TEMA 2.4: Análisis y especificación de requisitos

---

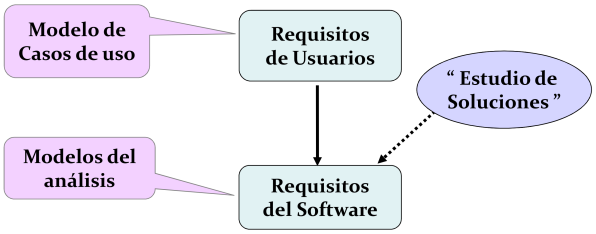
1. INTRODUCCIÓN

El análisis de requisitos es la fase de la ingeniería de requisitos en la que se examinan los requisitos para delimitarlos y definir exactamente cada uno de ellos.

Se trata, fundamentalmente, de:

- Detectar y resolver conflictos entre requisitos
- Delimitar el SW y establecer con qué elementos externos interacciona
- Elaborar los requisitos del sistema para obtener, a partir de ellos, los requisitos del SW a desarrollar

El **objetivo principal** del análisis de requisitos es refinar, estructurar y describir los requisitos para una comprensión más precisa y fácil de mantener que ayude a estructurar el sistema completo (modelos del análisis).



Actividades durante el análisis

- Clasificación de los requisitos: establecer un conjunto de categorías y situar cada requisito de ellas
- Priorización de los requisitos: determinar la importancia relativa de cada requisito en relación con los demás
- Modelado conceptual: representar los requisitos con un lenguaje o notación que "comprendan" aquellos que van a tratar con ellos
- Situación de los requisitos en la arquitectura del sistema: establecer qué elementos del sistema SW van a satisfacer los distintos requisitos (permite descubrir nuevos requisitos)
- Negociación de los requisitos: detectar y resolver problemas, definir de manera precisa los límites del sistema y cómo este debe interaccionar con su entorno

Especificación de requisitos

Consiste en completar la descripción de los requisitos del sistema a desarrollar. Se genera un documento de especificación de requisitos del SW.

Una especificación debe ser:

- Completa
- Verificable
- Consistente
- Modificable
- Susceptible de permitir seguimientos
- Utilizable durante las fases de operación y mantenimiento
- No debe contener ambigüedades

2. ANÁLISIS Y ESPECIFICACIÓN ORIENTADA A OBJETOS

El análisis orientado a objetos examina y representa los requisitos desde la perspectiva de los objetos que se encuentran en el dominio.

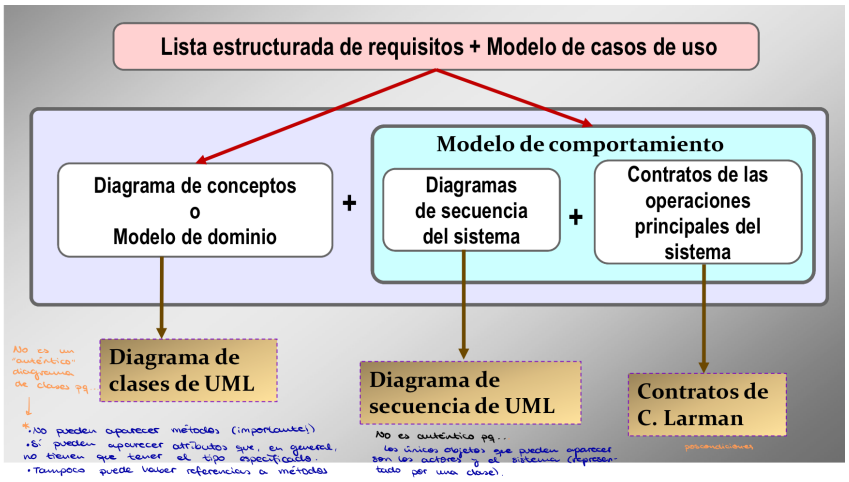
Los métodos de análisis orientado a objetos se centran en obtener dos tipos de modelos: 1) estáticos o de estructura y 2) dinámicos o de comportamiento. El lenguaje más usado para representarlos es UML.

¿Por qué usar análisis orientado a objetos?

