

Resolución de conflictos para el modelado estático orientado a aspectos

Laura Zeligueta, Marcelo Palma, Fernando Pincioli

Instituto de Investigaciones
Facultad de Informática y Diseño
Universidad Champagnat
Mendoza, Argentina, +54 (261) 424-8443
zeligueta@uch.edu.ar; mpalma@uch.edu.ar; fpincioli@aconcaguasf.com

Resumen

En este proyecto se está trabajando en el establecimiento de los lineamientos para la fase de requisitos de software del ciclo de vida del desarrollo de software (SDLC), poniendo especial énfasis en la vista estática y con el empleo del paradigma de orientación a aspectos, de manera de obtener los beneficios que ofrece [1].

Es el quinto de la misma línea de investigación de desarrollo de software orientado a aspectos (AOSD) que se vienen desarrollando en forma consecutiva desde el año 2013 en el Instituto de Investigaciones de la Facultad de Informática y Diseño de la Universidad Champagnat y que continúan de un proyecto que fuera desarrollado en la Facultad Regional Mendoza de la Universidad Tecnológica Nacional.

Palabras clave: orientación a aspectos, aspectos tempranos, requisitos de usuario, requisitos funcionales, separación de incumbencias, encapsulamiento de incumbencias, composición de incumbencias, resolución de conflictos, AOP4ST.

Contexto

Dentro de las líneas de investigación del Instituto de Investigaciones de la Facultad de Informática y Diseño de la Universidad Champagnat, se encuentra la línea de “Ingeniería de Software”. En ella, se llevaron a cabo los siguientes proyectos dedicados al

desarrollo de software orientado a aspectos (AOSD):

- “*Procesos de desarrollo de software de calidad basados en aspectos*”, realizado en conjunto con la UTN Facultad Regional Mendoza durante 2011 y 2012 [2].
- “*Definición de criterios para la detección temprana de aspectos en el modelado de negocios y el desarrollo de los requisitos*” llevado a cabo por completo en la Universidad Champagnat, desde 2013 a 2016 y presentado en WICC 2015 [3].
- “*Modelado de procesos de negocio orientados a aspectos con BPMN*”, de la Universidad Champagnat y realizado en los años 2016 a 2018, y que fue presentado en WICC 2016 [4] y 2017 [5].
- “*Ingeniería de requisitos orientada a aspectos en AOP4ST*”: este proyecto abordó la fase de requisitos de usuario y se desarrolló entre 2018 y 2020 [6].

El proyecto actual, que se denomina “Ingeniería de requisitos de software orientada a aspectos en AOP4ST”, se inició en julio de 2020 y ya se produjeron algunas publicaciones: [1] [7] [8]. Cuenta con el financiamiento de la Universidad Champagnat y de la empresa Aconcagua Software Factory S.A., una de las principales fábricas de software de la Argentina.

Introducción

AOP4ST, sigla derivada de Aspect-Oriented Process for a Smooth Transition, es un proceso marco para el SDLC. En su versión actual, está enfocado en las fases tempranas,

early aspects [9], compuesta por diferentes modelos: de negocio, de requisitos de usuario y de requisitos de software. Este último considera tres vistas: funcional, estática y de estados.

El acrónimo AOP4ST corresponde Aspect-Oriented Process for a Smooth Transition, que es un proceso marco para el desarrollo de software que pretende [8]:

- a) Ofrecer un proceso marco, no específico y liviano, de modo que permita su empleo con diferentes modelos del ciclo de vida del desarrollo del software (SDLC).
- b) Cubrir las etapas tempranas del SDLC, desde el modelado de negocio hasta la obtención de una especificación de requisitos completa y coherente.
- c) Emplear herramientas y técnicas estándares, de amplia difusión en la industria, para facilitar su adopción inmediata.
- d) Emplear notaciones estándares, para lograr modelos y especificaciones comprensibles y no ambiguas, que puedan contar con soporte de herramientas de software disponibles en el mercado.
- e) Desarrollar los productos intermedios de las etapas tempranas del SDLC empleando el paradigma de la orientación a aspectos.
- f) Obtener las incumbencias en forma progresiva a lo largo de todos los modelos.
- g) Mantener la separación de incumbencias a lo largo de todos los modelos.
- h) Mantener la trazabilidad bidireccional de las incumbencias de punta a punta.
- i) Obtener las incumbencias en forma natural a lo largo de todos los modelos, de manera de no afectar la obtención de los objetivos de cada modelo.

