

# AOP4ST – Aspect-Oriented Process for a Smooth Transition

Fernando Pincirolí

Instituto de Investigaciones  
Facultad de Informática y Diseño  
Universidad Champagnat  
Mendoza, Argentina, +54 (261) 424-8443  
pincirolifernando@uch.edu.ar

## Resumen

En el paradigma de orientación a aspectos actualmente existen numerosas propuestas e ideas para su aplicación en las etapas tempranas del ciclo de vida de desarrollo de software. Pero todas ellas presentan algún inconveniente: no son estándares, no cubren todo el ciclo de vida, no poseen herramientas de soporte automatizado, no contemplan prácticas difundidas en la industria, etc.

El objetivo de nuestra línea de investigación consiste en establecer un proceso que sirva de base para su aplicación real en la industria y que permita aprovechar los beneficios de la orientación a aspectos en las etapas tempranas del ciclo de vida, proveyendo, además la posibilidad de una transición suave desde las prácticas actuales hacia las necesarias para aplicar el paradigma orientado a aspectos.

**Palabras clave:** orientación a aspectos, procesos de desarrollo de software, early aspects, modelado de negocio, modelado de requerimientos, modelado de requisitos, separación de incumbencias, composición de incumbencias, resolución de conflictos, weaving, AOP4ST.

## Contexto

Esta línea de investigación continúa el trabajo que los miembros del equipo vienen realizando en el área de ingeniería de software más los avances alcanzados en el proyecto de investigación “Procesos de desarrollo de software de calidad basados en aspectos”, llevado a cabo en la UTN Facultad Regional Mendoza, y que ahora se está profundizando en el proyecto “Definición de criterios para la detección temprana de aspectos en el modelado de negocios y el desarrollo de los requisitos” de la Universidad Champagnat de la provincia de Mendoza.

## Introducción

A lo largo de la historia del desarrollo de software nos hemos enfrentado a numerosos cambios paradigmáticos que a nivel académico no presentan mayores dificultades, pero que en la aplicación práctica en la industria provocaron impactos importantes. En ella, los cambios tecnológicos se van incorporando de manera escalonada debido a que no es posible modificar fácilmente la realidad tecnológica en la que se encuentran inmersas, ya sea por la dependencia que tienen con tecnologías de desarrollo, por la formación de sus recursos humanos, por el alto costo de cambiar de tecnología sus productos, etc.

Por otra parte, el desarrollar software de calidad es una tarea difícil y a la que no se puede agregar complejidad innecesaria sin que esto impacte negativamente en la marcha y en el resultado de los proyectos. Por esta razón es que no se pueden incorporar elementos puramente teóricos o con escasa difusión a los procesos de desarrollo, agregando de esta manera factores de riesgo adicionales, sin considerar su impacto en la dificultad real de conseguir recursos humanos con las habilidades y experiencia necesarias en las herramientas y técnicas a emplear, en las mismas herramientas y técnicas disponibles, etc., cuestiones todas que en definitiva son las que terminan determinando el éxito o del fracaso de los proyectos.

Hoy existe una vastísima diversidad de ideas propuestas que ofrecen alternativas de técnicas y herramientas para el desarrollo de sistemas orientados a aspectos pero que, al mismo tiempo, no aportan una solución completa a la problemática mencionada.

Por todo lo dicho es que nos hemos enfocado en tomar de entre todas esas ideas aquellas que, a nuestro criterio y por nuestra experiencia académica y en la industria, son factibles de implementar al permitir realizar una transición gradual desde las técnicas actualmente difundidas hacia el empleo práctico de los conceptos de la orientación a aspectos. Planteándolo de forma un tanto singular, podríamos verlo como una “aspectización” de las prácticas actuales de la ingeniería de software.

De esta manera, lo que procuramos es aprovechar la formación actual de los profesionales de sistemas que se desempeñan en las diferentes empresas y organizaciones, mantener el empleo de las herramientas de modelado que hoy usan, poner el foco en las prácticas que realmente dan resultados, etc. pero al

mismo tiempo con la incorporación de los conceptos de la orientación a aspectos que efectivamente aportan valor, de modo de poder llevar adelante una transición gradual hacia el cambio de paradigma.

Estamos convencidos de que la orientación a aspectos presenta beneficios en todas las etapas del desarrollo de software, tal como coinciden otros autores [Gra03], de modo que con nuestra propuesta no es necesario que las organizaciones estén hoy mismo desarrollando software con lenguajes orientados a aspectos. Por supuesto que cuando estén en condiciones de hacerlo podrán aprovechar todos los beneficios que aporta el enfoque de aspectos, pero mientras tanto, e insistiendo con nuestra idea de lograr una transición suave, se podrán obtener aquellos resultados que se encuentran en las etapas tempranas del desarrollo de software.

