (~hier Detailgrad)

- … identify all currently relevant use cases at a very high level (low precision, high accuracy)
- … work out the details (add precision)

A Template for Fully Dressed Use Cases

(Created by Alistair Cockburn, alistair.cockburn.us)

Use Case Section	Purpose / Guidelines
Use Case Name	Start with a verb.
Scope	The system under design.
Level	“summary goals” → “user goals” → “subfunction”
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	*Who cares about this use case, and what do they want?*
Preconditions	*What must be true on start, and worth telling the reader?*
Minimal Guarantees	*The fewest promises the system makes to the stakeholders.*
Success Guarantee	*What must be true on successful completion, and worth telling the reader?*
Main Success Scenario	A typical path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variation List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing and timing of implementation.
Miscellaneous	Such as open issues.

A Template for Fully Dressed Use Cases

(Created by Alistair Cockburn, alistair.cockburn.us)

Use Case Section	Explanation
Stakeholders and Interests	Someone or something with an interest in the behavior of the system under discussion. E.g. company stakeholders, customers, vendors, and government regulatory agencies,…
Preconditions	It announces what the system will ensure is true before letting the use case start. Since it is enforced by the system and known to be true, it will not be checked again during the use case execution; e.g. user has logged in.
Minimal Guarantees	The fewest promises the system makes to the stakeholders, particularly when the primary actor's goal cannot be delivered. They hold when the goal is delivered, but they are more important when the main goal is abandoned. E.g. the system logged all performed steps.
Success Guarantee	States what interests of the stakeholders are satisfied after a successful conclusion of the use case, either at the end of the main success scenario or at the end of a successful alternative path. The success guarantees are written additionally to the minimal guarantees.

Excerpt of a Fully Dressed Use Cases

Name: Buy Stocks over the Web

Primary Actor: Purchaser

Scope: Finance Package (PAF)

Level: User goal

Stakeholders and Interests:

Purchaser - wants to buy stocks and get them added to the portfolio.

Stock agency - wants full purchase information.

Precondition:

User is logged in.

Minimal guarantee:

Sufficient logging information will exist so that the PAF can detect that something went wrong and ask the user to provide details.

Success guarantee:

Web site has acknowledged the purchase; the logs and the user's portfolio are updated.

Excerpt of a Fully Dressed Use Cases

Name: Buy Stocks over the Web

Scope: Finance Package (PAF)

...

Main Success Scenario:

1. Purchaser selects to buy stocks over the web
2. PAF gets name of web site to use (A, B,…) from user
3. PAF opens web connection to site, retaining control
4. Purchaser browses and buys stock from the web site
5. PAF intercepts responses from the web site and updates purchaser's portfolio
6. PAF shows the user the new portfolio standing

Excerpt of a Fully Dressed Use Cases

Name: Buy Stocks over the Web

Scope: Finance Package (PAF)

...

Extensions:

2a. Purchaser wants a web site PAF does not support

2a1. System gets new suggestion from purchaser, with option to cancel

4a. Web site does not acknowledge purchase, but puts it on delay

4a1. PAF logs the delay, sets a timer to ask the purchaser about the outcome

...

Guidelines for Use Cases

- During early requirements work keep the user interface out
  (Focus on intent!)
- Write terse use cases
- Write black-box use cases
  I.e. describe what the system must do and not how.
  E.g. write:
  "The system records the sale."
  and do not write:
  "The system writes the sale to a database."
- Take an actor or actor-goal perspective
  Focus on the users or actors of a system, asking about their goals and typical situations; focus on understanding what the actor considers a valuable result

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- Negotiate a supplier contract
- Handle returns
- Log in
- Move piece on game Board

?

Some Rules of Thumb for Finding Use Cases

- Does it achieve results of measurable value (VAL) w.r.t. the business?
- Is it a task performed by one person in one place at one time, in response to a a business event, which adds measurable business value and leaves the data in a consistent state?
  Elementary Business Process (EBP)
- Is it just a single step (Size)?

Which of these is a valid use case?

- Negotiate a supplier contract
- Handle returns
- Log in
- Move piece on game Board

?

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- Negotiate a supplier contract
- Handle returns
- Log in
- Move piece on game Board

?

Much broader and longer than an EBP.

