IN77J – Orientación al Objeto para el e-business

5. Análisis

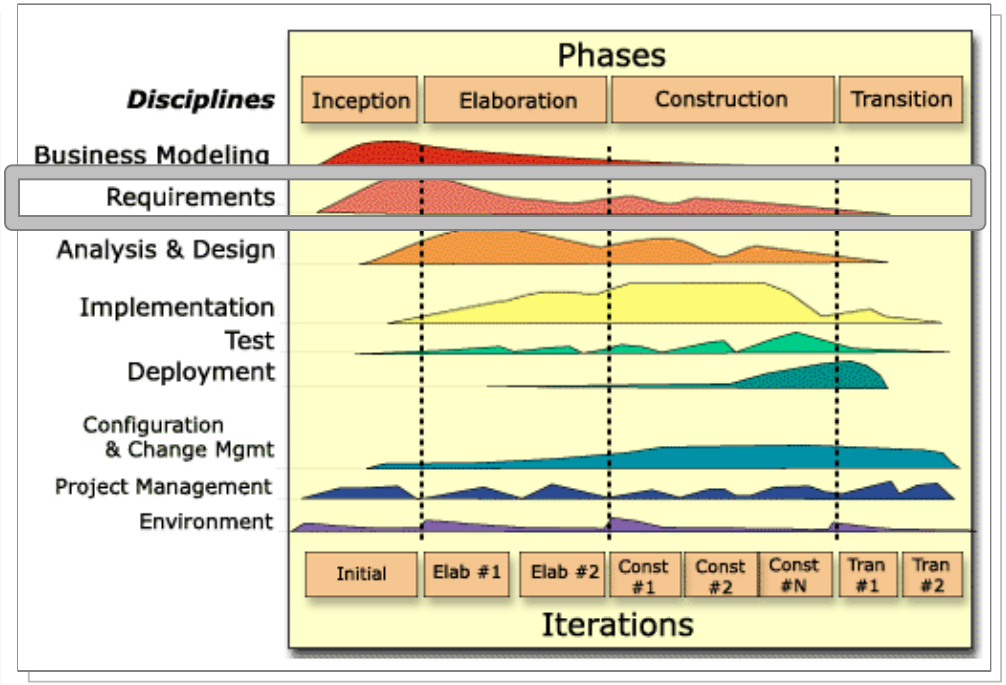
Temario

- 5. Análisis
  - Requerimientos
  - Modelo de Casos de Uso
  - Modelo de Dominio
  - Taller

Requerimientos

Propósito de los requerimientos:

- **Establecer y mantener un acuerdo con los clientes acerca de lo que el sistema debe hacer**
- **Ayudar a los desarrolladores a comprender los requerimientos del sistema**
- **Delimitar el sistema**
- **Proveer la base para la planificación del proyecto**
- **Proveer la base para la estimación del costo y el tiempo de desarrollo del sistema**
- **Definir una interfaz de usuario para el sistema**



Requerimientos: Artefactos Principales

- Especificación de requerimientos no funcionales: sistema operativo, tiempo de respuesta, nivel de confiabilidad, etc.
- Modelo de casos de uso: especificación de requerimientos funcionales
- Modelo conceptual: modelo de clases de negocio o de dominio
- Glosario

Modelo de Casos de Uso

- El modelo de casos de uso describe la funcionalidad del sistema
- Cada caso de uso describe un comportamiento del sistema, en interacción con uno o más actores
- Los casos de uso se describen mediante documentos de texto, con apoyo de diagramas UML:
  - Diagramas de casos de uso
  - Diagramas de actividades
  - Diagramas de máquinas de estados
  - Diagramas de clases

Casos de Uso

- Los casos de uso son una técnica para definir requerimientos creada por Ivar Jacobson en 1986, e incorporada en UML (pese a no ser una técnica orientada a objetos)
- *"Use cases are a means for specifying required usages of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do. The key concepts associated with use cases are actors, use cases, and the subject. The subject is the system under consideration to which the use cases apply. The users and any other systems that may interact with the subject are represented as actors. Actors always model entities that are outside the system. The required behavior of the subject is specified by one or more use cases, which are defined according to the needs of actors."*

