Tema 4: Captura y Análisis de Requisitos

Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática

Objetivos

Este tema tiene como objetivos principales:

- Conocer la actividad de captura de requisitos, sus fuentes y problemas asociados.
- Entender los diferentes puntos de vista en un sistema de información.
- Aprender diferentes tipos de técnicas de captura de requisitos y cómo aplicarlas según el sistema.
- Entender la importancia del análisis de requisitos dentro de la fase de descubrimiento de requisitos.
- Conocer las tareas implicadas dentro del análisis de requisitos.
- Aplicar diferentes métodos para el análisis de requisitos para resolver posibles conflictos, solapamientos y errores.
- Comprender por qué es importante establecer los límites de un sistema y modelar su contexto.
- Introducirse en el Lenguaje Unificado de Modelado (UML) y aprender cómo puede usarse para desarrollar modelos del sistema.
- Comprender los conceptos del modelado de objetos y datos a través de casos de usos, diagramas de clase, de interacción y de estados.

Análisis de Requisitos

La actividad del análisis de requisitos tiene como objetivos principales detectar y resolver conflictos entre los requisitos, permitir descubrir los límites del software y su interacción con el entorno y, elaborar los requisitos del sistema de cara a la posterior obtención de los requisitos software

Las siguientes secciones describen en profundidad varios aspectos sobre la actividad del Análisis de requisitos: 1) las tareas que se recomiendan dentro de esta actividad, 2) las técnicas que se utilizan para el análisis de requisitos.

En el análisis se pueden emplear métodos débiles para comprobar rápidamente la calidad de los requisitos obtenidos durante la educción. Entre ellos estudiaremos los más utilizados: checklist de análisis y la matriz de interacción.

Por último, profundizaremos en el modelado conceptual en el análisis, el cual nos permite comprender el sistema existente. Las técnicas de modelado conceptual son la principal herramienta del análisis de requisitos.

María Teresa García Valverde - Tlf: (+34) 968 27 86 57

Técnicas para análisis de requisitos

Entre las técnicas de análisis de requisitos se encuentran los métodos débiles de análisis. Estos métodos son un conjunto de técnicas simples que permiten comprobar rápidamente la calidad de los requisitos obtenidos durante la educción. Las técnicas más utilizadas son el checklist de análisis y la matriz de interacción.

Listas de Comprobación (CheckList)

Un checklist de análisis es, simplemente, un conjunto de preguntas que el analista debe considerar para cada requisito individual. Estas preguntas están relacionadas con atributos de calidad.

Un ejemplo de checklist elaborado por Kotonya y Sommerville (1998) sería el siguiente:

Atributo de calidad a considerar	**Pregunta**
Independencia del diseño	• ¿La lista de requisitos anticipa el diseño o incluye información de implementación?
Concisión	• ¿Cada requisito es simple o, por el contrario, podría dividirse en varios requisitos? • ¿Existe algún requisito que no parezca añadir ninguna información útil acerca del sistema a desarrollar?
Realizabilidad	• ¿Es realizable el requisito en la plataforma de implementación? • ¿Existe algún requisito irrealizable con la tecnología actual? **NOTA**: Para responder a esta pregunta, es necesario conocer los aspectos técnicos de la plataforma donde se implementará el sistema.
Consistencia externa	• ¿Existe algún requisito que contradiga requisitos organizativos explícitamente formulados?
Ambigüedad	• ¿Es posible interpretar de varias formas un requisito?
Verificabilidad	• ¿Es posible idear algún caso de prueba que permita establecer que el requisito NO SE CUMPLE?

Otro ejemplo similar sería:

Diseño prematuro	¿Requiere el requerimiento un diseño prematuro o información sobre su implementación?
Requisitos atómicos / combinados	¿La descripción del requerimiento lo es de un único requerimiento o puede descomponerse en varios requerimientos?