El presente proyecto de investigación pretende establecer lineamientos para la vista estática del modelo de requisitos de software a partir de la vista funcional realizada en trabajos anteriores, dejando la vista de estados como próximo objeto de estudio. Procura presentar una visión interna del dominio del problema al descomponerlo en las clases específicas que lo

constituyen, a diferencia del modelo de casos de uso, que presenta una visión externa del sistema [7].

Considerando las actividades propias del enfoque orientado a aspectos, mantiene la separación de incumbencias considerando los mismos paquetes que existen en la vista funcional del modelo de requisitos y elaborando dentro de ellos los diagramas de clases que materializan la vista estática.

En cuanto a la composición aplica un procedimiento sintáctico que combina las clases que inicialmente poseen el mismo nombre en las diferentes incumbencias. El total de los atributos, operaciones y relaciones es el resultado de la unión de todos estos elementos desde todas las incumbencias individuales y que terminan apareciendo por completo, y solo una vez, en la composición resultante. El proceso presenta distintas estrategias según se trate de incumbencias pares e impares [8].

En cuanto a la resolución de conflictos establece algunos acuerdos y consideraciones previas y preventivas tales como criterios de diagramación y nomenclatura.

Finalmente, propone buenas prácticas de manera de asegurar los objetivos de la fase del ciclo de vida en cuestión, la calidad del producto software en desarrollo y la obtención de los beneficios que son la razón del empleo del paradigma de orientación a aspectos.

Estos beneficios apuntan al desarrollo de un producto de software final, como así también de los productos intermedios que permiten producirlo, más modular, mantenible, reusable, extensible, comprensible, etc. [8], al administrar en forma separada las incumbencias que están desparramadas y enredadas en cada uno de los niveles de abstracción a lo largo del ciclo de vida completo.

Desde ya que los principales aportes a los requisitos tempranos (early aspects) con el

empleo de modelado estático, obtenidas mediante nuestro estudio de mapeo sistemático [1] y otros estudios de mapeo complementarios [10] [11] [12] [13], son fuentes obligadas de referencia y de las que nos nutrimos en este proyecto.

Líneas de Investigación, Desarrollo e Innovación

Los proyectos de investigación que se vienen desarrollando en nuestra Facultad presentan cuatro ejes de investigación, donde los tres primeros se ven integrados con el cuarto:

1. Modelado de procesos de negocio orientados a aspectos.
2. Separación y composición de incumbencias con resolución de conflictos.
3. Especificación de requisitos y gestión de incumbencias con casos de uso e historias de usuario.
4. Procesos de desarrollo de software orientados a aspectos.

Resultados y Objetivos

Se consideraron resultados obtenidos en proyectos anteriores como insumos para el trabajo actual, como así también la línea de trabajo que sigue AOP4ST. Tras haber detectado las primeras incumbencias pares e impares en el modelo de negocio [4] [5], con el empleo del enfoque asimétrico tradicional de la orientación a aspectos [14] [15] y el enfoque simétrico que se propuso más tarde [16], se sigue el desarrollo y la administración de los requisitos de usuario, que permiten enriquecer el modelo con nuevas incumbencias [17]. Estos requisitos de usuario son especificados mediante elementos vistas: funcional, estática y de estados [3].

La vista estática constituye el objeto de estudio de nuestro proyecto de investigación actual. Tiene como objetivo el profundizar los aportes que ella debe realizar a AOP4ST desde su rol en el proceso marco completo.

