

M-TAR: Metodología para el Tratamiento de Aspectos en Requerimientos

Xavier S. Medianero P.¹, Sérgio Crespo², Clifton Clunie³

¹ Estudiante, ² Profesor Investigador, Universidad Tecnológica de Panamá

² Profesor Investigador en UNISINOS

^{1,3}{xavier.medianero, clifton.clunie}@utp.ac.pa,

²crespo.unisinos@gmail.com

Resumen- Este artículo propone una metodología cuyo propósito es el tratamiento de aspectos en la etapa de diseño de la Ingeniería de Requerimientos, mediante procesos de identificación, integración y especificación de los componentes de los aspectos, encapsulación de funcionalidades y resolución de conflictos. La metodología propuesta presenta una normativa basada en otros métodos de Ingeniería de Requerimientos Orientadas en Aspectos, incorporando elementos de los enfoques de puntos de vistas, por casos de uso, orientación a objetivos, entre otros. Este trabajo contiene una versión inicial para la metodología basada en cuatro etapas que permite la integración de los aspectos y sus componentes como extracción de identificadores, influencias y dependencias presentes en los modelos rutinarios de la especificación de requerimientos.

Palabras Claves- Ingeniería de Requerimientos; Programación Orientada a Aspectos; Desarrollo de Software.

1. Introducción

En todo proceso de desarrollo de software se tiene como prioridad cumplir con un nivel de calidad sin perjudicar el costo o el tiempo del mismo, incluyendo tareas de análisis, diseño e implementación del mismo. La ingeniería de software es una disciplina que nace como medida para tratar de garantizar un producto de calidad que cumpla con los estándares exigidos mediante un procedimiento estructurado y cíclico de continua mejora de software tanto en diseño, implementación y mantenimiento del mismo.

En el desarrollo de software basado en programación orientada a aspectos (POA), para obtener un software que cumpla con estándares de calidad, es necesario analizar y diseñar el software enfocado a la identificación temprana de aspectos y de esta manera incluirlo en todos los componentes que sean necesarios.

La Ingeniería de Software consta de varias etapas, cada una de ellas presenta un grado significativo de importancia para el proceso, siendo importante destacar el fuerte impacto de la Ingeniería de

Requerimientos en todo el proceso de desarrollo de software en las etapas posteriores y en los resultados finales, por tal motivo este trabajo se enfoca a esta etapa del proceso.

En la Ingeniería de Requerimientos se realizan las interpretaciones de las necesidades y requerimientos del cliente, se crean los documentos con la información detallada del software, se desarrolla el

modelo lógico del sistema entre otras actividades.

La especificación correcta de funcionalidades y la descripción de los componentes del software en la Ingeniería de Requerimientos permitirá crear una base robusta para las etapas posteriores reduciendo la complejidad de las mismas. La incorporación de la Programación Orientada a Aspectos, uno de los actuales paradigmas de programación cuya función es la encapsulación de funciones en elementos denominados aspectos en beneficio de procesos de mantenimiento, depuración y reducción del acoplamiento, en la ingeniería de requerimientos, permitirá incorporar desde etapas tempranas este paradigma y sus beneficios al diseño y productos de la fase.

Según [1], el problema radica en la necesidad de una metodología con normas más acopladas que constituyan un procedimiento para el mejoramiento del diseño y estructura del sistema integrando los aspectos desde una etapa temprana, de tal forma que sea posible obtener un diseño más robusto que sirva de base para las etapas futuras, mejorando de esta forma la calidad a través de un procedimiento sistemático. La metodología que se presenta en este trabajo radica en la necesidad de incorporar a las etapas de análisis y diseño de sistemas de software atributos de la programación orientada a aspectos y de esta forma, abstraer todas las ventajas de la misma para consolidar una base sólida para etapas posteriores de la Ingeniería de Software.

En este artículo se propone una primera versión de una metodología para la consideración de aspectos en la etapa inicial de la Ingeniería de Software, la misma combina métodos y pasos provenientes de otras metodologías como la Ingeniería de Requerimiento Orientada a Aspectos, a Componentes y Aspectos, Modelos de Puntos de Vistas, Ingeniería de orientación a Aspectos utilizando Casos de Uso.

