Use-Case Modeling

CSE 3311 & 5324
Christoph Csallner
University of Texas at Arlington (UTA)

Several slides in this set are copied almost verbatim (with permission) from Dave Kung (UTA)

Main Take-Aways

- **Use-case = Text story**
- Use-case = Text story
- Use-case = Text story

- Use-case diagram
  - Can formalize some, but not all aspects of use cases
- **Primary task is to write text, not draw diagrams**

Who Wants What?

- Who uses the system and **what are their goals?**
- What is the root goal?
- Not: “What are the tasks?”

Finding Use-Cases

1. Choose the system boundary
   - Which things are outside the scope of the system?
2. Identify the primary actors
   - Those who have goals fulfilled through the system
   - Who starts, stops, administers the system?
   - Which other software systems call the system?
     - Etc., see CL, page 83
3. Identify the goals of each primary actor
   - For each primary actor, make a list of his/her/its goals
4. Define use-cases that satisfy user goals
   - Name use-cases according to user goals
     - E.g., “process sale”, see CL, page 84

Terminology

- Actor
  - Person, computer system, organization, etc.
- Scenario = Use-case instance
  - Sequence of actions between actors and the system that yields an observable **result of value** to a particular actor
- Use-case
  - Set of related success and failure scenarios
- Use-case model
  - Set of all use-cases
- Use-case diagram
  - Visualizes high-level interaction between actors and system

Use-Case

- *The key concept: A use case is a business process*
  - It is an abstraction (of a business process)
- Must be initiated by (or begin with) an actor
- Must accomplish some business task—for the actor
- Must end with an actor—preferably an actor action to acknowledge the accomplishment of the business task

Actor

- Denotes a **business** *role* played by (and on behalf of) a set of business entities or stakeholders

- Interacts with the system
  - Actors are not part of the system

- Often a human
  - but can also be a piece of hardware, a system, or another component of the system

Use-Case Abstraction Levels

- Use case name only
  - Verb noun phrase
- Brief
  - One paragraph on the main success scenario
- Casual
  - One informal paragraph for each scenario
- Fully dressed
  - All scenarios in detail with pre- and post-conditions

Use-Case Template (part 1)

- Name
  - Start with a verb
- Level
  - User-goal or (shared) sub-use-case
- Primary actor
- Stakeholders and interests
  - All use-case content should satisfy these interests
- Pre-Condition
  - **Assumption:** What must be true to start this use case?
- Post-Condition
  - **Guarantee:** What must be true on completion?

Use-Case Template (part 2)

- Main success scenario
  - **Basic flow (“happy path”),** easy to understand
  - Defer branches to:
- Extensions
  - Alternate flows
  - **Largest part of a use-case**

- Related non-functional requirements
- Open issues, etc.
  - Example: CL, pages 68—72

Scenario

- Either “main success” or “extension”

- Sequence of steps or actions of three kinds
  - Interaction between actors (including the system)
  - Validation (by the system)
  - State change by the system (store, update)

Extensions

- **Each extension answers a “what if” question about the main success scenario**
  - A branch from the main success scenario
- Name extension with the main success scenario step plus a fresh identifier
  - E.g.: “3a” is an extension that starts at main success scenario step 3
- The extensions of a use-case should cover most possible branches / variations, even rare ones

- **Write the “what if” question as something that can be detected by the system**

Example Use Case

- CL, pages 68 – 72
- From POS domain: point of sale

- **Name**: Process sale
- **Scope**: NextGen POS application
- **Level**: User goal
- **Primary Actor**: Cashier

Process Sale (2)

- **Stakeholders and Interests**:
  - **Cashier**: Wants accurate, fast entry, and no payment errors (cash shortage deducted from cashier's salary)
  - **Salesperson**: Wants sales commissions updated.
  - **Customer**: Wants fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.
  - **Company**: Wants accurate transaction record and satisfy customer interests. Wants payment receivables recorded. Wants sales capture even if server components (e.g., remote credit validation) unavailable. Wants automatic fast update of accounting and inventory.

Process Sale (3)

- **Stakeholders and Interests (continued)**:
  - **Manager**: Wants quick override operations, and easily debug Cashier problems.
  - **Government Tax Agencies**: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.
  - **Payment Authorization Service**: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

Process Sale (4)

- **Preconditions:**
  - Cashier is identified
  - Cashier is authenticated

- **Postconditions (Success Guarantee):**
  - Sale is saved
  - Tax is correctly calculated
  - Accounting and Inventory are updated
  - Commissions recorded
  - Receipt is generated
  - Payment authorization approvals are recorded

Process Sale (5)

- **Main Success Scenario (or Basic Flow)**:
  1. Customer arrives at POS checkout with goods and/or services to purchase
  2. Cashier starts a new sale
  3. Cashier enters item identifier
  4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules
  5. *Cashier repeats steps 3-4 until indicates done*

Process Sale (6)

- **Main Success Scenario (continued)**
  6. System presents total with taxes calculated
  7. Cashier tells Customer the total, and asks for payment
  8. Customer pays and System handles payment
  9. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory)
  10. System presents receipt
  11. Customer leaves with receipt and goods (if any)

