

La arquitectura de software como disciplina científica

The software architecture as scientific discipline

Yamila Vigil Regalado

Universidad de las Ciencias Informáticas

yvigil@uci.cu

Erich Fouces Cabana

DESOFTE S.A

Resumen

En el presente artículo se aborda un conjunto de factores que debe cumplir un campo del conocimiento para que sea categorizado como disciplina científica. Analiza como la Arquitectura de Software (AS) logra el cumplimiento de estos factores, a través del establecimiento de una comunidad que demuestre el quehacer científico y una práctica profesional. Realiza un análisis bibliográfico desde los inicios de la disciplina, y los principales hallazgos que determinan su objeto de estudio en la actualidad. Y por último analiza el nivel de madurez de la disciplina teniendo en cuenta los expertos.

Palabras clave: Arquitectura de software, disciplina científica, estilos arquitectónicos.

Abstract

The present article deals with a set of factors that must meet a field of knowledge in order to be categorized as a scientific discipline. An analysis such as Software Architecture achieves compliance with these factors, through the establishment of a community that demonstrates the scientific work and professional practice. It makes a bibliographical analysis from the beginnings of the discipline, and the main findings that determine their object of study at the present time. And finally analyzes the level of maturity of the discipline, taking into account the experts.

Key words: *Scientific discipline, software architecture, architectural styles.*

Introducción

Para que un campo del conocimiento se catalogue como disciplina científica, debe estar asociado a este un conjunto de elementos que lo avalen. Primeramente debe contar con un cuerpo de conocimientos sustentados por un alto índice de publicaciones científicas, las cuales den muestra del quehacer de una comunidad. En segundo lugar, debe tener asociado una práctica profesional, encargada de materializar este campo de conocimientos en los distintos problemas que se presentan en la rama. Y por último debe contener un método exclusivo y único para su fundamentación.

La AS, como disciplina bien delimitada, es mucho más nueva de lo que generalmente se sospecha. Los primeros pasos se observaron en las reflexiones de 1968, de Edsger Dijkstra, acerca del establecimiento de una estructuración correcta de los sistemas de software antes de lanzarse a programar. Más tarde en la conferencia de la OTAN en 1969, P.I.Sharp, formula un conjunto de apreciaciones sorprendentes, comentando las ideas de Dijkstra, estableciendo la diferencia entre ingeniería y arquitectura de software. Hasta la década de los 90's, el término fue visto de distintas maneras, sobretodo muy ligado al diseño, se hablaba de un nivel de abstracción, sin embargo aún no estaba en su lugar los elementos de juicio que permitieran reclamar la necesidad de una disciplina y una profesión particular. (Dijkstra, 1968)

Materiales y métodos

Las fuentes bibliográficas consultada para la realización de este artículo, ha sido las publicaciones realizadas por el Instituto de Ingeniería de Software, de la Universidad de Carnegie Mellon, conocido por sus siglas en inglés (SEI), quién ha sido líder

y participe del surgimiento de la arquitectura de software. Además ha concentrado un grupo de personajes claves en la concepción de la arquitectura como disciplina científica, entre los que podemos nombrar, Mary Shaw, David Garlam y Paul Clements.

Resultados y Reflexión

Luego de un análisis bibliográfico de las fuentes, se reconoce a “*Studying Software Architecture Through Design Spaces and Rules*”, de Thomas G. Lane, como el primer libro publicado por el SEI, sobre el tema de la AS, en el año 1990. En este Lane establece una definición de AS, basándose en el concepto expuesto por Mary Shaw, en el año anterior, en su libro “*Larger Scale Systems Require Higher-Level Abstractions*” presentando en el 5to Seminario Internacional sobre la especificación y el diseño del software, y publicado por la IEEE Computer Society. Lane define la AS como:

...“Arquitectura de software es el estudio de la estructura a gran escala y el rendimiento de los sistemas de software. Aspectos importantes de la arquitectura de un sistema incluye la división de funciones entre los módulos del sistema, los medios de comunicación entre módulos, y la representación de la información compartida.” (Lane, 1990)

Se observa como la concepción de componentes (“división de funciones entre los módulos del sistema”), y de conectores (“medios de comunicación entre módulos”), van tomando su espacio dentro de la definición. En este libro, Lane brinda un concepto interesante, “espacios de diseño”, los cuales proporcionan un marco para la confección de normas que pueden ayudar a un diseñador en la selección de un arquitectura apropiada para las necesidades funcionales de un nuevo sistema. Resulta “interesante”, debido a que se observa como el rol del arquitecto de software, aún no es reconocido, esta tarea es asignada directamente al diseñador del software.

Luego de este aporte, se observa una tendencia marcada hacia la búsqueda de un modelo de estructuración de software, y se diseñan un conjunto de modelos de dominios, basados en diseños genéricos, como por ejemplo DSSA (*Domain-Specific Software Architectures*) elaborado por Mettala y Graham, en el año 1992. (Erik Mettala, 1992)

No es hasta el lanzamiento del libro “*An Introduction to Software Architecture*”, de Mary Shaw y David Garlan, en el año 1994, donde plantean, que debido al aumento del tamaño y complejidad de los productos de software, el problema principal no radica ya, en los algoritmos y las estructuras de datos, sino que se dirige a la organización de los componentes que conforman el sistema, introduciendo así, la necesidad de la creación de la AS, como disciplina científica, cuyo objeto de estudio no es más que la determinación de un conjunto de paradigmas que establezcan una organización del sistema a alto nivel, la interrelación entre los distintos componentes que lo conforman y los principios que orientan su diseño y evolución. (David Garlam, 1994)

En esta ponencia se introduce por primera vez el término “architectural styles”, y se definen una gran gama de ellos, entre los que se encuentran: Tubería y filtros, Repositorio, Arquitectura en Capas, Arquitectura basada en eventos, etc.

