

Plan de Trabajo Final - Ingeniería de Sistemas

Herramienta Web Colaborativa de Asistencia al Diseño de Arquitecturas Guiado
por Atributos de Calidad

Alumnos

Luis Esteban Pouzo
Pablo Ignacio Marzal

Director

Dr. J. Andrés Díaz Pace

Facultad de Ciencias Exactas

Universidad Nacional del Centro de la Provincia de Buenos Aires

UNCPBA



Septiembre 2013

1 - Introducción

Actualmente muchas organizaciones demandan la construcción de sistemas de software grandes y complejos para el logro de sus objetivos. Estos sistemas requieren de la combinación de diferentes tecnologías de software y hardware para alcanzar un funcionamiento acorde con las necesidades de la organización. Lo anterior, exige a los desarrolladores de estos sistemas poner especial atención y cuidado en el diseño de la arquitectura, que dará sustento al funcionamiento de sus sistemas.

Si una arquitectura de software es deficiente en su concepto o diseño, habrá grandes probabilidades de construir un sistema que no alcanzará a cumplir el total de los requerimientos establecidos. Esto, indudablemente, generará un re-trabajo complicado o, peor aún, podrá llevar al fracaso del sistema de software cuando se encuentre en operación. De esta manera, es necesario conocer y comprender los elementos que deben atacarse al diseñar una arquitectura de software.

El desarrollo de la arquitectura de software es una de las etapas fundamentales en el desarrollo de software, pues es aquí donde los profesionales aportan todos sus conocimientos, creatividad y experiencia para crear una propuesta de solución que cumpla con los requerimientos funcionales y atributos de calidad establecidos para el sistema en desarrollo.

En este trabajo nos concentramos en la etapa de diseño de una arquitectura, basándonos en el método de diseño guiado por atributos o ADD, el cual recibe como entrada una lista de atributos de calidad que deben cumplirse, y que haciendo uso de distintas tácticas arquitectónicas produce como resultado diseños alternativos de la arquitectura enfocados a hacer cumplir los atributos pretendidos.^[1]

Existe actualmente un número reducido de trabajos en los que se hallan mapeado ADD a una aplicación o herramienta con el fin de servir como asistente en el diseño de las arquitecturas. Una de las más relevantes es la herramienta ArchE (Architecture Expert) desarrollada por el Carnegie Mellon Software Engineering Institute. La misma incorpora el conocimiento de teorías relativas a atributos de calidad y utiliza estas teorías para predecir la respuesta de una arquitectura, en términos de atributos de calidad, ante determinadas situaciones.^[2]

Los atributos de calidad se tratan en ArchE como frameworks de razonamiento, que constituyen una abstracción de los atributos de calidad de la arquitectura, los mismos implementan algoritmos que permiten evaluar su cumplimiento sobre la misma, realizando dicha evaluación por medio de los escenarios definidos. ArchE, una vez terminado el análisis, dará como resultado un conjunto de tácticas o mejoras para la arquitectura con el fin de satisfacer los atributos de calidad. Actualmente sólo incluye dos frameworks, uno relativo a performance y el otro relativo a modificabilidad.

Una de las características más importantes que posee ArchE es la de permitir reevaluar de forma global todo nuestro diseño arquitectónico en cada decisión que tomamos además de ser capaz de calcular de forma automática y con gran rapidez el nivel de cumplimiento de los distintos atributos de calidad

identificados. Sin embargo, realizando un análisis más profundo, se pueden observar ciertas limitaciones, las cuales se describen brevemente de la siguiente forma:

- Es una aplicación de escritorio que se instancia en una máquina cliente, haciendo de la misma un cliente pesado y con solo la posibilidad de ser ejecutada por un único usuario de manera simultánea.
- No es posible deshacer el resultado de aplicar una táctica.
- En ArchE no es posible acceder al interior del framework de razonamiento, analizar la base teórica y las fórmulas empleadas o incluso añadir nuevas reglas o modificar las ya existentes para adaptarlas mejor a su organización o a su experiencia.
- Desarrollar nuevos frameworks de razonamiento que implementen algoritmos para analizar y modificar arquitecturas puede resultar demasiado arduo para un arquitecto.
- Arche solo presenta una única vista gráfica para ver el estado de la arquitectura, cuya usabilidad va decayendo y se va tornando demasiado compleja a medida que se van incorporando nuevos escenarios al sistema.
- El framework de razonamiento de performance es considerado como un “anizador” sin proveer soporte para tácticas.