[especificación UML 2.0]

Definición de Caso de Uso

- *"A UseCase is a kind of behaviored classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants, that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling."*

[especificación UML 2.0]

Definición de Caso de Uso

- *"Use cases can be used both for specification of the (external) requirements on a subject and for the specification of the functionality offered by a subject. Moreover, the use cases also state the requirements the specified subject poses on its environment by defining how they should interact with the subject so that it will be able to perform its services."*

- *"The behavior of a use case can be described by a specification that is some kind of behavior, such as interactions, activities, and state machines, or by pre-conditions and post-conditions as well as by natural language text where appropriate."*

[especificación UML 2.0]

Otras Definiciones

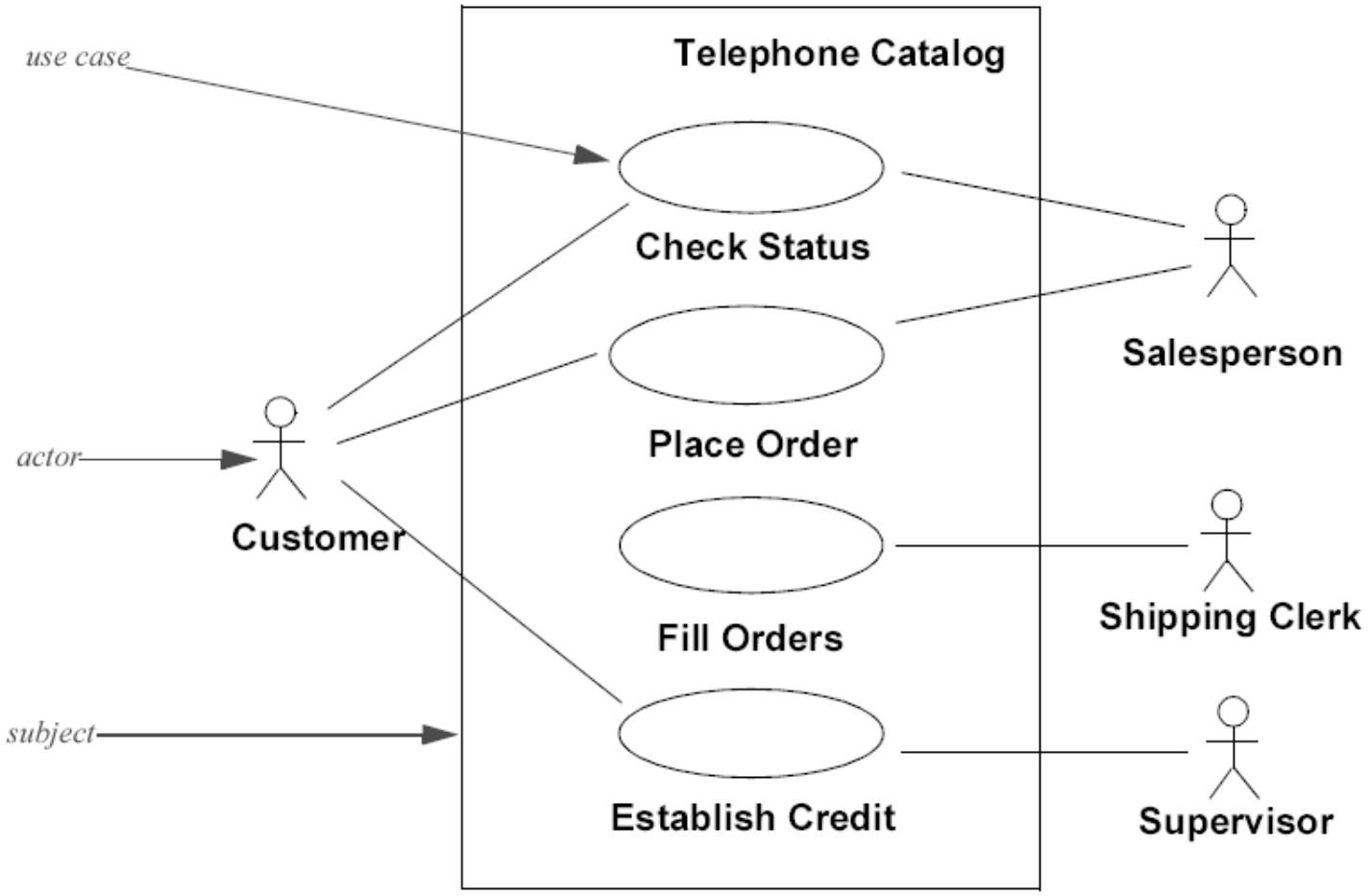
- *"A use case is a narrative document that describes the sequence of events of an actor (an external agent) using a system to complete a process"* [Jacobson, 1992]
- *"They are stories or cases of using a system. Use case are not exactly requirements or functional specifications, but they illustrate and imply requirements in the stories they tell."* [Larman, 1998]
- *"A description of set of sequences of actions, including variants, that a system performs that yield an observable result of value to an actor."* [Booch, 1999]
- *"You apply use cases to capture the intended behavior of the system you are developing, without having to specify how that behavior is implemented. Use cases provide a way for your developers to come to a common understanding with your system's end users and domain experts. In addition, use cases serve to help validate your architecture and to verify your system as it evolves during development."* [Booch, 1999]