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- ~~Negotiate a supplier~~
- Handle returns
- Log in
- Move piece on game Board

Achieves value; seems like an EBP; size is ok.

?

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- ~~Negotiate a supplier contract~~
- Handle returns
- Log in
- Move piece on game Board

?

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- ~~Negotiate a supplier contract~~
- Handle returns
- ~~Log in~~
- Move piece on game Board

?

Does not achieve value.

Finding use cases during the initial requirements analysis

Which of these is a valid use case?

- ~~Negotiate a supplier contract~~
- Handle returns
- ~~Log in~~
- Move piece on game Board

This is just a single step; fails “size test”.

?

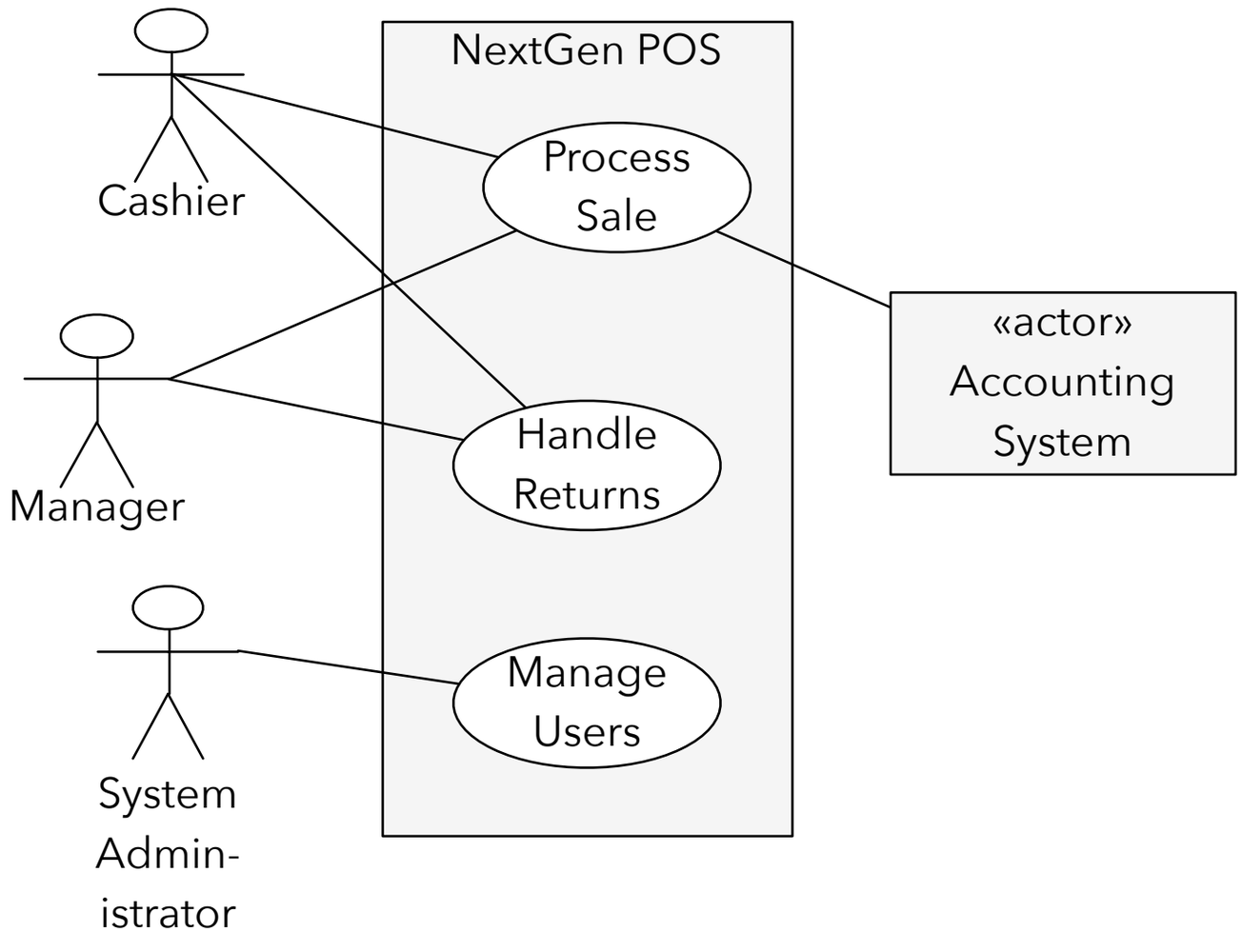
Finding use cases during the initial requirements analysis

Which of these is a valid use case?

