Use Cases

S. Thiel[1]

[1]Department of Computer Science
Concordia University

May 30, 2017

Outline

Why Use Cases [1, p.65]

- Lack of user involvement is bad
- Capturing goal is good
- Simple systems work best
- Use Cases address all these

Why **NOT** Use Cases [1, p.93]

- Application Servers
- database Products
- Back-end Systems
- Generally, feature-driven, or specifically not-user-driven systems may not fit well with Use Cases

Use Cases As Stories [1, p.61-62]

- Use Cases are text stories
- Use Cases are about actors using the system to meet goals
- Use Cases act as input for many artifacts
- Focus on clear and simple text, not the diagrams!

Use Case Definitions [1, p.62-63]

- **actor**: "is something with a behaviour. such as a person (identified by a role), computer system, or organization" [1, p.63]
- **scenario**: "is a specific sequence of actions and interactions between actors and the system" [1, p.63]; one way that the story can unfold. Also called **use case instance**
- **use case**: "is a collection of related sucess and failure scenarios" [1, p.63]
- **Main Success Scenario**: is the story of everything going as planned, successfully
- **Alternate Scenario**: is a story of an important deviation from the **Main Success Scenario**

RUP's Definition of Use Cases [1, p.64]

"A set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor." [1, as quoted, p.64]

Use Cases as Requirements [1, p.65]

- Primarily functional or behavioral requirements
- They indicate what the system will do
- Can describe other requirements as well
- In many processes, Use Cases are central mechanism to discover and define requirements
- Use Cases are more than just requirements

Use Cases as Contracts[1, p.65]

- Cockburn wrote **the** book on Use Cases
- Cockburn defined Use Cases in great detail, and described them as a contract to follow

Use Case Actors[1, p.66]

- **Primary Actor**: Has user goals fulfilled through using the system
  - Ususally a person
  - Can be a system or organization
- **Supporting Actor**: Provides a service to the system
  - usually a computer system.
  - Can technically be an organization or a person.
- **Offstage Actor**: Has an interest in the behaviour of the use case, but is neither **Primary** nor Supporting.
  - identifies (external) interests to make sure their needs are met.
  - e.g. government tax organization making sure your payment processing is legal

Types of Use Case[1, p.66]

- **brief** : Short summary, usually just the **Main Success Scenario**
- **casual** : Maybe a few paragraphs, covering the basics, one paragraph per scenario, including some **Alternate Scenarios**.
- **fully dressed** : All steps and variations written in detail, includes supporting sections, usually in a tabular format.

Fully Dressed Use Cases[1, p.67-81]

- In UP, a small number of the most important use cases might get written up like this in an initial iteration
- Cockburn's template is super-poluar for wtiting this out

Fully Dressed Pieces

- **Scope**:
- **Level**:
- **Primary Actor**:
- **Stakeholder and Interests List**:
- **Preconditions**:
- **Postconditions**:
- **Main Success Scenarios**:
- **Extensions**:

Use Case Model [1, p.64]

- A reminder that Use Cases are about text, not UML!
- Use Case Model is a quick way to show written Use Cases, provides a **context diagram**
- RUP considers the UCM a requirements artifact.
- Use Cases aren't OO. They are not about OO analysis, but...
- OOA/OOD can use Use Cases as input.

