



Centro de Investigación en Matemáticas, A.C.

Maestría en Ingeniería de Software



Especificación de Arquitectura de Software Basada en Dominios

Mario González Salazar

Dr. Carlos Montes De Oca Vázquez

Tabla de contenidos

1	Introducción.....	3
2	Planteamiento del problema	4
2.1	Antecedentes	4
2.2	Definición del problema	4
3	Trabajo previo.....	5
3.1	Formas alternativa de descripción de arquitectura.....	5
3.2	ADLs	5
3.2.1	ADLs para SPL.....	5
3.2.2	ADLs extensibles	5
3.3	Trabajo previo sobre dominio	5
3.4	Trabajo previo sobre otros aspectos de SPL.....	6
4	Investigación propuesta.....	7
4.1	Prepropuesta de investigación.....	7
4.2	Objetivos de la investigación	7
4.3	Justificación	7
4.4	Framework Conceptual.....	7
4.4.1	Hipótesis de investigación.....	7
5	Resumen	8
	Referencias	9

1 Introducción

El propósito de este trabajo es presentar una prepropuesta doctoral sobre la especificación de arquitecturas basándose en elementos relevantes al dominio. Tomando en cuenta que la arquitectura es el campo de diseño que más madurez y frutos le ha dado a la ingeniería de software [3] y que sea ha identificado la necesidad de una intervención directa de elementos de dominio en la definición de arquitecturas donde muchos de los lenguajes de definición de arquitectura (ADL) carecen de este enfoque [5].

2 Planteamiento del problema

2.1 Antecedentes

Desde el surgimiento del concepto de arquitectura en ingeniería de software se han explorado formas de plasmarla o describirla, una de estas formas son los ADLs, donde en su concepto inicial un ADL está compuesto por componentes, conectores y la estructura de interconexión entre estos [16]. La idea es que estos ADLs vinieran a cubrir carencias que otros lenguajes tenían para poder describir una arquitectura, por diversas razones los ADLs de primera generación no fueron bien aceptados por la industria. Esto dio pie a un análisis [5] donde como resultado se obtuvo que la descripción de arquitectura y por lo tanto los ADLs deben de considerar además de aspectos tecnológicos (el principal y en muchos casos único enfoque de muchos ADLs), aspectos del dominio de aplicación y de negocio.

2.2 Definición del problema

¿Cuáles son los elementos clave que los medios de especificación de arquitectura deben contemplar de los dominios de aplicación?

3 Trabajo previo

A continuación se presenta información sobre trabajo previo. Es importante mencionar que mucho del trabajo previo está de alguna forma relacionado con líneas de producto de software (SPL), ya que la especificación de arquitectura para este enfoque de desarrollo es la más cercana al ideal de la arquitectura según Nenad Medvidovic [5].

3.1 Formas alternativas de descripción de arquitectura

Un ADL no es la única herramienta que existe para describir la arquitectura de un sistema. Se ha utilizado desde diagramas hechos al vuelo sin un significado previamente definido, diagramas del Lenguaje Unificado de Modelado (UML) y especializaciones de este, hasta notaciones con un alto grado de formalidad [3]. Incluso procesos de ingeniería inversa para a partir de un sistema ya existente obtener su arquitectura [11]. Notaciones con distintos niveles de abstracción y distintos niveles de formalidad en cada abstracción [9].

3.2 ADLs

En un análisis y comparación de ADLs en su capacidad de describir aspectos relevantes de la arquitectura dio como resultado que hasta ese momento no existía un solo lenguaje que no tuviera alguna carencia en alguno de estos aspectos [16]. Estudios posteriores encontraron que una posible causa de la poca aceptación de los ADLs era que estaban enfocados en aspectos tecnológicos y dejaban fuera aspectos de dominio y negocio, cuya representación era importante en la arquitectura [5].

3.2.1 ADLs para SPL

Las SPL tienen como característica especial dentro de su arquitectura la necesidad de modelar partes comunes y variables. Varios ADLs han sido creados para cubrir estas características como Koala, Mae o extensiones que permiten la creación de nuevos tipos de representaciones [15]. Un enfoque ampliamente aceptado para modelar estas partes comunes y variables es el enfocado en características donde las partes comunes y variables se agrupan formando una característica para el usuario, lenguajes como ADLARS permiten tomar estas características y plasmarlas en la arquitectura [8].

