

## Evaluando la arquitectura de software

### Parte 2. Métodos de evaluación

Por Omar S. Gómez

En el número anterior (SG Año 03 No. 01, Pág. 40) dimos a conocer como una práctica recomendada llevar a cabo evaluaciones a la arquitectura durante el ciclo de vida del desarrollo de software. Algunos de los puntos más importantes que se trataron fueron el propósito de evaluar, los momentos y los tipos de técnicas de evaluación, las personas que intervienen en este proceso así como los resultados que se obtienen tras evaluar la arquitectura.

En el presente artículo damos a conocer una serie de métodos utilizados para evaluar la arquitectura de software. Estos métodos han demostrado su efectividad en diferentes dominios de aplicación, como son: sistemas financieros, sistemas de gobierno, sistemas en tiempo real, entre otros más. A continuación se describen cuatro métodos, que analizan un atributo de calidad específico en la arquitectura de software. Al final del presente artículo se incluye una tabla en la que se comparan los cuatro métodos con respecto a una serie de elementos.

#### Architecture Level Modifiability Analysis (ALMA)

El atributo de calidad que analiza ALMA en una arquitectura de software es la facilidad de modificación. La facilidad de modificación en un sistema de software es la facilidad con la cual éste puede ser modificado a cambios en el entorno, cambios en los requerimientos o cambios a la especificación funcional. ALMA es el resultado de los trabajos de investigación realizados por Bengtsson y Lassing [1].

ALMA es un método de evaluación orientado a metas; dependiendo de la meta, este método puede ser usado para predecir el costo de mantenimiento en una arquitectura, evaluar los riesgos al haber una modificación en esta, o comparar un conjunto de arquitecturas para determinar cuál es la más apropiada en soportar cambios.

A continuación se describe en qué consiste cada una de estas metas:

- Predicción del costo de mantenimiento. Consiste en estimar el esfuerzo requerido para modificar la arquitectura a cambios requeridos en un futuro.
- Evaluación de riesgos. Identifica los tipos de cambios que pueden ser complejos o que sean inflexibles en la arquitectura de software.
- Selección de un conjunto de arquitecturas. Compara dos o más arquitecturas propuestas y selecciona aquella que proporcione mayor soporte a cambios.

ALMA puede ser utilizado una vez que concluye la especificación de la arquitectura (evaluación clásica), no obstante este método puede ser usado si la arquitectura ha sido implementada (evaluación tardía). La técnica de evaluación principal que utiliza este método es el uso de escenarios de cambio. Los escenarios de cambio son usados para capturar eventos futuros en los que se requiere que el sistema sea adaptado. Antes de iniciar la evaluación es necesario que se cuente con la especificación de la arquitectura de software, así como los requerimientos no funcionales.

Este método se compone de cinco pasos que se muestran en la Figura 1 y que a continuación se describen.

1. Definir la meta de evaluación. Dependiendo del objetivo de la evaluación se selecciona alguna de las tres metas.
2. Describir la arquitectura de software. En este punto se colecta la información de las partes más relevantes de la arquitectura como son la descomposición de ésta en componentes, las relaciones entre componentes así como las relaciones que existen en el entorno del sistema.
3. Obtener escenarios. Una vez que se cuenta con la información de la arquitectura se procede a encontrar y definir los escenarios de cambio, estos escenarios son agrupados en categorías. Dependiendo de la meta de evaluación se seleccionan los escenarios. Por ejemplo, si la meta es estimar el esfuerzo de mantenimiento, se recomienda seleccionar aquellos escenarios que corresponden a los cambios que tienen una alta probabilidad de que ocurran durante la vida operacional del sistema. Esta actividad finaliza cuando se han identificado los principales escenarios de cambio.
4. Evaluar escenarios. En este punto se realiza un análisis de impacto que consiste en identificar los componentes afectados por los escenarios previamente definidos, determinar el efecto del cambio

sobre los componentes así como determinar los efectos del cambio en otros componentes. Los resultados de este análisis se deben expresar ya sea de manera cuantitativa o cualitativa.

