

Orientación a Aspectos y Orientación a Objetivos: Una propuesta para su integración*

Elena Navarro¹, Patricio Letelier², Isidro Ramos² y Bárbara Álvarez³

¹Departamento de Informática, UCLM, Albacete
enavarro@info-ab.uclm.es

²Departamento de Sistemas de Información y Computación, UPV, Valencia
{letelier, iramos}@dsic.upv.es

³Departamento Tecnologías de la Información y Comunicaciones, UPCT, Cartagena
Barbara.Alvarez@upct.es

1. Introducción

Existen diferentes paradigmas para tratar la separación de *concerns* (propiedades o áreas de interés) de un sistema con el propósito de facilitar evolución, adaptabilidad, comprensibilidad, etc. Estos *concerns* pueden ser características funcionales o no funcionales, como portabilidad o tolerancia a fallos. El Desarrollo de Software Orientado a Aspectos (AOSD) [1] es una de estos paradigmas. AOSD ofrece un conjunto de mecanismos que permite gestionar aquellos intereses que aparecen dispersos en el sistema, cruzando o cortando diferentes elementos del mismo (fenómeno referido como *crosscutting*). AOSD identifica estos *concerns* expresándolos como *aspectos* [5] y tratándolos de forma explícita. Recientemente diversas propuestas han surgido, tales como [13] que proponen una identificación de *aspectos candidatos*, en la etapa de captura y especificación requisitos (AORE).

Por otra parte, un enfoque que nace en la comunidad de Ingeniería de Requisitos es la Orientación a Objetivos [9] (OOR). Ésta permite, mediante un proceso de refinamiento y descomposición, pasar desde intereses u objetivos del sistema a requisitos asignables a agentes del mismo, es decir, a cualquier componente activo tales como humanos, dispositivos cualquier componente software del sistema.

El objetivo de este trabajo es presentar nuestra propuesta para la captura, especificación y organización de requisitos, integrando dos aproximaciones: AORE y OOR. Asimismo, se utiliza el estándar ISO/IEC 9126 [7] como punto de partida en el proceso de definición de objetivos. El enfoque aquí presentado es una de las actividades de ATRIUM [11], una metodología para la definición de arquitecturas software (AS) a partir de requisitos. PRISMA [12] es el modelo elegido para la especificación de AS debido a su habilidad para la generación automática de código y por sus capacidades reflexivas en tiempo de ejecución.

* Este trabajo ha sido subvencionado por el proyecto CICYT DYNAMICA (TIC2003-07804-C05-01 y TIC2003-07804-C05-02)

2. Modelo de objetivos: *concerns* del sistema

El Modelo de Objetivos (MO) de ATRIUM, previamente descrito en [10], está inspirado en KAOS [2] y NFR Framework.[3]. Desde el punto de vista exclusivo de orientación a objetivos nuestro trabajo se distingue por incluir el tratamiento tanto de requisitos funcionales como no funcionales en el MO. Así, no sólo se establece la funcionalidad del sistema sino también aquellas restricciones, calidad de servicio u objetivos de desarrollo, igualmente importantes para la definición de AS. Otra diferencia de nuestra propuesta es la incorporación de técnicas de Orientación a Aspectos para gestionar el *crosscutting* que suele aparecer en una Especificación de Requisitos del Software (SRS), especialmente cuando el sistema es complejo y/o de gran envergadura.

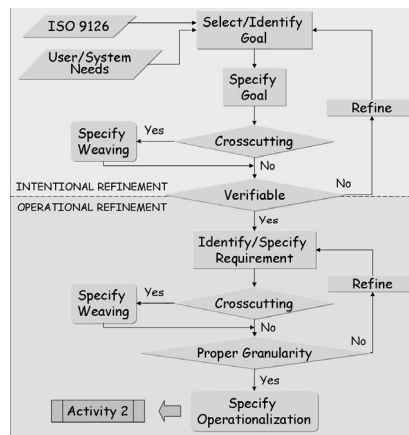


Fig. 1. Flujo de trabajo para la definición de objetivos/requisitos

La **Fig. 1** establece el flujo de trabajo para la elaboración del MO, junto con los artefactos de entrada utilizados en cada tarea. Aunque en la **Fig. 1** sólo se muestra un flujo secuencial de aplicación de las tareas, en la práctica el proceso es iterativo. Ello es debido a que las distintas tareas que lo integran, se ejecutan con cada nuevo objetivo que se identifica o selecciona.