3.2.2 ADLs extendibles

Dentro de los ADLs un problema común es la falta de flexibilidad, es decir el necesitar una característica única en la arquitectura y que el ADL no sea capaz de modelarla. Como respuesta a este problema surgen ADLs extendibles que agregan cierto grado de flexibilidad a la forma de modelar la arquitectura. xADL es uno de estos lenguajes extendibles que mediante el uso de esquemas permite lo que sería la creación de un ADL a la medida [13]. ALI utiliza meta-tipos para dar flexibilidad a la representación de componentes y conectores [2].

3.3 Trabajo previo sobre dominio

El estudio de dominios en software se realiza principalmente para mejorar la capacidad de reuso entre un grupo de sistemas pertenecientes a un dominio. FODA es uno de los métodos de análisis de dominio más conocido, el cual plasma en características las

partes comunes y variables de un conjunto de sistemas pertenecientes a un dominio [19]. ODM propone la creación de un proceso de análisis de dominio ajustable y configurable que permite su integración sencilla con una diversidad de tecnologías [18].

3.4 Trabajo previo sobre otros aspectos de SPL

Uno de los principales activos de las SPL es la arquitectura, la arquitectura de una SPL tiene varios aspectos relacionados con ella, aspectos como el modelo de negocio, o la administración del recurso humano pueden afectar el alcance de los productos [17]. La madurez y evolución de las líneas de productos también han sido un tema de investigación en las SPL, así como la facilidad y alternativas de adopción [14]. La evaluación de una SPL antes de ser implementada así como una vez que ha sido establecida es otro aspecto que está siendo investigado [10]. Uno de los temas distintivos y central de las SPL es la variabilidad el cual se utiliza como mecanismo para crear los distintos productos a partir de los activos base [7]. Otro tema relacionado al anterior es cómo se representa dicha variabilidad, uno de los enfoques más aceptados el de características (feature) [4], este enfoque puede llegar a un grado de casi generar los productos una vez que la variabilidad se ha modelado en un árbol de características, pero el modelo de características no está exento de problemas [1], como el hecho de que en lo conceptual puedas construir un producto con un conjunto de características pero esto no sea viable en su implementación. Es importante entender que no todos los dominios tiene potencial para adoptar un enfoque de SPL es por eso que también han surgido investigaciones en torno a cómo medir el potencial de un dominio para éstas [6].

4 Investigación propuesta

Esta sección muestra las posibilidades de investigación que aparentar ser viables con la literatura revisada hasta el momento. Dista mucho de ser una versión final de propuesta, pero ayuda a dar una idea de hacia donde puede guiarse la investigación.

4.1 Prepropuesta de investigación

La propuesta es determinar si es posible identificar elementos comunes para todos los dominios que sean relevantes en la definición de la arquitectura de un sistema de software y si es posible, clasificar estos elementos.

4.2 Objetivos de la investigación

Los objetivos que persigue la investigación son:

- Ayudar a entender cómo el dominio es relevante a la arquitectura en los sistemas de software.
- Encontrar elementos de dominio comunes, relevantes a la arquitectura y clasificarlos.
- Determinar si es posible definir una manera sistemática de evaluar cualquier dominio para encontrar información que apoye a las decisiones de diseño relevantes.
- Incluir esta manera sistemática en algún método/herramienta/lenguaje de especificación de arquitectura.

4.3 Justificación

La relevancia de la investigación reside en que de ser favorables las investigaciones brindarán las bases para acercar más la relevancia del dominio en una arquitectura sin la necesidad de crear para cada dominio un método/herramienta/lenguaje específico.

4.4 Framework Conceptual

Esta sección brindará el marco de trabajo sobre el cual está asentada la investigación, en esta versión sólo se presenta una primera versión de la posible hipótesis.

4.4.1 Hipótesis de investigación

Es posible encontrar los elementos clave que los medios de especificación de arquitectura deben contemplar de los dominios de aplicación.

5 Resumen

Este trabajo presentó una primera versión de una prepropuesta de investigación en la línea de arquitectura de software. Específicamente en la relevancia del dominio en la arquitectura. Se habló un poco de los antecedentes que hacen posible la propuesta, el trabajo relacionado alrededor de la arquitectura de software y del dominio al que esta pertenece. Por último se presentó la propuesta inicial con un conjunto de objetivos y una primera versión de la hipótesis.