Hasta el momento, estos son los resultados alcanzados:

Detección y separación de incumbencias: desde el modelo de negocio y hasta el modelo de requisitos de software en su vista funcional fue posible mantener separadas las incumbencias. Pero al entrar en la vista estática, se observa que la mayoría de las clases involucradas en las funcionalidades (incumbencias) que se mantuvieron separadas hasta aquí, están presentes en más de una funcionalidad. Dicho de otra manera, los elementos de las vistas funcional y estática se relacionan de manera “muchos a muchos”, con lo que aparece nuevamente el enredo. Así, para poder seguir manteniendo la separación de las incumbencias en la vista estática, se consideró el concepto de “módulo de caso de uso” [18], que contiene un diagrama de clases para cada incumbencia en forma separada y que, por supuesto, luego se deberán componer. En nuestro proyecto trabajaron por separado cuatro analistas de sistemas elaborando diagramas de clases correspondientes a procesos que pertenecen a un laboratorio bioquímico.

Composición de incumbencias: se realizó teniendo en cuenta el proceso que define AOP4ST [8] obteniendo un único modelo sobre los diagramas de clases desarrollados por los diferentes analistas. Dado que hay diferentes formas de realizar la composición del modelo de clases, se siguieron las alternativas de mezcla y sobreescritura propuestas por Clarke y Baniassad [15], a las que incorporamos algunas variaciones y a las que sumamos una tercera solución, de selección, que comprobamos que ofrece resultados más alentadores. Durante este proceso es posible la aparición de nuevas clases que podrían suscitar nuevas incumbencias. Esto obliga a que las vistas estática y funcional se elaboren en forma iterativa e incremental.

Resolución de conflictos: Finalizada la composición de los diagramas se identificaron, analizaron y clasificaron los conflictos. En primer lugar, se elaboró un conjunto de acuerdos y recomendaciones para reducir el número de conflictos. Luego, los modelos se

reconstruyeron con el uso del conjunto de acuerdos de modelado, y se volvieron a componer, con una importante mejora en el número de conflictos detectados después de la segunda composición.

Creemos que hay mucho que hacer sobre las actividades pre y post composición[19]. Pensamos profundizar en el estudio sobre el impacto de los cambios que hacemos durante la composición: adición, eliminación, modificación y derivación de elementos en el modelo compuesto[20]. La producción de patrones de modelado también será muy útil. Las reglas y convenciones para nombrar elementos son cruciales ya que los nombres de los elementos de modelado son la base de la composición. El estudio de los diferentes tipos de conflictos y su clasificación también será una gran contribución al diseño de técnicas que permitan abordarlos y resolverlos de manera más efectiva. La composición de las relaciones es también un área importante para explorar[21]. Finalmente, la automatización de nuestras reglas colaborará con una composición eficiente de modelos.

Formación de Recursos Humanos

El proyecto promueve la formación del equipo de profesores, alumnos y egresados de la Universidad Champagnat.

Además, se están elaborando diferentes tesis sobre el cuerpo de esta investigación:

1. Fernando Pincirolí, Doctorado en Ciencias Informáticas, Universidad Nacional de San Juan. Tesis defendida, con la que obtuvo el grado de Doctor en Ciencias de la Informática en noviembre de 2020.
2. Marcelo Palma, Maestría en Negocios y Tecnología, Universidad Champagnat. Tesis en desarrollo.
3. Marcelo Fransoy, Maestría en Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. Tesis en desarrollo.
4. Jerónimo Vargas, Maestría en Negocios y Tecnología, Universidad Champagnat. Tesis en desarrollo.
5. Pablo Ingrassia, Maestría en Negocios y Tecnología, Universidad Champagnat. Tesis en desarrollo.
6. Gustavo Albino, Maestría en Ingeniería de Software, Universidad Nacional de San Luis. Tesis en desarrollo.
7. Marcelo Fransoy, Especialización en Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. Tesis en desarrollo.

