

ANÁLISIS DE LA INGENIERÍA DE REQUISITOS ORIENTADA POR ASPECTOS SEGÚN LA INDUSTRIA DEL SOFTWARE

LUIS FERNANDO LONDOÑO*
RAQUEL ANAYA**
MARTA SILVIA TABARES***

RESUMEN

La aplicación de buenas prácticas en la gestión de requisitos de software es una condición fundamental para lograr productos de calidad. Por lo tanto, es importante que las compañías de desarrollo de software mantengan una investigación constante alrededor de nuevas técnicas que mejoren actividades de requisitos tales como levantamiento, especificación y modelado. En este artículo se presenta un estudio de los enfoques más representativos de la ingeniería de requisitos orientada por aspectos que podrían ayudar a resolver problemas que enfrentan las compañías de software durante las etapas tempranas de desarrollo. Se hace un análisis comparativo entre los enfoques, a partir de los criterios de alcance, trazabilidad, composición de requisitos, manejo de conflictos, mapeo, validación-verificación y escalabilidad. Los resultados obtenidos permiten detectar las oportunidades que el enfoque de aspectos ofrece para mejorar el proceso de los requisitos.

PALABRAS CLAVE: ingeniería de requisitos; requisitos; aspectos; intereses; intereses transversales; desarrollo de software orientado por aspectos; industrias del software.

* Gerente General, Avansoft S. A., Medellín, Colombia. lfondono@avansoft.com

** Doctor en Ingeniería de la Programación e Inteligencia Artificial, Universidad Politécnica de Valencia, España. Escuela de Ingeniería, Departamento de Sistemas, Universidad EAFIT. ranaya@eafit.edu.co.

*** Ph. D (c) en Ingeniería de Sistemas, Universidad Nacional de Colombia. Docente del Área de Ingeniería de Software y Bases de Datos, Escuela de Ingeniería de Antioquia. pfmstabare@eia.edu.co.

ABSTRACT

Applying good practices to manage software requirements is a basic condition to obtain quality products. Therefore, it is important that software companies maintain a constant research around of new techniques and mechanisms to improve requirements activities such as elicitation, specification, and modeling. In this paper, we present a study of most representative aspect-oriented requirement approaches in order to analyze the way how these could support issues confronted by software companies during early stages of the lifecycle. Thus, we make a comparative analysis among approaches under criteria defined such as scope, traceability, requirements composition, conflict management, mapping, validation-verification, and scalability. Results achieved allowing us to know opportunities offered by aspect-oriented approaches so as to improve the requirement process.

KEYWORDS: Requirement Engineering; requirements; aspects; concerns; crosscutting concerns; aspect oriented software development; software industries.

1. INTRODUCCIÓN

Buena parte de los problemas que surgen a lo largo del proceso de desarrollo se deben a la carencia de un proceso adecuado de definición y entendimiento del problema y a la definición poco clara de las necesidades del cliente. La importancia de la ingeniería de requisitos se comprueba en enfoques de mejoramiento del proceso como CMMI¹, que define la gestión y análisis de requisitos como unas de las prácticas básicas a la hora de evaluar el nivel de madurez de un proceso de desarrollo implantado en una organización. Los modelos de calidad establecen objetivos generales, prácticas y metas que sirven de guía a la organización y que, a su vez, ofrecen evidencias de que se tiene un proceso disciplinado y repetitivo de requisitos (*el qué*). La organización, por su parte, debe implementar o adoptar los métodos, prácticas y herramientas adecuados para llevar a cabo la compleja labor de captura, análisis y gestión de requisitos (*el cómo*) [1].

Las empresas de desarrollo de software deben mantener una investigación continua alrededor de mecanismos que les permitan aumentar la confiabi-

lidad de los requisitos, para disminuir los riesgos y los sobrecostos en el proceso de desarrollo. La investigación se debe orientar al uso de nuevas técnicas y enfoques que fortalezcan características tales como la agilidad en el tratamiento de los requisitos, la disminución de los conflictos entre los participantes, el reconocimiento oportuno de los errores o problemas en la identificación y especificación de los requisitos y el establecimiento de controles en su evolución en diferentes fases del ciclo de vida, etc.

