



Enfoque Orientado a Aspectos para una Arquitectura de Línea de Productos de Software

M. K. Cortés Verdín, C. Lemus Olalde, C. Montes de Oca Vázquez
Grupo de Ingeniería de Software, Centro de Investigación en Matemáticas, A. C.
Jalisco S/N, Col. Valenciana, Guanajuato, Gto., C.P. 36240, México.

Resumen: La Arquitectura de una Línea de Productos de Software (ALPS) es el activo más importante de una Línea de Productos. Una ALPS define no sólo los atributos de calidad de la línea de productos sino que también comprende la capacidad de reutilización, la derivación del producto, y la evolución de la línea de productos. Los enfoques orientados a aspectos buscan una adecuada separación de intereses, a fin de obtener software fácil de evolucionar, mantener, comprender, personalizar y reutilizar. De seguirse un enfoque orientado a aspectos en la definición de una ALPS, los intereses se podrán identificar, separar y modelar de una manera adecuada. Este trabajo presenta la primera versión de una metodología orientada a aspectos y los resultados preliminares de su aplicación en la definición de una arquitectura de líneas de producto para un Sistema de Tutores Académico.

Palabras Clave: software, línea de producto, arquitectura, orientado a aspectos.

Abstract: The Product Line Architecture (PLA) is the most important asset of a Product Line. The PLA defines not only the product line quality attributes but also encompasses the capability of reuse, product derivation, and product line evolution. Aspect-Oriented approaches seek proper separation of concerns in order to obtain evolvable, maintainable, comprehensible, customizable and reusable software. If an Aspect-Oriented approach is followed in the definition of a PLA, a proper identification, separation and modeling of the PLA's concerns might be obtained. This paper presents the first draft of an aspect oriented methodology and the preliminary results of its application in defining a Product Line Architecture for an Academic Tutoring System.

Keywords: software, product line, architecture, aspect-oriented.

Introducción

Una Línea de Productos de Software (LPS) es un conjunto de sistemas de software intensivo que comparten un conjunto administrado de características

que satisfacen las necesidades específicas de un determinado segmento de mercado o misión, que se desarrollaron a partir de un núcleo común de activos esenciales [1]. La ingeniería de LPS intenta capitalizar la reutilización mediante el establecimiento de un marco de trabajo para planificar, desarrollar y gestionar un núcleo común de activos esenciales. Entre esos activos, la Arquitectura de la Línea de Productos de Software (ALPS) es tal vez el más valioso, ya que permite desarrollar de manera rentable los productos de la LPS, modelar recursos comunes, y apoyar en la evolución de la LPS. La ALPS también debe ser lo suficientemente genérica para apoyar en la planeación de los productos dentro de la LPS. Además de los atributos de calidad de los productos, la PLA ha de exhibir sus propios atributos de calidad. Estos atributos se definen en el Modelo de Calidad CAFE [2] que incluye variabilidad, reutilización, facilidad de evolución, facilidad de gestión y mantenimiento por mencionar algunos.

Los enfoques orientados a aspectos tratan de aplicar el principio de separación de intereses (del inglés *concerns*) con el fin de obtener el software que sea más fácil de desarrollar, mantener, comprender, personalizar, y reutilizar [3]. Esta separación de intereses se puede aplicar desde las primeras fases del desarrollo de software. Los aspectos se encuentran presentes de manera dispersa (del inglés *crosscutting*), esto es, que se encuentran presentes en varias de estas fases y/o en los elementos de la arquitectura como serían los componentes. Estos intereses son conocidos como aspectos tempranos de acuerdo a [4] y [5].