María Teresa García Valverde - Tlf: (+34) 968 27 86 57

Requisitos innecesarios	¿Es el requerimiento un aditamento cosmético al sistema que no es necesario?
Uso de hardware no standard	¿Es necesario HW o SW no standard para implementar el requerimiento?
Conformidad con los objetivos propuestos	¿Es el requerimiento consistente con los objetivos de negocio planteados?
Requisitos ambiguos	¿Pueden interpretar personas diferentes de modo diverso el mismo requerimiento? ¿Qué interpretaciones son posibles?
Requisitos realistas	¿Es realista el requerimiento dada la tecnología en la que deberá implementarse el sistema?
Requisitos testeables	¿Podrá generarse un juego de pruebas para testear si el sistema incluye el presente requerimiento y es conforme a la especificación?

Uno de los principales problemas de esta técnica es la imposibilidad de localizar defectos en los requisitos. Es decir, se aplica el checklist, pero no detecta ningún requisito defectuoso. Esto puede deberse a dos factores principalmente. En primer lugar a que el analista posee mucha experiencia y ha confeccionado adecuadamente la lista preliminar de requisitos y, en segundo lugar, y más probable, a que el analista no es capaz de descubrir defectos en la lista preliminar de requisitos por que ha sido él mismo el que la ha confeccionado. La solución a este problema consistiría en que el checklist fuera aplicado por una persona distinta al analista, con lo cual evitaríamos el problema de la subjetividad del analista.

Matrices de interacción

La técnica de análisis matrices de interacción consiste en una matriz de doble entrada donde se cruzan todos los requisitos entre sí. Por cruzar, debe entenderse que para cualesquiera dos requisitos r1 y r2, se debe comprobar si:

- r1 se solapa con r2, esto es, r1 trata aspectos del sistema también tratados en r2. Este caso implicaría un problema de redundancia. Lo denotamos mediante S.
- r1 está en conflicto con r2, esto es, r1 y r2 son contradictorios. Este caso implicaría un problema de consistencia interna. Lo denotamos mediante C.

De esta forma la matriz quedaría de la siguiente forma, reflejando los posibles problemas de redundancia y consistencia:

María Teresa García Valverde - Tlf: (+34) 968 27 86 57

	r_1	r_2	...	r_n
r_1		C		
r_2				S
...				
r_n				

Una vez realiza la matriz debemos resolver los diferentes problemas S y C. Para ello realizamos los siguientes pasos:

- Si r1 y r2 están solapados: el analista debe condensar los requisitos r1 y r2 en un nuevo requisito r1+2, que refleje los contenidos de los requisitos r1 y r2. Esto implica condensar los requisitos, con lo cual se disminuye la redundancia y los requisitos tienden a dejar de expresarse como requisitos de usuario y se describen como requisitos del sistema.
- Si r1 y r2 están en conflicto, la solución es más compleja. Estos problemas pueden deberse a un problema de expresión, o algún defecto de orden menor en la lista preliminar de requisitos o a conflictos en la concepción del sistema software por parte de los usuarios, lo cual implicará que analista y usuarios negocien los requisitos para establecer una solución al conflicto.

La técnica de matrices de interacción adolece del mismo problema que los checklist, es decir, la imposibilidad del analista de verificar sus propias creaciones. Por tanto, lo más adecuado será que sea otra persona la encargada de detectar los conflictos y solapamientos. Además, para realizar este tipo de técnicas se debe tener un buen conocimiento de la aplicación para poder realizar dicha tarea. Por último, otro de los problemas de este tipo de técnicas es su dificultad para manejar grandes conjuntos de requisitos. Esto podemos resolverlo dividiendo el sistema en subsistemas más pequeños.

Modelado conceptual

El modelado conceptual de los requisitos busca la comprensión del problema antes de iniciar el diseño de la solución. Una de las principales técnicas de modelado de requisitos son los diagramas de casos de uso. Los diagramas de casos de uso son uno de los diagramas definidos por UML.

UML son las siglas de Unified Modeling Language. UML es un lenguaje visual que permite visualizar, especificar, construir y documentar los artefactos de sistemas software, incluyendo su estructura y diseño. Actualmente está aceptado como un estándar de facto y puede ser usado para el diseño software (para lo que surgió en principio), para modelar procesos de negocio, documentar un sistema, proceso u organización existente y para capturar requisitos.