El Desarrollo de Software Orientado por Aspectos (*Aspect Oriented Software Development –AOSD–*²) es un nuevo paradigma de desarrollo de software que se fundamenta en principios clásicos como la modularización y la separación de intereses (concerns) y centra su aplicación en el tratamiento de intereses transversales (crosscutting concerns). Surge básicamente como una propuesta de programación que en forma rápida fue extendiéndose en las otras fases del ciclo de desarrollo del software [2].

Específicamente, la ingeniería de requisitos orientada por aspectos (*Aspect-Oriented Software Requirement –AORE–*³) provee un conjunto de enfoques

¹ CMMI: Capability Maturity Model Integrated. <http://www.sei.cmu.edu/cmmi/>.

² <http://www.aosd.net>

³ <http://www.early-aspects.net/>



para gestionar intereses y requisitos transversales que podrían modularizarse para luego componerlos con otros intereses. Los enfoques que se han propuesto en esta línea tienen el propósito de fortalecer, en las etapas tempranas del desarrollo, el análisis de los intereses involucrados en un problema, así como el soporte a su evolución en la especificación del sistema, a la solución de conflictos y a la toma de decisiones de arquitectura [3-11].

En este artículo, se presenta un estudio de los enfoques más representativos de la ingeniería de requisitos orientada por aspectos que podrían ayudar a resolver problemas que ahora enfrentan las compañías de software durante las etapas iniciales de desarrollo. Se hace un análisis comparativo entre los enfoques con criterios definidos por grupos de requisitos y calidad de empresas de desarrollo, permitiendo conocer y evaluar nuevas características para mejorar el proceso de los requisitos. Los criterios seleccionados para la comparación son: alcance o nivel de cobertura en el tratamiento de requisitos, trazabilidad, composición de requisitos, manejo de conflictos, mapeo, validación-verificación y escalabilidad.

El artículo está organizado de la siguiente forma: en la sección 2 se provee una visión general de los enfoques estudiados. En la sección 3 se presenta el análisis comparativo de los enfoques. En la sección 4 se presentan trabajos relacionados. Finalmente, en la sección 5, se concluye y se describen trabajos actuales y futuros que apoyan la investigación.

2. LOS ENFOQUES DE LA INGENIERÍA DE REQUISITOS ORIENTADA POR ASPECTOS

Los enfoques orientados por aspectos asociados a la ingeniería de los requisitos proponen mecanismos que facilitan la identificación, separación y clasificación de intereses y la composición de intereses transversales. Para realizar el comparativo,

se estudiaron los siguientes enfoques: separación multidimensional de intereses [6], aspectos en modelos de objetivos de requisitos (ARGM) [12, 13], identificación de aspectos en los requisitos Theme/Doc [7, 8] y AOSD con casos de uso [9].

En 2.1 a 2.4 se presentan generalidades de los enfoques que se analizarán en la sección 3.

2.1 Separación multidimensional de intereses (MDSOC)

Este enfoque es pionero en el tratamiento de los aspectos en la fase de requisitos. Se basa en el enfoque de Puntos de Vista (*Viewpoints* [20]) y su fortaleza está centrada en el tratamiento de las reglas de composición de los intereses, así como en la definición y manejo de conflictos. La orientación multidimensional permite analizar el espacio de intereses del problema de forma homogénea. Para lograr esto define un interés como la agrupación de requisitos funcionales y no funcionales de cada punto de vista del sistema [8].