Diagrama de Casos de Uso

- Un diagrama de casos de uso representa una parte de la funcionalidad de un sistema, en la interacción con objetos externos a él
- Los diagramas de casos de uso muestran actores y casos de uso en conjunto con sus relaciones
- Un caso de uso muestra qué hace el sistema, no cómo lo hace

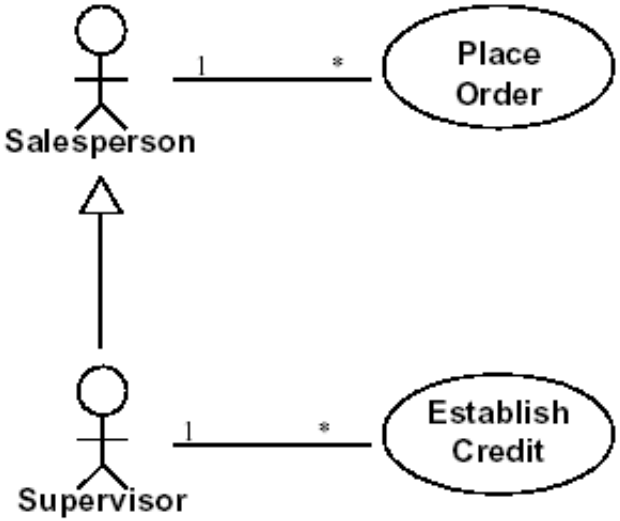
Diagrama de Casos de Uso

- Modela requerimientos funcionales, expresados en la interacción entre actores y módulos del sistema