En la fase de diseño de la arquitectura, los intereses se pueden encontrar presentes en varios elementos de la arquitectura. Los enfoques actuales para el diseño de la arquitectura de un sistema de software no distinguen entre los intereses arquitectónicos convencionales y aquellos intereses de la arquitectura que se encuentran presentes en múltiples componentes o elementos arquitectónicos. Si estos intereses no son considerados en el diseño de la arquitectura, puede que se de origen a



código espagueti, comprometiendo así la calidad del sistema previamente especificada. Por lo tanto, los enfoques orientados a aspectos para diseñar una arquitectura de software deben de identificar los intereses arquitectónicos así como los correspondientes componentes afectados por dichos intereses. El objetivo de nuestra investigación es desarrollar una metodología Orientada a Aspectos para elaborar una ALPS. Esta metodología seguirá un enfoque de identificación de aspectos presentes en etapas tempranas del desarrollo de software para determinar y modelar los intereses presentes en varios componentes, que llamaremos para motivos del presente artículo como *intereses dispersos* en una ALPS. La primera versión de dicha metodología se aplicó en la definición de la arquitectura para un portafolio de productos del Sistema Institucional de Tutorías de la Universidad Veracruzana (SITUV).

El presente documento está organizado de la siguiente manera. Procedemos a presentar la primera versión de la metodología. Seguida de los productos que conforman la LPS y los resultados preliminares de haber aplicado la metodología para definir la ALPS del SITUV. Finalizamos con mencionar trabajo futuro por realizar.

Metodología Orientada a Aspectos para la Definición de una arquitectura de Líneas de Productos de Software

La metodología propuesta es precedida por un trabajo de evaluación de los métodos existentes de arquitectura para líneas de productos y de arquitecturas de software orientadas a aspectos [6]. De dicho trabajo de investigación obtuvimos que los métodos de arquitecturas para líneas de productos no consideran el concepto de aspectos, y por el otro lado, los métodos de arquitecturas de software orientados a aspectos no toman en cuenta los conceptos propios de una línea de productos como son los atributos de calidad, generalidad, similitudes y variabilidad. Esta sección describe de manera muy breve las actividades propuestas por la metodología que satisface las cualidades de una ALPS (fácil de evolucionar, genérica, elementos comunes y variables). Una explicación a detalle de cada actividad está fuera del alcance de este documento por cuestiones de espacio pero el lector puede encontrar mayor información contactando a los autores. Sin embargo, trataremos de proporcionar al

lector con las contribuciones del uso de la metodología. En la **Tabla 1** se identifican los entregables e información de entrada de cada una de las actividades de la metodología.

Tabla 1: Actividades de la Metodología Orientada a Aspectos.

Actividad	Información de entrada	Entregable
Elaborar Caso de Negocio	Requerimientos de la LP; Requerimientos de los interesados en el sistema	Lista de productos; Caso de Negocio
Aplicar Ingeniería de Dominios	Lista de Productos	Modelos de: contexto, entidad-relación, funcional, intereses, características; diccionario del dominio
Ingeniería de requerimientos orientada a intereses para LPS	Modelos de: características, de intereses, caso de uso; Lista de productos	Especificación de requerimientos funcionales y no funcionales, composición de intereses, identificación y resolución de conflictos en la composición, mapeo de intereses
Elaborar Modelo de Calidad de la LPS	Especificación de requerimientos; Lista de productos; Análisis de variabilidad y similitudes	Modelo de Calidad de la LPS



Actividad	Información de entrada	Entregable
Desarrollo de la vista de módulos de la LPS	Modelo de Calidad; Lista de Productos; Mapeo de intereses; Análisis de variabilidad y similitudes;	Se obtienen vistas de descomposición y vistas de generalizaciones de uso que incluyen el modelado de los intereses y aspectos identificados en el mapeo
Desarrollo de la vista de componentes-y-conectores	Modelo de calidad; Vista de módulos; Lista de Productos; Mapeo de intereses; Análisis de variabilidad y similitudes	Diseño de Componentes y aspectos concretos
Desarrollo de la vista de asignación	Vista de Código y Componentes	Despliegue de los componentes en las distintas unidades de ejecución