2.2 Aspectos en modelos de objetivos de requisitos (ARGM)

Este enfoque integra las propuestas I* [13] y NFR (*Non-Functional Requirements Framework*) [14] agregando el concepto de aspectos. Utiliza un grafo en forma de V (llamado V-Graph) que contiene en sus vértices superiores objetivos funcionales (*goals*) y objetivos de calidad (*softgoals*) que pueden cruzar diferentes objetivos funcionales; en su vértice inferior se ubican las tareas u operaciones que debe realizar el sistema. Define un proceso sistemático e iterativo que va descomponiendo los objetivos hasta llegar al nivel de las tareas. Durante el proceso de descomposición se verifica el nivel de contribución de dichas tareas para el cumplimiento de los objetivos de calidad, lo cual permitirá detectar los conflictos que puedan presentarse y tomar acciones al respecto; si una tarea se encuentra relacionada con más de un objetivo, se convierte en candidata a aspecto [12].

2.3 Identificación de aspectos en los requisitos: Theme/Doc

Theme/Doc es la primera fase del enfoque Theme que describe un proceso de desarrollo orientado por temas. La aproximación se centra en el concepto de tema que representa una pieza de funcionalidad, asunto o interés que se quiere modelar de forma separada del sistema. Theme/Doc parte de una descripción textual de los requisitos en la cual se realiza un análisis gramatical para detectar las relaciones entre los requisitos y las acciones (*minor actions*); a partir de estas acciones se derivan las acciones de mayor granularidad (*major action*) o asuntos que representan la solución de software y que entran al proceso de diseño siguiendo las características de Theme/UML [7, 8].

2.4 AOSD con casos de uso (AOSD/UC)

Este enfoque trata los requisitos funcionales desde los casos de uso que representan la función básica del sistema (*peer use cases*). Los requisitos no funcionales se representan en casos de uso que extienden un *caso de uso de infraestructura*, el cual se parametriza al igual que el actor que lo usa (los parámetros representan el comportamiento que se cruzará en la funcionalidad de los casos de uso base). En el análisis y el diseño los casos de uso se representan en una estructura de composición que se identifica con el estereotipo *<use case slice>* y agrupa elementos de modelo que colaboran para lograr los requisitos del sistema (funcionales y no funcionales) [9].

3. ANÁLISIS COMPARATIVO DE LOS ENFOQUES

Para las industrias de desarrollo de software es importante lograr una investigación constante alrededor de nuevos enfoques de ingeniería de

requisitos que provean mecanismos ágiles y seguros para el levantamiento, especificación y modelado de los requisitos. El análisis se realizó en una empresa de desarrollo⁴ de la ciudad de Medellín, Colombia, la cual proporcionó el personal calificado de las áreas de investigación, calidad y desarrollo. Fueron capacitados en los enfoques considerados para el análisis, y algunos de ellos ya tenían conocimientos en *middlewares* y la programación orientada por aspectos. Los enfoques en estudio se aplicaron en tres casos reales de desarrollo de categoría media para lograr comprender con facilidad los modelos o métodos propuestos e identificar algunas fortalezas y debilidades de los enfoques. Los criterios usados para la comparación fueron definidos por el grupo con base en la necesidad de disminuir los riesgos durante la ingeniería de los requisitos.

3.1 Definición de los criterios

- **Alcance en el tratamiento de requisitos.** Capacidad o nivel de cobertura que un enfoque provee para favorecer el análisis de los diferentes tipos de requisitos que existen en un problema. Generalmente, los analistas dedican la mayor parte de los esfuerzos a la definición de requisitos funcionales y de estructura, prestando poca atención a las restricciones de contexto y a la definición de los atributos de calidad esperados del producto.
- **Trazabilidad.** Se refiere al soporte que un enfoque provee para controlar la evolución de los requisitos desde fases tempranas de desarrollo hasta la implementación. En otras palabras, controlar la trazabilidad (hacia delante y hacia atrás) por medio de caminos que garanticen la evolución de los requisitos en elementos del modelo de otras fases del proceso. La trazabilidad se considera una característica básica de un modelo de requisitos, puesto que permite tener un control de los cambios y un análisis del impacto de dichos cambios.