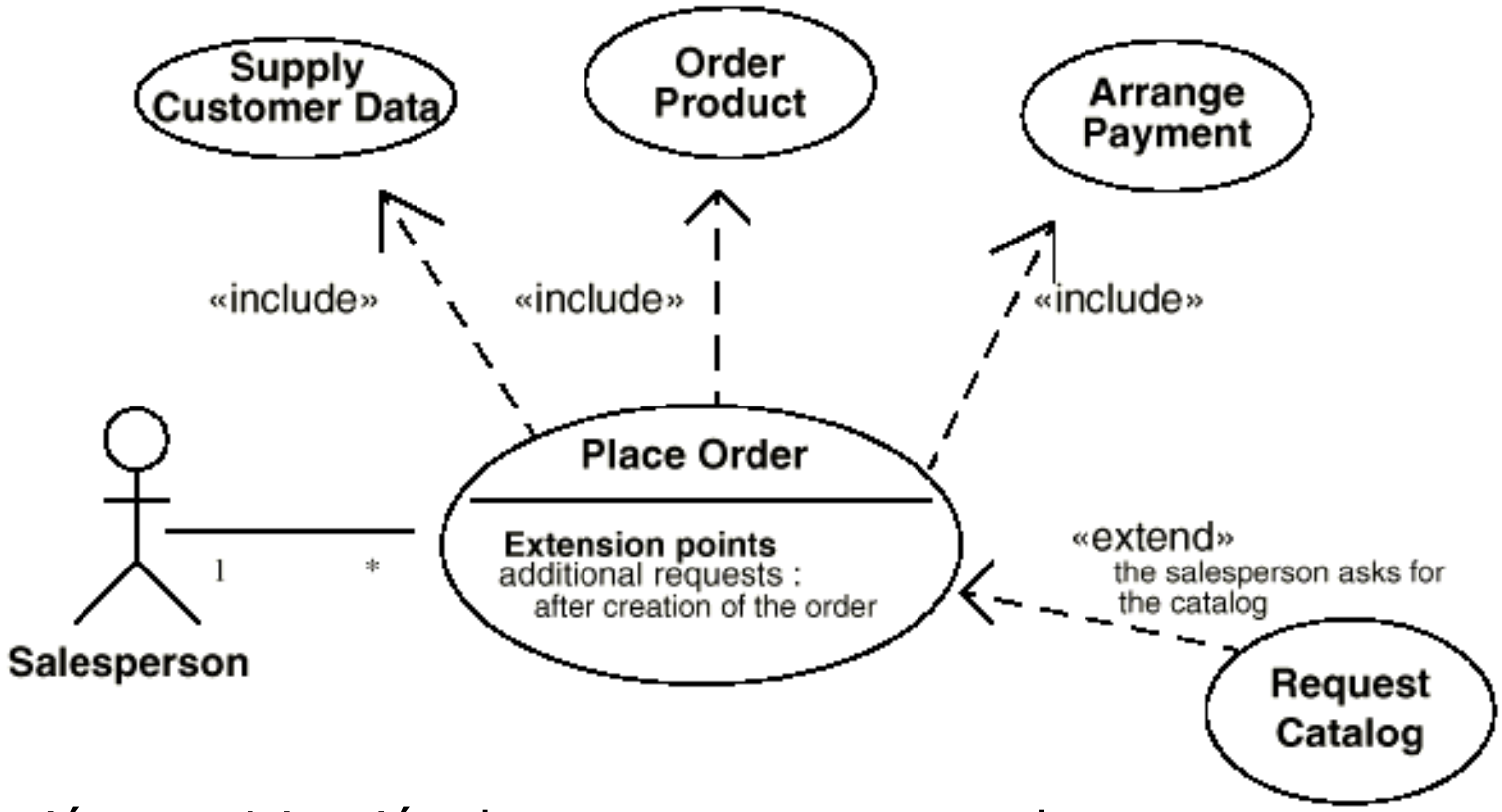


Actores

- Actor: *"An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects.*
- Generalización: *"A generalization from an actor A to an actor B indicates that an instance of A can communicate with the same kinds of use-case instances as an instance of B"*