- Los términos usados en los modelos están cercanos a los del mundo real
    - Facilita y mejora la obtención de requisitos
    - Acerca el espacio del problema al espacio de la situación
- Se modelan tanto elementos y propiedades estáticas como dinámicas del ámbito del problema
- Se manejan conceptos comunes durante el análisis, diseño e implementación del SW, por lo que
    - Hay una mejor transición entre fases
    - Se facilita el desarrollo iterativo
    - Se difumina la barrera entre el *qué* y el *cómo*

El modelo de análisis

La tarea principal es comprender, identificar y representar mediante modelos los requisitos del SW a desarrollar. Queremos representar *conceptualmente*. Un esquema de lo que queremos obtener sería el siguiente:



3. OBTENCIÓN DEL MODELO ESTÁTICO

El proceso general consta de:

- Identificar los principales conceptos y sus relaciones, y documentarlos
- Partir del modelo de CU, de la lista de requisitos y del glosario de términos
- Representarlos con un diagrama de clases de UML en el que podrá haber:
    1) Conceptos o clases conceptuales
    2) Asociaciones entre conceptos
    3) Generalizaciones de conceptos
    4) Atributos de los conceptos

Los pasos a seguir para construir esa representación serán:

1) Identificar e incorporar conceptos
2) Identificar e incorporar asociaciones
3) Identificar e incorporar generalizaciones
4) Identificar e incorporar atributos
5) Estructurar el modelo

1) Identificar e incorporar conceptos

Pasos a seguir: 1) identificar los conceptos, 2) seleccionar los conceptos relevantes en el problema y 3) representarlos como clases en el diagrama de conceptos.

Algunas estrategias para identificar correctamente los conceptos son:

- Establecer una lista de categorías de conceptos y rellenarla a partir de la información de la que se dispone.

  Por ejemplo, “actores y agentes participantes”, “lugares”, “organizaciones”, “cosas tangibles”, “cosas no tangibles”, “documentos físicos o virtuales”, “especificaciones, reglas y descripciones”, “transacciones”, “ítems de una transacción”, “eventos”, contenedores de cosas”, “ítems del contenedor”, “tipo o categoría de cosas” y “otros sistemas externos”.

- Encontrar los términos que se correspondan con sustantivos o frases nominales. Estos serán candidatos a conceptos.

  No hay que identificar sustantivos de forma mecánica (identificar sinónimos para evitar repeticiones, separar en conceptos y atributos…). Además, hay que tener presente que pueden generarse problemas a raíz de la ambigüedad del lenguaje natural.

2) Identificar e incorporar asociaciones

Una asociación es una conexión significativa y relevante entre conceptos. Pasos a seguir: 1) Identificar las posibles asociaciones, 2) representarlas en el diagrama y seleccionar las que sean válidas, 3) asignarles nombre, 4) identificar la multiplicidad.

Algunas estrategias para identificar correctamente las asociaciones son:

- Seguir una lista de categorías de relaciones entre conceptos.

  Véase

A es una parte física de B	A es una sub-unidad organizacional de B
A es una parte lógica de B	A usa o dirige B
A está contenido físicamente en B	A se comunica con B
A está contenido lógicamente en B	A se relaciona con una transacción B
A es una descripción de B	A es una transacción relacionada con otra transacción B
A es un elemento de línea en una transacción B	A está contiguo a B
A conoce / introduce / registra / presenta / captura B	A es propiedad de B
A es miembro de B	

- Identificar conceptos relacionados. Ver ejemplo Universidad (diapositiva 23).

Cuando una relación es redundante (es decir, se puede obtener directamente y por transitividad, por ejemplo) no hay que implementarla doblemente.

Una vez identificadas y seleccionadas las asociaciones, les damos nombre. Los nombres pueden repetirse si se refieren a conceptos distintos, lo importante es que no presenten ambigüedad. También podemos dotarlas de sentido ("navegabilidad"), pero no es necesario. Por último, asignamos multiplicidad.

