

QuaDAI

Augusto Pecho, Felipe Moreno

May 17, 2016

Contenido

Introducción

Motivación

Retos

Planteamiento del Problema

Objetivo

Contexto del método

Definición de Multimodelo

Un Multimodelo para especificar SPLs

Un Multimodelo para especificar SPLs: Puntos de Vista (I)

Un Multimodelo para especificar SPLs: Puntos de Vista (II)

Punto de vista de Calidad

Punto de vista de Variabilidad

Punto de vista Arquitectónico

Punto de vista de Transformaciones

Relaciones entre los puntos de vista

Contribución

Definición

¿Qué es QuaDAI?

Actividades de QuaDAI

- Evaluación de la arquitectura

- Transformación de la arquitectura de software

Relaciones entre transformaciones arquitectónicas y atributos de calidad

Selección de qué transformación aplicar

- Ejemplo de aplicación QuaDAI

- Arquitectura para control de misiones

- Patrones arquitectónicos: Backup Server

- Patrones arquitectónicos: Non ACK Backup (1/2)

- Requisitos de la arquitectura

- Evaluación de la arquitectura

- Evaluación de la arquitectura modificada: Opción Backup Server

- Evaluación de la arquitectura modificada: Opción Non-ACK Backup Server

Introducción

- Las Líneas de Producto Software (SPL) han surgido como un medio para mejorar la calidad del producto y el tiempo de comercialización.
- Un SPL es un enfoque de reutilización estratégica, planificada:
 - Dos procesos: **Ingeniería de Dominio & Ingeniería de Aplicaciones.**
 - Soporte para **elementos comunes** y **variabilidad.**
- En SPL la arquitectura juega un rol dual:

- La **arquitectura de la línea del producto (PLA)** contiene un conjunto de mecanismos variados que soportan la funcionalidad y requerimientos no funcionales (NFRs) de todo el conjunto de productos.
- La **arquitectura del producto (PA)** se deriva de la PLA mediante el empleo de sus mecanismos de variación arquitectónicos incorporados.

Motivación

- Los enfoques SPL existentes suelen centrarse en la vista de la variabilidad.
- Las variantes están más allá de **añadir/eliminar monótonamente** funcionalidades a la arquitectura PL
 - ¡Las **interacciones en la estructura y el comportamiento** de un producto de software que se desarrollará pueden tener un **impacto en su calidad** haciendo de este un producto inviable!
- La calidad es un factor crucial en el desarrollo SPL
 - Un defecto en la arquitectura PL puede afectar la calidad de muchos productos.

- Calidad tanto en la fase de Ingeniería de Dominio como Ingeniería de Aplicaciones.
- La **arquitectura** de software es un medio para lograr la **calidad del producto**.
- Algunas trabajos existentes aplican el **desarrollo de software dirigido por modelos (MDD)** para obtener arquitecturas del producto.
 - Sin tomar en cuenta explícitamente los atributos de calidad.

Retos

- La **derivación** de la arquitectura y configuración del producto es un **proceso complejo y que consume tiempo** (Rabiser et al., 2011)
 - Una de las tareas más **difíciles** de este proceso es conocer los **atributos de calidad** requeridos.
- Dado un conjunto de **puntos de variación de arquitectura** (arquitectura PL)
 - ¿Cómo podemos decidir cuáles deben ser seleccionados y cuáles no?
 - ¿Cómo podemos tomar esta decisión teniendo en cuenta los requerimientos no funcionales del producto?

- Una vez derivada, la **arquitectura del producto debe ser evaluada** para garantizar que reúne los niveles de atributo de calidad específico del producto.
- Cuando los niveles de calidad de los atributos no pueden alcanzarse, deben aplicarse ciertas **transformaciones arquitectónicas** para lograr este objetivo.

Planteamiento del Problema

- Hay una **falta de métodos genéricos e integrados** que permiten la **obtención**, la **evaluación** y la **mejora** de las arquitecturas de productos en entornos de desarrollo MDD SPL
 - Han surgido muchos métodos de derivación PA, aunque la mayoría de ellos no integran adecuadamente los **atributos de calidad** en el proceso de **derivación**
 - Una vez derivada, las arquitecturas de productos son evaluadas, por lo general, mediante el uso de **métodos de evaluación** de la arquitectura de software (no específicos para el desarrollo **SPL**)

Objetivo

Mejorar las prácticas actuales mediante la definición y validación empírica de un método para la obtención, evaluación y mejora de las arquitecturas de productos en entornos de desarrollo MDD SPL.