5. Interpretar resultados. Finalmente se interpretan los resultados así como las conclusiones del análisis dependiendo de la meta de evaluación seleccionada.

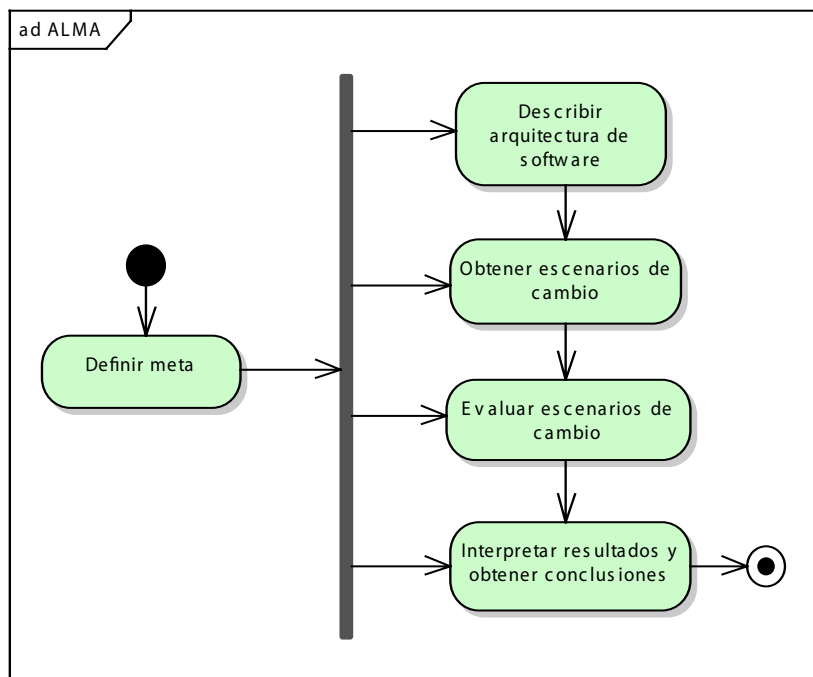


Figura 1. Método ALMA

Las personas involucradas durante la evaluación son el arquitecto de software y el equipo de desarrollo. ALMA se considera un método de evaluación maduro ya que éste ha sido validado en diferentes dominios de aplicación como son: sistemas de control embebido, sistemas médicos, telecomunicaciones y sistemas administrativos.

### Performance Assessment of Software Architecture (PASA)

PASA es el trabajo resultante de Williams y Smith [2] en el que utilizan un conjunto de técnicas para analizar el desempeño en una arquitectura de software. Entre estas técnicas se encuentran el uso de estilos arquitectónicos, anti-patrones, guías de diseño, modelos, así como técnicas empleadas en la ingeniería del desempeño de software.

El atributo de calidad que analiza PASA es el desempeño. El desempeño se interesa por conocer qué tanto tiempo le toma al sistema de software responder cuando uno o varios eventos ocurren, así como determinar el número de eventos procesados en un intervalo de tiempo dado.

PASA puede ser usado una vez que la especificación de la arquitectura ha concluido o ésta se encuentra implementada. La técnica de evaluación principal que utiliza PASA es el uso de escenarios. Estos proporcionan una medida de razonamiento con respecto al desempeño del sistema. Si se requieren análisis más detallados, los escenarios sirven como punto de partida para construir modelos de desempeño.

Para utilizar este método la arquitectura de software debe estar previamente documentada, en caso de que se encuentre parcialmente documentada es necesario extraer la información que falta de los miembros del equipo de desarrollo así como de códigos fuente y documentación existente.

Este método está compuesto de nueve pasos que se muestran en la Figura 2 y que a continuación se describen.

1. Presentar el método de evaluación. En este paso se elabora una presentación que contiene el objetivo de realizar la evaluación, en qué consiste el método PASA, qué información de la arquitectura es

necesaria para efectuar la evaluación así como los resultados de la evaluación. La presentación va dirigida a los administradores y al equipo de desarrollo.