La separación de los asuntos y requerimientos del sistema en aspectos es uno de los aportes más importantes del paradigma de POA [2], debido a esto se han creado disciplinas como lo son la AORE [3], SLAI [1], AOCRE [4] que proponen mecanismos para el análisis de los aspectos en la etapa de Ingeniería de Requerimientos.

El procedimiento que aplica M-TAR consta de cuatro etapas cíclicas para la identificación y especificación; el propósito de la misma es el de establecer una serie de pasos que refuerzan el tratamiento y la abstracción de los elementos de los aspectos dentro del proceso de diseño del sistema de la Ingeniería de Requerimientos en la aplicación de la Ingeniería de Software, mediante la utilización de prácticas rutinarias como casos de Uso, enfoque a objetivos y puntos de vistas.

2. Desarrollo de Software Orientado a Aspectos

El proceso de separación de asuntos es un procedimiento indispensable en la etapa de análisis y diseño de la ingeniería de requerimientos [5]; Esta técnica consiste en la separación de dominios del problema, logrando los siguientes beneficios: Disminuir fallas y evitar o facilitar la resolución de conflictos que podrán presentarse en etapas posteriores causadas por el mal entendimiento de las necesidades y objetivos del proyecto, reducir el grado de complejidad de las actividades de desarrollo [6].

De esta integración nace la disciplina AOSD (Desarrollo de Software Orientado a Aspectos) la cual fue creada para hacer frente a la

dificultad del código transversal debido a su complejidad y su dificultad para entenderse. Las principales ventajas que se logran mediante la integración de AOSD según [7], se tienen: reducción de esfuerzo en el desarrollo del software, un diseño más elaborado y robusto, facilita el mantenimiento y actividades de pruebas y depuración de código, abstrae y separa las funcionalidades encapsulándolas en aspectos, entre otras.

La disciplina AOSD se enfoca en la identificación, separación, representación y la composición de los asuntos de carácter transversal [1]. Estos asuntos son capturados en módulos independientes denominados aspectos.

A continuación se detalla la programación orientada a aspectos y los diversos enfoques de cómo son tratados en la AOSD.

3. Trabajos Relacionados

Trabajos relacionados en el área de la AOSD que intentan suplir la necesidad de una metodología sistemática para el tratamiento de aspectos en la Ingeniería de Requerimientos, entre ellos se tienen:

AORE (Ingeniería de Requerimiento Orientada a Aspectos)

AORE utiliza una metodología en la cual trata los asuntos como un conjunto coherente de requerimientos y donde los aspectos transversales son matrices conformado por los primeros elementos [6], de esta forma es posible determinar cuántas funcionalidades trabajan de manera negativa o positiva frente a otros asuntos.

Esta metodología basada en la manipulación de aspectos cuenta con las siguientes etapas [6, 7]:

Identificación de Asuntos: Separa los requerimientos funcionales y no funcionales del software. En esta etapa se utilizan Vistas, Enfoque a Casos de Uso, Orientación a Objetivos, Orientación a Aspectos, entre otras.

Especificación de Asuntos: Se evalúan la compatibilidad de cada uno de ellos como paso previo para su clasificación por funcionalidad.

Identificación y Especificación de Influencias: Los asuntos presenten en las matrices son alineados y clasificados según su nivel de influencia en otros elementos, creando un enfoque de vistas. En esta etapa se identifican aspectos candidatos, se especifican y priorizan los aspectos resultantes.

Manejo de Conflictos entre asuntos: Conforman las actividades que se llevarán a cabo para la solución de conflictos que surgen por las relaciones entre los asuntos.

Especificación de las dimensiones de los asuntos: Define cómo los aspectos serán soportados en la arquitectura según influencia y mapeo. En esta etapa se evalúa y se predice cómo se adaptarán los aspectos a las siguientes áreas del proceso ingenieril, en especial, frente al ciclo de cambio y redefinición de requerimientos; en este punto es necesario tomar en consideración los aspectos, las funciones y las decisiones para cada etapa del ciclo de vida.