La primera de estas tareas aborda la *Identificación/Selección de los Objetivos*. Una entrada de esta tarea es el modelo de calidad ISO/IEC 9126, el cual es utilizado como marco de trabajo de partida, ofreciendo al analista una clasificación inicial de alto nivel para los posibles *concerns* del sistema. En la medida que se realiza la captura de requisitos puede ser conveniente incluir nuevos *concerns* para incluir adecuadamente los objetivos/requisitos adicionales, o finalmente descartar aquellos no usados.

Las *Necesidades de Usuario/Sistema* constituyen otra de las entradas de esta tarea. Éstas nos ofrecen información para identificar nuevos objetivos/requisitos. Cada uno de los objetivos establecidos mediante este proceso quedará enmarcado dentro de la especificación de *concerns* establecida por el modelo ISO/IEC 9126, estableciendo una estructura de grafo. Esto nos permitirá conseguir dos mejoras, por un lado

facilitar la legibilidad de la especificación, y por otro, conducir el proceso de captura y análisis de requisitos.

En la tarea de *Especificación del Objetivo* se establecen no sólo los atributos que lo conforman, como son su nombre, prioridad, etc., sino también cuáles son las relaciones de composición (AND/OR/XOR) con sus objetivos padres.

Para cada objetivo, no sólo se especifican las relaciones de refinamiento, sino también aquellas relaciones de composición^(*) necesarias para gestionar el *crosscutting* que aparece en la especificación. Este proceso de refinamiento de objetivos, continúa hasta que el objetivo es asignable a un agente del sistema. En este momento, pasamos de un refinamiento intencional a un refinamiento operacional y con ello a la especificación de un requisito. Una diferencia importante entre un requisito y un objetivo estriba en que para los requisitos se exige que puedan ser verificables. Así, para la *Identificación/Especificación de requisitos*, se sigue un proceso similar al de los objetivos, mediante la definición de sus atributos y relaciones de composición. También, se establecen aquellas relaciones de composición que sean aplicables.

Tras la especificación de requisitos la siguiente tarea es la *Especificación de la operacionalización*, es decir, la definición del agente o conjunto de agentes que colaboran en su realización. En la definición del MO la operacionalización es únicamente una descripción de la solución propuesta para la realización, actuando ésta como entrada de la siguiente actividad de la metodología, la actividad *Definición de Escenarios*. Esta última es la encargada, mediante la descripción del escenario correspondiente, de elaborar una definición completa de la solución. La introducción de esa descripción en el MO nos permite incorporar las relaciones entre esa solución y los requisitos ya definidos en el modelo. Así, podemos reflejar cómo una solución puede colaborar positivamente en la realización de un requisito y negativamente en la realización de otros, consiguiendo con ello un análisis más exhaustivo del conjunto de soluciones.

3. Caso de Estudio

Esta sección ilustra cómo hemos aplicado nuestra propuesta en el contexto de un sistema real. Esta experiencia fue desarrollada gracias a la colaboración en el proyecto europeo *Environmental Friendly and cost-effective Technology for Coating Removal (EFTCoR)* [4]. El objetivo fundamental de este proyecto está relacionado con las operaciones de mantenimiento de cascos de buques. Principalmente, trata las operaciones de eliminación de revestimiento, limpieza y re-pintura del casco del buque por una familia de robots que realizan diferentes operaciones o la misma pero de distinta manera. La plataforma de teleoperación identificada está integrada por los subsistemas descritos en [6]. Nuestro caso de estudio se centra en la especificación de requisitos de la *Unidad de Control de los Dispositivos Robóticos (RDCU)*. Su

* En la presentación de nuestro trabajo hemos preferido utilizar el término “composición” en lugar de *weaving*, usado este último en el paradigma orientado a aspectos. En ingeniería de requisitos la semántica se acerca más a una composición de elementos que a un *weaving* determinando saltos en la ejecución.

definición arquitectónica es especialmente compleja debido a las restricciones que han de satisfacerse para permitir un comportamiento dinámico del sistema final. En tiempo de ejecución, cada herramienta de limpieza o dispositivo de posicionamiento podría ser reemplazado. Además, tanto el cambio de herramientas como la operación tienen severos requisitos de seguridad, proporcionando mecanismos para la parada si se prevé algún daño al operador o al equipamiento. Asimismo, cada operación de mantenimiento ha de ser programada para su realización dentro de unos límites temporales concretos. Algunos fragmentos de las necesidades establecidas para la RDCU descritos en [4]:

<i>Positioning systems and tool can work simultaneously. The RDCU is responsible for co-ordinating their actions according to their operational states, the mission parameters, and the current state of the environment.</i>
<i>(1) The operational commands for positioning systems and activating the tool can be easily and efficiently issued to the system.</i>
<i>(2) The system should react to such commands efficiently. In some cases, the execution time of the commands should be smaller than a given deadline</i>
<i>(1) The possibility of using different coating removal technologies (blasting, water pressure). Though the chosen technology for the EFTCoR project is blasting, the RDCU should be open to incorporate cleaning tools based on other technologies.</i>
<i>(2) The possibility of using the system for different maintenance tasks, including at least the fresh water washing before blasting and the painting after blasting.</i>
<i>(3) The possibility of using different positioning systems and different combinations of primary and secondary positioning systems.</i>
<i>(4) The possibility of using different tools for the same or different processes (already considered in the first point)</i>

A partir de las sentencias anteriores podemos apreciar la existencia del *crosscutting* en la especificación, como es el caso de los objetivos relacionados con la *Efficiency* y *Adaptability*. Ambos se aplican sobre otros objetivos como son el *ControlPositioning* o *ControlTools*. La aplicación de la tarea de *Identificación/Selección de Objetivos* usando como entrada los fragmentos anteriores, generó como resultado la lista de objetivos que aparecen en la Tabla 1 (en [10] se describen otros atributos aplicables).

El grafo de la **Fig. 2** muestra (parte de) el MO incluyendo los objetivos de la Tabla 1. Así, se observa como *Portability*, *Functionality*, y *Efficiency* son algunas de las características seleccionadas como *concerns* para el sistema EFTCoR. También dicha la figura nos muestra algunas de las relaciones de refinamiento que establecieron. Por ejemplo, la relación AND entre el objetivo *AdaptabilityWorkingEnvironment* y *AdaptabilityHullMaintenance-Operation* introducida para conseguir la satisfacción de *Adaptability*. Por otra parte, para el objetivo *AdaptabilityWorkingEnvironment* se ha establecido una relación de composición con el objetivo *ControlPositioning*.

Tabla 1. Descripción (parcial) de Objetivos para EFTCoR

GOAL	DESCRIPTION
Functionality	The system has to provide functions which meet stated and implied needs
Suitability	The system has to provide an appropriate set of functions for specified tasks and user objectives
CoordinateDevices	The RDCU has to coordinate robotic devices according to the current mission procedure and global system state
ControlPositioning	The RDCU has to control the positioning devices
ControlTools	The RDCU has to control the tools attached to positioning devices (blasting head, painting tool, etc)
Efficiency	The system should use efficiently their resources
TimeBehaviour	The system should respond with appropriate speed
PerformanceOperation	The system should respond with appropriate speed to the operation commands
Portability	The system should be able to be transferred from one environment to another
Replaceability	The system should be able to use different software product for the same purpose
Adaptability	The systems has to be adaptable for different specified environments
AdaptabilityWorkingEnvironments	The system operation has to be adaptable to different working environments
AdaptabilityHullMaintenanceOperation	The system operation has to be adaptable to different hull maintenance operations

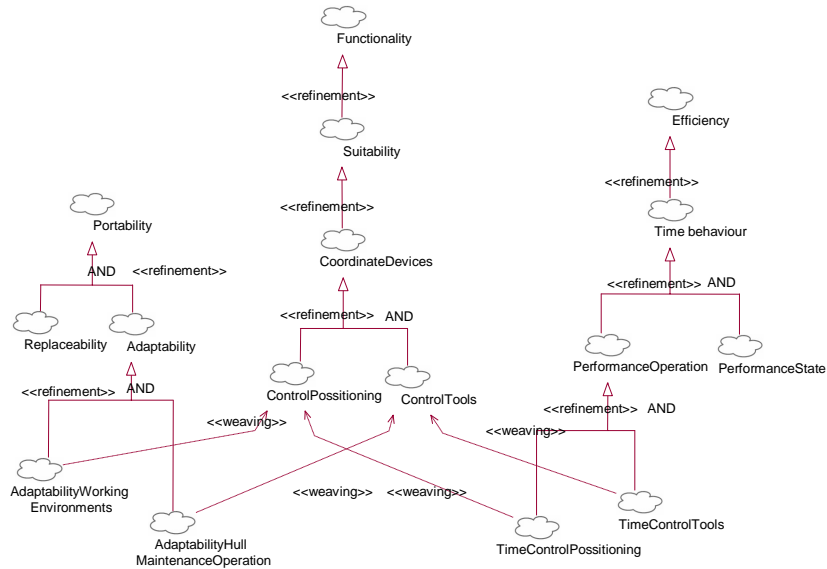


Fig. 2. (Vista parcial del) Modelo de Objetivos para EFTCoR.