Otra aplicación que fue pensada para servir como asistente en el diseño de arquitecturas y en la cual se realizó un análisis es la llamada DBots. Esta herramienta es un prototipo de asistente de diseño, que proporciona una infraestructura multiagente que mapea conceptos de diseños de arquitectura predecibles a modelos de planeamiento de redes de tarea jerárquica. Estos modelos representan los planes como tareas jerárquicas que pueden ser refinadas en subplanes. En este contexto el diseño de redes predecible y la red de tareas jerárquica ayudan a identificar soluciones tácticas para los atributos de calidad y luego proporcionan de forma separada los detalles para su materialización a través de patrones arquitectónicos.^[3]

De forma general podemos decir que DesignBots ayuda al arquitecto en la búsqueda de alternativas de diseños mediante la captura de conceptos de diseño de atributos de calidad en un modelo de planeamiento jerárquico, mixto-interactivo.

El objetivo de la herramienta es el de asistir al arquitecto permitiendo que este se enfoque en las decisiones que son claves para dar forma a la arquitectura, delegando a un motor de planeamiento el trabajo de búsqueda frecuente derivado de esas decisiones.

DesignBots divide el conocimiento arquitectónico entre sus agentes (los designbots). Diferentes tipos de designbots compiten en diferentes atributos de calidad, por ejemplo están los agentes orientados a performance y agentes orientados a modificabilidad. Estos atributos de calidad fueron en sí a los cuales se les dio más importancia.

Sin embargo DesignBots presenta ciertas limitaciones las cuales puede llegar a ser parte de trabajos futuros. De forma resumida podemos describirlas como siguen:

- Aun cuando se equipen los designbots con suficientes tácticas y patrones ellos no siempre serán capaces de emular la experiencia de diseño. Para hacer frente a este problema técnico, el motor de planeamiento debe incorporar más conocimiento heurístico de expertos.
- DesignBots presenta una limitada interfaz gráfica, lo que conlleva en un considerable esfuerzo intelectual para utilizar la herramienta y comprender los resultados
- Otro punto importante es la falta de lógica de diseño acerca de las acciones que va realizando el motor de planificación por lo que, al igual que el ítem anterior, exige un notable trabajo intelectual.

Las distintas aplicaciones anteriormente mencionadas fueron las que motivaron, en gran medida, el desarrollo de este trabajo.

2 - Objetivos

El principal objetivo es el desarrollo de una herramienta para asistir al arquitecto en la toma de decisiones para problemas que se presentan al diseñar una arquitectura.

Como se mencionó en el apartado anterior, las herramientas que motivaron este trabajo se centraron en los atributos de calidad de performance y modificabilidad por lo que se pretende continuar con el mismo enfoque. De manera similar a dichas herramientas, se podrán ejecutar análisis sobre los distintos escenarios, presentando posibles cambios en la arquitectura y métricas para estos cambios, con las cuales poder comparar y decidir cuales de estos pueden ser (o no) aplicados para la obtención de un diseño óptimo según el criterio del diseñador.

El diseño de arquitecturas será de forma visual mediante la utilización de elementos web que reaccionarán y se actualizarán automáticamente con los diferentes cambios que el sistema o los usuarios vayan realizando a través del tiempo. Dichos elementos web estarán organizados en las diferentes partes o secciones con las que el usuario puede interactuar. Además otro punto que será de gran interés es el de posibilitar a los usuarios a que los mismos puedan deshacer los cambios que ellos hayan realizado sobre la arquitectura.

Adicionalmente se presenta como otro desafío el poder utilizar esta herramienta de manera compartida, y así posibilitar el trabajo en equipo, ya sea en un mismo espacio físico como puede ser una oficina, o de forma remota pudiendo realizarse dicho trabajo entre equipos o grupos de personas.

3 - Trabajo propuesto

Dado que se pretende que la herramienta pueda ser usada, en forma simultánea, por un gran número de usuarios, los cuales pueden (o no) estar trabajando sobre una misma arquitectura, se deberá utilizar un esquema en el que soporte múltiples solicitudes; las cuales serán tratadas imparcialmente (se

atiende la solicitud, se procesa y se entrega un resultado). Además, la información circulará a través de la web, por lo que la integridad y seguridad en datos y transacciones es otro punto a considerar. Esto nos lleva al empleo de una entidad que gestione y controle dichos datos y transacciones (centralización del control). La incorporación de agentes externos no debe reflejar un alto costo en la herramienta (escalabilidad). Por último, los clientes que utilicen la herramienta no deberán ser afectados por el reemplazo o actualización de componentes de la misma (fácil mantenimiento).

Teniendo en cuenta lo anteriormente mencionado se optó por la construcción de la herramienta bajo un esquema cliente-servidor web (o aplicación web), donde la aplicación cliente (un usuario utilizando un explorador de internet) solicitará recursos al servidor y mostrará los resultados obtenidos. Los recursos provistos por el servidor serán las distintas secciones que compondrán la herramienta y las acciones que los usuarios podrán ejecutar en cada una de las secciones; las cuales, en términos generales, serán:

- Administración de proyectos de arquitecturas.
- Edición de una arquitectura.
- Visualización de una arquitectura.
- Panel de análisis de una arquitectura respecto a los escenarios.