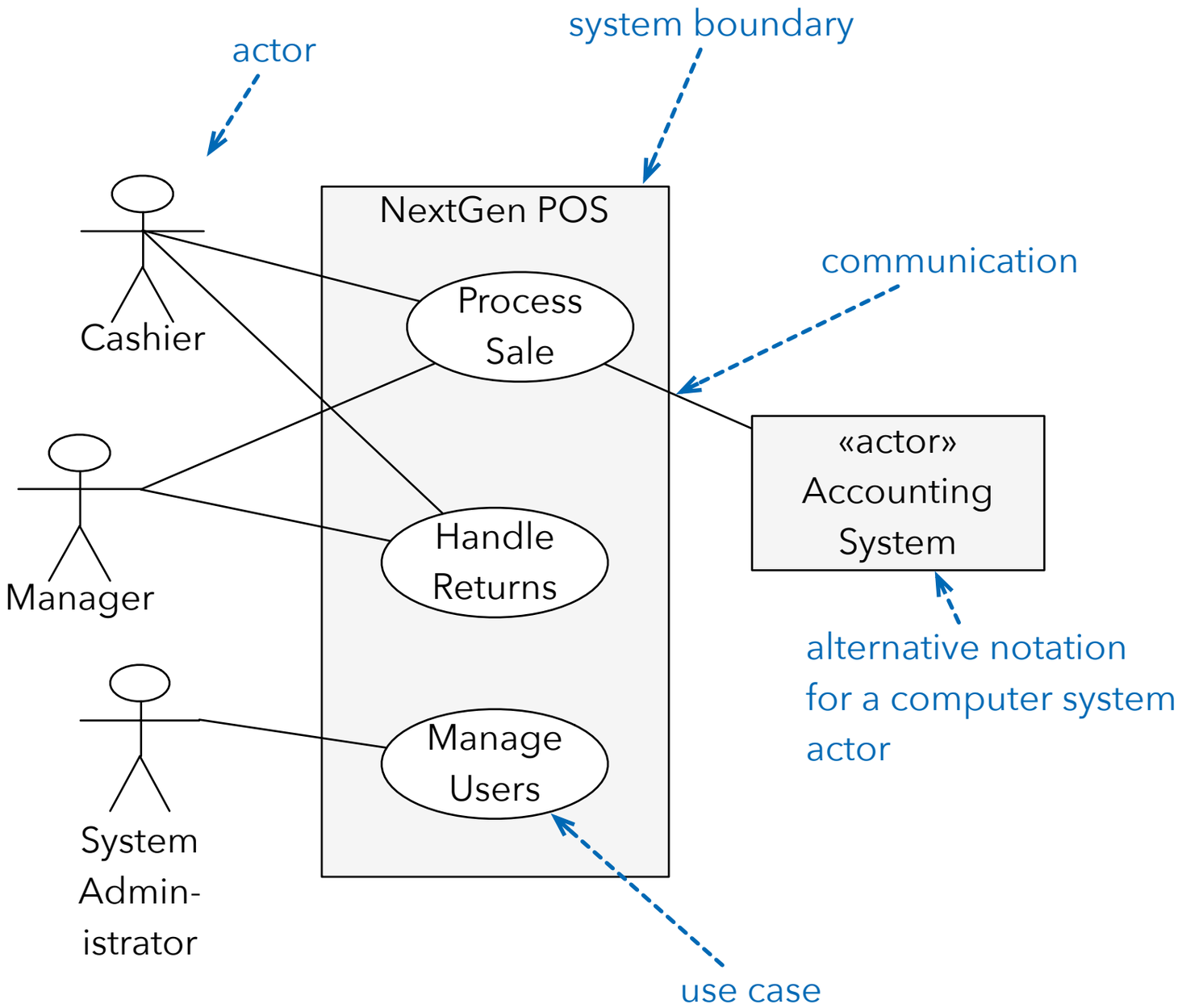
- ~~Negotiate a supplier contract~~
- Handle returns
- ~~Log in~~
- ~~Move piece on game Board~~

UML use case diagrams provide a notation to illustrate the names of use cases and actors (roles), and the relationship between them. (To depict the context.)