SLAI (Léxico Estructurado para la Identificación de Aspectos)

Esta metodología tiene como objetivo principal la identificación de aspectos potenciales del software en la etapa de diseño de la Ingeniería de Software [1].

Su meta es la de reconocer los posibles puntos transversales de los requerimientos funcionales y no funcionales del software.

Su enfoque está basado en la utilización de casos de uso para el

proceso de integración de aspectos.

La Metodología de SLAI se basa en procesos separados para el manejo de requerimientos y así llevar a cabo la identificación de aspectos.

Para el primer grupo de requerimiento se identifican los actores, usuarios y sistemas que interactúan en el mismo, se crea un repositorio de identificadores determinantes considerados como palabras claves que denotan acciones. Al agregar funcionalidades se verifica si existen en el repositorio, de ser así, la misma puede llegar a considerarse como un aspecto candidato. Este procedimiento es cíclico y busca en optimizar los resultados.

Para el segundo grupo de requerimientos se utilizan funciones ya reconocidas e identificadas, se analizan las necesidades del cliente para extraer cualquier otra y luego la identificación es similar al de los requerimientos funcionales.

AOCRE (Ingeniería de Requerimientos y Componentes Orientada a Aspectos)

Se enfoca a la identificación y especificación de funciones transversales encapsulando los requerimientos en aspectos y a su vez relacionando cada uno de los componentes del sistema [4].

Dentro de sus objetivos está el análisis del comportamiento de los componentes basados en los aspectos que lo cortan, el enfoque de los aspectos dentro del sistema encapsulados por componentes. Se basa en un procedimiento de especialización desde lo más grande a lo más pequeño [4] es decir, un procedimiento de descomposición de los aspectos.

La metodología de AOCRE se basa en la identificación y determinación de aspectos mediante el ciclo de especialización jerárquico de funcionalidades, este proceso se realiza por cada componente, Una vez seleccionado los aspectos candidatos se comparan frente a los requerimientos funcionales y no funcionales para su última aprobación [8].

El Modelo ViewPoint (Puntos de Vista)

Está basado en la metodología de AORE a la cual se le integran los enfoques por puntos de vistas según funcionalidades y actores, para un tratamiento de aspectos más detallado y mejorado [9].

El modelo por puntos de vista se basa en procedimientos de extracción de funciones a partir de los requerimientos del sistema, para luego clasificar las vistas por estas funcionalidades, realizar diagrama de casos y escenarios, como también, la identificación de los requerimientos no funcionales. Posteriormente se realiza una resolución de conflictos para determinar qué función debe tener privilegios frente a otra dependiendo de las influencias, dependencias y estimaciones de los *stakeholders*.

La integración de vistas y aspectos hace que estos enfoques creen una estructura más sólida para el tratamiento de los requerimientos [9].

4. Metodología

MTAR es una propuesta sistemática para la identificación y especificación de aspectos en la Ingeniería de Requerimientos, que consta de cuatro etapas y está basada en metodologías como: AORE [3, 6], AOCRE [4], SLAI [1] y Modelo de Puntos de Vistas [9]. El proceso de identificación de aspectos está basado, por ende, en

Casos de Uso, Puntos de Vistas y enfoque a objetivos.

Cada una de estas etapas forman un procedimiento cíclico, que consta de los siguientes pasos y normativas:

4.1 Definición de Requerimientos

En esta etapa se llevan a cabo los siguientes pasos:

Enfoque a Vistas e Identificación de elementos claves: Involucra la orientación a vistas como primer requisito. Dependiendo de los requerimientos funcionales del sistema y de los actores que se ven involucrados en cada una de las funcionalidades, ya sean: usuarios, sistemas casi o completamente automatizados, se crea un enfoque a vistas por objetivos. De tal forma que de paralelo al procedimiento de creación de vistas se diseña una tabla con los elementos detallados en la Tabla 1.

Tabla 1. Elementos a tomar en consideración de la Orientación a Vistas.

