Supplemental Modelling:
Use Case Diagrams

Software Systems – Design – L5T1
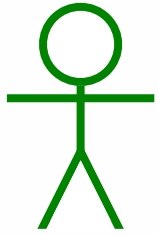
Dr. Vadim Zaytsev aka @grammarware, November 2020

UNIVERSITY OF TWENTE.

Fm Formal Methods & Tools

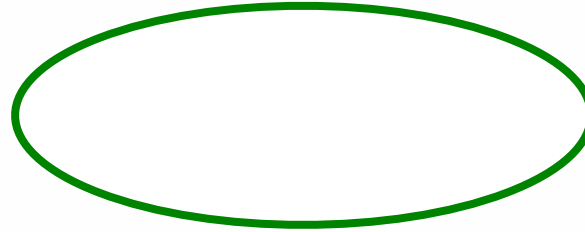# Supplemental Modelling

- Recall:
  - structural modelling is a static view
  - behavioural modelling is a dynamic view
- What's left?
  - Use Case Diagrams
  - Deployment Diagrams
  - Information Flow Diagrams
- We'll add version control

UNIVERSITY
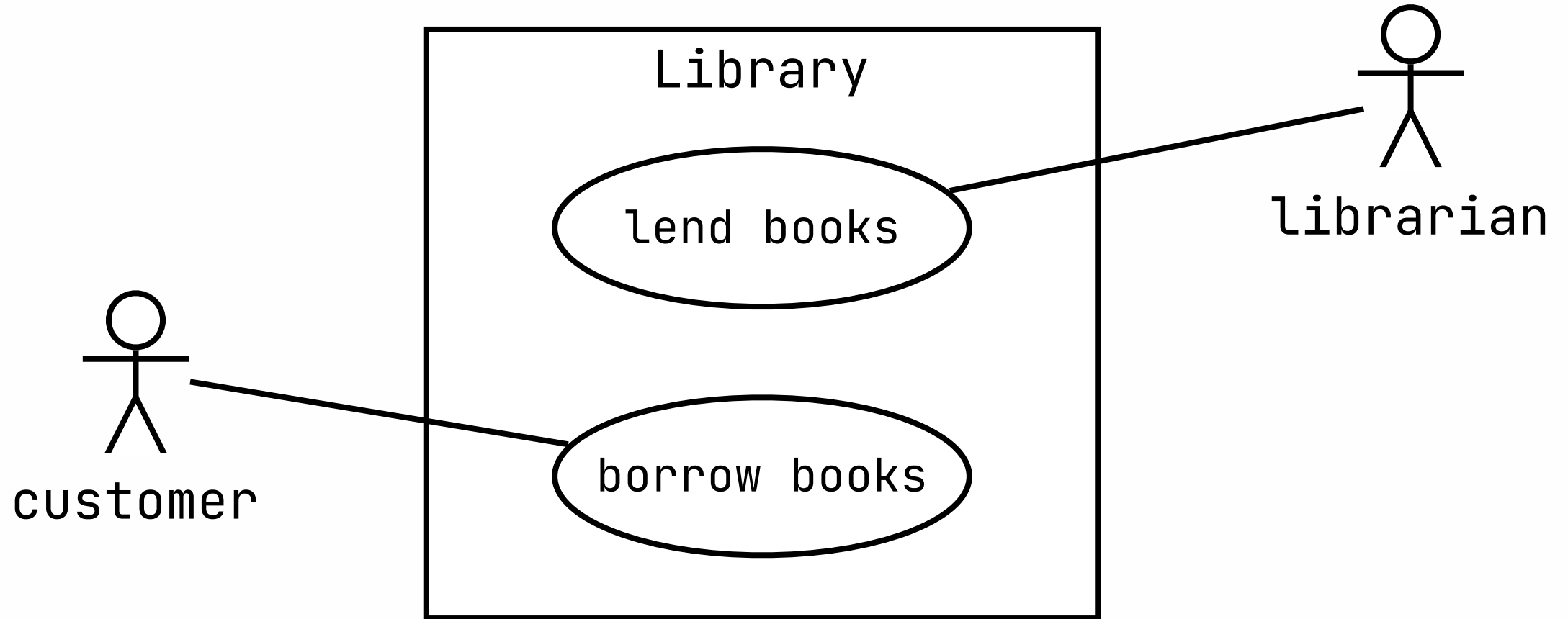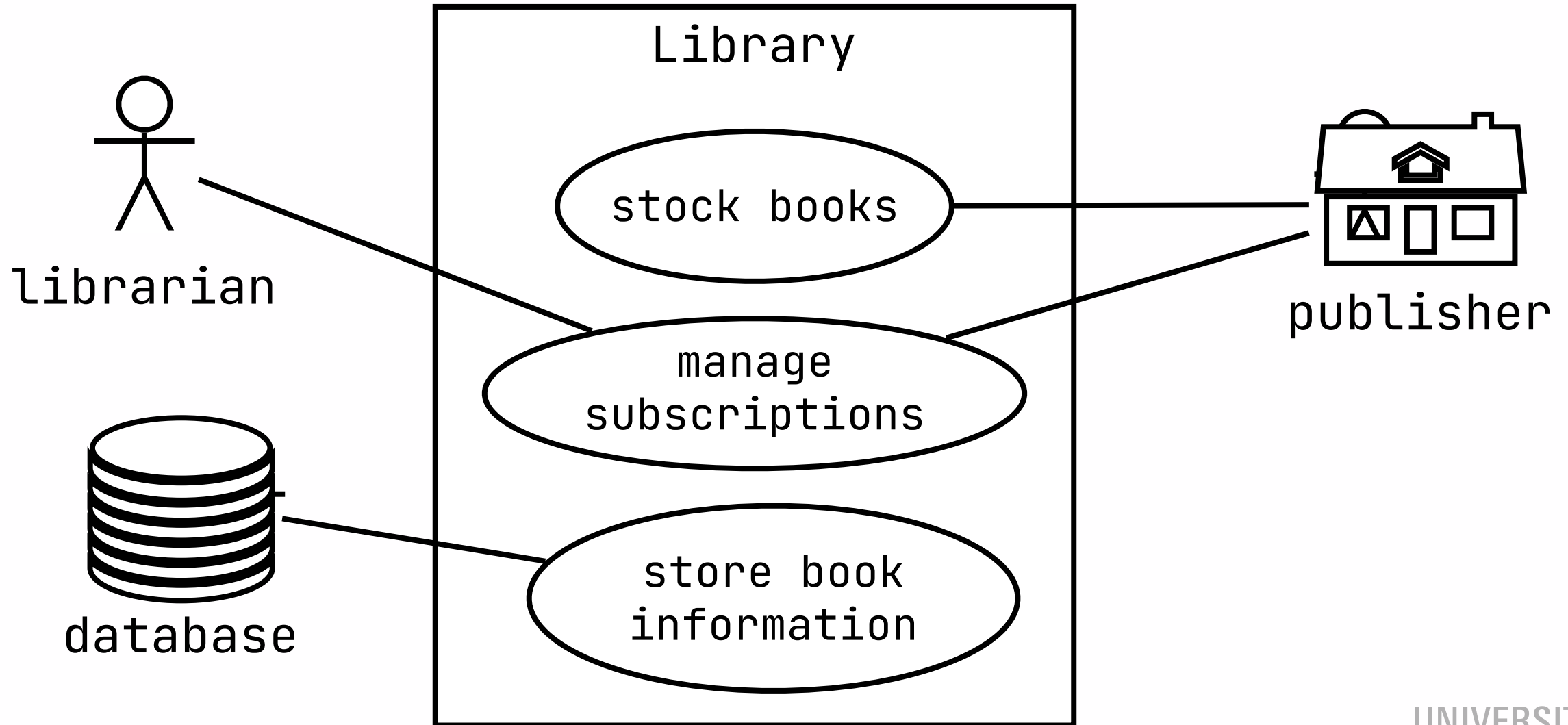OF TWENTE.