A Partial Use Case Model

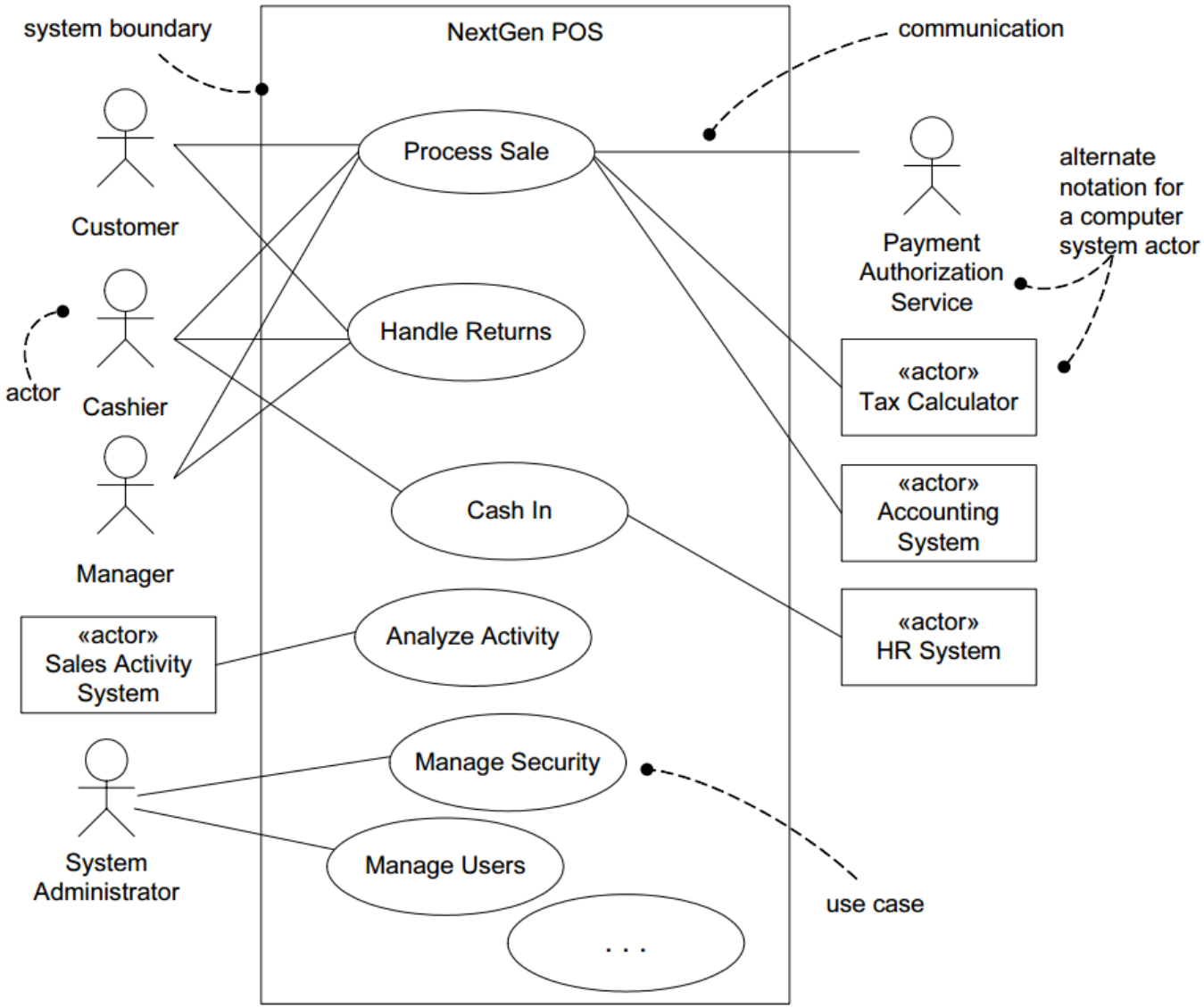

Figure: A Partial Use Case Model [1, p.90].

Use Case Model Guidelines

- Draw it simply while writing up actor-goal list.
- Primary actors on the left
- Supporting actors on the right

Use Case Model Actors

- Stick-Figure or box with **stereotype** "actor" are interchangeable
- I like stick figures
- Indicate services by calling them "Service" or using the "system" **stereotype** on the stick-figure.
- Not everyone focuses on the distinction between people and things
- Primary vs. Supporting actors is more important

Activity Diagrams [1, p.92]

- Useful for explaining complex work-flows (many parties, complex interaction, maybe concurrency)
- An alternate or supplemental notation for a Use Case
- When a Use Case is simple, Use Case text is enough

References I

[1] Craig Larman.
Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development.
Addison Wesley, 3rd edition, 2013.