Referencias

- [1] C. Kaestner, S. Apel, S.S. ur Rahman, M. Rosenmueller, and D. Batory. On the Impact of the Optional Feature Problem: Analysis and Case Studies. **Software Product Line Conference (SPLC)**, September 2009.
- [2] Rabih Bashroush, Ivor Spence, Peter Kilpatrick, T. John Brown, Wasif Gilani, Mathias Fritzsche, "ALI: An Extensible Architecture Description Language for Industrial Applications," ecbs, pp.297-304, 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008), 2008.
- [3] Richard N. Taylor , Andre van der Hoek, Software Design and Architecture The once and future focus of software engineering, 2007 Future of Software Engineering, p.226-243, May 23-25, 2007.
- [4] S. Trujillo. Feature Oriented Model Driven Product Lines . Ph.D. Dissertation, Department of Computer Sciences, The University of the Basque Country, Spain, March 2007.
- [5] Nenad Medvidovic , Eric M. Dashofy , Richard N. Taylor, Moving architectural description from under the technology lamppost, Information and Software Technology, v.49 n.1, p.12-31, January, 2007.
- [6] Mark Dalgarno, Mobile Games Development Whit Software Product Lines, Software Acumen 2006.
- [7] Bachmann, F. & Clements, P. Variability in Software Product Lines (CMU/SEI-2005-TR-012, ADA450337). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.
- [8] R. Bashroush, T.J. Brown, I. Spence, P. Kilpatrick, "ADLARS: An Architecture Description Language for Software Product Lines," sew, pp.163-173, 29th Annual IEEE/NASA Software Engineering Workshop, 2005.
- [9] María Cecilia Bastarrica, Marcelo López, Sergio F. Ochoa, Pedro O. Rossel. "Using the Integrated Notation for Defining Product Line Architecture". Proceedings de la First Conference on the Principles of Software Engineering (PRISE'04), Buenos Aires, Argentina. Noviembre 2004.
- [10] Frank van de Linden, Jan Bosch, Erik Kamsties, Kari Kansala, Henk Obbink, Proceedings of the Third Conference Software Product Line Conference (SPLC 2004), Springer Verlag LNCS 3154, pp. 110-129, September 2004.
- [11] Robyn R. Lutz , Gerald C. Gannod, Analysis of a software product line architecture: an experience report, Journal of Systems and Software, v.66 n.3, p.253-267, 15 June 2003.

- [12] Geppert, B. & Weiss, D. "Goal-Oriented Assessment of Product-Line Domains," 180-188. Proceedings of the Ninth International Software Metrics Symposium (METRICS'03). Sydney, Australia, September 3-5, 2003. Los Alamitos, CA: IEEE Computer Society, 2003.
- [13] Eric M. Dashofy, André van der Hoek, and Richard N. Taylor. An Infrastructure for the Rapid Development of XML-based Architecture Description Languages. In Proceedings of the 24th International Conference on Software Engineering (ICSE2002), Orlando, Florida.
- [14] Bosch, J. "Maturity and Evolution in Software Product Lines: Approaches, Artifacts, and Organization," 257-271. Software Product Lines: Proceedings of the Second Software Product Line Conference (SPLC2). San Diego, CA, August 19-22, 2002. Berlin, Germany: Springer, 2002.
- [15] Eric M. Dashofy , André van der Hoek, Representing Product Family Architectures in an Extensible Architecture Description Language, Revised Papers from the 4th International Workshop on Software Product-Family Engineering, p.330-341, October 03-05, 2001.
- [16] Nenad Medvidovic , Richard N. Taylor, A Classification and Comparison Framework for Software Architecture Description Languages, IEEE Transactions on Software Engineering, v.26 n.1, p.70-93, January 2000.
- [17] Bass, L.; Clements, P.; Cohen, S.; Northrop, L.; & Withey, J. Product Line Practice Workshop Report (CMU/SEI-97-TR-003, ADA327610). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.
- [18] Mark A. Simos, Organization domain modeling (ODM): formalizing the core domain modeling life cycle, ACM SIGSOFT Software Engineering Notes, v.20 n.SI, p.196-205, Aug. 1995.
- [19] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study, 1990.