2. Presentar la arquitectura. A continuación el equipo de desarrollo presenta la arquitectura actual de una manera general sin entrar en detalles.
3. Identificar casos de uso críticos. Se seleccionan los casos de uso que son importantes para la operación del sistema como son aquellos que demandan una respuesta de tiempo rápida para el usuario así como los que presentan algún riesgo de desempeño.

Es importante aclarar que un caso de uso puede contener uno o varios escenarios, estos describen las secuencias de acciones requeridas para ejecutar el caso de uso. Los escenarios deben ser especificados en UML como diagramas de secuencia.

4. Seleccionar los escenarios de desempeño principales. Por cada caso de uso crítico se debe identificar los escenarios que son importantes con respecto al desempeño. Por ejemplo, seleccionar aquellos escenarios que son ejecutados frecuentemente así como aquellos que son críticos desde la percepción de desempeño del usuario.
5. Identificar objetivos de desempeño. Por cada escenario de desempeño principal, se debe especificar al menos un objetivo de desempeño. Los objetivos de desempeño pueden ser expresados de distintas maneras. Por ejemplo, expresarlos en tiempo de respuesta, capacidad de respuesta, restricciones o utilización de recursos. Cabe señalar que el objetivo de desempeño debe ser cuantitativo para que este pueda ser medido.
6. Clarificar la arquitectura y discutirla. En este paso se estudia la arquitectura más a detalle por lo que se tienen reuniones con el arquitecto y miembros del equipo de desarrollo para aclarar dudas con respecto a las interacciones entre componentes si existe información acerca del desempeño del sistema se colectan métricas.
7. Analizar la arquitectura. En este paso se utilizan diferentes técnicas para analizar el desempeño de la arquitectura. Por ejemplo, se identifican estilos arquitectónicos con la finalidad de detectar efectos negativos en el desempeño, se identifican anti-patrones de desempeño que documentan problemas comunes de desempeño y se sugieren formas de cómo resolverlos, se elaboran modelos de desempeño con el objetivo de identificar problemas en la arquitectura. También se pueden utilizar técnicas de Ingeniería del desempeño de software para elaborar modelos más precisos.
8. Identificar alternativas. Si se encontraron problemas de desempeño, se identifican alternativas de solución para satisfacer los objetivos de desempeño. Estas alternativas pueden incluir el cambio de estilo arquitectónico, modificar la arquitectura para eliminar anti-patrones de desempeño, o alternativas para optimizar la interacción entre componentes.
9. Presentar resultados. Finalmente se elabora un documento con los resultados de la evaluación que incluye los objetivos de la evaluación, hallazgos encontrados, pasos específicos a seguir y recomendaciones.