Process Sale (7): Extension 1

***a**. At any time, Manager requests an override operation [star * means the branch can occur at any step of the main success scenario]:

1. System enters Manager-authorized mode
2. Manager or Cashier performs one Manager-mode operation. e.g.:
   - cash balance change
   - resume a suspended sale on another register
   - void a sale
3. System reverts to Cashier-authorized mode

Process Sale (8): Extension 2

*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

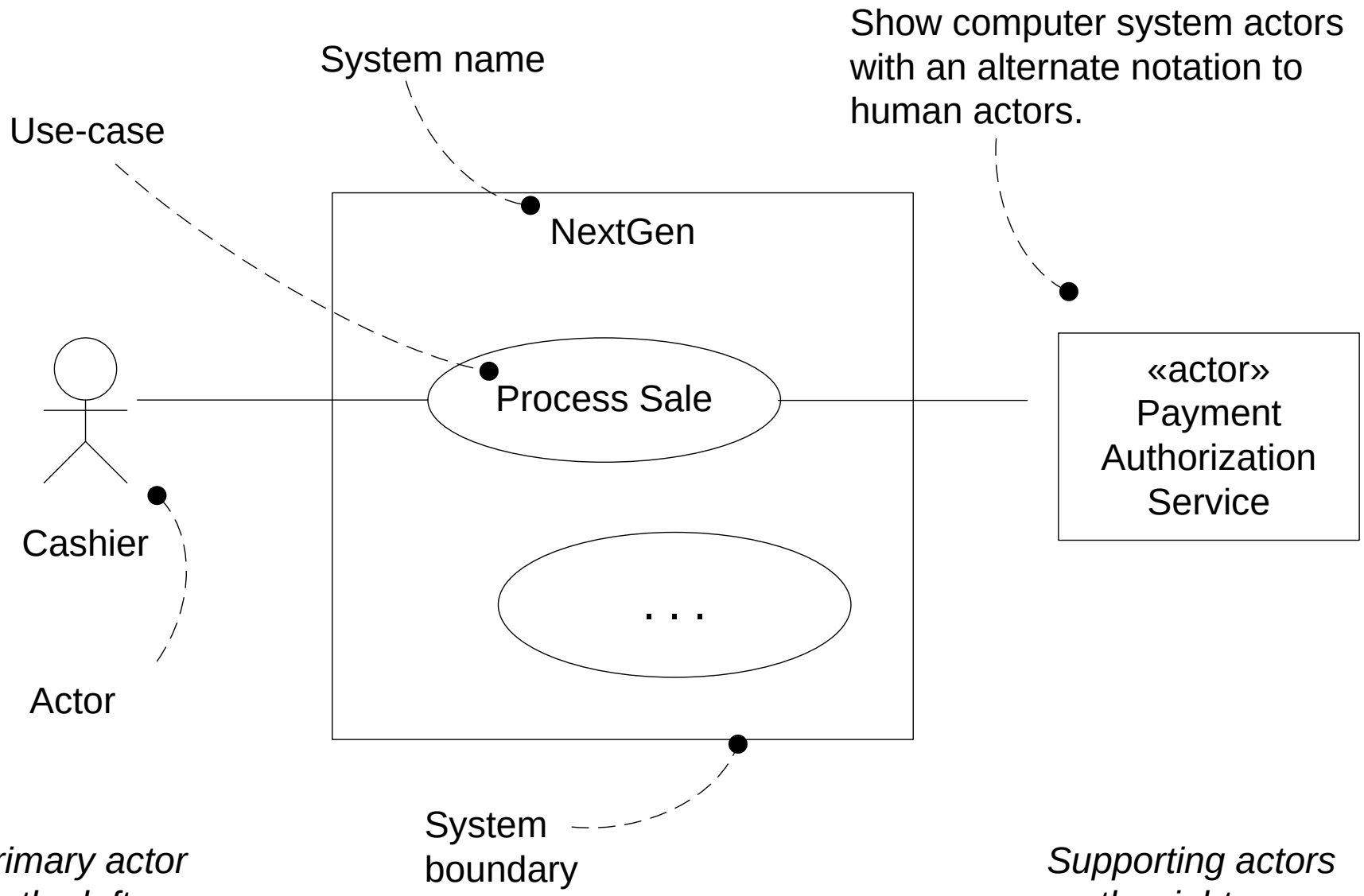
– ...

IN-CLASS EXERCISE: ADD AN EXTENSION TO A USE-CASE

Add an Extension to a Use-Case

- Get together with your team
- You have N minutes to complete a use-case
  1. Pick a use-case you defined for your project
  2. Make sure the use-case has a main success scenario (likely you did that for your project inception)
  3. Identify the most important extensions of the use-case: (“what if X goes wrong during the main success scenario”)
  4. Describe one key extension (as a detailed use case extension)
- Be prepared to present your extension to the class

Optional: Use-Case Diagram



CL Figure 6.4

Writing Tips

- “Essential”
  - Focus on user intentions & system responsibilities
  - Leave out many UI details
  - CL, page 80,...

- Terse
  - Condense writing to key information
  - Delete noise words