Relaciones en Casos de Uso



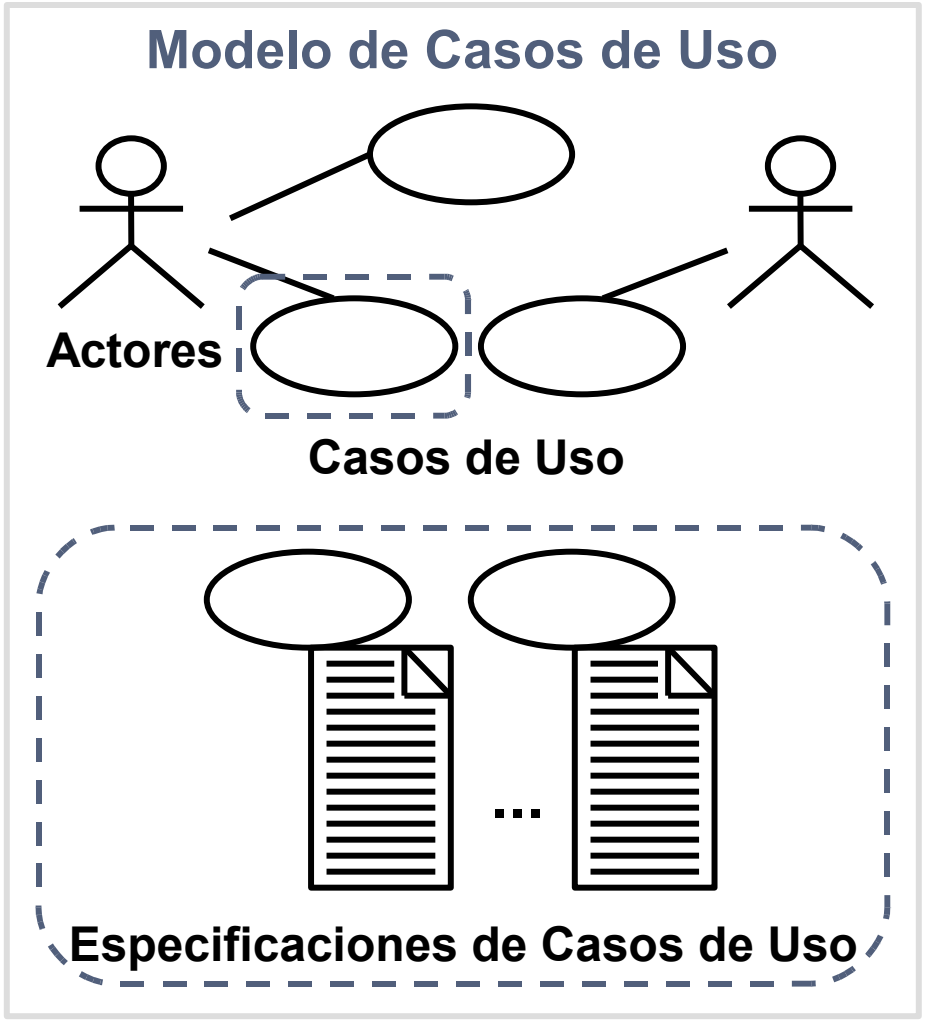
- Asociación: participación de un actor en un caso de uso
- Extensión: un caso de uso extiende el comportamiento de otro
- Generalización: especialización entre casos de uso o actores
- Inclusión: un caso de uso incluye el comportamiento de otro

Algunas Recomendaciones

- Fuente: Scott W. Ambler
- Casos de Uso
  - Usar verbos para nombres de casos de uso
  - Usar terminología del dominio para los nombres de los casos de uso
- Actores
  - Dibujar los actores hacia las afueras del diagrama
  - Usar nombres singulares para los actores, con significado en el negocio
  - Actores modelan roles, no posiciones
  - Usar <<system>> para indicar que un actor es un sistema
  - Los actores no interactúan entre ellos
- Relaciones
  - Evitar flechas en relaciones entre actores y casos de uso (para evitar confusión con flujo de datos)
  - ¡Evitar descomposición funcional!

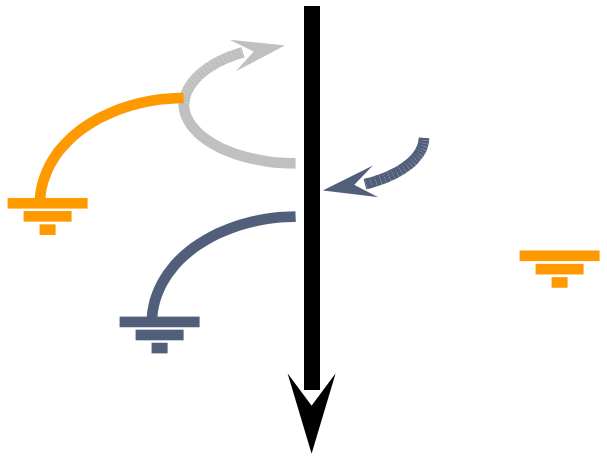
Especificación de Casos de Uso

- Plantilla RUP
  - Nombre del Caso de Uso
    - Breve Descripción
  - Flujo de Eventos
    - Flujo Básico
    - Flujos Alternativos
  - Requerimientos Especiales
  - Pre-condiciones
  - Post-condiciones
  - Puntos de Extensión
- Otros: diagramas, etc.