Referencias

- [1] F. Pincirolí, J. L. Barros-Justo, and R. Forradellas, "Systematic Mapping Study: on the coverage of aspect-oriented methodologies for the early phases of the software development life cycle," *J. King Saud Univ. - Comput. Inf. Sci.*, 2020.
- [2] F. Pincirolí, "Consideraciones para un proceso de desarrollo de software de calidad orientado a aspectos," in *Sexto Encuentro de Investigadores y Docentes de Ingeniería EnIDI 2011*, 2011.
- [3] F. Pincirolí, "AOP4ST – Aspect-Oriented Process for a Smooth Transition," in *WICC 2015 - XVII Workshop de Investigadores en Ciencias de la Computación*, 2015.
- [4] F. Pincirolí and L. Zeligueta, "El modelo de negocio en AOP4ST," in *WICC 2016 - XVIII Workshop de Investigadores en Ciencias de la Computación*, 2016.
- [5] F. Pincirolí and L. Zeligueta, "Modelado de negocios orientado a aspectos con AOP4ST," in *WICC 2017 - XIX Workshop de Investigadores en Ciencias de la Computación*, 2017.
- [6] F. Pincirolí and M. Palma, "Desarrollo de requisitos en 'Aspect-Oriented Process for a Smooth Transition,'" in *XXII Workshop de Investigadores en Ciencias de la Computación - WICC 2020*, 2020.
- [7] F. Pincirolí, "Modeling the Static View in Aspect-Oriented Software Development," no. section 6.
- [8] F. Pincirolí, Proceso marco orientado a aspectos en las etapas tempranas del ciclo de vida del desarrollo de software para una transición en la industria, Universidad Nacional de San Juan, San Juan, 2020.
- [9] A. Rashid, P. Sawyer, A. Moreira, and J. Araújo, "Early aspects: A model for aspect-oriented requirements engineering," *Proc.*

- IEEE Int. Conf. Requir. Eng.*, vol. 2002-Janua, pp. 199–202, 2002.
- [10] A. Magableh, Z. Shukur, and N. M. Ali, “Systematic review on aspect-oriented UML modeling: A complete aspectual UML modeling framework,” *J. Appl. Sci.*, vol. 13, no. 1, pp. 1–13, 2013.
- [11] A. A. Magableh, “Systematic review of aspect-oriented formal method,” *Int. J. Comput. Appl. Technol.*, vol. 56, no. 2, pp. 132–140, 2017.
- [12] M. Guessi, L. B. R. Oliveira, and E. Y. Nakagawa, “Modeling Aspect-oriented Software Systems Using UML: A Systematic Mapping,” no. November 2016, 2010.
- [13] M. Guessi, L. B. Ruas Oliveira, and E. Yumi Nakagawa, “Extensions of UML to Model Aspect-oriented Software Systems,” *CLEI Electron. J.*, vol. 14, no. 1, pp. 1–18, 2011.
- [14] M. Katara and S. Katz, “A concern architecture view for aspect-oriented software design,” *Softw. Syst. Model.*, vol. 6, no. 3, pp. 247–265, Aug. 2007.
- [15] S. Clarke and E. Baniassad, *Aspect-oriented analysis and design. The Theme approach*. Boston: Addison-Wesley, 2005.
- [16] W. Harrison, H. Ossher, and P. Tarr, “Asymmetrically vs. Symmetrically Organized Paradigms for Software Composition,” vol. 22685, 2002.
- [17] F. Pincioli, “Requisitos de usuario y gestión de la demanda en AOP4ST,” in *XX Workshop de Investigadores en Ciencias de la Computación - WICC 2018*, 2018.
- [18] I. Jacobson and P. Ng, *Aspect-oriented software development with use cases*. Addison-Wesley, 2005.
- [19] R. Reddy, R. France, F. Fleury, and B. Baudry, “Model composition - a signature based approach,” *Proc. Asp. Oriented Model. Work. held with Model. 2005*, p. 2, 2005.
- [20] K. S. Farias de Oliveira, “Empirical Evaluation of Effort on Composing Design Models,” Pontificia Universidade Católica do Rio de Janeiro, 2012.
- [21] S. A. H. Elasri, E. Elabbassi, Semantic integration of UML class diagram with semantic validation on segments of mappings, 2018.