Use Case Diagrams

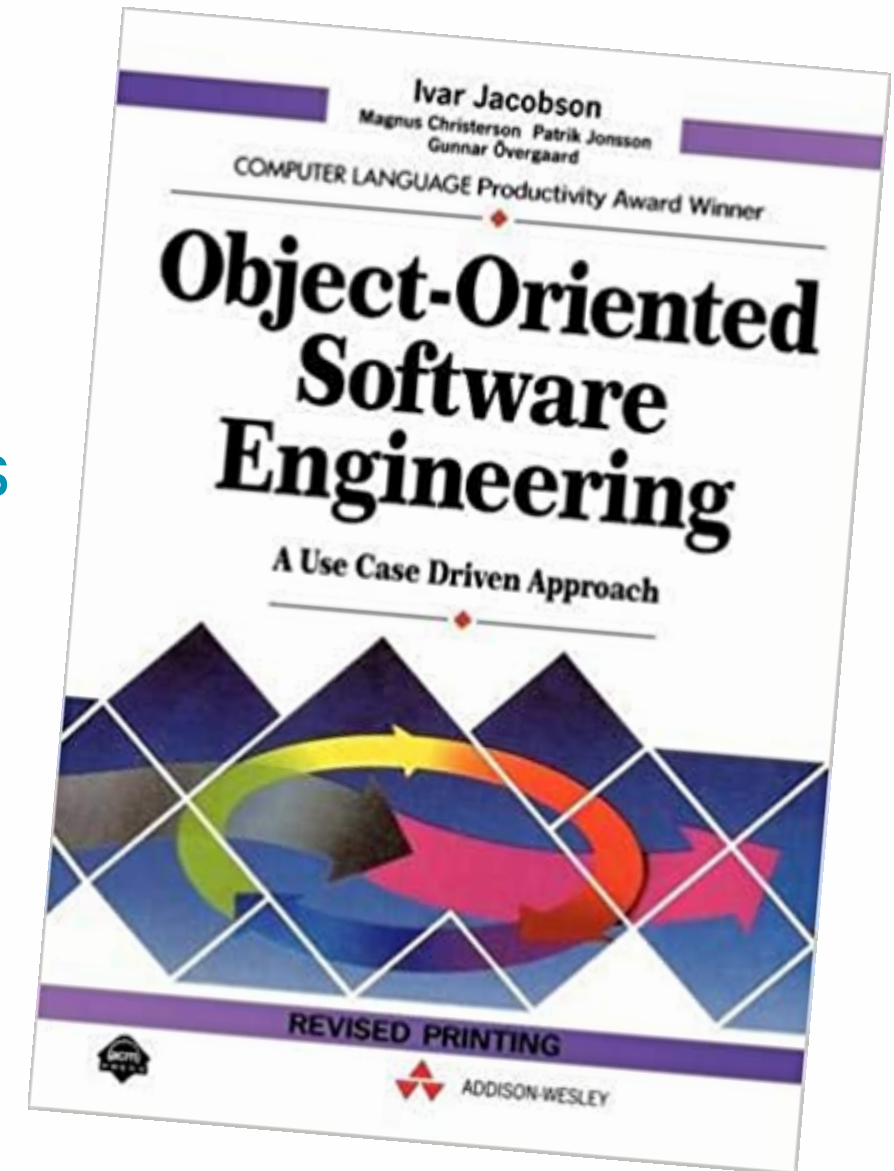actor

use case

system

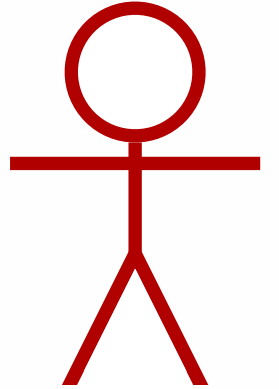# Library UCD

Library UCD – Alternative

Use Case Driven SE

- Easy to collaborate
  - with analysts and domain experts
- Concretising system architecture
- Concretising system requirements
- Getting closer to implementation

  - and test stories

Ivar Jacobson
Magnus Christerson Patrik Jonsson
Gunnar Övergaard

COMPUTER LANGUAGE Productivity Award Winner

Object-Oriented Software Engineering

A Use Case Driven Approach

REVISED PRINTING

ADDISON-WESLEY

UNIVERSITY
OF TWENTE.