- El método debe:
 - Ser independiente de los **lenguajes de descripción arquitectónicos** (ADL) o dominios.
 - Proveer un camino **eficiente** y **efectivo** para obtener arquitecturas de producto que cumplan con los requisitos funcionales y no funcionales (NFRs).
 - Ser percibido como **fácil de usar** y **útil** por los usuarios.

Contexto del método

- Software Engineering and Information Systems (ISSI) Research Group
- Proyecto MULTIPLE: Multimodeling Approach for Quality-Aware Software Product Lines. Financiado por el Ministerio de Ciencia e Innovación.
- Proyecto CALIMO: Integration of quality in MDD. Financiado por: Conselleria de Educación, Generalitat Valenciana
- Proyecto Quality Driven-Model Transformations. Financiado por la Universidad Politécnica de Valencia
- Proyecto TwinTIDE: Towards the Integration of Transectorial IT Design and Evaluation. Financiado por la COST action de la Unión Europea.

Definición de Multimodelo

Un Multimodelo para especificar SPLs

- Un **multimodelo** es un conjunto de modelos interrelacionados que representa diferentes puntos de vista de un sistema particular.
- Un **punto de vista (viewpoint)** es una abstracción que produce una especificación del sistema restringido a un conjunto particular de problemas.
 - En cualquier punto de vista, es posible definir un modelo (**modelo de punto de vista**) del sistema que contiene sólo los objetos que son visibles desde ese punto de vista
- Las **relaciones** pueden ser definidas entre los elementos en diferentes puntos de vista:

- **Relaciones de Impacto:** Un elemento del modelo A en un punto de vista, impacta (positiva o negativamente) en un elemento B en otro punto de vista.
- **Relaciones de Composición:** Un elemento del modelo A en un punto de vista, a ser descompuesto en elementos B, C, etc. en otros puntos de vista

Un Multimodelo para especificar SPLs: Puntos de Vista (I)

- Representar a los diferentes **puntos de vista** de un conjunto de productos que se pueden derivar de la SPL
- El multimodelo usado en este método contiene **4 puntos de vista** y las relaciones entre ellos:
 - **Variabilidad**: Contiene los elementos comunes y variaciones dentro del SPL.
 - **Arquitectónico**: Contiene la variabilidad arquitectónica de la arquitectura PL. Se puede definir el uso de diferentes estilos (por ejemplo, componente-y-conector, módulo, asignación).

- **Calidad:** Contiene los atributos de calidad y los NFRs tanto del SPL como del producto en fase de desarrollo.
- **Transformaciones:** Contiene las posibles transformaciones de la arquitectura.

Un Multimodelo para especificar SPLs: Puntos de Vista (II)

- **Ingeniería de Dominio:**

- El multimodelo representa:
 - * Impactos y limitaciones entre variaciones.
 - * Varibilidad arquitectónica.
 - * NFRs
 - * Transformaciones arquitectónicas.

- **Ingeniería de Aplicación:**

- El multimodelo representa:

- * Características seleccionadas del punto de vista de Variabilidad.
- * Elementos seleccionados del punto de vista Arquitectónico.
- * Elementos seleccionados del punto de vista de Calidad.
- * Las transformaciones arquitectónicas que afectan a los elementos seleccionados en el punto de vista de la Calidad.

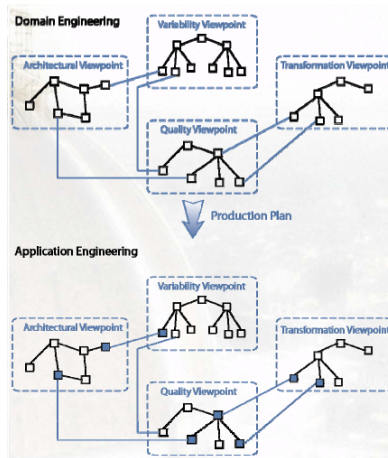


Figure 1: Ingenierías

Punto de vista de Calidad

- Representado por un **Modelo de Calidad** para SPLs donde podemos:
 - Definir las **relaciones de impacto** entre los atributos de calidad.
 - Definir los **NFRs** para el SPL y el producto específico (como restricciones sobre el modelo de calidad).
 - Seleccionar los NFRs y establecer la prioridad de los atributos de calidad de un producto dado (durante la configuración).