Aspectos	Intervenciones	Acciones	Funciones Afectadas	Condiciones	Función Transversal
Nombre	Inicio Durante Fin	Acción Acción Acción específica	Función Caso de Uso	Parte del Proceso Condiciones	Función Global

Dependiendo a la concurrencia de los actores en el sistema, principalmente usuario se contabiliza en escala de 1-9 su relevancia sobre las funcionalidades del sistema. Una funcionalidad puede tener más de una vista asociada.

Elaboración de Casos de Uso: Con las vistas elaboradas se diseñan los casos de uso. Este procedimiento se realiza de la forma usual.

Contabilización de identificadores determinante de asuntos. Aplicando el método en [11], los identificadores (verbos y sustantivos significativos en el nombre de los casos de uso) se almacenan en un repositorio, de tal forma que sea posible determinar la frecuencia de cada uno de ellos dentro del procedimiento. Además, se incorpora una influencia de los identificadores sobre la cardinalidad de las funcionalidades y vistas, de tal forma que sea posible asociar también cuántos cortes transversales serán posibles realizar a una funcionalidad determinada. Retomando las etapas de definición de requerimientos, a partir de la segunda ejecución de la misma se realizan las siguientes actividades:

Actualización de Casos de Uso. Consiste en adaptar los casos de uso de la primera etapa incorporando el resultado final de las otras etapas del proceso en su primer fase. Este procedimiento es necesario debido a la segmentación de aspectos en sub-aspectos mediante el enfoque de especialización (top-down). El procedimiento cíclico involucra que se realice la sub-etapa de contabilización de identificadores determinante de asuntos de manera similar.

4.2 Identificación de Aspectos

Identificación de Influencias Mediante las influencias determinadas en la identificación de vistas se profundiza la identificación de influencias basadas en los diagramas de caso de uso, de tal forma, obtener qué función influye sobre otra.

A diferencia del procedimiento de AORE, las influencias en este

paso son de manera más específicas en lugar de globales, se determinan por los diagramas.

Aplicación de la matriz de descomposición. Una vez determinadas las influencias, se crea la matriz de descomposición, la cual determinará la cardinalidad de dependencias de un asunto determinado, siendo así posible, determinar los cortes transversales y los posibles aspectos candidatos. El procedimiento para determinar esta matriz será a partir de la regla de descomposición detallada dentro de la revisión bibliográfica del trabajo.

Determinación de Requerimientos No Funcionales Los requerimientos no funcionales se determinarán por su adaptabilidad en relación con el sistema en desarrollo. Este procedimiento se complementa con la identificación de RNF (Requerimientos no funcionales) a partir de licitaciones con los usuarios del sistema.

A continuación un listado de posibles funcionalidades que representan, generalmente, RNF:

- Disponibilidad de Servicios
- Nivel de Seguridad
- Rendimiento del Sistema
- Tiempo de Servicio
- Confiabilidad de Procesos
- Compatibilidad
- Capacidad de Usuarios Múltiples
- Casos Legales
- Adaptabilidad a la Red

En esta etapa se deben identificar los aspectos por nombres, determinar sus funciones, verificar los casos de uso que son impactados por los aspectos y especificar la función transversal (cross-cutting).

Considerando los elementos de los aspectos, podemos ubicar ciertos puntos críticos en la primera etapa (una vez los elementos de la POO se encuentren identificados).

Como apoyo a la regla de descomposición se utiliza el diccionario léxico: mediante la frecuencia de cada identificador se pueden determinar aspectos candidatos, este procedimiento tiene más relevancia para sistemas complejos.

A partir de la segunda intervención, la Determinación de RNF y la matriz de descomposición no se aplican debido a que su relevancia es mínima.

4.3 Especificación de Aspectos

Fragmentación de Aspectos La fragmentación de aspectos por el enfoque de especialización (top-down) se lleva a cabo después de la identificación de los cortes transversales, consiste en que, una vez identificados los aspectos en la etapa ii, se crea un listado de cada uno de ellos y de su funcionalidad asociada, se intenta descomponer las funciones en funciones más pequeñas y se trata de encontrar un aspecto asociado; de ser posible se segmenta el aspecto. En la Tabla 2 se muestran elementos que deben especificarse de los aspectos para facilitar el procedimiento de fragmentación.