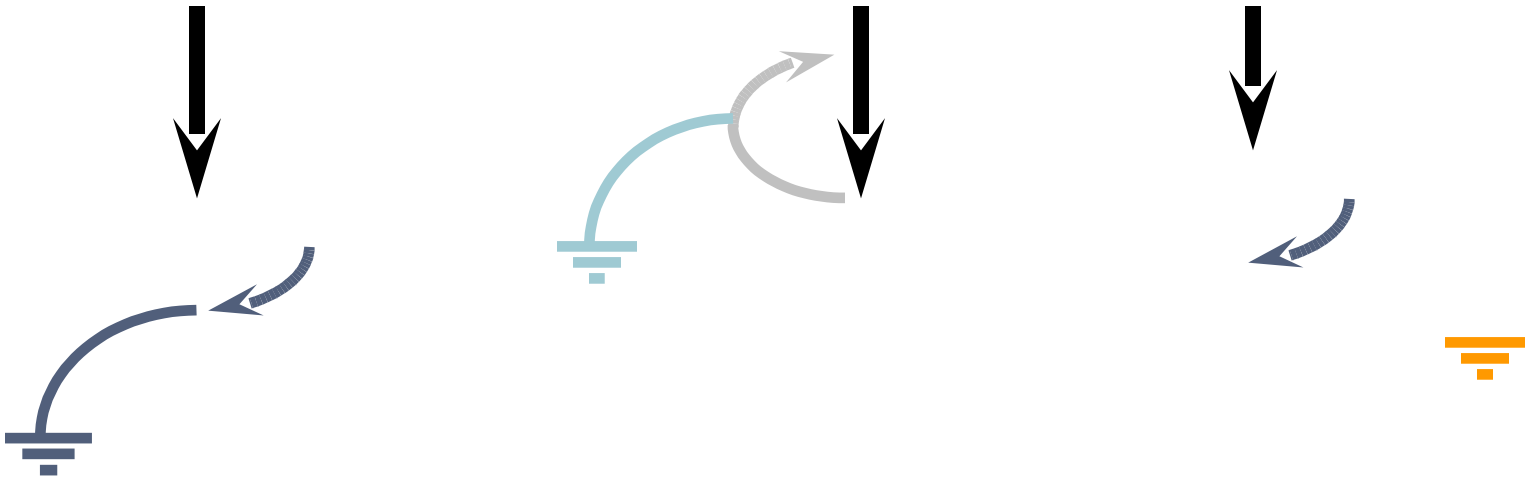
Flujo de Eventos

- Un flujo normal (básico)
- Varios flujos alternativos
  - Variantes regulares
  - Casos especiales (no queda stock, no tiene crédito disponible, etc.)
  - Flujos de excepción para manejar situaciones de error



Escenarios

- Un escenario es una secuencia de acciones específica del sistema en discusión
- Un escenario es una instancia de un caso de uso



Ejemplo 1 (de A. Cockburn)

USE CASE 1: BUY STOCKS OVER THE WEB

Primary Actor: Purchaser
Scope: Personal Advisors / Finance package ("PAF")
Level: User goal
Stakeholders and Interests:
Purchaser - wants to buy stocks, get them added to the PAF portfolio automatically.
Stock agency - wants full purchase information.
Precondition: User already has PAF open.
Minimal guarantee: sufficient logging information that PAF can detect that something went wrong and can ask the user to provide details.
Success guarantee: remote web site has acknowledged the purchase, the logs and the user's portfolio are updated.
Main success scenario:
1. User selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwabb, etc.) from user.
3. PAF opens web connection to the site, retaining control.
4. User browses and buys stock from the web site.
5. PAF intercepts responses from the web site, and updates the user's portfolio.
6. PAF shows the user the new portfolio standing.
Extensions:
2a. User wants a web site PAF does not support:
    2a1. System gets new suggestion from user, with option to cancel use case.
3a. Web failure of any sort during setup:
    3a1. System reports failure to user with advice, backs up to previous step.
    3a2. User either backs out of this use case, or tries again.
4a. Computer crashes or gets switched off during purchase transaction:
    4a1. (what do we do here?)
4b. Web site does not acknowledge purchase, but puts it on delay:
    4b1. PAF logs the delay, sets a timer to ask the user about the outcome.
    4b2. (see use case *Update questioned purchase*)
5a. Web site does not return the needed information from the purchase:
    5a1. PAF logs the lack of information, has the user *Update questioned purchase*.