3) Incorporar generalizaciones

Ya tenemos los conceptos y las asociaciones, nos tocan las generalizaciones. Los pasos a seguir son:

1) Identificar posibles generalizaciones

- A partir de la descripción del problema y de las clases conceptuales identificadas, encontrar clases conceptuales con elementos comunes
- Definir las relaciones de superclase (concepto general) y subclase (concepto mas específico)

2) Validar las estructuras encontradas. Una subclase potencial debería estar de acuerdo con

- La regla del 100% (conformidad con la definición de la superclase, todas sus características las tiene que tener la subclase)
- La regla "es-un" (conformidad con pertenencia al conjunto que define la superclase, un elemento de ella es un elemento de alguna de las subclases)

3) Representarlas en el modelo conceptual

Algunas estrategias para identificar correctamente las generalizaciones son:

Para crear subclases conceptuales a partir de superclases Especialización

- La subclase tiene atributos adicionales de interés
- La subclase tiene asociaciones adicionales de interés
- La subclase funciona, reacciona o se manipula de manera diferente a la superclase o a alguna subclase

Para crear superclases conceptuales a partir de subclases potenciales Generalización

- Cuando las subclases presentan variaciones de un concepto similar
- Las subclases cumplen con las reglas del "100%" y "es-un"
- Todas las subclases tienen el mismo atributo que se puede factorizar en la superclase
- Todas las subclases tienen la misma asociación que se puede factorizar en la superclase

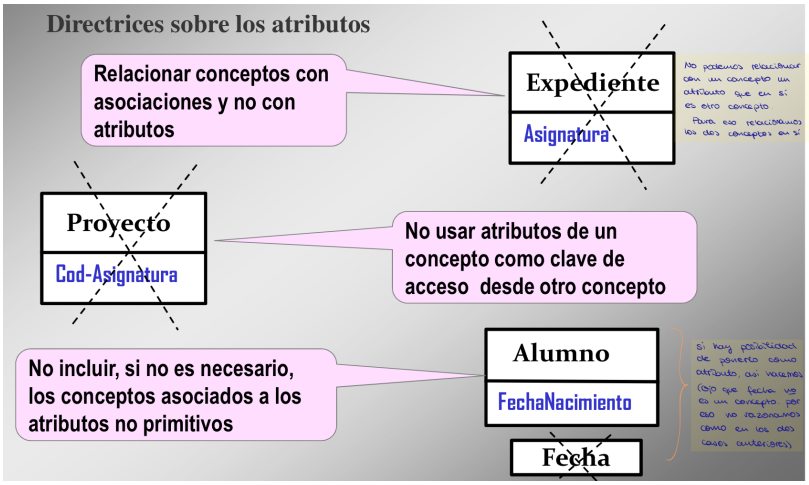
4) Agregar atributos

Pasos a seguir: 1) identificar atributos desde casos de uso, lista de requisitos y otras fuentes de información (documentos, impresos…) y 2) representarlos en el diagrama, en los conceptos o en las relaciones que correspondan.

Los tipos de atributo válidos son:

- Primitivos o valores puros de datos: entero, real, carácter, booleano, cadena
- No primitivos: nombre de persona, número de teléfono, hora, fecha…

A tener en cuenta:



5) Estructurar el modelo

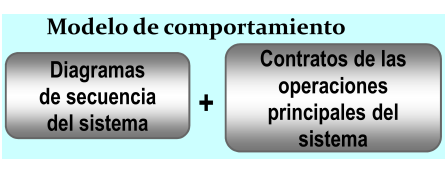
Se hace mediante diagramas de paquetes. Un paquete es una división del modelo agrupando conceptos que tienen una fuerte relación entre sí. Esto facilita el modelado y la posterior representación mediante diagramas. Los <<import>> entre paquetes NO pueden general ciclos.

Guía para estructurar el diagrama de conceptos o modelo de dominio:

- Elementos que están en el mismo área de interés (relacionados por conceptos)
- Están juntos en una jerarquía de clases (generalizaciones)
- Participan en los mismos casos de uso
- Están fuertemente asociados

4. OBTENCIÓN DEL MODELO DE COMPORTAMIENTO

Es un estudio adicional del dominio del problema en el que se añaden los requisitos funcionales al modelo del análisis. “Qué hace el sistema sin explicar cómo lo hace”.