El servidor podrá ser utilizado por múltiples usuarios de manera concurrente, como así también definir grupos de usuarios, los cuales podrán trabajar sobre uno o más proyectos de arquitectura. Además, dichos usuarios, pertenecientes al mismo grupo, tendrán la posibilidad de realizar tal trabajo de manera colaborativa sobre un proyecto de arquitectura en particular y cada uno de ellos verá los cambios realizados por sus pares en tiempo real.

Las funciones de la herramienta, correspondientes al análisis, se basarán en dos de los atributos de calidad propuestos por el SEI, modificabilidad y performance.^[4] Para el primero de ellos, serán creados algoritmos de exploración que analizarán y aplicarán las modificaciones resultantes de dichos análisis, para un sub conjunto de tácticas correspondientes a tal atributo de calidad.

Del atributo performance se incluirá el análisis del mismo sobre una arquitectura haciendo uso de una aplicación externa que simulará el comportamiento temporal de sistemas de tiempo real.

4 - Algoritmos de exploración

Los distintos algoritmos que contendrá la herramienta pueden ser divididos en dos categorías. Aquellos que buscan la mejor solución y aquellos que buscan una cercana a la óptima. Ambos utilizarán un valor numérico (costo) como criterio de comparación preestablecido entre resultados sugeridos así como también entre ellos y los escenarios a evaluar.^[5] Sin embargo será el usuario quien decida considerar finalmente si utilizar o no las sugerencias de la herramienta.

Ambas categorías contendrán heurísticas para acortar el espacio de búsqueda de los algoritmos y tácticas que realizarán las transformaciones que afectarán a los parámetros del modelo de atributo de calidad subyacente; proveyéndonos de una solución arquitectónica para cumplir dicho atributo.

Aquellos algoritmos que buscan la mejor solución recorrerán un subconjunto de los elementos de la arquitectura, que serán provistos por las heurísticas, realizando las transformaciones correspondientes al atributo de calidad bajo análisis. Con cada elemento se analizará el costo de su aplicación sobre la arquitectura y cómo este afecta a los escenarios involucrados, quedándonos siempre con la arquitectura que minimice los costos de los escenarios.^[6]

Para aquellos que buscan una solución aproximada se realizarán las transformaciones sobre la arquitectura tomando el elemento de menor costo, el cual será obtenido del subconjunto que retorne las heurísticas (sin analizar cada elemento) y de esta manera obtener un solución cercana a la óptima.

Al finalizar los algoritmos habrán modificado la arquitectura, unos con un análisis más profundo, obteniendo un conjunto de escenarios de menor o igual costo.^[7]

5 - Cronograma de trabajo

A continuación se listan las distintas actividades a llevar a cabo para el desarrollo del trabajo propuesto junto con una estimación del tiempo necesario para cada una.

Actividad	Duración estimada
Relevamiento bibliográfico de ADD, atributos de calidad y tácticas	4 semanas
Análisis de tecnologías para la implementación de la herramienta	3 semanas
Implementación de algoritmos	8 semanas
Desarrollo de la herramienta	12 semanas
Testeo funcional de la herramienta y prueba con casos de estudios y realización de apuntes.	5 semanas
Redacción del informe final	En paralelo con las actividades anteriores

6 - Bibliografía

1. Wojcik, R.; Bachmann, F.; Bass, L.; Clements, P.; Merson, P.; Nord, R.; & Wood, B. “Attribute-Driven Design (ADD), Version 2.0 (CMU/SEI-2006-TR-023).” Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. (2006).

<http://www.sei.cmu.edu/publications/documents/06.reports/06tr023.html>

2. Andres Diaz-Pace, Hyunwoo Kim, Len Bass, Phil Bianco, and Felix Bachmann. “Integrating Quality-Attribute Reasoning Frameworks in the ArchE Design Assistant”. Software Engineering Institute, Carnegie Mellon University 4500 Fifth Avenue, Pittsburgh, PA-15213-2612, USA (2008)

http://www.sei.cmu.edu/library/assets/QoSA08_ArchERFI_preproceedings1.pdf

3. J. Andrés Díaz-Pace, Marcelo R. Campo. “Exploring Alternative Software Architecture Designs: A Planning Perspective”; Software Engineering Institute, Carnegie Mellon University. (2008)

4. Bass, L.; Clements, P.; & Kazman, R. “Software Architectures in Practice, 2nd ed.” Boston, MA; Addison-Wesley. (2003).

5. Recordar que la herramienta se basa en ADD que usa los escenarios como entrada de evaluación.

6. El costo de comparación depende de la transformación realizada.

7. El Arquitecto es quien realmente tiene conocimiento sobre la conveniencia de las modificaciones realizadas y los costos obtenidos. Es él quien sabe a ciencia cierta si la arquitectura resultante maximiza el diseño.