Ejemplo 2 (de A. Cockburn)

USE CASE 2: GET PAID FOR CAR ACCIDENT

Primary Actor: The Claimant
Scope: The insurance company ("MyInsCo")
Level: Summary
Stakeholders and Interests:
the claimant - to get paid the most possible
MyInsCo - to pay the smallest appropriate amount
the dept. of insurance - to see that all guidelines are followed.
Precondition: none
Minimal guarantees: MyInsCo logs the claim and all activities.
Success guarantees: Claimant and MyInsCo agree on amount to be paid, claimant gets paid that.
Trigger: Claimant submits a claim
Main success scenario:
1. Claimant submits claim with substantiating data.
2. Insurance company verifies claimant owns a valid policy
3. Insurance company assigns agent to examine case
4. Insurance company verifies all details are within policy guidelines
5. Insurance company pays claimant and closes file.

Main success scenario:
1a. Submitted data is incomplete:
    1a1. Insurance company requests missing information
    1a2. Claimant supplies missing information
2a. Claimant does not own a valid policy:
    2a1. Insurance company declines claim, notifies claimant, records all this, terminates proceedings.
3a. No agents are available at this time
    3a1. (What does the insurance company do here?)
4a. Accident violates basic policy guidelines:
    4a1. Insurance company declines claim, notifies claimant, records all this, terminates proceedings.
4b. Accident violates some minor policy guidelines:
    4b1. Insurance company begins negotiation with claimant as to degree of payment to be made.

Ejemplo 3 (de A. Cockburn)

