

Derivando el diseño a partir de especificaciones de requisitos basadas en Casos de Uso

Luis Roqué Fourcade, Liliana Arakaki
 Departamento de Informática – Facultad de Ciencias Físico Matemáticas y Naturales
 Universidad Nacional de San Luis
 Ejercito de Los Andes 950, CP D5700HHW, San Luis, Argentina, +54 (0266) 4424027
 email: araroq@unsl.edu.ar, liliana.arakaki@yahoo.com

Resumen

A pesar de que los enfoques basados en Casos de Uso han demostrado ser muy efectivos en la actividad de captura de requisitos, su falta de formalidad los deja muy alejados de las expresiones de diseño de solución para los casos modelados.

Esa falta de formalidad en la expresión de los Casos de Uso, convierte a la tarea de derivación de un diseño de solución en una tarea ambigua y librada únicamente al buen criterio del analista/diseñador.

El presente trabajo, describe líneas de investigación y desarrollo que intentan dar soporte al proceso de derivación de diseños de solución para especificaciones de requisitos basadas en Casos de Uso, sin resignar las ventajas que se obtienen de la expresión informal y cercana al negocio que distinguen a los enfoques basados en Casos de Uso.

Palabras clave: Caso de Uso - Diseño - Derivación de Diseño - OOD - UML - Modelado - MDA - QVT - XSLT

Contexto

Las líneas de investigación presentadas en este Trabajo, se desarrollan en el marco del proyecto de investigación Ingeniería de Software: Conceptos, Métodos y Herramientas en un Contexto de “Ingeniería de Software en Evolución”, de la Facultad de Ciencias Físico-Matemáticas y Naturales de la Universidad Nacional de San Luis, bajo el número de proyecto 22F822.

Este proyecto, reconocido por el programa de Incentivos, es continuación de otros proyectos de investigación que han permitido establecer importantes vínculos con otras universidades a nivel nacional e internacional.

Introducción

La falla para alcanzar un grado de satisfacción aceptable en los destinatarios de un Producto de Software constituye el eje de una polémica que ha existido siempre y que perdura aún hoy en ámbitos de la Ingeniería de Software.

Entre otras cosas, la Ingeniería de Software se ha ocupado de producir y evolucionar métodos y técnicas destinadas a descubrir y especificar de manera ordenada y sistemática un conjunto de requisitos, especificar el diseño de un Producto de Software que los satisfaga y, finalmente, construir el producto y administrar su ciclo de vida.

Sin embargo, a pesar de que “especificar el diseño de un Producto de Software que satisfaga dichos requisitos”, no se refiere en absoluto a una tarea trivial, ninguno de los métodos actuales se ocupa de proveer un soporte que asista de manera efectiva a los desarrolladores en la mencionada tarea.

En particular, los enfoques basados en Casos de Uso han demostrado ser muy efectivos en la actividad de captura de requisitos, sin embargo, su falta de formalidad los deja muy alejados de las expresiones de diseño de solución para los casos modelados y convierte a la tarea de derivación en una de naturaleza ambigua y librada únicamente al buen criterio del analista/diseñador.

En el presente trabajo, introducimos líneas de investigación y desarrollo que intentan dar soporte al proceso de derivación de diseños de solución para requisitos basados en Casos de Uso.

Para este objetivo, estas líneas de investigación actualmente abarcan las siguientes áreas:

- ✓ Especificaciones de Requisitos Basadas en Casos de Uso
- ✓ Modelo de Procesamiento de Requerimientos
- ✓ Patrones Arquitecturales y de Diseño
- ✓ Clasificación de Conocimiento de Negocio

Especificaciones de Requisitos Basadas en Casos de Uso

Un sistema no existe de manera aislada, sino que interactúa con elementos externos, humanos o mecánicos, que lo utilizan para lograr algún objetivo. Un Caso de Uso especifica el comportamiento de un sistema o parte del mismo mediante la descripción escrita de las interacciones que se llevan a cabo con los elementos externos, o

actores, en respuesta a un evento iniciado por uno de ellos, el actor principal. El conjunto de interacciones es descrito de manera textual y en lenguaje natural, como una secuencia de acciones ejecutadas por el sistema, destacando una secuencia o flujo normal y luego todas las secuencias alternativas posibles.