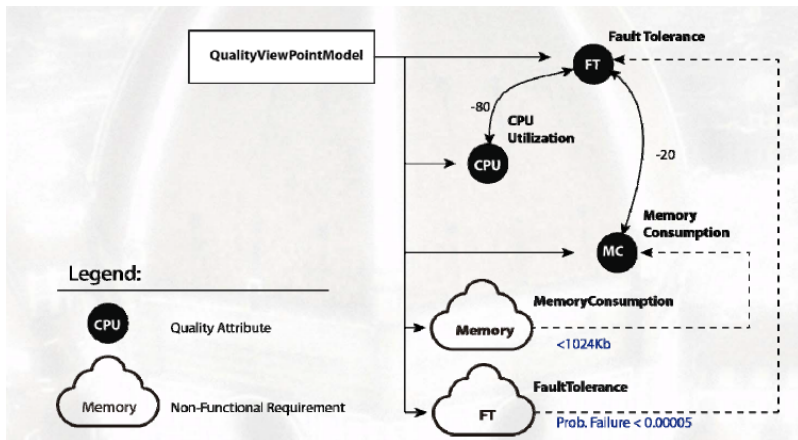


Figure 2: Modelo de Calidad

Punto de vista de Variabilidad

- Expresa los puntos comunes y la variabilidad dentro de la línea de productos.
- Su elemento principal es la **característica**, que es un aspecto visible para el usuario o característica de un sistema.
- La vista de variabilidad del multimodelo ha sido definida usando una variante del modelo basado en función de cardinalidad.

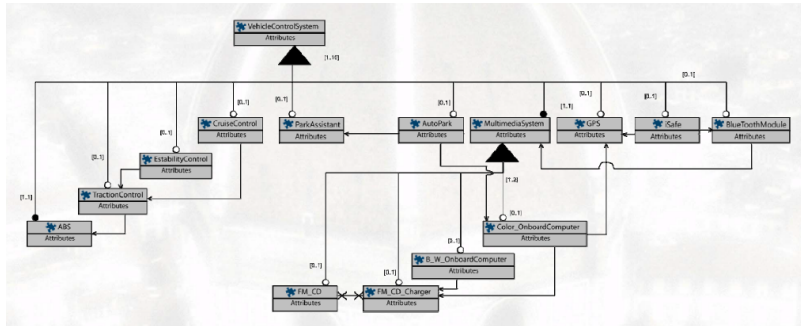


Figure 3: Modelo de Variabilidad

Punto de vista Arquitectónico

- Representa los mecanismos de variación incorporadas de la arquitectura PL independientemente del ADL o el dominio. Su expresa utilizando el lenguaje de variabilidad común (CVL)

