

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación



Tesis de Licenciatura en
Ciencias de la Computación

Hacia un modelo más flexible para la implementación de la auto reparación de sistemas de software basada en Arquitectura

Director: Santiago Ceria

Alumno	LU	Correo electrónico
Chiocchio, Jonathan	849/02	jchiocchio@gmail.com
Tursi, Germán Gabriel	699/02	gabrielkursi@gmail.com

Resumen

Deber incluir un resumen en Castellano e Inglés no más de 200 palabras en donde se exponga con claridad los objetivos y los alcances del trabajo.

1. Introducción

Breve referencia del trabajo mencionando la parte novedosa del trabajo.

1.1. Self Healing

1.2. Descripción de Sistemas basados en Componentes

Un problema fundamental de las arquitecturas de los sistemas basados en componentes ha sido encontrar la notación apropiada para definir dichos sistemas.

Un buen lenguaje para la descripción de arquitecturas permite generar una documentación clara sobre los componentes del sistema, que luego servirá como base a los desarrolladores, permitiendo a su vez razonar sobre las propiedades del sistema y automatizar su análisis y hasta puede llegar a utilizarse para la generación automática del sistema.

Una forma de describir dichas arquitecturas es mediante el modelado de objetos, si bien este método ha sido ampliamente aceptado y utilizado en la industria, tiene varios inconvenientes, el más importante y bloqueante es que no proveen un soporte directo para describir propiedades no funcionales, esto hace dificultoso razonar sobre propiedades críticas del sistema, como son la performance y confiabilidad. Ésta es la razón principal que ha motivado el avance de los ADLs (Architecture Description Language).[GMW99]

La descripción de arquitecturas de sistemas basada en ADLs ha avanzado muchísimo en las últimas 2 décadas, al punto de que ya permiten definir una base formal para la descripción y el análisis de los mismos.

1.3. Acme

Acme es uno de los ADLs más reconocidos y utilizados, ha sido desarrollado en la Carnegie Mellon University, más precisamente por el proyecto [Architecture Based Languages and Environments \(Able\)](#), liderado por el Dr. David Garlan.

Acme es un pilar fundamental dentro del proyecto Able, ya que todo el proyecto gira en torno a la arquitectura de software de los sistemas, y es Acme quien permite describir dichas arquitecturas, por lo tanto todos los restantes subproyectos dentro de Able utilizan Acme en menor o en mayor medida.

1.4. Rainbow

1.4.1. Introducción a Rainbow

1.4.2. Tácticas y Estrategias

1.4.3. ZNN: Testeando Rainbow

1.5. Escenarios de Atributos de Calidad y QAW

1.6. ATAM

2. Limitaciones de Rainbow

Tabla del Google Doc [expandida](#)

3. Conclusiones

Alcances de los resultados obtenidos, ventajas y desventajas, futuros trabajos, etc.

Bibliografia

- [GIO/82] Gioan A. “Regularized Minimization Under Weaker Hypotheses”, applied Mathematics Optimization, Springer Verlag, Volumen 8 numero1 - pag 59-68.1982.
- [GMW99] Garlan, D., Monroe R. T., Wile D. “Acme: Architectural Description of Component-Based Systems”.