Actor

- One role of a user
- Has a business role (stakeholder or system)
- Named by noun
- Interacts with a function of the system (= use case)
- Triggers use case(s)
- Provides system input(s)
- Expects system output(s)

UNIVERSITY
OF TWENTE.

Use Case

- System function
  - usually a process
  - can be manual or automated
- Patterned
  - \<verb\>
  - \<verb\> \<noun\>
- Has associations

UNIVERSITY
OF TWENTE.
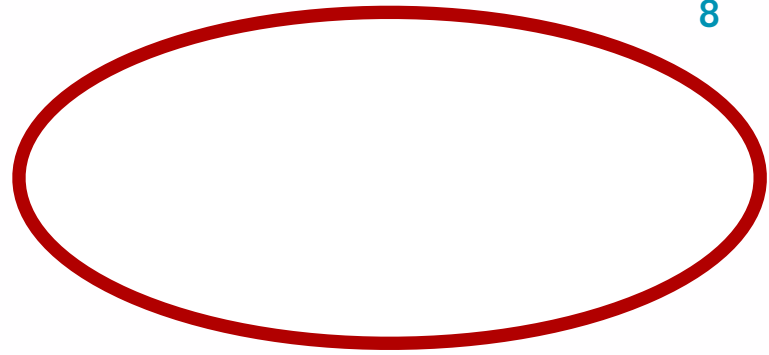
System Boundary

- • (or Subject/Business Boundary)
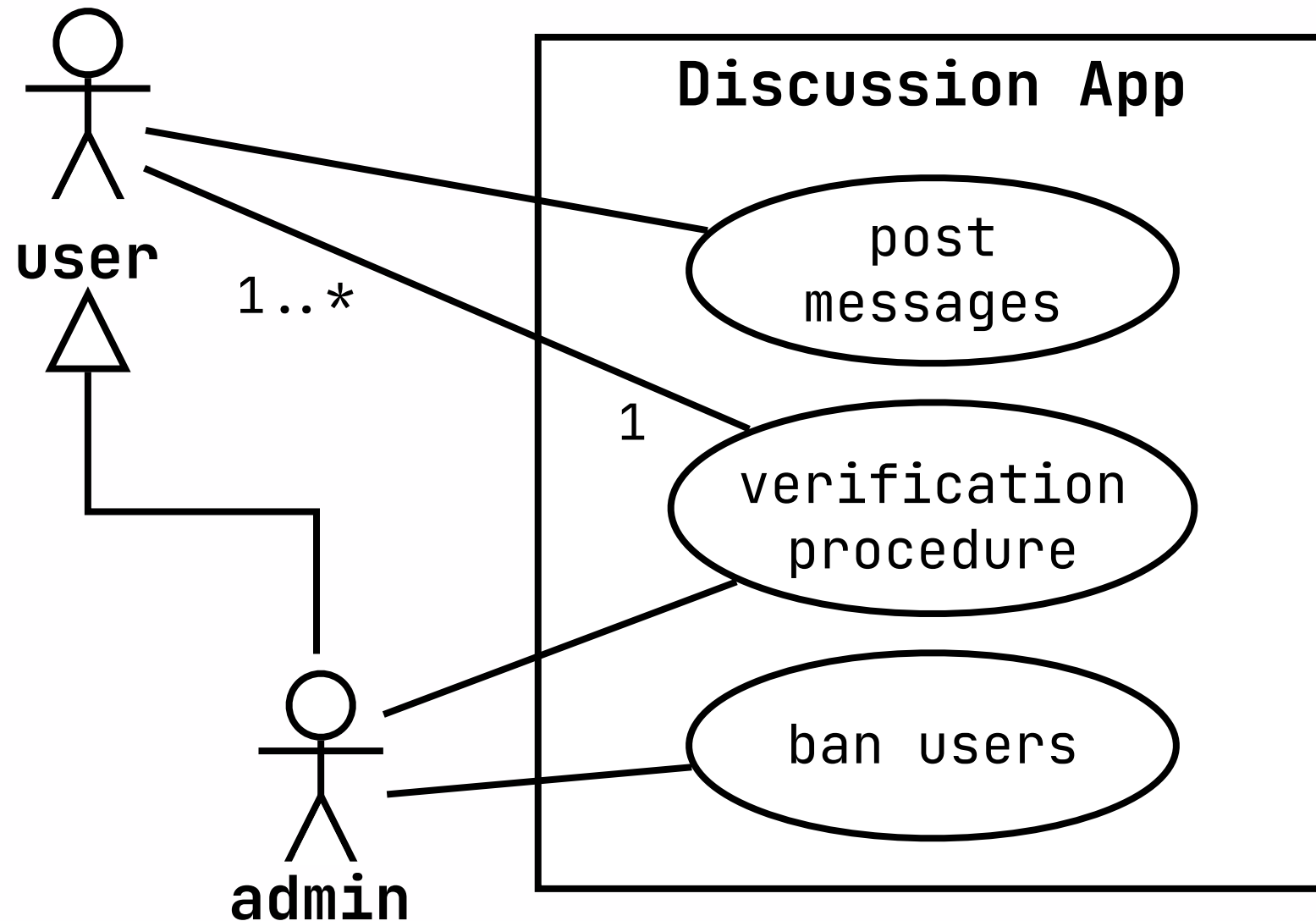
- • Often one system per diagram