⁴ Avansoft S. A. www.avansoft.com



- **Composición de requisitos.** Es la capacidad que un enfoque tiene para componer requisitos individuales en requisitos de mayor nivel de abstracción. Esta capacidad de composición es importante, porque permite describir las necesidades desde el enfoque del negocio y no desde una perspectiva de solución de ingeniería, lo cual facilita la comunicación entre el cliente y el desarrollador.
- **Manejo de conflictos.** Es la manera como el enfoque facilita el análisis y solución de conflictos entre los participantes del proyecto de desarrollo para dar apoyo a las decisiones y gestionar los riesgos. El manejo de conflictos es clave durante el levantamiento y análisis de requisitos, ya que se deben contemplar las consideraciones de los involucrados como clientes y usuarios finales, las condiciones de negocio y de tecnología y otros que pueden dar origen a conflictos que deben ser resueltos.
- **Soporte para mapeo.** Capacidad del enfoque para facilitar la transformación de un requisito en artefactos de una etapa siguiente. En la medida que el enfoque facilite el proceso de mapeo o transformación de los artefactos de una fase a otra, se estará logrando una mejor trazabilidad.
- **Validación-verificación.** Capacidad de revisar y evaluar los resultados intermedios durante el proceso, es decir, validar si se están haciendo las cosas como se debe y verificar que los productos generados corresponden a lo que se ha solicitado.
- **Escalabilidad.** Capacidad del enfoque para ajustarse a diferentes tipos de proyecto, tanto por su tamaño como por su misma naturaleza.

3.2 Evaluación de los criterios

Cada criterio se valoró según la siguiente escala: (A) cumple plenamente (valor 5 puntos); (B) cumple en alto grado (valor 4 puntos); (C) cumple aceptablemente (valor 3 puntos); (D) cumple deficientemente (valor 2 puntos); (E) no cumple (valor 1 puntos); (NA) no aplica (valor 0 puntos). La tabla 1 muestra la valoración cuantitativa realizada a cada enfoque después de la experiencia de aplicación. Los totales por criterio indican la fortaleza de las aplicaciones de la AORE y los totales por enfoque indican cuál enfoque podría ser el más propicio para orientar los desarrollos de software en empresas de desarrollo.

Tabla 1. Evaluación de los enfoques en estudio

Criterio	MDSOC	ARGM	Theme/ Doc	AOSD/UC	Total por criterio
Alcance	4	5	4	3	16
Trazabilidad	3	4	5	5	17
Composición de requisitos	5	4	4	4	17
Manejo de conflictos	4	5	0	0	9
Mapeo	2	2	5	4	13
Validación-verificación	3	4	3	0	10
Escalabilidad	1	2	2	3	8
Total por enfoque	22	26	23	19	