## **Líneas de Investigación, Desarrollo e Innovación**

Nuestro proyecto presenta 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.
4. Procesos de desarrollo de software orientados a aspectos.

## **Resultados y Objetivos**

Con respecto a la línea de **modelado de procesos de negocio orientado a**

**aspectos** aprovechamos el impulso que tuvo con la amplia aceptación de la propuesta de AO4BPMN [Cha10], además de otras ideas [Cap10, Jal11, Nog11]. El enfoque liviano de AO4BOMN nos permitió tomar sus ideas para no salirnos de la notación estándar BPMN 2.0 y aprovechar así las herramientas de soporte automatizado ampliamente difundidas. En este momento nos encontramos elaborando prácticas de modelado recomendadas, resolviendo cuestiones de composición de incumbencias y resolución de conflictos y elaborando un weaver para componer modelos de BPMN completos a partir de modelos de BPMN orientados a aspectos.

En cuanto a la línea de **separación y composición de incumbencias con resolución de conflictos**, estamos trabajando en su aplicación sobre varios modelos: el de negocio antes mencionado, el de requerimientos, el de especificaciones de requisitos con un enfoque funcional mediante el empleo de casos de uso [Chi07, Cla05, Ead06, Fil04, Kic97, Van07] y con un enfoque estático y el de estados. Concebimos la separación de incumbencias como el resultado de una evolución progresiva a lo largo de los modelos mencionados y poniendo énfasis también en las incumbencias simétricas que hoy no cuentan con toda la base de conocimiento existente en las asimétricas [Bal12, Col12, Har02, Bal11]. Debemos realizar la composición de manera diferente dependiendo del modelo en particular, destacándose la composición de modelos de clases y de estados. La resolución de conflictos es otro punto clave que de atención al presentar características diversas dependiendo de los modelos a componer.

La línea de **especificación de requisitos y gestión de incumbencias con casos de uso** tiene por un lado la ventaja de que la técnica de casos de uso

está ampliamente difundida y, por consiguiente, hay una buena base acerca de su aplicación en el paradigma de aspectos [Jac05, Ara03, Eck96, Gom04, Paw04, Som08, Wam03], pero como contrapartida, en muchas oportunidades se confunde la técnica de casos de uso con la descomposición funcional llevando a conclusiones imprecisas, de modo que para nuestro proyecto la problemática se centra en asegurarnos la aplicación de la técnica de manera estándar, dando soporte completo a la orientación a aspectos, y hacerlo de forma integrada con los modelos mencionados y con otros modelos complementarios que son fundamentales en el ciclo de vida del desarrollo de software, por ejemplo, los modelos de estados, de pruebas, etc.

Por último, en la línea de **procesos de desarrollo de software orientados a aspectos** estamos elaborando una propuesta de un proceso completo para las etapas tempranas del ciclo de vida, desde el modelado de negocio hasta la culminación de la especificación de requisitos con la obtención de los beneficios de la orientación a aspectos mediante el empleo de técnicas y notaciones estándares y de herramientas de soporte automatizado de amplia difusión en el mercado, más la consideración de cuestiones prácticas que se dan en equipos de trabajo en proyectos de desarrollo reales. Por esta razón es que decidimos llamarlo AOP4ST, al ser un proceso realmente orientado a aspectos pero que por sus características especiales permite aplicarlo en la realidad y lograr una incorporación suave del paradigma en los proyectos. Esta línea, en la que venimos trabajando desde hace un tiempo [Pin11], es integradora de las otras tres. Las principales conclusiones que estamos obteniendo son logradas, también, por la aplicación práctica en proyectos de desarrollo de software de complejidad

diversa para organizaciones de envergadura en empresas de ingeniería de software mendocinas.

## Formación de Recursos Humanos

El proyecto apunta a la formación del equipo de profesores del área de Ingeniería de Software de la Universidad Champagnat y de alumnos participantes tanto del proyecto en sí como de sus avances y resultados.

Por el momento existe una tesina de grado en curso de Noemí Morales: “Software para implementación de mecanismos de composición de incumbencias transversales en BPMN”. Dirección: Fernando Pincirolí.

También estamos colaborando en la definición de una tesis de maestría y dos tesinas de grado sobre los temas alcanzados por nuestro proyecto.

