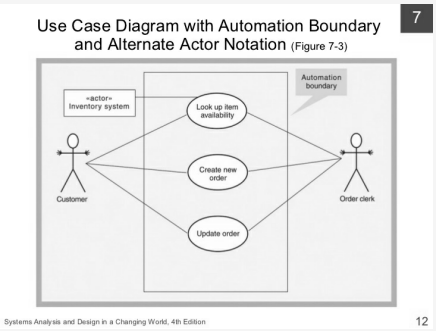
Labelled use case automation boundary



Use Cases

Define requirements activity

Lists steps defining interactions between a role (aka UML "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time.

Benefits

Short summary of what the system will offer.

Provides a project planning skeleton, to be used to build initial priorities, estimates, team allocation and timing.

Provides everyone with an agreement as to what the system will/won't do.

Provides context for requirements

Helps in investigating small details which could cause big problems.

Answers often detailed / tricky / ignored business questions: “What are we supposed to do in this case?”

Shows that the investigators have thought through every user’s needs, every user system goal, every business variant involved.

Limits

not good for non-interaction based requirements

Clarity depends on the skill of the writer(s).

Sometimes use cases are complex to write and to understand

No fully standard definitions of use cases

Use case names

Name a use case with a verb-noun phrase that states the actor's goal. Action, Name.

User goal technique (Use case method)

1. Identify all potential users for a system
2. (Optional) Classify users by functional role (shipping, marketing, sales) and operational level (operational, management, executive)
3. Interview each user and determine what goals they have when using the system
4. Make a preliminary list of use cases for each type of user
5. Look for duplicates and inconsistencies across users
6. Identify when multiple users need the same use case
7. Review completed list with users and other stakeholders for validation

Event Decomposition

This approach looks for all events that would lead to the information system being used. Each event typically leads to a use case. Simplify events to ones that have a clearly defined start and end, and achieve a clear business purpose. These are: Elementary Business Processes (EBPs) = use cases.
Keeps attention on the macro scale purpose of the system, not internal details
Events can be:
- External – caused by an actor
- Temporal – done at fixed time intervals
- State – triggered by an internal condition, e.g. low inventory
Steps:
a. Identify relevant external events
b. For each, name a use case
c. Identify relevant temporal events

Event Decomposition (cont)

d. For each, name a use case and define when it occurs
e. Identify relevant state events
f. For each, name a use case
g. Omit trivial use cases (like log in), but keep system controls

Develop a use case diagram

1. ID all stakeholders who need a use case diagram.
2. Determine what each stakeholder or user needs to review in a use case diagram. Could be subsystem, user focused, for use cases with the includes relationship, and for use cases that are of interest to specific stakeholders.
3. For each potential communication need select the use cases and actors to show and draw the use case diagram.
4.Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users

CRUD - Create, Read or Report, Update and Delete

1. ID all data entities or domain classes involved
2. For each verify a use case has been created that creates a new instance, updates existing instances reads or reports values of instances and deletes (archives) an instance
3. If a needed use case has been overlooked add a new one and ID stakeholders
4. With integrated apps make clear app responsible for adding and maintain data and which merely uses the data

By **Natalie Moore** (NatalieMoore)

cheatography.com/nataliemoore/
www.jchmedia.com/

Published 29th May, 2015.
Last updated 13th May, 2016.

CRUD

Data Entity	CRUD	Resulting Use Case
Customer	**Create**	Add New Customer
	Read/Report	Find Customer
		Generate Customer List
	Update	Update Customer Information
	Delete	Delete inactive customer
Order	**Create**	Create New Order
	Read/Report	View Order
		View Order History
	Update	Update Order
	Delete	Cancel Order
Inventory Item	**Create**	Add New Inventory Item
	Read/Report	View Inventory Item
		Find Inventory Item
		Generate Inventory Item List
	Update	Update Inventory Item
	Delete	Delete Inventory Item
Shipment	**Create**	Create New Shipment Request
	Read/Report	View Shipment Details
		Generate Shipment Report
	Update	Update Shipment Request
	Delete	Cancel Shipment Request

UML - Unified Modeling Language

1. General-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.
2. UML has been evolving since the second half of the 1990s and has its roots in the object-oriented methods developed in the late 1980s and early 1990s.

Why

Visualise through an assortment of types of diagrams: Activities, Components and how they interact with other software, How system will run, How entities interact with others, external user interface

What

UML diagrams include (*part of unit): Component, Activity*, Sequence*, Class*, Use case*, Communication

By **Natalie Moore**
(NatalieMoore)

cheatography.com/nataliemoore/
www.jchmedia.com/

Published 29th May, 2015.
Last updated 13th May, 2016.