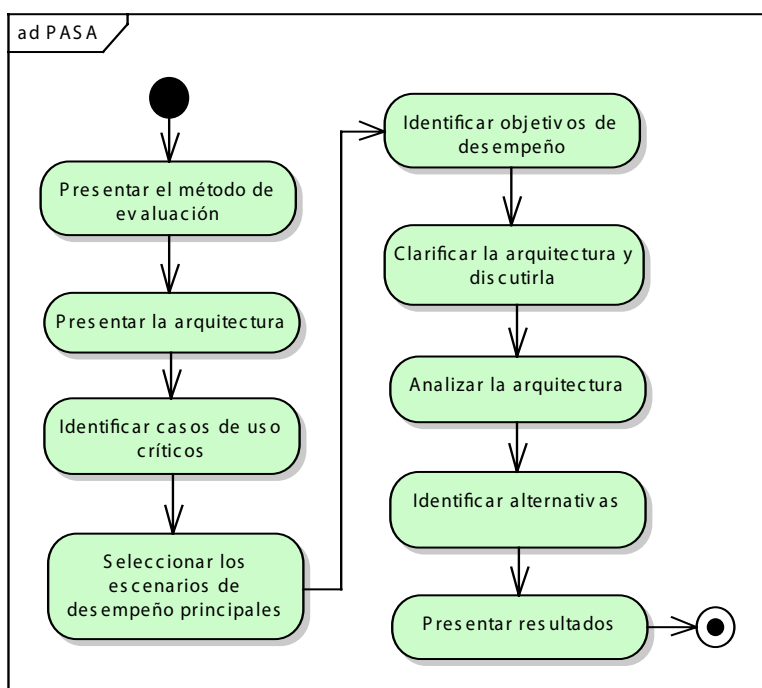


Figura 2. Método PASA

Las personas involucradas durante la evaluación son el arquitecto, el equipo de desarrollo y en algunos momentos de la evaluación los administradores responsables del proyecto. La duración de la evaluación puede ser llevada a cabo intensivamente en una semana. PASA se considera un método de evaluación maduro ya que ha sido probado en varios dominios de aplicación tales como sistemas basados en Web, aplicaciones financieras y sistemas en tiempo real.

### Scenario based Architecture Level Usability Analysis (SALUTA)

SALUTA es el primer método desarrollado para evaluar en la arquitectura de software la facilidad de uso del sistema. La facilidad de uso es la facilidad con la cual el usuario puede aprender a operar, preparar entradas e interpretar las salidas de un sistema o componente [3]. SALUTA es el resultado de los trabajos de investigación hechos por Folmer y Gorp [4].

Este método hace uso de un marco de referencia elaborado por los mismos autores en el que se expresan las relaciones que existen entre la facilidad de uso y la arquitectura de software [5]. Este marco de referencia incluye un conjunto integrado de soluciones de diseño tales como patrones de diseño y propiedades que tienen un efecto positivo sobre la facilidad de uso en un sistema de software. SALUTA analiza cuatro atributos que están directamente relacionados con la facilidad de uso en un sistema de software, estos son: facilidad de aprendizaje, eficiencia de uso, confiabilidad y satisfacción. Se recomienda efectuar la evaluación una vez que la especificación de la arquitectura ha finalizado y ésta no se ha implementado.

La técnica principal de evaluación que utiliza este método es el uso de escenarios de uso. Estos se agrupan en uno o varios perfiles de uso. En un perfil de uso se representa la facilidad de uso requerida por el sistema de software. Para efectuar la evaluación se requiere que la especificación de la arquitectura se encuentre disponible, así como el documento que incluya los requerimientos no funcionales relacionados con la facilidad de uso.

SALUTA está compuesto de cuatro pasos que se muestran en la Figura 3 y que a continuación se describen.

1. Crear perfiles de uso. En este punto se identifican los usuarios y se organizan en categorías. A

continuación se identifican las tareas más significativas del sistema, se identifica el contexto donde será usado el sistema, se determinan los valores para los atributos: facilidad de aprendizaje, eficiencia de uso, confiabilidad y satisfacción. Para reflejar la diferencia en la prioridad de los atributos, se asigna a cada uno un valor numérico no repetido entre 1 y 4. Por último se seleccionan los escenarios más representativos que tienen mayor prioridad con respecto a sus atributos.

2. Describir la facilidad de uso proporcionada. Se colecta la información de la arquitectura de software para determinar el nivel de soporte en los escenarios de uso. Esto se realiza efectuando un análisis basado en patrones o basado en propiedades. En ambos análisis se utiliza el marco de referencia para determinar la presencia de patrones o propiedades de facilidad de uso en la arquitectura a evaluar.
3. Evaluar escenarios. En este paso se evalúa cada uno de los escenarios que forman parte del perfil de uso. Por cada escenario se analizan los patrones y propiedades previamente identificados. Se recomienda usar el marco de referencia para analizar cómo un patrón o propiedad particular afecta a un atributo específico en un escenario de uso. Finalmente en este paso se expresan de manera cuantitativa los resultados del análisis, que indican el grado de soporte de facilidad de uso en cada uno de los escenarios.
4. Interpretar resultados. En este paso se obtienen los resultados de la evaluación que contienen el grado de facilidad de uso que soporta la arquitectura evaluada.