- • Split & classify use cases in large complex systems

- • Can be nested

UNIVERSITY
OF TWENTE.

Link

- Communication

- Participation
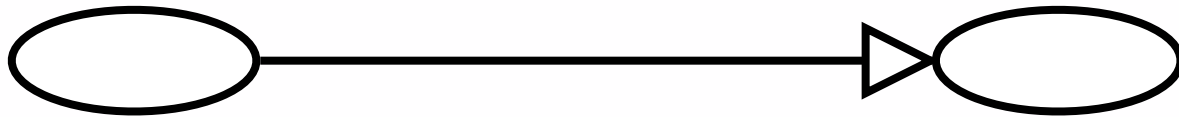
- Association

UNIVERSITY
OF TWENTE.

Use the Rest of UML!

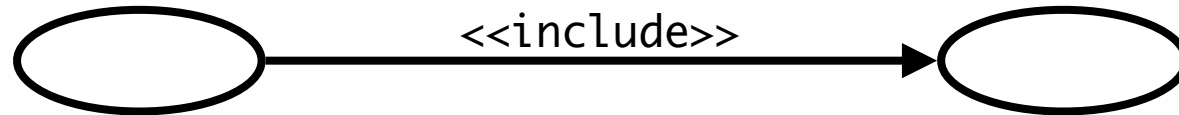# **Generalise, Include, Extend**

- Generalisation = enhancement
  - include refined use cases
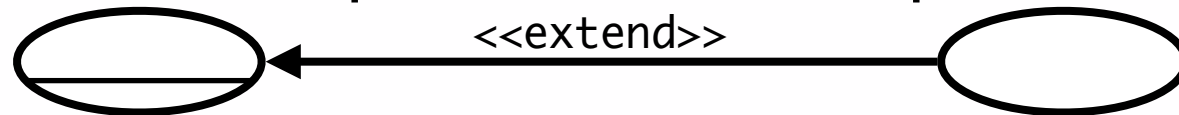