La técnica de Casos de Uso, por su capacidad expresiva y facilidad de uso, es muy valiosa en la captura de requisitos; cada Caso de Uso describe en términos de los usuarios un requisito funcional del sistema sin comprometerlo con criterios formales de solución o aspectos tecnológicos. Pero tal flexibilidad descriptiva resulta en especificaciones informales que dificultan el proceso de derivación de la solución y de validación de los requisitos.

Sin embargo, el espacio entre la expresión informal de una especificación de requisitos basada en Casos de Uso y la correspondiente expresión formal de un modelo de diseño ha sido base de numerosas propuestas de trabajo. El objetivo de estas propuestas es expresar el conocimiento con un nivel suficiente de formalidad que sirva de base a técnicas de derivación de solución sin resignar la riqueza del vocabulario del dominio ni la facilidad de uso, propios de la técnica de Casos de Uso.

Un ejemplo de tales propuestas sugiere el uso de Plantillas Genéricas para la descripción de Casos de Uso [1], organizando el conocimiento de manera sistemática y categorizada. Esta propuesta permite "plantear soluciones genéricas, estandarizadas y validadas a problemas similares, en las etapas de captura de requisitos, análisis y diseño"[1].

Otra propuesta interesante sugiere el uso de Esquemas de Operación como suplemento a los Casos de Uso [2].

Un Esquema de Operación describe declarativamente los efectos de una Operación de Sistema por medio de una representación abstracta del estado del sistema antes y después de ejecutar la operación y de todos los eventos enviados al exterior.

Los estados inicial y final son especificados mediante pre y pos-condiciones expresadas en OCL (Object Constraint Language) [4], que especifican los estados en términos de objetos, atributos y asociaciones del Modelo de Clases del Dominio.

Una Operación de Sistema es una acción particular del sistema que se ejecuta atómicamente y que conjuntamente con otras acciones integran una secuencia de un Caso de Uso [2][3].

El formalismo de OCL permite obtener una base más rigurosa sobre la que se pueden razonar las propiedades del sistema, verificar y validar soluciones en base a la especificación y encontrar el

nivel de granularidad correcto que nos permita una derivación segura de un modelo de diseño.

Claramente es posible, a partir de un Esquema de Operación derivar, una especificación de diseño expresada como un modelo de colaboración.

Así, la posibilidad de identificar Operaciones de Sistema, especificarlas mediante Esquemas de Operación y derivar el correspondiente modelo de colaboración, las convierte en un enfoque valioso para el objetivo perseguido por las líneas de investigación que proponemos en este trabajo.

Modelo de Procesamiento de Requerimientos¹

A partir de la definición de Caso de Uso proporcionada por OMG [5], podemos tratar un Caso de Uso como una especificación de un tipo de requerimiento que un Actor hace a un Sistema.

De este modo, las especificaciones de posibles secuencias de acciones especifican posibles tratamientos del mencionado requerimiento.

Derivar el diseño de una arquitectura de solución para el tipo de requerimiento en cuestión, requiere un cambio en nuestro punto de vista:

- ✓ de un punto de vista "centrado en el Actor" a uno "centrado en el Sistema" y
- ✓ de un punto de vista funcional a uno orientado a objetos

En este punto es posible reorganizar y/o formalizar algunos aspectos de la especificación del Caso de Uso en cuestión, de modo que resulte menos 'brusco' y menos librado al azar el paso a una primera versión de una especificación de diseño.

Hay muchas propuestas formuladas en ese sentido (algunas de las cuales referiremos) pero, en este punto y a modo de ejemplo, formularemos una diferente que consideramos hace un aporte en una dirección diferente: identificar un modelo o patrón de tratamiento del referido requerimiento.

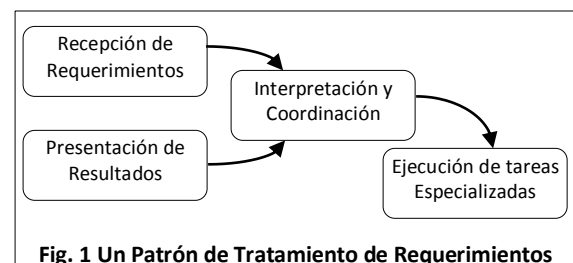


Fig. 1 Un Patrón de Tratamiento de Requerimientos

La figura 1 bosqueja un posible patrón que se caracteriza por su simplicidad pero que arroja una luz importante al proceso de derivación.