(Use-case diagram = dt. Anwendungsfall-Diagramm)
(Point-of-Sale System (POS) = dt. Kassensystem)



UML use case diagrams provide a notation to illustrate the names of use cases and actors, and the relationship between them.



On Use Cases

- The UML use case diagram is trivial to learn.

- UML use case diagrams are an organization method to improve communication and comprehension of the use cases and to reduce duplication of text. Organizing use cases into relationships has no impact on the behavior or requirements of the system.

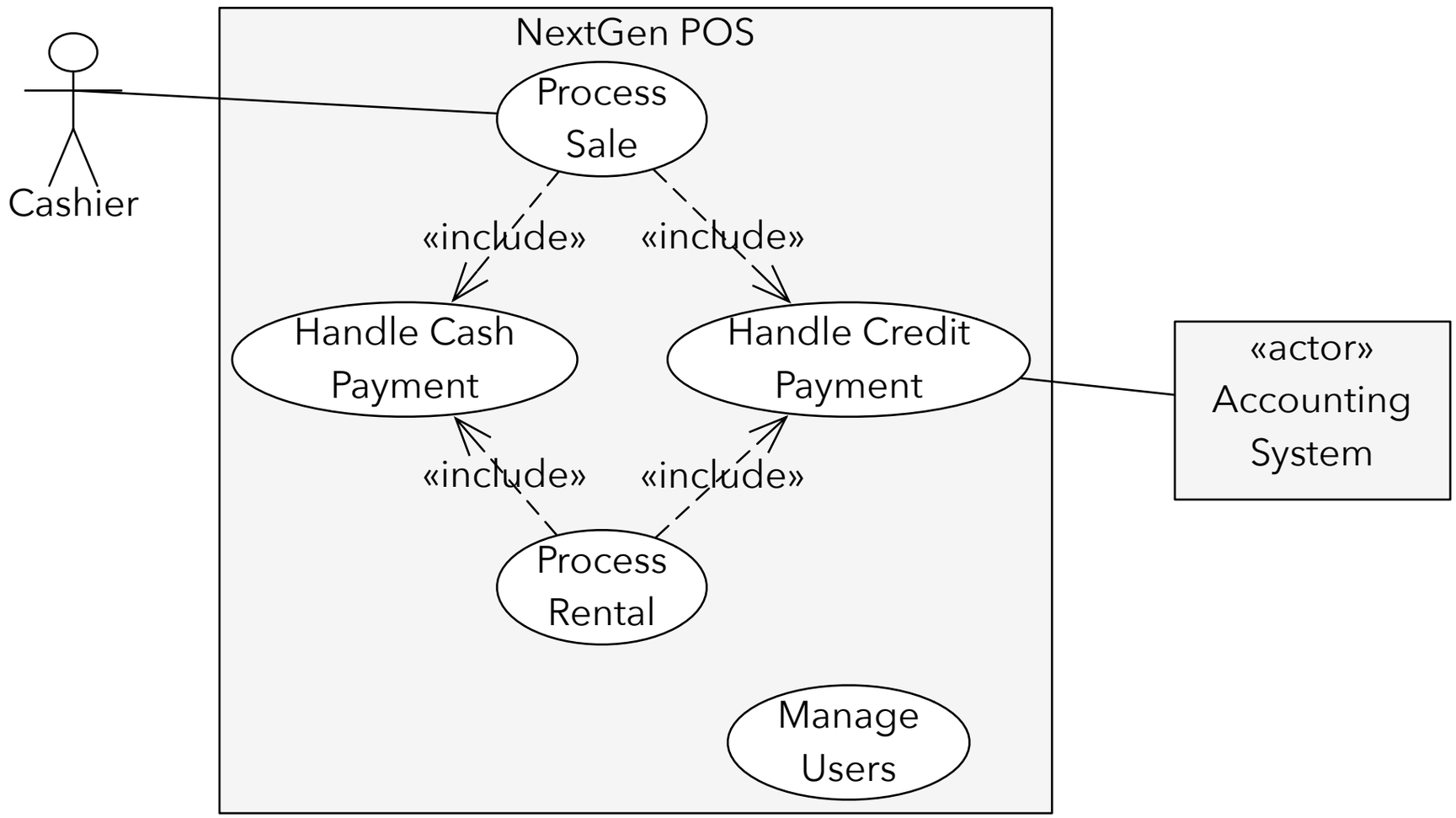
Identifying and writing good use cases requires practice.