4.1. DIAGRAMA DE SECUENCIA DEL SISTEMA

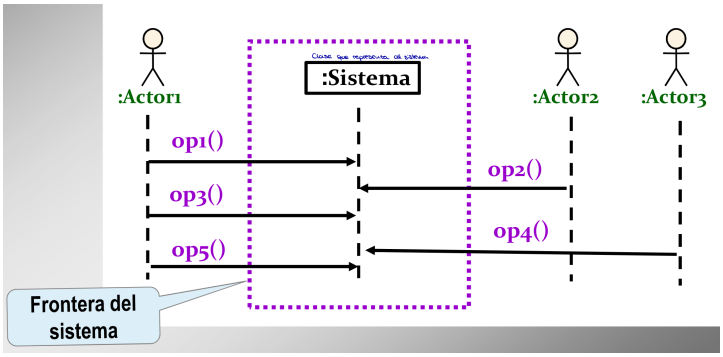


Diagrama de secuencia de UML en el que se muestran cómo los eventos generados por los actores provocan la ejecución de una operación por el sistema, siento este visto como una caja negra (sólo podemos representar una clase que es el sistema y a los actores. No podemos poner cómo se elabora la respuesta del sistema ante una operación, si acaso, ponemos sólo la respuesta).

Se representa la línea de vida de los objetos, actores, las respuestas que dan a las distintas operaciones tanto los actores como el sistema… No se representa ninguna interacción entre actores. Los actores inician operaciones hacia el sistema y ya. No es necesario indicar el orden de las operaciones.

Como mínimo, un diagrama de secuencia por cada diagrama de caso de uso. Los pasos a seguir, para todos los CU, son:

1) Identificar los actores que inician las operaciones

2) Asignar un nombre a todo el sistema

3) Identificar y nombrar las operaciones principales del sistema a partir de las descripciones de los CU

4) Determinar los parámetros de las operaciones

5) Incluir las operaciones en la clase que identifica al sistema

6) Hacer una descripción informal de la funcionalidad de cada operación

Ver ejemplo cajero (diapositiva 41).

4.2. CONTRATOS

Un contrato es un documento que describe lo que una operación se propone lograr, sin decir cómo se conseguirá. Define la especificación de una operación sin entrar en su implementación y suele redactarse con un estilo declarativo (no usamos ninguna herramienta UML).
Una plantilla sería:

Nombre	<<Nombre de la operación y sus parámetros>>
Responsabilidad	<<Descripción informal de las responsabilidades que debe cumplir la operación>>
Tipo	<<Concepto, clase o interfaz responsable de la operación>>
Notas	<<Notas de diseño, algoritmo...>>
Excepciones	<<Casos excepcionales>>
Salida	<<Mensajes o datos que proporciona>>
Precondiciones	<<Suposición acerca del estado del sistema o de los objetos del modelo conceptual antes de ejecutar la operación>>
Poscondiciones	<<Estado del sistema o de los objetos del modelo conceptual después de la ejecución de la operación>>

Las excepciones son condiciones que se comprueban una vez hemos empezado a ejecutar la operación, por lo que la comprobación la hace esta misma. Si la excepción salta, se deja de ejecutar la operación; si no salta, se cumple la poscondición.

Las precondiciones se comprueban antes de empezar a ejecutar la operación (y ya no se hacen más comprobaciones de esa condición). Nos da igual quién realice la comprobación. Puede darse el caso de que la operación ni siquiera llegue a ejecutarse.

Una condición la ponemos o bien como excepción o bien como precondición, pero nunca como ambas (no queremos redundancia). La diferencia (fundamental) radica en el momento de la comprobación de la condición. Generalmente, da igual qué opción escoger pero, a veces, dependiendo de la implementación, una cosa es mejor que otra.

Algunas directrices de cara a la elaboración de un contrato:

- El nombre de la operación viene del diagrama de secuencia del sistema correspondiente
- Comenzar con las responsabilidades, describiendo informalmente el propósito de la operación, continuar con las poscondiciones y finalizar con las demás secciones, especialmente con las precondiciones y excepciones
- Las poscondiciones deben describir los cambios de estado de un sistema no sus acciones (espíritu escenario-telón), estos son
  - Creación y destrucción de objetos
  - Creación y destrucción de enlaces
  - Modificación de atributos

  "Los objetos y enlaces que se pueden crear y destruir son los que están en el modelo conceptual"
- Las poscondiciones deben expresarse mediante una frase verbal en pretérito

Para especificar las poscondiciones, hay que identificar en el diagrama de conceptos los objetos que intervienen en la operación.

Ver ejemplo Universidad (diapositiva 51).