Lo que se pretende con la elaboración del caso de negocio del SITUV es identificar los productos que definirán el alcance de la línea de productos. Una vez identificada dicha lista de productos se procede a realizar las actividades de ingeniería de dominios (en esta etapa principalmente se realiza un análisis del dominio), que consiste en la elaboración de una serie de modelos de: contexto, estructura, funcional, entidad-relación, y características (FODA [9]). Este modelo de características establece una jerarquía de las mismas mediante la siguiente clasificación: características de capacidad, de ambiente de operación, de tecnología del dominio y de tecnología de implementación. Durante esta misma etapa se hace el modelado de intereses del dominio. Esto se hace principalmente siguiendo el

enfoque Cosmos [10]. La siguiente actividad consiste en ingeniería de requerimientos, considerando los intereses modelados durante el análisis de dominio, sólo que aquí se analiza su impacto en los requerimientos mediante la composición de intereses. Esta composición permite identificar y solucionar conflictos entre intereses. Otro resultado importante es la correspondencia de los intereses hacia el resto del ciclo de desarrollo, ya sea que éstos correspondan a funciones, decisiones de diseño o componentes. Esta composición de requerimientos está guiada por la variabilidad y similitud entre los productos del portafolio. Por lo que es importante también hacer un análisis de variabilidad y de similitudes en esta etapa.

El modelo de calidad de la LPS está basado en el Modelo de Calidad de LPS del proyecto CAFÉ [2]. Este modelo establece nueve atributos de calidad propios de una LPS. De esta modelo se eligieron tres atributos de calidad siguiendo un criterio de factibilidad de la LPS para SITUV: 1) facilidad de reutilización, 2) facilidad de evolución, y 3) facilidad de derivación. Se agregó el atributo de generalidad pues se considera importante para una ALPS. Además de lo anterior, durante la etapa de ingeniería de dominios y de ingeniería de requerimientos se encontraron los atributos de seguridad y rendimiento. De esta manera se elaboró un modelo de calidad con estos seis atributos de calidad. El modelo consiste principalmente en un metamodelo relacionando patrones a través de escenarios abstractos. Elementos del modelo son: atributos de calidad, métricas para cada atributo de calidad, medios para el logro de los atributos de calidad, los escenarios abstractos (cuando se elige una respuesta específica se tienen los escenarios concretos) y los patrones o estilos arquitectónicos correspondientes.

Con los resultados anteriores, se procede al diseño de la ALPS. Se consideran tres tipos de vistas principales [11]: módulos, componentes-y-conectores y asignación. Durante esta última etapa es importante la correspondencia obtenida de la actividad de ingeniería de requerimientos, donde se identifican los módulos y aspectos que deberán de desarrollarse así como las decisiones de diseño que deben estar presentes en la ALPS. Además, el modelo de calidad de la LPS es otro recurso importante pues determina los patrones arquitectónicos, estrategias y tácticas específicas para el logro de los atributos de calidad de la LPS.



Productos y requerimientos identificados en el SITUV.

Requerimientos de los productos: Del caso de negocio se obtuvieron los requerimientos del portafolio de los productos: a) registrar de sesiones de tutoría, b) elaborar programa de tutores, c) reporte de tutores en el periodo, d) elaborar reporte de seguimiento del alumno, e) apoyar Programa Integral de Desarrollo, f) seguimiento al Programa Integral, g) registro y consulta del progreso académico del alumno, h) cálculo de productividad y Programa de Estímulos al Desempeño del Personal Académico, i) tutoría a posgraduados (registrar sesión, reportes, desarrollo académico), j) interfase con el Sistema de Información de Desarrollo (SID) Académico y el Sistema Institucional de Tutores (SIT).

Además de los requerimientos de los productos se identificaron los siguientes requerimientos de la coordinación académica de la institución: k) detección de alumnos en riesgo, l) detección de materias con mayores dificultades para los alumnos, m) acceso a programas individuales de tutores y programas de coordinación, n) acceso a reportes de tutores, o) registro y acceso al perfil del tutor, p) tareas de los tutores y alumnos.

La lista de productos identificados en la línea de productos se muestra en la **Tabla 2**.

Tabla 2: Productos y Variantes Identificadas en la LP del SITUV.