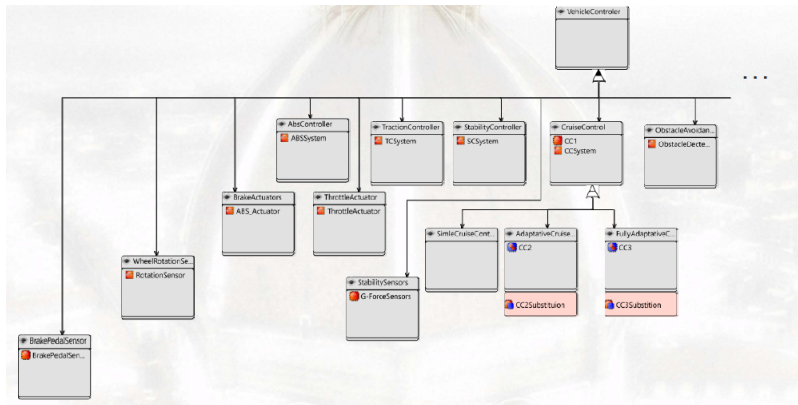


Figure 4: Modelo Arquitectónico

Punto de vista de Transformaciones

- Representa las **decisiones de diseño** en los procesos de transformación de modelos que integran el plan de producción para MDD SPL.
- Las **alternativas** aparecen en un proceso de transformación de modelos cuando un conjunto de construcciones en el modelo de origen admite **diferentes representaciones** en el modelo de destino.
 - Transformaciones alternativas podrían generar modelos alternativos de destino que pueden tener la **misma funcionalidad**, pero pueden **diferir** en los **atributos de calidad**.

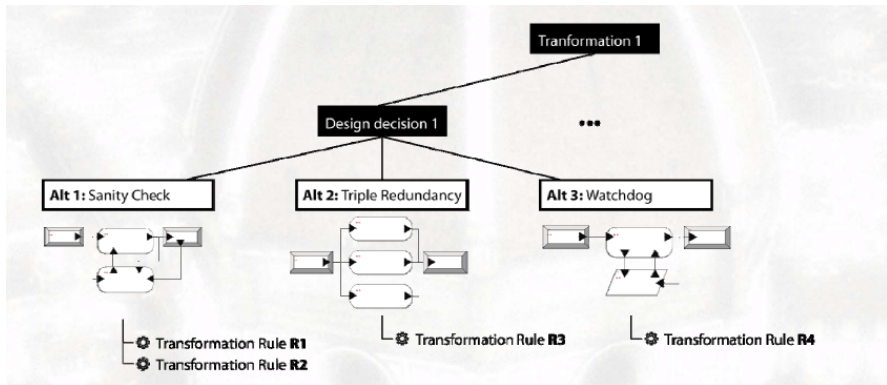


Figure 5: Modelo de Transformaciones

Relaciones entre los puntos de vista

- El multimodelo puede ser usado para definir las relaciones entre los elementos en diferentes **modelos de puntos de vista o vistas**. Esto permitirá analizar las propiedades sobre el SPL como un conjunto.
 - Se usa en las diferentes tareas que integran el proceso de derivado y evaluación.

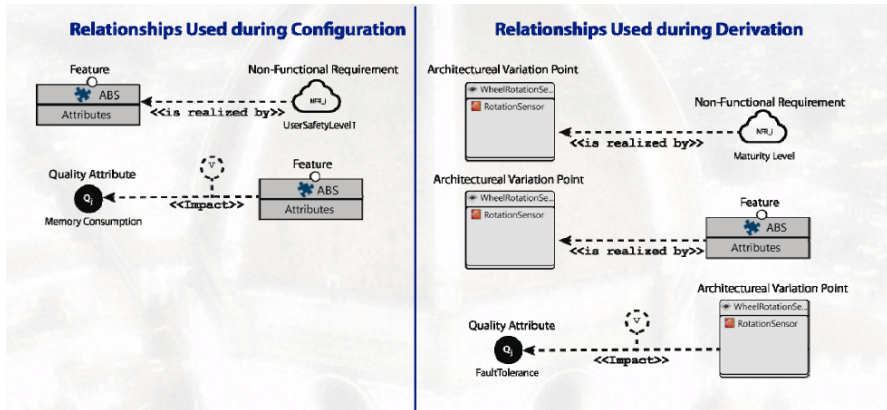


Figure 6: Relaciones entre puntos de vista

Contribución

- El multimodelo soporta diferentes puntos de vista con distintos propósitos.
 - Permite la definición de relaciones con diversa semántica, entre distintos puntos de vista.
 - Expresa el impacto que las variaciones pueden causar en la calidad del producto.
 - Conduce la derivación, evaluación y mejora de PAs mediante la integración del **punto de vista de Calidad** como un **artefacto activo** en el proceso de desarrollo de SPL.
- El **punto de vista Arquitectónico** (variabilidad arquitectónica) añade una **nueva capa de abstracción** que **desacopla** la solución del **ADL**.

- Un conjunto de directrices de diseño que ayuda al desarrollador de la transformación en la definición de las transformaciones del modelo basado en la Calidad.