3.3 Análisis comparativo

- Alcance en el tratamiento de los requisitos.** Una de las principales fortalezas de los enfoques aspectuales en general es que rescatan la importancia de los atributos de calidad como elementos de primera clase desde la fase de requisitos. La propuesta mejor evaluada fue ARGM, puesto que en todo momento permite analizar la relación entre las características funcionales del sistema y su impacto en el cumplimiento de los atributos de calidad, los cuales se han analizado utilizando el marco NFR [14]. MDSOC y Theme/Doc parten de un espacio multidimensional de intereses, pero no presentan consideraciones especiales para los atributos de calidad. La propuesta que realiza AOSD/UC considera atributos de calidad como casos de uso de infraestructura que no logran cumplir de forma adecuada las exigencias de especificación de un atributo de calidad.
- Trazabilidad.** En MDSOC se define una trazabilidad directa entre intereses y requisitos. Algunas matrices de intereses contra intereses pueden usarse para el trazado. ARGM define una trazabilidad jerárquica entre objetivos y tareas para tomar decisiones de diseño (requieren mayor elaboración). Los enfoques mejor evaluados fueron Theme/Doc y AOSD/UC; en el primero se define una trazabilidad directa entre requisitos y acciones, y el segundo utiliza enlaces de trazado UML o estereotipo de relaciones de dependencia específicas entre los artefactos de requisitos y sus correspondientes artefactos en el diseño.
- Composición de los requisitos.** Todos los enfoques orientados por aspectos se caracterizan por sus aportes en la composición de artefactos. En realidad, ofrecen un proceso sistémico de composición; la diferencia radica en la forma o elementos que utilizan para realizarla. MDSOC provee un conjunto de reglas de composición que determinan cómo los intereses e intereses transversales pueden componerse. ARGM posee una visión descendente, partiendo de los objetivos de negocio que representan los requisitos de alto nivel que se van descomponiendo progresivamente; así, las operaciones o tareas representadas en los *softgoals* son compuestas en los objetivos (*goals*) funcionales del sistema. Theme/Doc parte de listas detalladas de requisitos para descubrir acciones básicas que luego, en un proceso ascendente, van a formar las acciones o temas principales. AOSD provee un artefacto de composición o colaboración llamado *<use case slice>* en el cual se representan cada caso de uso en diferentes fases del ciclo de vida y los artefactos que colaboran para su función.
- Manejo de conflictos.** MDSOC provee mecanismos para la identificación de conflictos y la asignación de pesos de acuerdo con las tablas de contribución entre intereses. ARGM provee un proceso de resolución de conflictos que se realiza de manera incremental mediante la regla de remoción de enlaces de contribución negativa, lo cual facilita su aplicación. Theme/Doc y AOSD/UC no especifican ningún mecanismo para identificación y solución de conflictos.
- Soporte para mapeo.** MDSOC propone una matriz que sugiere de qué modo los elementos identificados como decisiones, funciones y aspectos pueden correlacionarse en fases siguientes del ciclo de vida. En ARGM las descomposiciones de los *goals* y *softgoals* proponen una primera aproximación para correlacionar algunas tareas en aspectos. Theme/Doc establece reglas de mapeo entre los elementos de Theme/Doc y su evolución en elementos de Theme/UML. AOSD/UC hace un mapeo directo de elementos UML, por ejemplo, un caso de uso del análisis mapea directamente a una realización de caso de uso en el diseño. En general, se observa una gran deficiencia en reglas de correlación de los intereses transversales desde los requisitos hacia la arquitectura y la implementación.
- Validación-verificación.** Tanto en MDSOC como en Theme/Doc se sugiere la verificación



“walking through” de las diferentes actividades para validar los resultados. Multidimensional provee las matrices de contribución para validar la composición. ARGM define heurísticas que facilitan la verificación de inconsistencias, ya que cada vez que se establece una relación se verifica el impacto que tuvo sobre las relaciones asociadas. Theme/Doc provee el análisis gramatical sobre una lista de requisitos proporcionando consistencia de los requisitos. Ninguna de las propuestas proveen listas de chequeo que faciliten la validación de los requisitos de acuerdo con los artefactos de representación creados.

- **Escalabilidad.** La escalabilidad es una característica asociada directamente con el soporte que provean las herramientas de apoyo al enfoque. En general, se observa que las herramientas de apoyo aún no se encuentran preparadas para el manejo de intereses transversales o aspectos en el nivel del diseño desde fases tempranas del desarrollo. Por su cercanía con los enfoques tradicionales, AOSD/UC resulta ser una de las que pueden ser soportadas más fácilmente por las herramientas CASE actuales.

4. TRABAJOS RELACIONADOS

Chitchyan *et al.* han realizado una recopilación detallada de los principales enfoques aspectuales y no aspectuales. Para el análisis, sus enfoques se agrupan en tres grandes categorías: requisitos, arquitectura, y análisis y diseño; realizan la comparación con base en los criterios de trazabilidad, composición, evolución y escalabilidad [10]. Ellos retoman gran parte de ese trabajo restringiendo el estudio a sólo algunas aproximaciones orientadas a requisitos y enfatizan el tratamiento de asuntos transversales funcionales y no funcionales [4]. Bakker *et al.* definen criterios generales que, desde una perspectiva más práctica, permiten reconocer algunas de las aproximaciones de requisitos [5].