Producto	Capacidades y alcance de los requerimientos
Producto 1	100% interfases (Internet); Requerimientos a → g, j (50% dentro del alcance de la LP)
Producto 2	100% interfases (opción de sincronización con SID y SIT; Requerimientos a → g, 10 (50% dentro del alcance de la LP)
Producto 1 Variante 1	Producto 1 + Requerimiento h (56% dentro del alcance de la LP)

Producto	Capacidades y alcance de los requerimientos
Producto 1 Variante 2	Producto 1 + Requerimientos h, i (63% dentro del alcance de la LP)
Producto 2 Variante 1	Producto 2 + Requerimiento h (56% dentro del alcance de la LP)
Producto 2 Variante 2	Producto 2 + Requerimientos h, i (63% dentro del alcance de la LP)
Producto 3	100% interfases (Internet) y 100% de los requerimientos dentro del alcance de la LP.

Resultados preliminares de haber aplicado metodología

La metodología se aplicó a un caso de estudio, donde el SITUV fue puesto a consideración para elaborar una arquitectura de línea de productos. Se utilizó el modelo de Kano [7] para definir el portafolio de productos con una mayor oportunidad de satisfacer las necesidades del cliente [8]. Es sumamente importante detectar y modelar los aspectos lo más temprano en el proceso de desarrollo. Es por eso que se decidió empezar con ese modelado desde la fase de ingeniería de dominios. La identificación y modelado de aspectos inicia con el modelado de intereses, y de estos intereses los que están dispersos o entrelazados son los que se modelarán e implementarán como aspectos. Sin embargo, los aspectos que pueden identificarse en etapas tempranas, tienden a diluirse conforme se avanza en el proceso. Esto motivó a que se hiciera un modelado de intereses principalmente y se dejara la identificación de aspectos hasta iniciar la etapa de diseño de la ALPS. Otra cuestión que motivó esta decisión es que conforme se avanza en el proceso de desarrollo de software, se introducen nuevos intereses y dimensiones de intereses. El modelado de intereses en la etapa de ingeniería de dominios se basó en Cosmos [10]. Fue así que a la par de que se desarrollaban las actividades propias de la



ingeniería de dominios, se realizaba la identificación y modelado de intereses.

Con respecto a la etapa de ingeniería de requerimientos, se continuó con el modelado de intereses y las actividades propias de esta etapa: identificación, análisis y especificación de requerimientos. Como se trata de una LPS también fue necesario considerar las variaciones y similitudes de la LPS. Durante esta etapa se continúa con el modelado de intereses y se procede a la composición de los requerimientos a fin de identificar y solucionar conflictos. Por ejemplo, se identificaron los intereses de rendimiento y seguridad, los cuales tienen una influencia hasta cierto punto negativa entre sí. Los conflictos se analizan y se resuelven y cada interés se lleva a función, decisión de diseño o aspecto. Esta etapa está basada en [12], comprobando de esta manera, que el modelado de intereses iniciado desde la ingeniería de dominios y continuada durante la ingeniería de requerimientos, proporciona entradas claras y útiles para la etapa de diseño de la ALPS. Actualmente,

- Se trabaja en la representación de aspectos en las vistas arquitectónicas.
- Una vez desarrollada la representación de aspectos en las vistas, ésta proporcionará la base para especificar el mecanismo de composición o entrelazado de intereses.

Una vez logrado lo anterior el trabajo futuro consiste en realizar la evaluación de la ALPS, especificar las transformaciones correspondientes si es necesario. Posteriormente desarrollar un caso de estudio adicional que permita validar los resultados obtenidos.