¹ Utilizamos aquí la palabra *Requerimiento* en lugar de *Requisito* para distinguir 'lo que se espera cumplir' (Requisito) de 'lo que se ha dado y es necesario procesar' (Requerimiento)

Este modelo, que conserva aún el enfoque funcional, tipifica conjuntos de responsabilidades organizando el conocimiento desde el punto de vista de tratamiento de requerimientos. Esta organización puede actuar como una guía efectiva a la hora de descubrir un conjunto de objetos que, asignándoles tales responsabilidades, sean capaces de colaborar para satisfacer el requerimiento.

Es posible identificar responsabilidades como:

- ✓ Recepción de Requerimientos: Presentación de Interfaces, Validación de Entrada, etc.
- ✓ Interpretación y Coordinación: Interpretación de Requerimientos, Selección de Plan de Acción, Coordinación de Colaboraciones, etc.
- ✓ Ejecución de Tareas Especializadas: Representación de Objetos de Negocio, Manejo de Ciclo de Vida de Objetos de Negocio, Coordinaciones Especializadas de Colaboraciones, etc.
- ✓ Presentación de Resultados: Presentación de Interfaces

Esta simple tipificación de responsabilidades en el tratamiento de requerimientos, nos permite inferir una colaboración en la que estarán presentes mínimamente, los siguientes componentes:

- ✓ el Actor que formula el requerimiento,
- ✓ un Gestor que posee todo el conocimiento acerca de cómo gestionar el requerimiento,
- ✓ un conjunto de objetos de negocio involucrados en la gestión y
- ✓ gestores de ciclo de vida de los objetos de negocio

La construcción de un modelo de colaboración y/o de secuencia a partir de esta información resulta prácticamente trivial.

Patrones Arquitecturales y de Diseño

En la actualidad ya nadie duda acerca del impacto que tiene la calidad del diseño en todos los procesos que componen el ciclo de vida de un Producto de Software, desde que es concebido como idea hasta que es retirado de producción.

En particular, el proceso que nos ocupa “derivación de diseños de solución para requisitos basados en Casos de Uso”, se ve impactado de manera especial, porque se trata del proceso mismo por el cual es derivado el diseño de solución.

Si aspiramos a contar con un proceso de derivación del diseño de solución que sea controlado y guiado por técnicas precisas que garanticen la calidad del resultado obtenido, podemos asegurar que el uso de patrones arquitecturales y de diseño constituye un recurso esencial en cualquier técnica que pretenda dar soporte al proceso.

Patrones Arquitecturales expresan esquemas de organización estructurales fundamentales de los sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen normas y directrices para organizar las relaciones entre los mismos [6].

Por lo tanto, un patrón arquitectural, actúa como marco de referencia contra el cual contrastar decisiones de diseño.

El patrón arquitectural subyacente a MVC (Model-View-Controller) [6] define con precisión un modelo arquitectural compuesto de 3 categorías de componentes (Modelo, Vista y Controlador) y define, también de manera precisa las relaciones y comunicaciones entre componentes.

De modo que, actúa como guía en el proceso de descubrimiento de componentes y como marco contra el cual contrastar decisiones.

Otra característica distintiva es su cercanía al negocio, es decir que, está definido de manera tal que es muy simple mapear conocimiento de negocio a las categorías definidas. Otros, como por ejemplo el expresado por JEE [7], son mucho más cercanos a la implementación, es decir que, define componentes arquitecturales y de diseño que son fácilmente mapeados a implementación.

Como es natural suponer, la definición de un patrón arquitectural más especializado que herede características de ambos patrones es motivo de trabajos que se vienen realizando en el marco de las líneas de investigación que proponemos.

El aporte que haría el uso de patrones arquitecturales y de diseño a posibles técnicas de derivación de diseño a partir de especificaciones de requisitos basadas en Casos de Uso es claro.

Si por ejemplo, como sugerimos en párrafos anteriores, inferimos una primera versión de un diseño de solución como una colaboración, en un necesario proceso de refinamiento de diseño, sería posible resolver delegaciones y/o abstracciones de comportamiento guiados por especificaciones de patrones arquitecturales y de diseño.

Además, al ser el diseño de solución obtenido instancia de los patrones utilizados, el objetivo de asistencia automatizada al proceso de derivación, es mucho más factible.