El presente trabajo retoma algunas de las características definidas en los trabajos anteriores, pero se evaluaron desde las necesidades y prácticas de la industria del software. Así, este artículo hace parte de un conjunto de productos de investigación para usar y aplicar los enfoques aspectuales en el proceso de desarrollo de la industria del software y, en especial, en las etapas tempranas del desarrollo. Actualmente, se hace la validación del enfoque denominado AORE++ [15] en proyectos reales de la industria de software; este enfoque es una extensión al enfoque multidimensional propuesto por Moreira *et al.* [6]. Además, se está trabajando en el desarrollo de dos proyectos importantes: 1) definición de un marco de trabajo que integre los enfoques aspectuales como soporte al proceso de desarrollo de aplicaciones de software industriales con diferentes consideraciones del contexto del negocio [16]; 2) adaptación del enfoque AORE++ al desarrollo de software dirigido por modelos para crear el ambiente propicio para la transformación y la trazabilidad de intereses en diferentes niveles de abstracción bajo la arquitectura MDA [17, 24].

5. CONCLUSIONES

En este artículo se realizó un análisis comparativo de algunos de los enfoques de requisitos orientados por aspectos desde el enfoque de las prácticas de la industria del software local de la ciudad de Medellín, Colombia. Al estudiar y aplicar las aproximaciones en casos reales, se pudo concluir que los enfoques de aspectos pueden contribuir a la solución de algunos de los problemas que se presentan con mayor frecuencia en el proceso de ingeniería de los requisitos dentro de una empresa de software. También fue posible identificar algunas desventajas que aún tienen los enfoques con respecto a las prácticas actuales.

En cuanto al tratamiento uniforme de los requisitos, en las empresas de desarrollo los analistas usan diferentes técnicas para identificar, separar y

clasificar los requisitos. Los criterios usados para la aplicación de las técnicas son variados y dependen en gran medida de la experiencia del analista. Muchas veces, los requisitos no funcionales o los atributos de calidad adquieren importancia sólo en el momento de modelar la arquitectura, limitando el tratamiento uniforme de los requisitos y de las contribuciones entre intereses. Los enfoques aspectuales introducen un nuevo concepto de tratar los requisitos desde el concepto de interés. La identificación y separación de intereses e intereses transversales permiten un tratamiento uniforme de los requisitos; son reconocidos como elemento de primera clase desde el espacio del problema hasta la implementación. Además, las relaciones entre los intereses proveen insumos importantes para las decisiones de arquitectura y diseño.

En cuanto al análisis de requisitos desde la perspectiva del cliente y la perspectiva del uso de la técnica, las empresas de desarrollo usan técnicas de levantamiento de requisitos donde el análisis de los requisitos se hace desde la perspectiva del cliente. Esto está orientado a lograr buenos niveles de especificación y validación de los requisitos, lograr modelos de análisis muy cercanos al producto requerido e identificar los riesgos del desarrollo en etapas posteriores. Los enfoques aspectuales soportan esta labor desde el uso de las técnicas, para así involucrar las diferentes visiones de los clientes, proporcionando estándares de separación y clasificación de los intereses según las necesidades de todos los participantes del proyecto. Acercar estos enfoques al análisis del dominio del problema es importante para lograr una mejor acogida por parte de los clientes y usuarios finales, quienes son protagonistas en el proceso de requisitos.

Para una empresa de software es muy importante contar con prácticas de aplicación flexible de acuerdo con las condiciones de un proyecto o aplicación específico. En el estudio se detectó que, dada la diversidad de escenarios para originar los requisitos, no todas las aproximaciones son adecuadas para todos los escenarios.