4. Conclusiones y trabajos futuros

En este trabajo hemos presentado resumidamente un enfoque para la captura y análisis de requisitos en la que se integran orientación a objetivos y orientación a aspectos. Gracias a la OOR, se mantiene una trazabilidad explícita desde los objetivos hacia los requisitos, y desde estos últimos hacia las operacionalizaciones que se materializan en los componentes software. Por otro lado, el enfoque orientado a aspectos permite gestionar de manera elegante y efectiva el *crosscutting* que puede aparecer en la SRS cuando se entremezclan objetivos/requisitos. Definiendo relaciones de composición entre los elementos de un objetivo/requisito se consigue una representación óptima, desde la cual puedan incluso presentarse los objetivos/requisitos en su forma original si fuese necesario. Se ha definido un flujo de trabajo en el cuál se integran ambas aproximaciones, detallando una serie de tareas que ofrecen guías para la elaboración y organización de la especificación de requisitos. Otra de las ventajas que ofrece nuestra propuesta se refiere a la utilización del ISO/IEC 9126 como punto de partida en cuanto al establecimiento de posibles *concerns* (intereses del sistema). Con nuestro enfoque es posible ajustarse a las recomendaciones del IEEE 830-1998 [8] en cuanto a contenido de la SRS pero contando con significativas ventajas en cuanto a flexibilidad y versatilidad en la organización de los requisitos.

Es importante destacar que el concepto de aspecto no aparece explícitamente en el MO, es decir, no existe ningún constructor manifiesto para él, tal y como ocurre el paradigma Orientado a Aspectos, al menos en un nivel de diseño o implementación.

En el MO un *aspecto candidato* es aquel objetivo/requisito que presenta relaciones de composición con varios otros objetivos o requisitos (más de uno al menos).

Diversos retos quedan todavía por resolver. Uno de ellos está relacionado con las relaciones de composición entre objetivos/requisitos. Concretamente, con la idoneidad del conjunto de relaciones ofrecido y su semántica. Para ello, la problemática ofrecida por el caso de estudio facilitará la verificación y validación del mismo. Nuestro interés es que para las relaciones de composición y los objetivos/requisitos identificados se mantengan relaciones de trazabilidad hacia artefactos arquitectónicos, según el proceso planteado por ATRIUM.

Referencias

- [1] Aspect-Oriented Software Development, <http://www.aosd.net>
- [2] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed Requirements Acquisition. In *Science of Computer Programming*, 20, (1993), 3-50.
- [3] L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishing, 2000
- [4] EFTCOR: Environmental Friendly and cost-effective Technology for Coating Removal. European Project within the 5th Marco de trabajo Program (GROWTH G3RD-CT-00794), 2003.
- [5] T. Elrad, R. E. Filman and A. Bader, "Aspect-oriented programming: Introduction", *Communications of the ACM*, Vol. 44 , Issue 10, pp. 29 – 32, October, 2001.
- [6] C. Fernández, J.A. Pastor, P. Sánchez, B. Álvarez, A. Iborra: "Co-operative Robots for Hull Blasting in European Shiprepair Industry. Robotics and Automation Magazine (RAM)", special issue on Industrial Robotics Applications & Industry-Academia Cooperation in Europe. New Trends and Perspectives. (To appear in September 2004)
- [7] ISO/IEC Standard 9126-1 Software Engineering- Product Quality-Part1: Quality Model, ISO Copyright Office, Geneva, June 2001
- [8] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications, en Volumen 4: Resource and Technique Standards, The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection.
- [9] A. van Lamsweerde "Goal-Oriented Requirements Engineering: A Guided Tour", Invited Paper for 5th IEEE International Symposium on Requirements Engineering, Toronto, August, 2001, pp. 249-263.
- [10] E. Navarro, I. Ramos and J. Pérez: "Goals Model-Driving Software Architecture", 2nd International Conference on Software Engineering Research, Management and Applications, May 5-8, 2004, Los Angeles, CA, USA.
- [11] E. Navarro, I. Ramos and J. Pérez: "Software Requirements for Architected Systems". Proc. 11th IEEE International Conference on Requirements Engineering, pp. 365-366, September 8-12, 2003, Monterey, CA, USA.
- [12] J. Pérez, I. Ramos, J. Jaén, P. Letelier and E. Navarro, "PRISMA: Towards Quality, Aspect Oriented and Dynamic Software Architectures", Proc. 3rd IEEE International Conference on Quality Software, Dallas, Texas, USA, pp. 59-66, November 6 - 7, 2003.
- [13] A. Rashid, P. Sawyer, A. Moreira and J. Araújo, "Early Aspects: a Model for Aspect-Oriented Requirements Engineering", Proc. International Conference on Requirements Engineering, September 9-13, 2002, Essen, Germany.