UML use case diagram provide a black-box view on a system.

They are particularly useful during the early phases of a software project.

The Include Relationship in UML Use Case Diagrams

For partial behavior that is common across several use cases (e.g. “Pay by Credit” occurs in “Process Sale”, “Process Rental”,…) it is desirable to separate it into its own sub-function use case and indicate its inclusion.

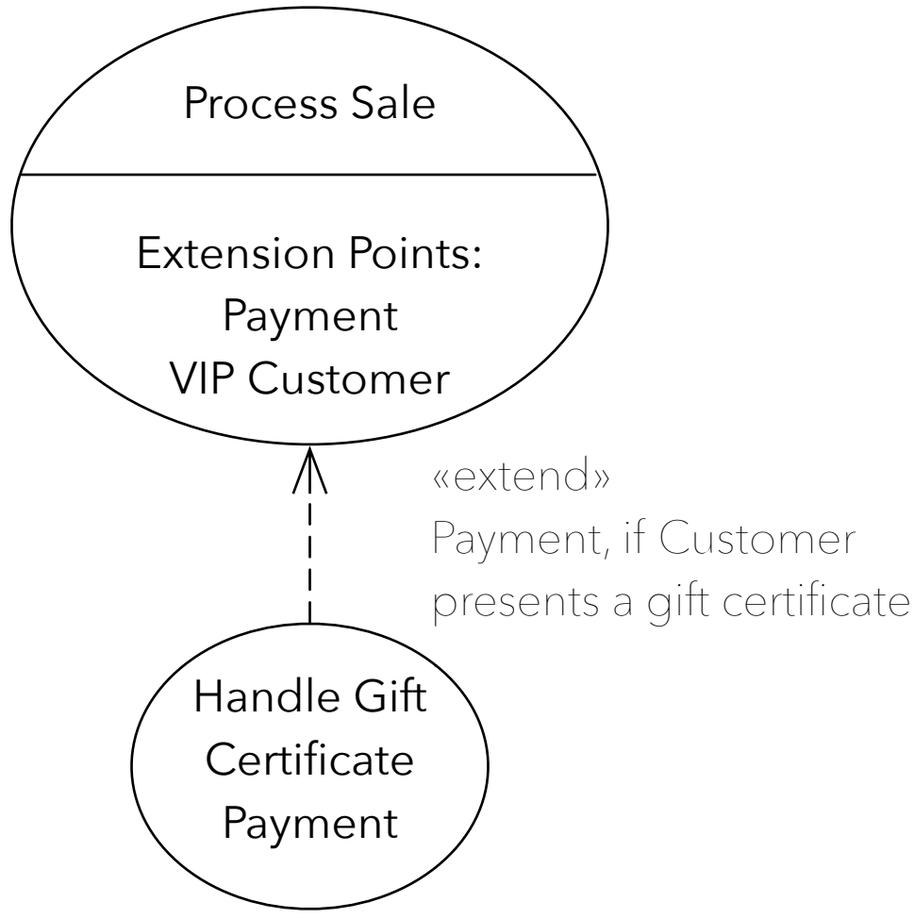


The primary purpose is to avoid repetition or to decompose extremely long use cases to make them comprehensible.

The Extend Relationship in UML Use Case Diagrams

The extend relationship can be used to describe where and under what condition an extending or additional use case extends the behavior of some base use case.

