



Sistemi ICT per il Business Networking

Use Cases

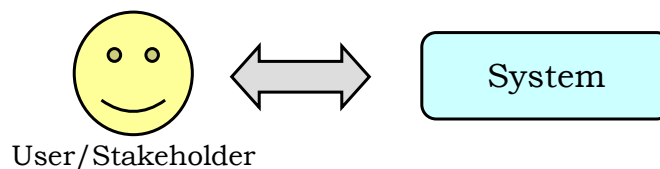
Docente: Vito Morreale (vito.morreale@eng.it)

25 March 2006

1

Use cases

- **Use cases:** stories of using a system
- A **technique** to understand and describe functional requirements
- **Use-Case Model:** the set of all use cases
 - a model of the system's functionality and environment



25 March 2006

ICT Systems for Business Networking

2

User Goals

- Customers and end users have **goals** (also known as **needs** in the UP)
- There are **several ways** to capture these goals and system requirements → **Use cases** are a mechanism to help keep it simple and understandable for all stakeholders
- **Example:** A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased 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 case: scenario

- An **actor** is something with behavior, such as a person (identified by role), computer system, or organization
- A **scenario** is a specific sequence of actions and interactions between actors and the system under discussion. It is one particular story of using a system
- **Example:** the scenario of successfully purchasing items with cash
- A scenario represents **something that can happen**. Variations (e.g. failing to purchase items because of a credit card transaction denial) lead to **alternate scenarios** ...
- ... but all these scenarios share a **GOAL** (e.g. purchase some items)

Use cases

- **Use case:** a collection of related success and failure scenarios that describe actors using a system to support a goal

Handle Returns

Main Success Scenario: A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item ...

Alternate Scenarios:

If the credit authorization is reject, inform the customer and ask for an alternate payment method.

If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).

If the system detects failure to communicate with the external tax calculator system, ...

Use cases

- Definition by RUP: A set of use-case instances (scenarios), where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor.
 - the system behavior should **emphasize providing value to the user**

Use cases vs. others

- The **feature and function list approach** to capturing requirements can contribute to a negative outcome because it does not encourage the stakeholders to consider the requirements in a larger context of **using the system in a scenario** to achieve some observable result of value, or some goal
- **Use cases** place features and functions in a **goal-oriented context**
- **Key idea**: do requirements work with a **focus** on how a system can add value and fulfill goals

25 March 2006

ICT Systems for Business Networking

7

Use cases as requirements

- Use cases are **requirements**: primarily **functional requirements** that indicate what the system will do
- Use cases define **a promise or contract** of how a system will behave

- **Important**. Use cases are **TEXT** documents, not diagrams, and use-case modeling is primarily an act of **WRITING TEXT**, not drawing

- UML defines a **use case diagram** to illustrate the names of use cases and actors, and their relationships

25 March 2006

ICT Systems for Business Networking

8

Use case formats

- Use cases are written in **different formats**, depending on need
- **Degrees of formality:**
 - **Brief:** terse one-paragraph summary, usually of the main success scenario
 - **Casual:** informal with multiple paragraphs that cover various scenarios
 - **Fully dressed:** all steps and variations are written in detail and there are supporting sections, such as preconditions and success guarantees

Use case: brief format

A *customer* arrives at a checkout with items to purchase. The *cashier* uses the **POS system** to record each purchased 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 case: casual format

Handle Returns

Main Success Scenario: A *customer* arrives at a checkout with items to return. The cashier uses the **POS system** to record each returned item ...

Alternate Scenarios:

If the credit authorization is reject, inform the *customer* and ask for an alternate payment method.

If the item identifier is not found in the *system*, notify the *Cashier* and suggest manual entry of the identifier code (perhaps it is corrupted).

If the *system* detects failure to communicate with the external *tax calculator system*, ...

Use case: fully-dressed format

- Various **format templates** are available for fully-dressed use cases
- However the most widely used and shared format is the template available at www.usecases.org
 - Primary Actor: the actor that interact with the system to fulfill a goal
 - Stakeholders and Interests
 - Preconditions
 - Success Guarantee (Postconditions)
 - Main Success Scenario (or Basic Flow)
 - Extensions (or Alternative Flows)
 - Special Requirements
 - Technology and Data Variations List
 - Frequency of Occurrence
 - Open Issues

Pre- and Post-conditions

- **Preconditions:** what must always be true before beginning a scenario in the use case
 - Not tested within the use case
 - Conditions that are assumed to be true
 - Typically, a scenario of another use case that has successfully completed, such as logging in
- **Success guarantees (or postconditions):** what must be true on successful completion of the use case (i.e. either the main success scenario or some alternate path)
 - should meet the needs of all stakeholders

Main Success Scenario and Steps

- Also called the "**happy path**" scenario or the "**Basic Flow**"
- **The typical success path** that satisfies the interests of the stakeholders
- **May not** include any conditions or branching
- **Suggestion:** defer all conditional and branching statements to the Extensions section