Para las empresas de software es importante tener una forma flexible de integrar nuevas técnicas y tecnologías en sus marcos metodológicos. El aprendizaje y adopción de nuevos paradigmas se puede convertir en un riesgo, aunque con la madurez de uso de los nuevos enfoques los analistas pueden lograr disminución de costos y riesgos en etapas avanzadas del desarrollo. Para realizar la ingeniería de los requisitos, las empresas de desarrollo siguen modelos tradicionales, pero en realidad la experiencia de los analistas en el uso de las técnicas se convierte en el factor más importante de éxito. Así, más que trabajar en un enfoque aspectual específico, es importante contar con listas de chequeo, instructivos o módulos complementarios a las herramientas CASE que ayuden a seleccionar y utilizar el enfoque aspectual más apropiado para todo el proceso de desarrollo o en alguna fase específica.

Para que estas nuevas prácticas en la ingeniería de requisitos puedan ser adoptadas con éxito por las organizaciones de software, es necesario considerar aspectos que van más allá de los elementos técnicos. Kauppinen *et al.* dicen que las consideraciones de tipo humano (motivación, participación activa tanto de personal técnico como de clientes, disposición al cambio, etc.) y las consideraciones de infraestructura (entrenamiento “*just on time*”, uso adecuado de herramientas, divulgación de lecciones aprendidas, etc.) son factores centrales que deben considerarse, si se desea que las organizaciones adopten estos nuevos enfoques [25].

Uno de los grandes problemas que ahora enfrentan las empresas de desarrollo es el soporte a la trazabilidad y el mapeo de los requisitos durante el proceso de desarrollo. Las herramientas CASE proveen elementos para el trazado de los elementos de modelo, pero aun así la trazabilidad es una práctica que se desvirtúa con facilidad durante el proceso. Además, las herramientas soportan superficialmente la transformación del diseño a la implementación. Algunas herramientas o métodos conceptuales se han propuesto para mejorar el uso de esta práctica en la industria de software [18, 19]. Los enfoques



aspectuales proveen mecanismos de separación y composición de intereses y requisitos para facilitar la evolución y el cambio durante el proceso. Los enfoques como MDA (Model-Driven Architecture) soportan la separación arquitectónica y transformación de modelos en diferentes niveles de abstracción [21-23].

En la actualidad, la integración de los enfoques de aspectos en los marcos metodológicos de la industria del software se hace al nivel de *middlewares* y *frameworks* para la programación (p. ej., JBoss AOP, Spring, etc.). En la adopción de los enfoques de la AORE, no se han registrado aún trabajos en el campo industrial, solo en aplicaciones académicas (casos de estudio). Generalmente, esto ocurre debido al escepticismo de las empresas de desarrollo en la adopción de nuevas técnicas para mejorar la ingeniería de requisitos y a la poca disponibilidad de personal para su investigación y aplicación.

RECONOCIMIENTOS

Este trabajo hace parte del proyecto MMEDUSA, realizado en convenio por la Universidad EAFIT, y Avansoft S. A., con el apoyo de Colciencias, cuyo propósito es definir un marco metodológico para desarrollo de aplicaciones utilizando la aproximación de aspectos.