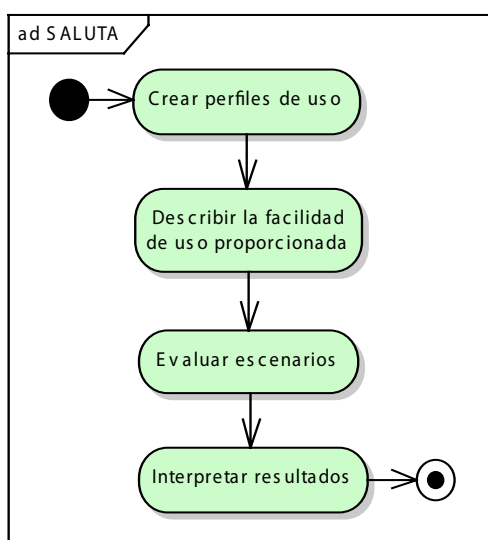


Figura 3. Método SALUTA

Las personas involucradas durante la evaluación son el arquitecto de software, ingenieros de requerimientos o ingenieros responsables por la facilidad de uso. SALUTA es un método que se encuentra en etapa de maduración ya que solamente se ha probado en algunos casos de estudio, sin embargo los resultados que se han obtenido han sido satisfactorios.

### Survivable Network Analysis (SNA)

SNA [6] es un método desarrollado por el CERT (Computer Emergency Response Team) que forma parte del SEI (Software Engineering Institute). Este método ayuda a identificar la capacidad de supervivencia en un sistema analizando su arquitectura. La supervivencia es la capacidad que tiene un sistema para completar su misión a tiempo ante la presencia de ataques, fallas o accidentes. Un ejemplo de la definición anterior es la siguiente: un cajero automático debe garantizar al usuario los servicios esenciales aun cuando este se encuentre en presencia de algún ataque externo o falla interna.

SNA utiliza tres propiedades clave para evaluar la supervivencia en un sistema. Estas son:

1. Resistencia. Es la capacidad del sistema para repeler ataques, fallas o accidentes.
2. Reconocimiento. Es la capacidad de detectar ataques, fallas o accidentes y si estos ocurren evaluar los daños.
3. Recuperación. Es la capacidad de mantener en operación los servicios esenciales en presencia de ataques, fallas o accidentes.

Este método puede ser usado en el proceso de construcción de la arquitectura (evaluación temprana),

una vez que la construcción de esta ha terminado o si la arquitectura se encuentra implementada.

La técnica de evaluación que utiliza SNA es el uso de escenarios. SNA hace uso de dos tipos de escenarios. El primer tipo son los escenarios normales de uso, éstos se componen de una serie de pasos donde los usuarios invocan a servicios y obtienen acceso a activos tales como bases de datos. El segundo tipo de escenarios son los de intrusión, en los que se representan diferentes tipos de ataques al sistema. Para llevar a cabo la evaluación, se requiere que se cuente con la especificación de la arquitectura. Si hace falta documentación de la especificación se procede a completarla.

SNA está compuesto de cuatro pasos que se muestran en la Figura 4 y que a continuación se describen.

1. Definición del sistema. En este paso se obtiene la misión del sistema, se discute el entorno de uso en términos de capacidades y ubicaciones de los usuarios, se analizan posibles riesgos que el sistema pueda presentar en condiciones adversas, finalmente se analiza la arquitectura de software.
2. Definición de las capacidades esenciales. A continuación se seleccionan los servicios esenciales y los activos que usan. Se deben de seleccionar aquellos servicios y activos que sean críticos para garantizar en condiciones adversas la misión del sistema. Una vez seleccionados, estos se trazan a través de la arquitectura para identificar los componentes que participan en proporcionar los servicios esenciales.
3. Definición de las capacidades que comprometen al sistema. En este paso se selecciona un conjunto representativo de posibles ataques basados en el entorno de operación del sistema. Se definen los escenarios de intrusión y se trazan a través de la arquitectura para identificar componentes que comprometan la supervivencia del sistema logrando así el acceso y daño a éste.
4. Analizar la supervivencia. Finalmente se identifican los escenarios de intrusión que contienen los componentes esenciales y que comprometen la supervivencia del sistema. Por cada componente identificado se procede a analizarlo en términos de las capacidades de resistencia, reconocimiento y recuperación. Estos análisis se colocan en una tabla llamada mapa de supervivencia que contiene por cada escenario de intrusión las estrategias de supervivencia actuales y las recomendadas con respecto a cada una de las capacidades.