Referencias

- [1] P. Clements, and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison-Wesley, USA, 2001.
- [2] Anastasopoulos M., J. Bayer J., "Product Family Specific Quality Attributes", Fraunhofer Institut Experimentelles Software Engineering, IESE Report No. 042.02/E, Version 1.0, Kaiserslautern, Germany, 2002.
- [3] T. Elrad, R. E. Filman, A. Bader, "Aspect-oriented programming", *Communications of the ACM*, vol. 44, no. 10, October 2001, pp. 29-32.
- [4] E. Baniassad, P.C. Clements, J. Araujo, A. Moreira, A. Rashid, and B. Tekinerdogan, "Discovering Early Aspects", *IEEE Software*, vol. 23, no. 1, January/February 2006, pp. 61-70
- [5] Early Aspects web site: <http://www.early-aspects.net/>
- [6] K. Cortes Verdín and C.Lemus Olalde, "Assessment of Product Line Architecture and Aspect-Oriented Software Architecture Methods", 1st Workshop on Aspect-Oriented Product Line Engineering in 5th International Conference on Generative Programming and Component Engineering (GPCE'06), Portland, Oregon, USA, 2006.
- [7] N. Kano, "Attractive quality and must-be quality", *The Journal of the Japanese Society for Quality Control*, April 1984, pp. 39-48.
- [8] Karen Cortes Verdín, Lorena López Lozada, "Definition of a Software Product Line Portfolio Using the Kano Model", *SIS 07, Simposio de Ingeniería de Software 2007*. 24 y 25 de Septiembre, Morelia, México.ENC, 07.
- [9] Kang K. C., Cohen S. G., Hess J.A., Novak W.E., and Peterson A. S., "Feature-Oriented Domain Analysis (FODA) Feasibility Study", Software Engineering Institute, Carnegie Mellon University, CMU/SEI-90-TR-21, Pittsburgh, Pennsylvania, USA, 1990.
- [10] Stanley M. Sutton, Jr., Isabelle Rouvellou, "Modeling of software concerns in Cosmos", *Proceedings of the 1st international conference on Aspect-oriented software development 2002*, Enschede, The Netherlands April 2002.
- [11] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford, *Documenting Software Architectures: Views and Beyond (The SEI Series in Software Engineering)*, Addison-Wesley, USA, 2003
- [12] A. Moreira, J. Araujo, and A. Rashid, "A Concern-Oriented Requirements Engineering Model", *Proceedings of the Int'l Conference on Advanced Information Systems Engineering (CAiSE) 2005*, Porto, Portugal, 2005.

M. K. Cortés Verdín

Profesor de Tiempo Completo de la Facultad de Estadística e Informática de la Universidad Veracruzana. Licenciada en Informática de la Universidad Veracruzana en 1989. Obtuvo el grado de Master of Science in Information Systems Engineering en 1995 del Instituto de Ciencia y Tecnología de la Universidad de Manchester (UMIST), UK, el grado de Maestría en Ingeniería de Software se obtuvo en 2005 del Centro de Investigación en Matemáticas A.C. (CIMAT); México y actualmente es candidato a Doctor en el Programa de Ciencias de la Computación del CIMAT, México.

C. Lemus Olalde

El Dr. Lemus se ha desempeñado como Director de Centro de Cálculo e Informática, Gerente de Desarrollo,



Líder de Proyectos, Investigador, Instructor de cursos en Licenciatura y Maestría, Consultor y Asesor en Sistemas Computacionales. El Dr. Lemus recibió su título de Doctor en Ciencias Computacionales de la Universidad de Tulane en Nueva Orleáns, en 1996. Sus intereses de investigación incluyen reutilización de software y arquitecturas de software. Actualmente se desempeña como profesor e investigador titular en el Departamento de Ciencias de la Computación del CIMAT ubicado en la Ciudad de Guanajuato, Guanajuato.

C. Montes de Oca Vázquez

El Dr. Carlos Montes de Oca Vázquez es Investigador Titular en el Centro de Investigación en Matemáticas A.C. (CIMAT) en Guanajuato, Gto., donde hace investigación, da clases en los programas de maestría y doctorado, y hace proyectos de consultoría y vinculación con la industria. Su área de interés es la Ingeniería de Software, en particular la educación en Ingeniería de Software y las tecnologías enfocadas a producir Software con calidad, a tiempo y dentro del presupuesto. El Dr. Montes de Oca obtuvo la licenciatura en Ingeniería en Sistemas Computacionales en el ITESM, en Monterrey, la maestría en Ciencias de la Computación en la Universidad Autónoma Metropolitana, en la Ciudad de México, y el Doctorado en la Universidad Estatal de Louisiana, en los Estados Unidos de América.