El tema del rol de arquitecto dentro del proceso de concepción y desarrollo del software, no se expone a la luz, sin embargo, en el último capítulo del libro, titulado: “Pasado, presente y futuro”, exponen un grupo de áreas de interés para el estudio posterior de la disciplina, dentro de las que se destaca, “lograr un mejor entendimiento del rol del arquitecto en el ciclo de vida del proceso.”

En este mismo año, 1994, el cual podría ser considerado como “el año de oro de la Arquitectura de Software”, se realiza otro acontecimiento arquitectónico relevante. Mary Shaw y David Garlan, lanzan el concepto de Lenguajes de Descripción Arquitectónica (ADLs), en su libro “*Characteristics of Higher Level Languages for Software Architecture*”. En el estudio se señala las alternativas que hasta el momento se poseen para la definición de la AS de un sistema. En primer lugar, a través de la modularización de las herramientas existentes de programación y de los módulos de conexión entre ellas y, en segundo lugar, describir sus diseños usando diagramas informales y frases idiomáticas. A partir de estas reflexiones, definen un conjunto de regularidades y propiedades específicas, que constituyeron las bases de los ADLs. (Mary Shaw, 1994)

A partir de este hallazgo la disciplina ha ido fomentándose de forma incremental, destacándose un conjunto de acontecimientos como el libro “*Coming Attractions in Software Architecture*”, de Paul Clements, en el año 1996, donde define cinco temas fundamentales en torno de los cuales se agrupa la disciplina; diseño o selección de la arquitectura, representación de la arquitectura, evaluación y análisis, etc. (Clements, 1996)

Otro suceso relevante fue el lanzamiento de la tesis de Roy Fielding, en el año 2000, en la cual presenta el modelo REST, quien establece definitivamente el tema de las tecnologías de Internet y los modelos orientados a servicios y recursos en el centro de las preocupaciones de la disciplina. En el mismo año se publica la versión definitiva de la recomendación IEEE Std 1471, que procura homogeneizar y ordenar la nomenclatura de descripción arquitectónica y homologa los estilos como un modelo fundamental de representación conceptual. (Fielding, 2000)

A continuación se refleja en una tabla las publicaciones realizadas por el SEI relacionadas con la AS, desde 1990 hasta la fecha, que refleja el establecimiento de una comunidad que conduce la investigación y el desarrollo científico en la disciplina.



Tabla 1. Publicaciones por año realizadas por el SEI acerca del la arquitectura de software.

Conclusiones

La AS como ciencia, se basa en el tratamiento de los estilos, el desarrollo de los lenguajes de descripción arquitectónicos, la formulación de metodologías y los patrones de diseño. El arquitecto de software tiene entre sus tareas fundamentales, el diseño o selección de la arquitectura, su representación, evaluación y análisis. Por lo que se puede avalar el establecimiento de la AS como disciplina científica, sin lugar a duda.

Sin embargo, a pesar de poseer ya un campo del conocimiento delimitado, la AS se encuentra, reconocidamente, en una etapa aún formativa. Sus teóricos no se encuentran todavía en condiciones de asegurar que con las herramientas, propiedades y modelos definidos se pueda realizar un software de la más alta calidad posible. Por el contrario, los mejores entre los

arquitectos consideran que su disciplina es tentativa y que se encuentra en estado de flujo. Pero aunque lo que resta por hacer es formidable, con lo que se lleva hecho ya hay un enorme repertorio de ideas, experiencias e instrumentos que ayudan a pensar de qué manera, aquí y ahora, se pueden mejorar las prácticas.

Referencias Bibliográficas

Lane, Thomas G. *Studying Software Architecture Through Design Spaces and Rules*. University Carnegie Mellon : Pittsburgh, Pennsylvania 15213, November 1990. <http://www.sei.cmu.edu/pub/documents/90.reports/pdf/tr18.90.pdf>. CMU/SEI-90-TR-18.

D. Garlan, M. Shaw. *An Introduction to Software Architecture*. January 1994. <http://www.sei.cmu.edu/publications/documents/94.reports/94.tr.021.html>. CMU/SEI-94-TR-021.

Mary Shaw, David Garlan. *Characteristics of Higher-Level Languages for Software Architecture*. Pensilvania. : Software Engineering Institute, 1994. CMU/SEI-94-TR-023.

Shaw, Mary. *Larger Scale Systems Require Higher-Level Abstractions*. s.l. : IEEE Computer Society, May 1989. págs. 143-146. Vol. Volume 14 Number 3., Proceedings of Fifth International Workshop on Software Specification. ACM SIGSOFT Software Engineering Notes.

Fielding, Roy Thomas. *Architectural Styles and the design of network-based software architectures*. University of California. Irvine : s.n., 2000. Tesis Doctoral.

Clements, Paul. *Coming attractions in Software Architecture*. Pensilvania, EE.UU. : Software Engineering Institute, University Carnegie Mellon, January, 1996. CMU/SEI-96-TR-008.

Erik Mettala, Marc H. Graham. *The Domain-Specific Software Architecture Programs*. s.l. : Special Report, June 1992. CMU/SEI-92-SR-009.