Definición

¿Qué es QuaDAI?

QuaDAI es un método genérico e integrado para la derivación y la mejora de las arquitecturas. Se basa en un artefacto (el multimodelo) que representa los puntos de vista de SPL (Software Product Line) y un proceso que consiste en un conjunto de actividades llevadas a cabo por las transformaciones del modelo.

QuaDAI ha sido diseñado teniendo en cuenta los puntos débiles de los métodos de evaluación arquitectura existentes con el fin de mejorar su facilidad de uso y eficacia.

En QuaDAI, un multimodelo permite la representación explícita de las relaciones entre las entidades en distintos puntos de vista.

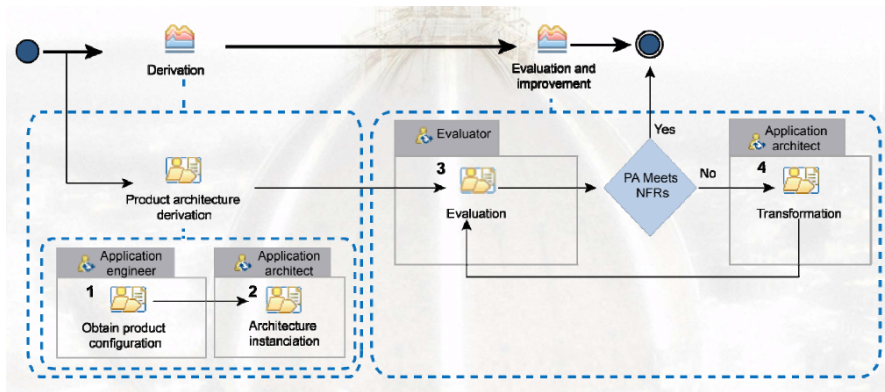


Figure 7: Proceso QuaDAI

Actividades de QuaDAI

Hay dos actividades que son llevadas a cabo por transformaciones de modelos

1. Evaluación de la arquitectura: Se aplican las métricas definidas en el modelo de calidad para comprobar si la arquitectura software cumple con los requisitos no funcionales.
2. La transformación de arquitectura software: Aplica las transformaciones arquitectónicas a una arquitectura software para alcanzar los requisitos de no funcionales

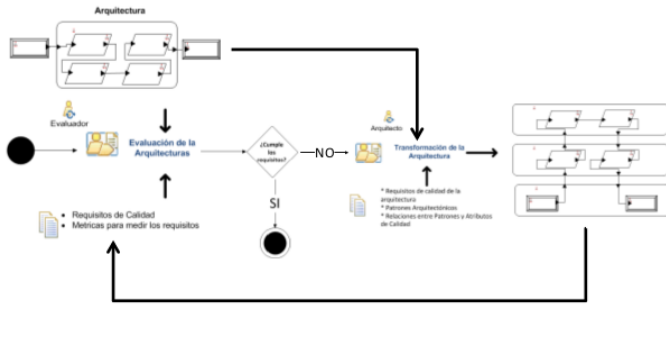


Figure 8: actividades

Evaluación de la arquitectura

- La evaluación de la arquitectura es llevada a cabo por el Evaluador.
- Toma como entradas:
 1. El/los modelo/s arquitectónico/s que representa/n la arquitectura software
 2. La vista de calidad de calidad del multimodelo donde están definidos:
 - Los requisitos no funcionales que la arquitectura ha de cumplir
 - Las métricas que miden los atributos asociados a los distintos requisitos no funcionales

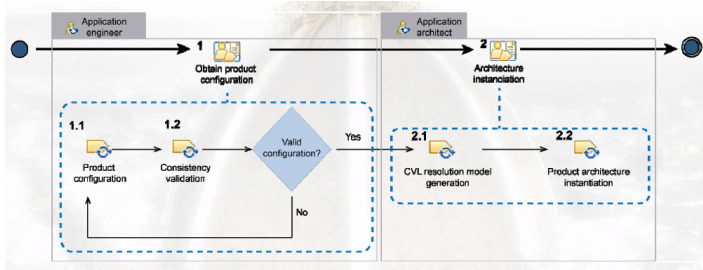


Figure 9: Derivación de Arquitectura de Producto

- Genera como salida un modelo de evaluación que contiene los valores de las métricas para los distintos requisitos no funcionales.