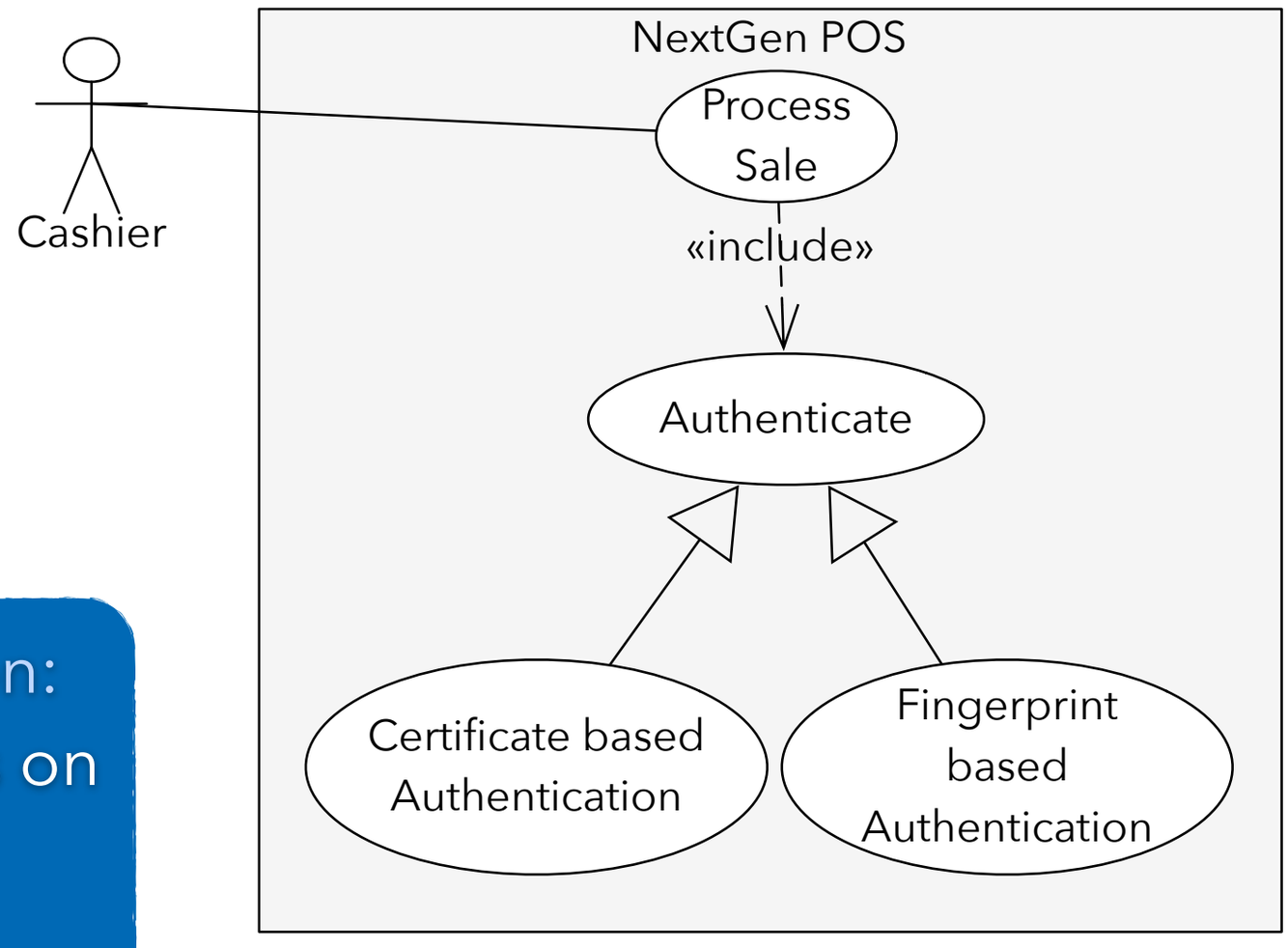
Recommendation:
Focus on the textual description. (They are rarely used in practice.)



The Inheritance Relationship in Use Case Diagrams

The inheriting use case replaces one or more of the courses of action of the inherited use case.