Aunque UML define muchos diagramas para modelar distintos aspectos/vistas del sistema, en esta asignatura nos centraremos en los casos de uso y diagramas de casos de uso.

Puntos clave

En este apartado se muestran aquellos puntos más importantes que el alumno debería tener claros al finalizar este capítulo:

Primera parte:

- Las técnicas contextuales trabajan sobre las personas, se ocupan del lenguaje tácito.
- Las técnicas guiadas por modelos proveen un modelo específico del tipo de información a capturar, y utilizan este modelo para orientar el proceso de elicitación.
- El prototipado es una herramienta para clarificar requisitos poco claros. Además de para la captura de requisitos también son utilizados en su validación.
- Cuando sea posible, también son útiles otras técnicas para la captura de requisitos como la ingeniería inversa y la reutilización.

Segunda parte:

- El análisis de requisitos persigue analizar los requisitos para detectar y resolver conflictos entre requisitos y descubrir los límites del sistema y su interacción con el entorno.
- El análisis de requisitos conllevo su clasificación, modelado conceptual, localización y posterior negociación.
- En el análisis se pueden emplear métodos débiles para comprobar rápidamente la calidad de los requisitos obtenidos durante la educción. Los más utilizados son el checklist de análisis y la matriz de interacción.
- El modelado conceptual en el análisis permite comprenden el sistema existente.

Tercera parte:

- Un modelo es una vista abstracta de un sistema que prescinde de algunos detalles del mismo. Pueden desarrollarse modelos del sistema complementarios para presentar otra información sobre dicho sistema.
- Los modelos de objetos describen las entidades lógicas del sistema y su clasificación y agregación.
- Los casos de uso describen como las personas interactúan con el sistema.
- Los diagramas de clases ofrecen una perspectiva conceptual del sistema, que nos aporta conocimiento sobre el dominio.
- Los modelos de secuencia muestran las interacciones entre actores y objetos en un sistema se utilizan para modelar el comportamiento dinámico.

María Teresa García Valverde - Tlf: (+34) 968 27 86 57

- Los modelos de colaboración resaltan la organización estructural de los objetos que envía y reciben mensajes. Muestran explícitamente las relaciones entre roles
- Los diagramas de estado modelan clases cuyo comportamiento es dinámico y cuyos objetos atraviesan una serie de estados en respuesta a estímulos recibidos, junto con sus respuestas y acciones

María Teresa García Valverde - Tlf: (+34) 968 27 86 57

Bibliografía

(Booch et al., 2006) *El lenguaje unificado de modelado*. Grady Booch, James Rumbaugh, Ivar Jacobson. Pearson educación, 2006.

(Durán, 2000) Tesis doctoral: *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Amador Durán Toro. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2000.

(Kendall & Kendall, 2005) *Análisis y Diseño de Sistemas.* Kenneth E. Kendall & Julie E. Kendall. Pearson Educación, 2005.

(Leffingwell, 2003). *Managing Software Requirements: A Use Case Approach*, 2/E. Dean Leffingwell, Don Widrig. Pearson Education, 2003

(Muñoz C., 2002) *Auditoría en sistemas computacionales*, Carlos Muñoz Razo. Pearson Educación, 2002.

(Nuseibeh et al., 2000). *Requirements Engineering: A Roadmap*. Bashar Nuseibeh and Steve Easterbrook. Proceedings of the Conference on the Future of Software Engineering. ACM, 2000

(Sanchez Alonso et al., 2011) *Ingeniería del software Un enfoque desde la guía SWEBOK*. Salvador Sánchez, Miguel Ángel Sicilia, Daniel Rodríguez. 2011. ISBN: 978-84-9281-240-0.

(Sommerville, 2005) *Ingeniería del Software*. Ian Sommerville. Pearson Educación, 2005.

María Teresa García Valverde - Tlf: (+34) 968 27 86 57