Transformacion de la arquitectura de software

- En el caso en que la arquitectura software no cumpla con los requisitos no funcionales entonces se aplicará la transformación de la arquitectura software.
- Es llevada a cabo por el arquitecto empleando transformaciones de modelos
- Toma como entradas:
 1. El/los modelo/s arquitectónico/s que representa/n la arquitectura software
 2. La importancia relativa de cada uno de los atributos de calidad, definidos por el arquitecto, en forma de pesos en un rango de 0 a 1

3. El multimodelo

- La vista de transformaciones donde se encuentran definidas las transformaciones arquitectónicas
- Las relaciones entre las distintas transformaciones arquitectónicas y los atributos de calidad (como una transformación arquitectónica mejora o empeora un determinado atributo de calidad)
- Genera como salida una nueva arquitectura software en la que se ha tratado de mejorar los atributos de calidad definidos por el arquitecto.

Relaciones entre transformaciones arquitectónicas y atributos de calidad

- Las relaciones entre las distintas transformaciones arquitectónicas y los atributos de calidad son definidas por el experto del dominio.
- Para ello se utiliza una técnica de Trade-Off denominada Analitical Hierarchy Process (AHP).
 1. La salida de AHP es una matriz que expresa la importancia relativa en relación con cada atributo de calidad
 2. Los valores de esa matriz son almacenados en el multimodelo y son utilizados por el proceso de transformación para decidir que transformación arquitectónica se ha de aplicar en cada caso.

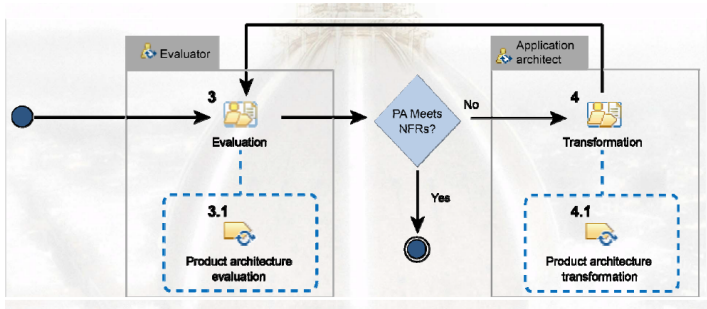


Figure 10: Evaluación y Mejora del Proceso

Selección de que transformación aplicar

- El proceso de transformación (definido en QVT) selecciona que transformación se ha de aplicar en cada caso en base a:
 1. La importancia relativa de cada uno de los K atributos de calidad (Q), definidos por el arquitecto, en forma de pesos en un rango de 0 a 1
 2. Los valores de salida del AHP expresa la importancia relativa (I) de cada transformación arquitectónica (j) en relación con cada atributo de calidad (Q)
- La decisión se toma en base a la expresión:

$$R_j = \sum_{i=0}^{k-1} Q_i * I_{ij}$$

Figure 11: ecuacion

Donde R es el ranking para cada una de las transformaciones arquitectónicas consideradas

Ejemplo de Aplicación de QuaDAI

Ejemplo de aplicación QuaDAI

- El sistema de nuestro ejemplo es un **sistema** para el **control** de los movimientos de tropas en **misiones** en el campo de batalla. Hay distintos tipos de nodos en este sistema:
 1. **Nodo Servidor (Comandante)**: es el nodo que da soporte a los encargado de tomar decisiones y de transmitir las ordenes a los nodos cliente. Además está en comunicación con otros nodos Servidor para enviar y recibir ordenes.
 2. **Nodos Cliente (Soldados)**: Realizan consultas y actualizaciones de la base de datos del servidor.

- Se dispone de un canal de comunicación cifrado
 1. Con un ancho de banda de **9600bd**.
 2. Solamente **un nodo** puede hacer broadcast en cada instante.

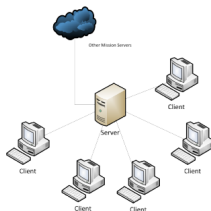


Figure 12: topologia de red Cliente-Servidor

Arquitectura para control de misiones

Patrones arquitectonicos: Backup Server

Problema:

- El sistema depende totalmente del nodo Servidor. Si el nodo servidor falla el sistema deja de funcionar.