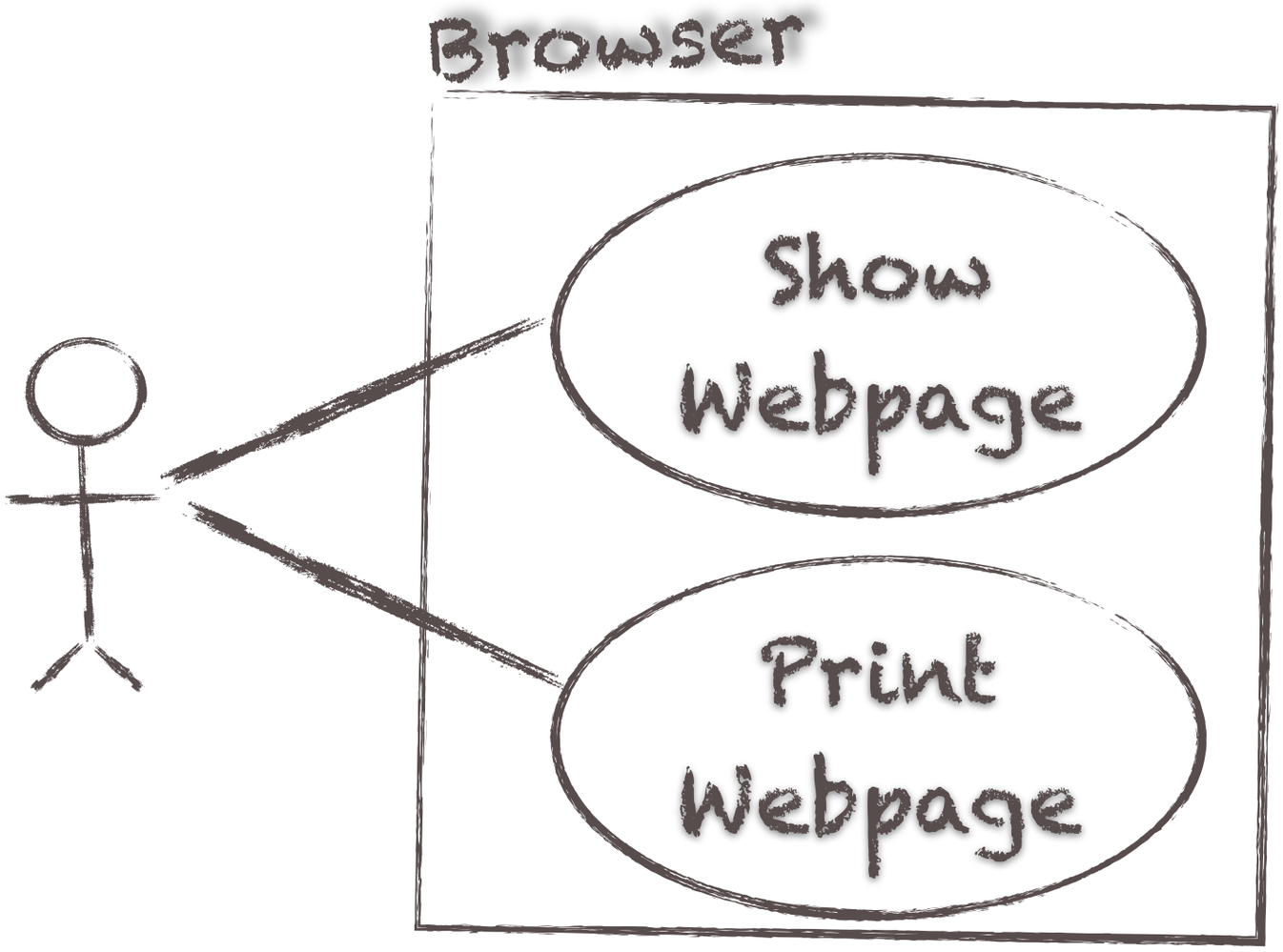
The inheriting use case overrides the behavior of the inherited use case.



Recommendation:
Don't use it; focus on the textual description.

Inheritance between use cases is not very common.

Think Agile



Do not draw UML Use Case Diagrams with a low value.

> ***The UML […] is just a diagramming notation.***
>
> *It's useless to learn UML and perhaps a UML CASE Tool, but not really know how to create an excellent OO design, or evaluate and improve an existing one.*

Craig Larman; 2005

Applying UML and Patterns

Use Cases Summary

- Use cases are used to capture (functional) requirements
  - Textual formats: “brief”, “casual” and “fully dressed”
  - Graphical format: UML use case diagrams

TECHNISCHE UNIVERSITÄT DARMSTADT

- The goal of this lecture is to enable you to systematically carry out small(er) commercial or open-source projects.