REFERENCIAS

1. Anaya R., Londoño L. y Hurtado J. Una estrategia para elevar la competitividad de las industrias de software PYMES. Memorias del 9° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software. IDEAS, La Plata, Argentina, 2006.
2. Filman R., Elrad T., Clarke S. and Akşit M. Aspect-oriented software development. Addison Wesley, 2005.
3. Araujo J., Baniassad E., Clements P., Moreira A. and Tekinerdogan B. Early aspects: the current landscape, Technical Notes, Carnegie Mellon University CMU/SEI-2005-TN-xxx. 2005.
4. Chitchyan R., Rashid A. and Sawyer P. Comparing requirement engineering approaches for handling crosscutting concerns. Anais do WER05 Workshop em Engenharia de Requisitos, Porto, Portugal, Junho 13-14, 2005, pp. 1-12.
5. Bakker J., Tekinerdoğan B. and Akşit M. Characterization of early aspects approaches. Proceedings of the Early Aspects Workshop at AOSD, 2005.
6. Moreira A., Rashid A. and Araujo J. Multi-dimensional separation of concerns in requirements engineering. International Conference on Requirements Engineering, 2005. IEEE Computer Society.
7. Baniassad E. and Clarke S. Finding aspects in requirements with Theme/Doc, presented at Workshop on Early Aspects (held with AOSD 2004), Lancaster, UK, 2004.
8. Baniassad, E. and Clarke, S. Aspect-oriented analysis and design: The Theme Approach. Addison-Wesley, 2005.
9. Jacobson I. and Ng P. W. Aspect-oriented software development with use cases. Addison Wesley Professional, 2005.
10. Chitchyan R., Rashid A., Sawyer P., Bakker J., Pinto M., Garcia A., Tekinerdogan B., Clarke S. and Jackson A. Survey of aspect-oriented analysis and design approaches, AOSD-Europe-ULANC-9, AOSD-EUROPE network of excellence. May 2005
11. Tabares M. S., Anaya R. y Arango F. La ingeniería de requisitos orientada a aspectos: una experiencia de aplicación en un sistema de ayuda en línea. Revista DYNA No. 153, noviembre 2007. ISSN 0012-7353.
12. Yu Y., Sampaio do Prado Leite J. C. and Mylopoulos J. From goals to aspects: discovering aspects from requirements goal models, presented at International Conference on Requirements Engineering, Kyoto, Japan, 2004.
13. Yu E. Modelling your system goals: the i* approach. London, UK: British Computer Society, Requirements Engineering Special Interest Group, 2005.
14. Chung L., Nixon B.A., Yu E., and Mylopoulos J. Non-functional requirements in software engineering, Kluwer Academic Publishers, 2000.
15. Ospina C. A., Parra C. A., Londoño L. F. y Anaya R. Una extensión del modelo AORE multidimensional. Memorias del X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software. IDEAS 2007, Isla Margarita, Venezuela.
16. Quintero J. B., Hernández D. y Anaya R. Experiencia práctica de la aplicación de aproximaciones

- orientadas por aspectos en el desarrollo de un portal temático. *Revista Avances en Sistemas e Informática*. Volumen 4, No. 1, Junio 2007, pp. 109-116.
17. Tabares M. S., Moreira A., Anaya R., Arango F. and Araújo J. A. Traceability method for crosscutting concerns with transformation rules. *Proceedings IEEE Early Aspects at ICSE: Workshop in Aspect-Oriented Requirements Engineering and Architecture Design (EARLYASPECTS'07) in 29th ICSE, US. 2007*. Digital Object Identifier 10.1109/EARLYASPECTS.2007.2.
 18. Tabares M. S., Barrera A., Arroyave J. D y Pineda J. D. Un método para la trazabilidad de requisitos en el proceso de desarrollo unificado. *Revista EIA, ISSN 1794-1237 Número 8, diciembre 2007*, pp. 68-82.
 19. Asuncion H. U., François F. and Taylor R. N. An end-to-end industrial software traceability tool. *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. 2007*.
 20. Sommerville I. and Sawyer P. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, vol. 3, pp. 101-130, 1997.
 21. MDA-Guide, OMG Document v1.0.1, omg/03-06-01.
 22. Mellor S. J., Clark A. N. and Futagami T. Guest editors' introduction: Model-Driven Development. *IEEE Software* 20(5): 14-18 (2003).
 23. Quintero J. B. y Anaya R. MDA y el papel de los modelos en el proceso de desarrollo de software. *Revista EIA, ISSN 1794-1237, número 8, p. 131-146. Diciembre 2007*.
 24. Tabares M. S., Anaya R. y Arango F. Un esquema de modelado para soportar la separación y transformación de intereses durante la ingeniería de requisitos orientada por aspectos. Presentado en el Tercer Congreso Colombiano de Computación, pendiente de publicación en la *Revista Avances en Sistemas e Informática*. Vol 5, No. 1. Año 2008.
 25. Kauppinen M., Vartiainen M., Kontio J., Kujala S. and Sulonen R. Implementing requirements engineering processes throughout organizations: success factors and challenges. *Information and Software Technology* 46 (2004) 937-953.