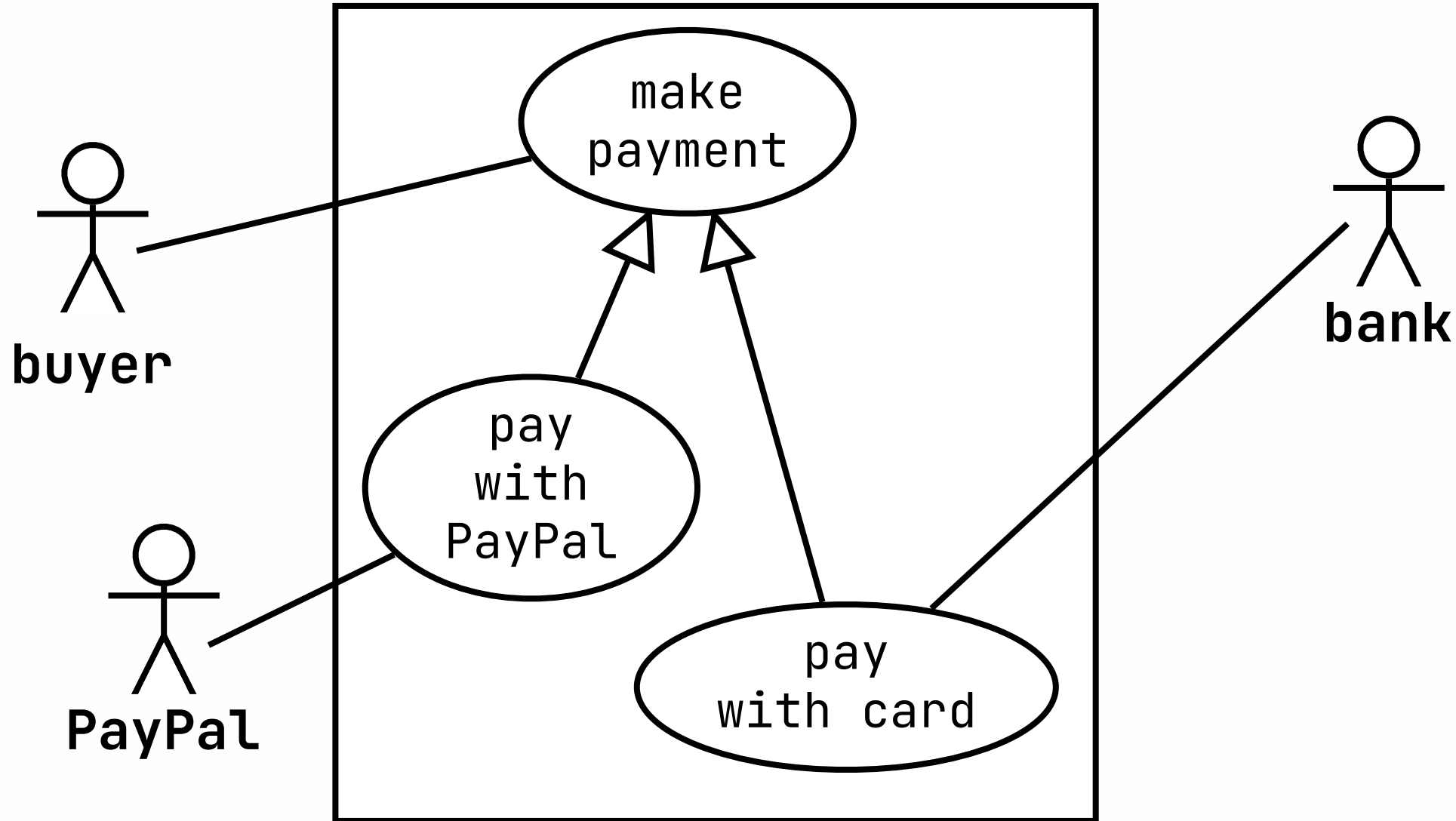- Inclusion = containment