USE CASE 3: REGISTER ARRIVAL OF A BOX

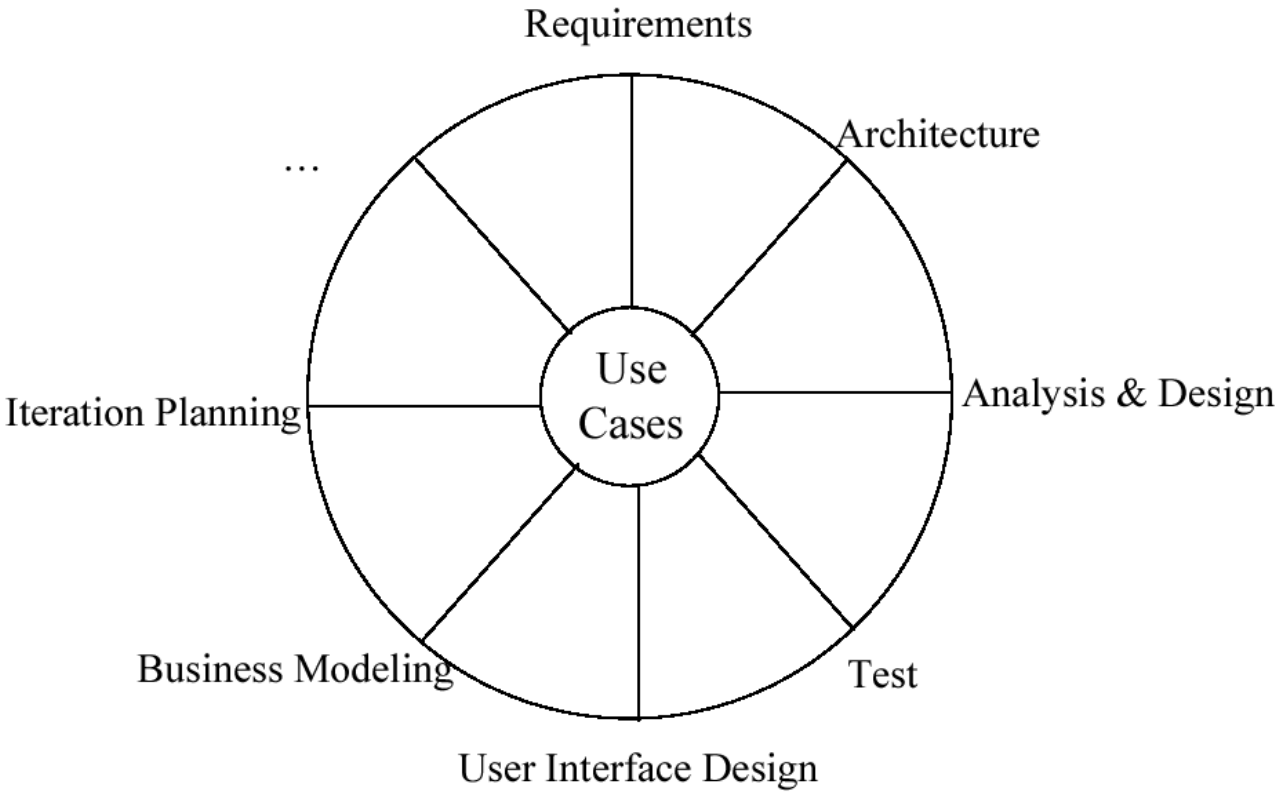
RA means "Receiving Agent".
RO means "Registration Operator"
Primary Actor: RA
Scope: Nightime Receiving Registry Software
Level: user goal
Main success scenario:
1. RA receives and opens box (box id, bags with bag ids) from TransportCompany TC
2. RA validates box id with TC registered ids.
3. RA maybe signs paper form for delivery person
4. RA registers arrival into system, which stores:
    RA id
    date, time
    box id
    TransportCompany
    <Person name?>
    # bags (?with bag ids)
    <estimated value?>
5. RA removes bags from box, puts onto cart, takes to RO.

Extensions:
2a. box id does not match transport company
4a. fire alarm goes off and interrupts registration
4b. computer goes down
 leave the money on the desk and wait for computer to come back up.

Variations:
4'. with and without Person id
4". with and without estimated value
5'. RA leaves bags in box.

No sólo Requerimientos

- Los Casos de Uso tienen una utilidad que va más allá del manejo de requerimientos



Modelo de Dominio

- El modelo de dominio (en ocasiones llamado modelo conceptual) presenta las entidades de negocio con las que el sistema debe interactuar
- El modelo de dominio se representa mediante un diagrama de clases, con las clases correspondientes a las entidades de negocio, sus relaciones y sus atributos
- Algunas referencias:
  - http://www.aptprocess.com/whitepapers/DomainModelling.p
  - http://www.craiglarman.com/book_applying/domain%20moc pdf

Modelo de Dominio en RUP

- RUP define el modelo de dominio como *"a standalone subset of the Business Analysis Model that focuses on concepts, products, deliverables, and events that are important to the business domain"*
- Y luego agrega *"You can choose to develop a standalone subset of the Business Analysis Model that focuses on explaining products, deliverables, or events that are important to the business domain. Such a model describes only the significant information in the business and does not include the responsibilities that people carry. Such a model is often referred to as a Domain Model."*

Modelo de Datos

- Si el sistema hace uso de una base de datos, es importante crear un modelo de datos que represente su estructura de manera visual (tablas, columnas, índices, llaves, relaciones, etc.)
- En UML no existe un diagrama específicamente diseñado para este efecto, por lo que suele utilizarse el diagrama de clases para representar tablas
- Si se trabaja con tecnologías de tipo mapeo objeto-relacional (EJB entity beans, Hibernate, TopLink, JDO, etc.), el modelo de datos debe manejarse sincronizadamente con el modelo de clases persistentes