Tabla 2. Modelo de especificación de aspectos.

Aspectos	Intervenciones	Acciones	Funciones Afectadas	Condiciones	Función Transversal
Nombre	Inicio Durante Fin	Acción Acción Acción específica	Función Caso de Uso	Parte del Proceso Condiciones	Función Global

Como otro método de segmentación de aspectos está su división (representación y especificación de *advices*, puntos de corte, puntos de unión) que se hace a partir de los siguientes pasos: Especificación de componentes de los aspectos: Como se muestra en la Figura [1], la especificación de los *advices*, puntos de corte y puntos de unión se lleva a cabo a través del siguiente procedimiento:

Cross-cutting: se visualizan en la relación directa entre los aspectos y los casos de uso que interceptan, es decir, dependiendo de la funcionalidad del caso de uso se conceptualizan dentro de una función transversal; representa la funcionalidad transversal en cada uno de los casos de uso al elemento que la genera.

Join-Point: Se asocian con un aspecto determinado, contiene los métodos que forman parte de la funcionalidad transversal. Se propone su identificación mediante la detección del diagrama de caso de uso involucrado al aspecto determinado, su ubicación debe estar entre el caso de uso afectado por el crosscutting y el anterior.

Cut-Point: Son el conjunto de los join-point, pero colocados en todos los casos de uso que intervienen en el proceso, es decir, la ubicación de los puntos de corte se dan en todos los casos de uso que ejecutarán en su siguiente etapa el corte transversal. También es necesario analizar los elementos anteriores al elemento de caso de uso analizado.

Advice: deben ubicarse en el caso de uso al que el aspecto interviene, los *advices* serán colocados de forma intermediaria entre los predecesores del caso de uso y el mismo, para los eventos before(), in() y after() en caso de ser necesarios, si no es así, simplemente se prescinden de ellos.

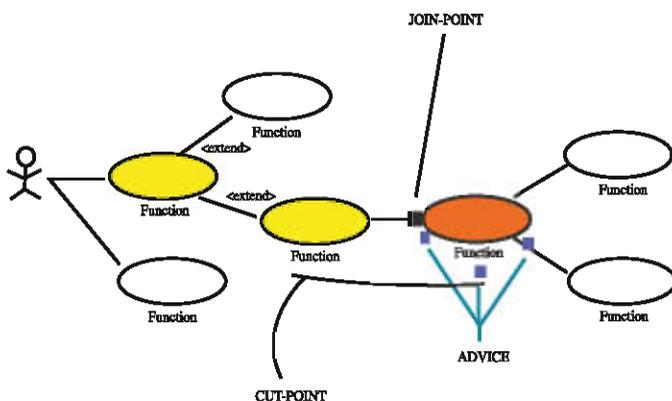


Figura 1. Ubicación de los puntos de corte dentro de un diagrama de casos de uso.

Clasificación de Aspectos: La clasificación de aspectos se realiza para filtrar los aspectos reales de posibles funciones y decisiones. A este proceso se le denomina "Mapeo".

En el *mapeo*, a cada aspecto ya definido, se le asocia las funcionalidades sobre las que influye y se separan de las funciones y decisiones del proceso.

Control de Redundancia: Una vez segmentado los aspectos, es necesario revisar si estos no causan redundancia con otros aspectos identificados, de ser así, se eliminan. Esta redundancia se puede visualizar con el listado de funcionalidades de cada aspecto. A partir de la segunda intervención, los procedimientos de segmentación de aspectos y especificación no son contemplados, ya que, la relevancia es mínima.

4.4 Resolución de Conflictos

En la resolución de conflictos se utiliza la "tabla de conflictos para *stakeholders*" aplicada en la metodología AORE y ViewPoint, modificada por el enfoque a actores del sistema.

