

## Desarrollo de incumbencias en el modelado de la vista estática

Fernando Pincirolí, Laura Zeligueta, Marcelo Palma,  
Iván Cappello y Elías Motta

Instituto de Investigaciones  
Facultad de Informática y Diseño  
Universidad Champagnat  
Mendoza, Argentina, +54 (261) 424-8443  
pincirolifernando@uch.edu.ar; zeliguetalaura@uch.edu.ar; mpalma@civitas.com.ar;  
ivancappello.e@gmail.com; eliasmotta1@gmail.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].

Este proyecto 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, y a lo largo de poco más de siete años, 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 algunas 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 siguientes resultados alcanzados y los objetivos pendientes:

**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 relación 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.

**Composición de incumbencias:** en este momento nos encontramos trabajando en este punto, teniendo en cuenta el proceso que define AOP4ST [8]. Dado que hay diferentes formas de realizar la composición del modelo de clases, se siguieron las alternativas de mezcla y sobrescritura propuestas por Clarke y Baniassad [15], a las que en 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. La composición desde lo semántico, que es la parte más difícil de resolver y a la que se debe prestar la mayor atención, requiere un estudio más profundo y que está pendiente en nuestra agenda de trabajo.

**Resolución de conflictos:** se está trabajando fuertemente en este punto, con cuatro analistas elaborando los mismos modelos por separado para analizar luego los conflictos que se generen, diseñar reglas de trabajo que apunten a reducirlas, volver a elaborar los mismos modelos y, finalmente, evaluar el grado de reducción de los conflictos y la consecuente necesidad de ajustar el conjunto de reglas.

## Formación de Recursos Humanos

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

Además, ya se han defendido tesis y se están elaborando otras sobre el cuerpo temático de esta investigación:

1. Tesis de Fernando Pincioli, Doctorado en Ciencias Informáticas, Universidad Nacional de San Juan, con la que se obtuvo el grado de Doctor en Ciencias de la Informática en noviembre pasado.
2. Tesis de Marcelo Palma, Maestría en Negocios y Tecnología, Universidad Champagnat.
3. Tesis de Jerónimo Vargas, Maestría en Negocios y Tecnología, Universidad Champagnat.
4. Tesis de Gustavo Albino, Maestría en Ingeniería de Software, Universidad Nacional de San Luis.
5. Tesina de Elías Motta, Licenciatura en Sistemas de Información, Universidad Champagnat.

6. Tesina de Javier Amutio, Licenciatura en Sistemas de Información, Universidad Champagnat.

## Referencias

- [1] Pinciroli, F., Barros Justo, J.L. and Forradellas, R. (2020). Systematic Mapping Study: on the coverage of aspect-oriented methodologies for the early phases of the software development life cycle. *Journal of King Saud University, Computer and Information Sciences*, Elsevier.
- [2] Pinciroli, F. (2011). Consideraciones para un proceso de desarrollo de software de calidad orientado a aspectos. Sexto Encuentro de Investigadores y Docentes de Ingeniería, EnIDI 2011, San Rafael.
- [3] Pinciroli, F. (2015). AOP4ST - Aspect-Oriented Process for a Smooth Transition. XVII Workshop de Investigadores en Ciencias de la Computación, WICC 2015, Salta.
- [4] Pinciroli, F. and Zeligueta, L. (2016). El modelo de negocio en AOP4ST. XVIII Workshop de Investigadores en Ciencias de la Computación, WICC 2016, Concordia, pp. 423-426.
- [5] Pinciroli F. and Zeligueta, L. (2017). Modelado de negocios orientado a aspectos con AOP4ST. XIX Workshop de Investigadores en Ciencias de la Computación, WICC 2017, Buenos Aires.
- [6] Pinciroli, F. and Palma M. (2020). Desarrollo de requisitos en “Aspect-Oriented Process for a Smooth Transition”. XXII Workshop de Investigadores en Ciencias de la Computación, WICC 2020, El Calafate.
- [7] Pinciroli, F. (2019). Modeling the Static View in Aspect-Oriented Software Development. III Congreso Internacional de Ciencias de la Computación y Sistemas de Información, CICC SI 2019, Mendoza.
- [8] Pinciroli, F. (2019). Proceso marco orientado a aspectos en las etapas tempranas del ciclo de vida del desarrollo de software para una transición en la industria. Tesis del Doctorado en Ciencias de la Informática, Universidad Nacional de San Juan, 2019.
- [9] Rashid, A., Sawyer, P., Moreira, A. and Araújo, J. (2002). Early aspects: a model for aspect-oriented requirements engineering. *IEEE Joint Int. Conference on Requirements Engineering*.
- [10] Magableh, A., Shukur, Z. and Ali, N.M. (2013). Systematic review on aspect-oriented UML modeling: A complete aspectual UML modeling framework. *Journal of Applied Sciences*, vol. 13, no. 1, pp. 1–13.
- [11] Magableh, A.A. (2017). Systematic review of aspect-oriented formal method. *International Journal of Computers and Applied Technology*, vol. 56, no. 2, pp. 132–140.
- [12] M. Guessi, L., Oliveira, B.R. and Nakagawa, E.Y. (2010). Modeling Aspect-oriented Software Systems Using UML: A Systematic Mapping. *CLEI Electronic Journal* vol.14, no.1, p.3.
- [13] M. Guessi, L. Oliveira, B.R. and Nakagawa, E.Y. (2011). Extensions of UML to Model Aspect-oriented Software Systems. *CLEI Electronic Journal* vol.14, no.1, pp. 1-18.
- [14] Katara, M. and Katz, S. (2007). A Concern Architecture View for Aspect-Oriented Software Design. *Software and Systems Mod.*, vol. 6, no. 3, pp. 247–265.
- [15] Clarke, S. and Baniassad, E. (2005). *Aspect-Oriented Analysis and Design. The Theme Approach*. Addison-Wesley, Boston, 2005.
- [16] Harrison, W., Ossher, H. and Tarr, P. (2002). Asymmetrically vs. Symmetrically Organized Paradigms for Software Composition. *IBM Research Report RC22685*.
- [17] Pinciroli, F. (2018). Requisitos de usuario y gestión de la demanda en AOP4ST. XX Workshop de Investigadores en Ciencias de la Computación, WICC 2018, Buenos Aires, pp. 511-515.
- [18] Jacobson, I. and Ng, P. (2005). *Aspect-oriented software development with use cases*. Addison-Wesley, 2005.