Main Success Scenario and Steps

- The scenario records one of the following three step types:
 1. An **interaction** between an actor and the system
 2. A validation (usually by the system)
 3. A state change by the system (for example, recording or modifying something)
- **Step one** of a use case does not always fall into this classification, but indicates the **trigger event** that starts the scenario
- Common to **capitalize the actors' names** for ease of identification

25 March 2006

ICT Systems for Business Networking

15

Extensions (or Alternate Flows)

- **All the other scenarios**, both success and failure
- Often considerably **longer and more complex** than the Main Success Scenario section
- In thorough use case writing, **the combination of the happy path and extension scenarios** should satisfy "nearly" **all the interests of the stakeholders**
- Extension scenarios are **branches** from the main success scenario, and so can be notated with respect to it
- An extension has two parts: the **condition** and the **handling**
- **At the end of extension handling**, by default the scenario merges back with the main success scenario, unless the extension indicates otherwise

25 March 2006

ICT Systems for Business Networking

16

Extensions (or Alternate Flows)

- **Sometimes** a particular extension point is quite complex
- A motivation to express the **extension** as a separate use case
- It is desirable to describe an extension condition as possible during any steps: the **labels** *a, *b, ..., can be used

*a. At any time, System crashes:

In order to support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered at any step in the scenario.

1. Cashier restarts the System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

Other sections

- **Special Requirements:** non-functional requirement, quality attribute, or constraint relates specifically to a use case
 - E.g. performance, reliability, and usability, and design constraints (often in I/O devices)
- **Technology and Data Variations List:** how something must be done, but not what (early design decisions or constraints)
 - **technical constraint** imposed by a stakeholder regarding input or output technologies
 - E.g. the POS system must support credit account input using a card reader and the keyboard
 - **variations in data schemes**
 - E.g. using UPCs or EANs for item identifiers, encoded in bar codes

Use case: fully-dressed format

Use Case UC1: Process Sale

Primary Actor: Cashier

Stakeholders and Interests:

- Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.
- Salesperson: Wants sales commissions updated.
- Customer: Wants purchase and fast service with minimal effort. Wants proof of purchase to support returns.
- Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.
- 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.

Preconditions: Cashier is identified and authenticated.

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

Use case: fully-dressed format

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.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. 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).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

Use case: fully-dressed format

Extensions (or Alternative Flows):

- a. 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.
 - 1. Cashier restarts System, logs in, and requests recovery of prior state.
 - 2. System reconstructs prior state.
 - 2a. System detects anomalies preventing recovery:
 - 1. System signals error to the Cashier, records the error, and enters a clean state.
 - 2. Cashier starts a new sale.
 - 3a. Invalid identifier:
 - 1. System signals error and rejects entry.
 - 3b. There are multiple of same item category and tracking unique item identity not important (e.g., 5 packages of veggie-burgers):
 - 1. Cashier can enter item category identifier and the quantity.
 - 3-6a: Customer asks Cashier to remove an item from the purchase:
 - 1. Cashier enters item identifier for removal from sale.
 - 2. System displays updated running total.
 - 3-6b. Customer tells Cashier to cancel sale:
 - 1. Cashier cancels sale on System.

Use case: fully-dressed format

Special Requirements:

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
- Credit authorization response within 30 seconds 90% of the time.
- Somehow, we want robust recovery when access to remote services such the inventory system is failing.
- Language internationalization on the text displayed.
- Pluggable business rules to be insertable at steps 3 and 7.

Technology and Data Variations List:

- 3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
- 3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
- 7a. Credit account information entered by card reader or keyboard.
- 7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Use case: fully-dressed format

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- What are the tax law variations?
- Explore the remote service recovery issue.
- What customization is needed for different businesses?
- Must a cashier take their cash drawer when they log out?
- Can the customer directly use the card reader, or does the cashier have to do it?

Two-Column format

Use Case UC1: Process Sale

Primary Actor: ...
... as before ...

Main Success Scenario:

Actor Action (or Intention)

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.

Cashier repeats steps 3-4 until indicates done.

6. Cashier tells Customer the total, and asks for payment.
7. Customer pays.

System Responsibility

4. Records each sale line item and presents item description and running total.
5. System presents total with taxes calculated.
8. Handles payment.

**First proposed by
Rebecca Wirfs-Brock**

What's The Best Format?

- There **isn't** one best format
- **Some** prefer the one-column style, some the two-column
- **Sections** may be added and removed
- **Heading names** may change
- **The key thing:** to write the details of the main success scenario and its extensions in some form
- In *Writing Effective Use Cases* there are many usable formats

How should use cases be discovered?