Siguiendo el enfoque MDA (Model Driven Architecture) [8], es posible extender los Meta-modelos de Casos de Uso y de Diseño, definir Meta-modelos de las técnicas sugeridas y definir transformaciones QVT entre éstos de manera que sean el fundamento de posibles soportes automatizados al proceso de derivación.

Clasificación del Conocimiento de Negocio

La asistencia al proceso de derivación de diseños de solución a partir de especificaciones de requisitos basadas en Casos de Uso constituye un objetivo verdaderamente ambicioso.

Ya hemos comentado acerca de la importancia de observar y analizar el conocimiento representado por el Caso de Uso a la luz de patrones.

Sin embargo es posible mejorar aún más la calidad del soporte al proceso si logramos mejorar la capacidad de interpretación del conocimiento de negocio.

Es decir, si logramos por ejemplo establecer una tipificación del conocimiento, entonces estaremos seguramente en mejor posición de tomar decisiones arquitecturales y de diseño al contrastar mejor el conocimiento de negocio contra las especificaciones de los patrones.

Por ejemplo tipificar el conocimiento de negocio clasificándolo en Procesos y Objetos de Negocio, sería fundamental a la hora de derivar la primera versión de diseño, según el ejemplo que mencionamos, dado que una de las clases de objeto que participan en la colaboración son precisamente los que modelan Objetos de Negocio.

En el caso de Procesos de Negocio, la correspondencia no aparece tan trivial como en el caso de los Objetos de Negocio.

Sin embargo, una tipificación más detallada de los Procesos de Negocio, como diferenciar Procesos Síncronos de Asíncronos y/o Procesos que requieren mantener estado de los que no lo requieren, sumado al uso de recursos que mejoren la especificación de los Casos de Uso, como los mencionados en estas secciones, permiten suponer que es posible especificar técnicas que den soporte efectivo al proceso de derivación de especificaciones de diseño.

Líneas de Investigación y Desarrollo

Ya nos hemos referido a la efectividad de la técnica de Casos de Uso en la actividad de captura de requisitos, como así también a los inconvenientes resultantes de la falta de formalidad que los aleja de las expresiones de diseño de solución para los casos modelados.

Posteriormente nos hemos referido a lo que pueden aportar, propuestas tales como Plantillas Genéricas [1] o Esquemas de Operación [2][3].

También nos hemos referido a la posibilidad de, a partir de conocimiento identificado en las descripciones de Casos de Uso, derivar versiones iniciales de especificaciones de diseño, las cuales

pueden ser refinadas posteriormente mediante el uso de posibles técnicas basadas en patrones.

Finalmente, además de sugerir el uso de propuestas como *Plantillas Genéricas* [1] o *Esquemas de Operación* [2] y el uso de posibles patrones o modelos funcionales para mejorar la formalidad e interpretación del conocimiento en descripciones de Casos de Uso, mostramos como se pueden adoptar otros puntos de vista para identificar tipos de conocimiento útiles al proceso de derivación y dimos un ejemplo tipificando el conocimiento de negocio como Objetos y Procesos de Negocio.

A partir de estos fundamentos proponemos líneas de investigación y desarrollo que intentan dar soporte al proceso de derivación de diseños de solución para especificaciones de requisitos basados en Casos de Uso.

Estas líneas de investigación y desarrollo, proponen la definición de tales soportes a partir de:

- 1) Identificación y formalización de conocimiento en los Casos de Uso sin resignar beneficios que caracterizan a las descripciones de los mismos (uso y extensión de propuestas tales como Plantillas Genéricas [1] y Esquemas de Operación [2]).
- 2) Uso de patrones, como patrones funcionales, patrones arquitecturales (MVC, JEE, CORBA, etc.) y patrones de diseño para soportar la tarea de derivación de diseño.
- 3) Tipificación del conocimiento de negocio para establecer correspondencia entre los tipos de conocimiento y componentes de los patrones
- 4) Extensión de meta-modelos de Casos de Uso y de diseño, definición de meta-modelos de técnicas de derivación y especificación de transformaciones entre meta-modelos que sirvan de base a posibles soportes automatizados a actividades de derivación.