  - functionality is included
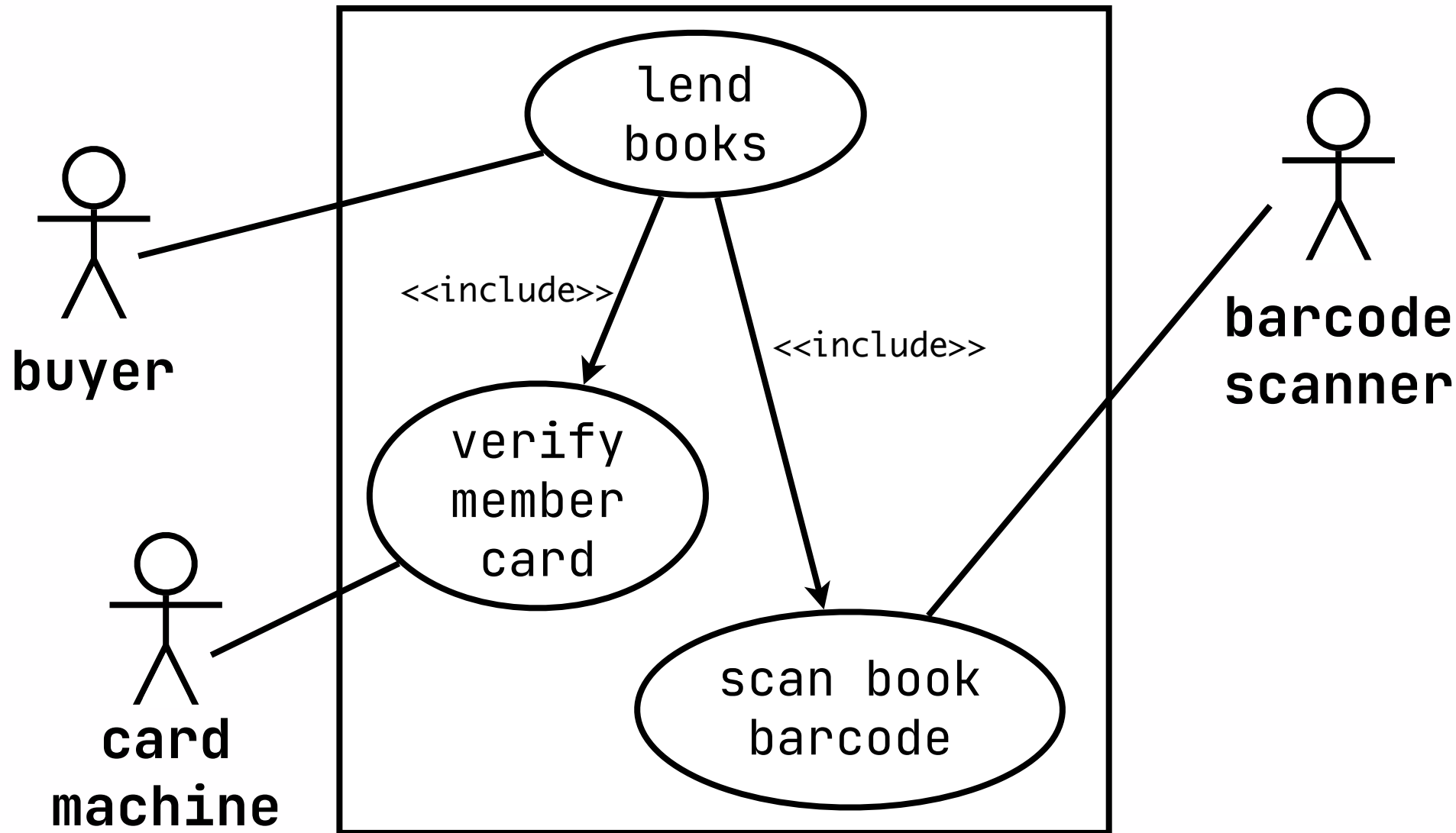
  <<include>>