Estructura:

- Un nodo cliente puede Promociona a Backup Server enviando una petición y recibiendo el ACK del servidor.
- El nodo Backup Server hace replica de todo el estado del servidor cada 10 minutos, monitoriza las comunicaciones (se guarda copia de todos los mensajes) y si el Server falla automáticamente promociona a Server.

Consecuencias:



Figure 13: backup1

- Aumenta la disponibilidad del sistema (el sistema pervive a la caída del servidor).
- Incrementa las comunicaciones a través del canal al tener que recibir replica del estado del servidor (55Kb).

Patrones arquitectonicos: Non ACK Backup (1/2)

Problema:

- El sistema depende totalmente del nodo Servidor. Si el nodo servidor falla el sistema deja de funcionar.

Estructura:

- Un nodo cliente promocionan a Non-ACK Backup enviando una petición al servidor.
- Un nodo Non-ACK Backup promociona enviando una petición y recibiendo el ACK del servidor.

- El nodo Non-ACK Backup solamente monitoriza las comunicaciones guardando copias de los mensajes. Si necesita promocionar a Server directamente por caída del Backup y del Server solicitará retransmisión del estado a los otros nodos cliente.
- Los nodos Backup Server hacen replica de todo el estado del servidor cada 10 minutos, monitorizan las comunicaciones guardando copia de los mensajes y si el Server falla, el nodo Backup mas antiguo automáticamente promociona a Server.

Consecuencias:

- Mejora la disponibilidad del sistema. El sistema pervive a una eventual caída del Server y del Backup Server antes de que este ultimo haya promocionado a Server.

- Mantener dos nodos Backup aumenta la carga de la red, dado que tienen que estar recibiendo copia del estado del servidor cada 10 minutos (55Kb cada copia).
- En caso de caída del servidor y de todos los Backups antes de que estos promocionen a Server hará que el nodo Non-Ack Backup tenga que solicitar al resto de nodos cliente el envío de su estado, aumentando el tiempo de promoción a Server.

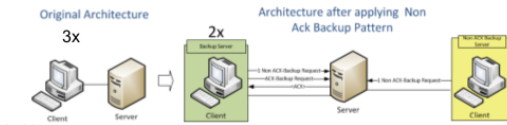


Figure 14: Arquitectura aplicando el patron Non Ack BackUp

Requisitos de la arquitectura

- NFR001 Fiabilidad: La probabilidad de fallo del sistema debe ser $<1\%$
- NFR002 Rendimiento: El canal de comunicación debe estar libre para transmitir al menos el 99% del tiempo.
- En nuestro caso nos interesará obtener la arquitectura para un sistema de control de misiones en la que se aseguren de manera simultánea ambos requisitos.

Evaluacion de la arquitectura

Analisis de la fiabilidad:

- El porcentaje de que un nodo falle, incluyendo al servidor es de un 5%.

Analisis de la prestaciones:

- El canal de comunicaciones solo se utiliza para transmisiones entre servidores y cliente.

Vamos a obtener los valores de las métricas mediante una hoja Excel que nos asiste en los cálculos.

Fiabilidad: probabilidad de fallo

- El canal de comunicaciones solo se utiliza para transmisiones entre servidores y cliente.

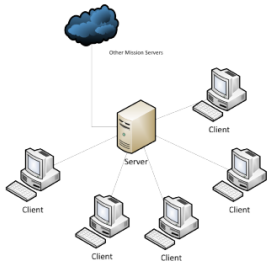


Figure 15: topologia de red Cliente-Servidor

Valor Obtenido:

5%

¿Cumple con el valor mínimo?

NO

Figure 16: fiabilidad

Valor Obtenido:	100%
¿Cumple con el valor mínimo?	SI

Figure 17: canal

Rendimiento: Porcentaje de Canal libre

- Se cuenta con un multimodelo que nos permite automatizar la selección de patrones arquitectónicos.
- Para simplificar las tareas se ha automatizado el proceso de trade off de decisión mediante una Excel que nos dará el patrón que se selecciona, las arquitecturas generadas en cada caso ya están disponibles

- Lo que tenemos que hacer en este paso es decidir la importancia relativa de los distintos atributos de calidad (La suma total ha de ser 1).