Basados en ese planteamiento y en experiencias realizadas y en curso en el ámbito académico e industrial proponemos líneas de investigación con dos orientaciones:

- 1) Especificación de técnicas para derivación de diseños de solución a partir de especificaciones de requisitos basadas en Casos de Uso
- 2) Provisión de bases para la automatización de las técnicas especificadas siguiendo el enfoque MDA [8] mediante:
 - ✓ Extensión de meta-modelos de Casos de Uso
 - ✓ Extensión de meta-modelos de modelos de diseño
 - ✓ Especificación de meta-modelos de técnicas de derivación

- ✓ Definición de transformaciones utilizando QVT (y posiblemente XSLT) entre los meta-modelos

Resultados y Objetivos

En lo referido a Técnicas de Derivación de Diseño a partir de Especificaciones de Requisitos basadas en Casos de Uso, se han formulado algunas hipótesis de trabajo:

- ✓ A partir de un modelo de tratamiento de requerimientos, derivación de primera versión de un Modelo de Secuencia asignando responsabilidades del modelo
- ✓ Aplicación de pasos de refinamiento aplicando técnicas de Delegación apoyadas en Patrones de Diseño simples como por ejemplo de Composición, Fachada, Comandos, etc.
- ✓ Estructuración arquitectural del diseño de solución obtenido basada en parámetros resultantes de la integración de patrones funcionales (Patrón de tratamiento de requerimientos), Patrones Arquitecturales (MVC, JEE) y tipificación de Conocimiento de Negocio.

En base a estas hipótesis de trabajo se ha formulado el desarrollo de casos experimentales (algunos ya terminados y otros aún en curso) en el ámbito académico y privado.

En lo referido a extensión de meta-modelos existentes y definición de meta-modelos de técnicas de derivación, se encuentran en desarrollo el estudio de meta-modelos UML involucrados en el alcance, y la especificación de necesidades de extensión, en función de resultados obtenidos a partir de casos experimentales. Esta actividad se está cumpliendo como parte del desarrollo de tesis de maestría correspondientes a la Maestría en Ingeniería de Software de la Universidad Nacional de San Luis y en el marco del proyecto de investigación.

También en el mismo marco de tesis de maestría y proyecto, se están analizando y especificando casos de transformaciones utilizando QVT y XSLT.

El objetivo a mediano plazo consiste en alcanzar un punto en el cual sea posible especificar formalmente algunas técnicas de derivación junto con las especificaciones de meta-modelo y posibles transformaciones.

Formación de Recursos Humanos

En el área de formación de recursos humanos, desde el año 2006 se vienen desarrollando varias actividades de capacitación y experimentación:

- ✓ Inclusión de un Laboratorio de experimentación en la aplicación manual de técnicas de derivación en:
 - Arquitectura Web: Materia optativa correspondiente a 5to. Año de la carrera Licenciatura en Ciencias de la Computación
 - Ingeniería Web: Materia correspondiente a 3er. Año de la carrera Tecnicatura Universitaria en Web
- ✓ Formulación de Trabajos Finales correspondientes a la carrera Tecnicatura Universitaria en Web. Estos trabajos recogen importante información experimental porque se desarrollan Productos de Software destinados al mercado local
- ✓ Actualmente se encuentra en curso el desarrollo de:
 - Un trabajo de tesis de licenciatura que intenta aplicar técnicas de derivación en el desarrollo del producto objetivo de la tesis
 - Dos trabajos de tesis de maestría con objetivos que incluyen extensión y definición de meta-modelos y especificación de transformaciones QVT y XSLT entre meta-modelos

Referencias

- [1] Marcela Daniele, Daniel Riesco, et al., Transformación de modelos aplicada a la definición genérica de Casos de Uso utilizando QVT (Query/View/Transformation) y RTG (Reglas de Transformación de Grafos), WICC 2011.
- [2] Shane Sendall, Alfred Strohmeier: From Use Cases to System Operation Specifications, UML 2000, Octubre 2000.
- [3] I. Jacobson, M. Griss and P. Jonsson. Software Reuse: Architecture Process and Organization for Business Success, Addison-Wesley 1997
- [4] J. Warmer and A. Kleppe, The Object Constraint Language: Precise Modeling With UML, Addison-Wesley 1998
- [5] OMG Unified Modeling Language Superstructure Specification, Mayo 2010, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>
- [6] Buschmann, Meunier, Ronhert, Sommerlad, Stal. Pattern Oriented Software Architecture – A System of Patterns, Febrero, 2001
- [7] Java Enterprise Edition (JEE), <http://www.oracle.com/technetwork/java/javae/overview/index.html>
- [8] OMG Model Driven Architecture (OMG MDA), <http://www.omg.org/mda/>