- Many levels of granularity
- A framework for identifying the use cases:
 - elementary business processes goals

Use Cases for Elementary Business Processes

- **EBP Use Case:** focus on use cases at the level of EBP
- **Elementary Business Process:** a task performed by one person in one place at one time, in response to a business event, which adds measurable business value and leaves the data in a consistent state
 - **Not a single small step** (e.g. "delete a line item" or "print the document")
 - The main success scenario is probably **five or ten steps**
 - It **doesn't take days and multiple sessions** ("negotiate a contract")
 - It is a task done during a **single session**
 - It is probably **between a few minutes and an hour** in length
 - It emphasizes adding **observable or measurable business value**
 - It comes to a **resolution** in which **the system and data are in a stable and consistent state**

25 March 2006

ICT Systems for Business Networking

27

Use Cases and Goals

- **Assumption:** actors have goals (or needs) and use applications to help satisfy them
- An EBP-level use case is called a **user goal-level use case**, to emphasize that it serves (or should serve) to fulfill a goal of a user of the system, or the primary actor
- **Recommended procedure:**
 1. Find the user goals
 2. Define a use case for each
- 1. The **EBP guideline** can be applied to decide if a goal or a use case is at a suitable level

25 March 2006

ICT Systems for Business Networking

28

Use Cases and Goals

- A Customer may be asked:
 - "What do you do?" (roughly a use case-oriented question)
 - Answers are more likely to reflect current solutions and procedures, and the complications associated with them
 - "What are your goals?"
 - Answers combined with an investigation to move higher up the goal hierarchy open up the vision for new and improved solutions, focus on adding business value, and get to the heart of what the stakeholders want from the system under discussion

25 March 2006

ICT Systems for Business Networking

29

Subfunction Goals and Use Cases

- **Subfunction Goals:**
 - Goals at a lower level
- Use cases should **only occasionally** be written for these subfunction goals:
 - It is not illegal to write use cases for subfunction goals, but it is not always helpful, as it adds complexity to a use-case model
 - There can be hundreds of subfunction goals for a system
- The most common motivation to express a subfunction goal as a use case is when the subfunction is **repeated** in or is a precondition for multiple user goal-level use cases

25 March 2006

ICT Systems for Business Networking

30

Finding Primary Actors, Goals, and Use Cases

1. Choose the **system boundary**
2. Identify the **primary actors** (those that have user goals fulfilled through using services of the system)
3. For each, identify their **user goals** and raise them to the highest user goal level
4. Define **use cases** that satisfy user goals and name them according to their goal
 - Usually, user goal-level use cases will be one-to-one with user goals, but there is at least one exception

25 March 2006

ICT Systems for Business Networking

31

Step 1: Choosing the System Boundary

- Defining what is outside:
 - the external primary
 - supporting actors

25 March 2006

ICT Systems for Business Networking

32

Steps 2 and 3: Finding Primary Actors and Goals

- In requirements, analysts brainstorm and generate a **mixture of primary actors and their user-level goals**
 - Sometimes, goals reveal the actors, or vice versa
- **Questions:**
 - Who starts and stops the system?
 - Who does user and security management?
 - Is there a monitoring process that restarts the system if it fails?
 - How are software updates handled?
 - Who does system administration?
 - Who evaluates system activity or performance?
 - Who evaluates logs? Are they remotely retrieved?
 - ...

25 March 2006

ICT Systems for Business Networking

33

Steps 2 and 3: Finding Primary Actors and Goals

- Record the primary actors and their user goals in an **actor-goal list**

Actor	Goal	Actor	Goal
Cashier	process sales process rentals handle returns cash in cash out ...	System Administrator	add users modify users delete users manage security manage system tables ...
Manager	start up shut down ...	Sales Activity System	analyze sales and performance data
...

25 March 2006

ICT Systems for Business Networking

34

Steps 2 and 3: Finding Primary Actors and Goals

- Another approach to aid in finding actors, goals, and use cases is to identify **external events**. What are they, where from, and why?

External Event	From Actor	Goal
enter sale line item	Cashier	process a sale
enter payment	Cashier or Customer	process a sale
...		

25 March 2006

ICT Systems for Business Networking

35

Step 4: Define Use Cases

- In general, **define** one EBP-level use case for each user goal
- **Name** the use case similar to the user goal starting with a verb

25 March 2006

ICT Systems for Business Networking

36

Actors

- **Actor**: anything with behavior, including the system under discussion itself when it calls upon the services of other systems.
- Actors will appear **in the action steps** of the use case text
- Actors are not only roles played by **people**, but **organizations, software, and machines**

Kinds of Actors

- **Primary actor** has user goals fulfilled through using services of the system
- **Supporting actor** provides a service (even information) to the system (e.g. a computer system, but could be an organization or person)
 - Objective: to clarify external interfaces and protocols
- **Offstage actor** has an interest in the behavior of the use case, but is not primary or supporting (e.g. a government tax agency)
 - Objective: to ensure that all necessary interests are identified and satisfied

Use case: examples

- [Use case example](#)
- [Use Case Template](#)
- [Use Case Template by Cockburn](#)

References

- **Use Case Model: writing requirements in context**
(Capitolo libro "Applying UML and patterns")

Other references

- Although many have made contributions to the use cases subject, the most influential, comprehensive, and coherent work in defining what use cases are (or should be) and how to write them came from [Alistair Cockburn](#), **Writing Effective Use Cases**.