La Tabla [3] es la tabla de conflictos para *stakeholders* con valoración de actores involucrados en el sistema, pero más significativo para esta evaluación son las influencias y dependencias de las funcionalidades en conflicto. Un porcentaje del máximo de influencias actuales se aplicará a cada funcionalidad en conflicto, de tal forma, que quien tenga más funciones dependientes merecerá más prioridad; en este punto es importante resaltar, que la magnitud de significancia es restada si una función en conflicto es predecesora de la misma, debido a que la otra función se ejecutará primero.

Tabla 3. Para cualquier intervención de esta etapa se aplica la misma metodología.

Funciones	Stakeholders	Actores	Número de	¿Antecede?
1 \ 2	Valoración	Valoración	Dependencias	

5. Conclusión

M-TAR es una metodología propuesta que está incorporando modelos para el tratamiento de aspectos en la Ingeniería de Requerimientos basada en AORE, SLAI, AOCRE y el Modelo ViewPoint, que son procedimientos donde se incluyen aspectos en el análisis de requisitos no funcionales y funcionales del sistema para el diseño del software.

Su enfoque está basado en el análisis de diagramas elaborados con casos de uso, componentes, orientación a objetivos y vistas; técnicas de descomposición de aspectos, abstracción de funcionalidades, entre otras.

La incorporación del paradigma de programación orientada a aspectos dentro de las etapas de diseño de la Ingeniería de Requerimientos permite agregar al modelo una estructura más sólida debido a que se podrá contar con todos los beneficios de la misma desde fases tempranas.

Las normas establecidas y la metodología conformada por la unión de características de otros modelos ha resultado un método alternativo para el tratamiento de aspectos.

Consta de procesos como Identificación de aspectos en casos de uso, segmentación de vistas, resolución de conflictos, método determinación de puntos de corte, puntos de unión, advice, entre otros elementos.

Referencias

[1] C. C. Budwell and F. J. Mitropoulos, "The SLAI Methodology: An Aspect Oriented Requirement Identification Process," in 2008 International Conference on Computer Science and Software Engineering, Wuhan, Hubei, 2008, pp. 296 - 301

- [2] H. Hu, et al., "An AOP Framework and Its Implementation Based on Conceptual Model," ISECS International Colloquium on Computing, Communication, Control and Management, p. 4, 2009 2009.
- [3] M. Aoyama and A. Yoshino, "AORE (Aspect-Oriented Requirements Engineering) Methodology for Automotive Software Product Lines," presented at the Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific, Beijing, 2008.
- [4] J. Grundy, "Aspect-oriented requirements engineering for component-based software systems," in IEEE International Symposium on Requirements Engineering, 1999., Limerick, 1999, pp. 84 - 91.
- [5] D. Dahiya and R. J. Sachdeva, "Understanding Requirements: Aspect Oriented Software Development," in Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06), Chicago, IL, 2006, pp. 303 - 308.
- [6] M. Tabares, et al., "Aspect Oriented Software Engineering: An Experience of Application in Help Desk Systems," Dyna rev.fac.nac.minas, vol. 147, 2007.
- [7] A. Rashid, et al., "Early aspects: a model for aspect-oriented requirements engineering," presented at the Proceedings. IEEE Joint International Conference on Requirements Engineering, 2002, 2002.
- [8] A. M. Reina and J. Torres. Components + Aspects: A General Overview. Revista Colombiana de Computación.
- [9] P. Yu-Ning and L. Qiang, "A Viewpoint-Oriented Requirements Elicitation Integrated with Aspects," presented at the World Congress on Computer Science and Information Engineering, 2009 WRI., Los Angeles, CA 2009



10^a Conferencia Internacional del Consorcio Latinoamericano y del Caribe de Escuelas de Ingeniería



Megaproyectos: Construyendo Infraestructura mediante Colaboración en Ingeniería, Integración Eficiente y Efectiva y Planificación Innovadora

Educación, Innovación, Tecnología, Diseño y Práctica

Julio 23-27, 2012

Ciudad de Panamá, Panamá

Organizado por:
LACCEI Consorcio Latinoamericano y del Caribe de Escuelas
de Ingeniería y la Universidad Tecnológica de Panamá

www.laccei2012panama.org.pa