- Extension = possible inclusion
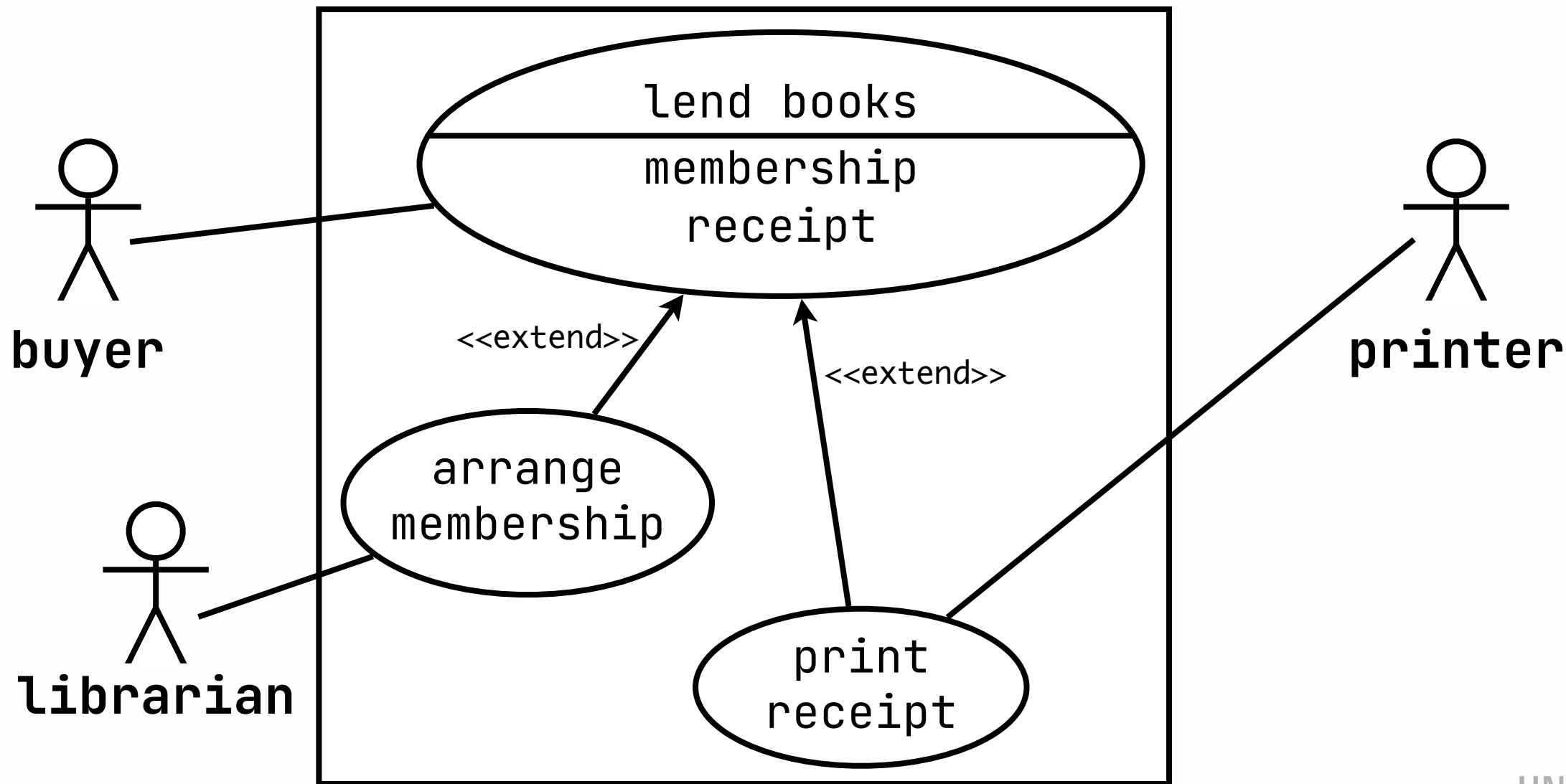 - extension points are explicit

  <<extend>>

Generalisation: "is a"

Inclusion: "part is same as"

Extension: "sometimes requires"

Conclusion

- Use Case Diagrams are supplemental
- Turn requirements into system usages
- Define what the system can do
- Specify how env and system interact
- Notation is simple
  - can be the first step
  - toward Activity Ds, Sequence Ds, …

UNIFIED
MODELING
LANGUAGE

UNIVERSITY
OF TWENTE.

Topics/Slides Disclaimer

- **Good** ✅
  - watch before Q&A
  - embrace reality
  - try out at labs
  - ask for feedback
  - apply to project
  - dig deeper
  - recall from slides

- **Bad** ❌
  - slides over videos
  - assumptions
  - blanks
  - timing

UML Unified Modeling Language

UNIVERSITY OF TWENTE.