## Referencias

[Ara03] ARAÚJO, J. y A. Moreira. An Aspectual Use-Case Driven Approach. VIII Jornadas Ingeniería del Software y Bases de Datos (JISBD), 2003.

[Bal11] BÁLIK, J. y V. Vranić. Sustaining Composability of Aspect-Oriented Design Patterns in Their Symmetric Implementation. En: 2<sup>nd</sup> International Workshop on Empirical Evaluation of Software Composition Techniques, ESCOT 2011, ECOOP 2011. Lancaster, julio de 2011.

[Bal12] BÁLIK, J. y V. Vranić. Symmetric Aspect-Oriented: Some

Practical Consequences. En: NEMARA 2012, International Workshop on Next Generation Modularity Approaches for Requirements and Architecture, AOSD 2012. Postdam, marzo de 2012.

[Cap10] CAPPELLI, et al. Reflections on the Modularity of Business Process Models: the Case for Introducing the Aspect-Oriented Paradigm. BPM Journal, 2010.

[Cha10] CHARFI, A. et al. Aspect-Oriented Business Process Modeling with AO4BPMN. Modelling Foundations and Applications. Lecture Notes in Computer Science Volume 6138, 2010.

[Chi07] CHITCHYAN, R. et al. Semantics-based composition for aspect-oriented requirements engineering. AOSD '07: Proceedings of the 6<sup>th</sup> International Conference on Aspect-Oriented Software Development, ACM, 2007.

[Cla05] CLARKE, S. y E. Baniassad. Aspect-Oriented Analysis and Design. The Theme Approach. Addison-Wesley, 2005.

[Col12] COLOMER COLLEL, D. Aspect-Oriented Modeling of Business Processes. Tesis de Maestría, Universidad Técnica de Darmstadt, 2012.

[Ead06] EADDY, M. An Empirical Assessment of the Crosscutting Concern Problem. Tesis doctoral de la Universidad de Columbia, 2006.

[Eck96] ECKLUND, E. et al. Change cases-Use cases that identify future

- requirements. Proceedings of OOPSLA 96, California, octubre de 1996.
- [Fil04] FILMAN, R. et al. Aspect-Oriented Software Development. Addison-Wesley, 2004.
- [Gom04] GOMAA, H. Designing software product lines with UML—From use cases to pattern-based software architectures. Addison-Wesley, 2004.
- [Gra03] GRAY, J. et al. Aspect-Oriented Domain-Specific Modeling. Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, 2003.
- [Har02] HARRISON, W. et al. Asymmetrically vs. Symmetrically Organized Paradigms for Software Composition. IBM Research Report. RC22685 (W0212-147), 20 de diciembre de 2002.
- [Jac03] JACOBSON, I. Use Cases and Aspects – Working Seamlessly Together. En: Journal of Object Technology, vol. 2, nro. 4, julio-agosto de 2003.
- [Jac05] JACOBSON, I. y P. Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2005.
- [Jal11] JALALI, A. Foundations of Aspect Oriented Business Process Management. Tesis. Universidad de Estocolmo, 2011.
- [Kic97] KICZALES, G. et al. Aspect-Oriented Programming. En: Proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finlandia. Springer-Verlag LNCS 1241, junio de 1997.
- [Nog11] NOGUEIRA SANTOS, F. et al. Using Goals to Identify Aspects in Business Process Models. En: Proceedings of the 2011 international workshop on Early Aspects - EA '11.
- [Paw04] PAWLAK, R. y H. Younessi. On Getting Use Cases and Aspects to Work Together. En: Journal of Object Technology, vol. 3, nro. 1, enero-febrero 2004.
- [Pin11] PINCIROLI, F. Consideraciones para un proceso de desarrollo de software de calidad orientado a aspectos. Sexto Encuentro de Investigadores y Docentes de Ingeniería, EnIDI. San Rafael. Argentina, 2011.
- [Som08] SOMÉ, S. y P. Anthonyamy. An Approach for Aspect-Oriented Use Case Modeling. En: Proceedings of the 13th International Workshop on Software Architectures and Mobility, pp. 27-34, 2008.
- [Van07] VANOLI, V. y C. Marcos. Early Conflicts: Análisis y Resolución de Conflictos Tempranos. Jornadas Chilenas de Computación (JCC07). XIX Encuentro Chileno de Computación (ECC07). Iquique, noviembre de 2007.
- [Wam03] WAMPLER, D. Use Cases as Aspects – An Approach to Software Composition. © Dean Wampler, 2003.