Tópicos especiales de Software

	A	B	C
1	Introduzca la importancia relativa de cada uno de los atributos de		
2	calidad en relación con los requisitos de la arquitectura		
3			
4	FIABILIDAD	1	
5	PORCENTAJE CANAL LIBRE	0	
6	SUMA TOTAL	1	
7			
8			
9			
10			
11	PATRON SELECCIONADO	NON ACK BACKUP	
12			

Figure 18: excel

Evaluación de la arquitectura modificada: Opción Backup Server

Análisis de la fiabilidad:

- La probabilidad global de fallo del sistema es la probabilidad acumulada de que fallen ambos nodos $P(\text{Backup}) * P(\text{Server})$.

Análisis de la prestaciones:

- Analizaremos el porcentaje del tiempo que estamos realizando tareas de sincronización entre servidores.
- El Backup hace una réplica cada 10 minutos (600s) y se requiere transmitir 55Kb (a 9600bps).

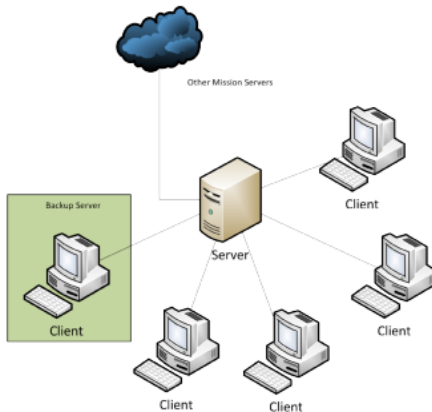


Figure 19: nube

$$\text{TraficoSincronización} = \frac{55 \text{ Kb} / 9600}{600} * 100$$

Figure 20: trafico

Vamos a obtener los valores de las métricas mediante una hoja Excel que nos asiste en los cálculos.

Valor Obtenido:	0,25%
¿Cumple con el valor mínimo?	SI

Figure 21: fia

Valor Obtenido:	99%
¿Cumple con el valor mínimo?	SI

Figure 22: canal

Fiabilidad: probabilidad de fallo

Rendimiento: porcentaje del canal libre

Evaluación de la arquitectura modificada: Opción Non-ACK Backup Server

Análisis de la fiabilidad:

- La probabilidad de fallo es la probabilidad acumulada de que fallen los cuatro nodos $P(\text{Backup})^2 * P(\text{Server}) * P(\text{Non-ACK})$.

Análisis de la prestaciones:

- Analizaremos el porcentaje del tiempo que estamos realizando tareas de sincronización entre servidores.
- El Backup hace una réplica cada 10 minutos (600s) y se requiere transmitir 55Kb (a 9600bps).

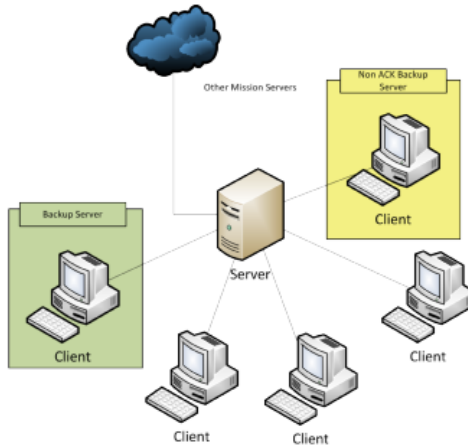


Figure 23: nube

$$\text{TraficoSincronización} = \frac{55 \text{ Kb} / 9600}{600} * 100$$

Figure 24: trafico

Vamos a obtener los valores de las métricas mediante una hoja Excel que nos asiste en los cálculos.

Valor Obtenido:	0,000625%
¿Cumple con el valor mínimo?	SI

Figure 25: fia

Valor Obtenido:	98%
¿Cumple con el valor mínimo?	NO

Figure 26: canal

Fiabilidad: probabilidad de fallo

Rendimiento: porcentaje del canal libre

FIN :D