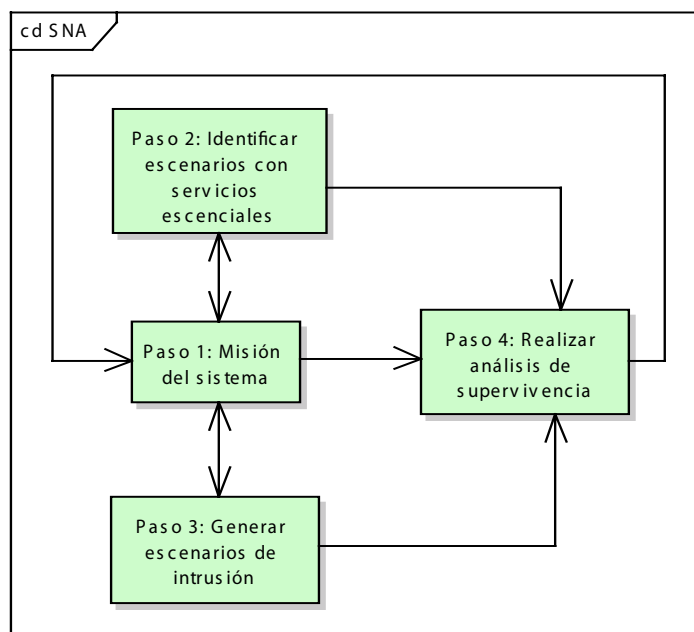


Figura 4. Método SNA

El resultado que se obtiene al final de la evaluación es un documento que incluye modificaciones recomendadas a la arquitectura acompañadas del mapa de supervivencia. Las personas involucradas durante la evaluación son el arquitecto de software, el diseñador principal, los propietarios del sistema y usuarios del mismo. SNA se considera un método maduro ya que se ha utilizado satisfactoriamente en sistemas comerciales y de gobierno. No obstante este método se mantiene en un proceso de mejoramiento continuo.

## Conclusiones

En esta segunda parte hemos presentado el método de evaluación ALMA que se interesa por predecir la facilidad de modificación en una arquitectura, PASA que analiza en la arquitectura el desempeño del sistema, SALUTA que analizando una serie de atributos de calidad predice la facilidad de uso y SNA que analiza en la arquitectura la supervivencia de un sistema ante la presencia de ataques, fallas o accidentes. Para concluir, en la Tabla 1 presentamos un cuadro comparativo entre los cuatro métodos de evaluación antes descritos.

	ALMA	PASA	SALUTA	SNA
<b>Meta</b>	Predecir el costo de mantenimiento, evaluar riesgos, comparación entre arquitecturas.	Analizar la arquitectura con respecto a los objetivos de desempeño de un sistema.	Predecir la facilidad de uso en un sistema analizando la arquitectura.	Identificar la capacidad de supervivencia en un sistema ante la presencia de ataques, fallas o accidentes.
<b>Atributo de calidad</b>	Facilidad de modificación	Desempeño	Facilidad de uso	Supervivencia
<b>Tipo de evaluación</b>	Clásica	Clásica	Clásica	Temprana / Clásica
<b>Técnica de evaluación</b>	Escenarios de cambio	Escenarios	Escenarios de uso	Escenarios normales de uso, Escenarios de intrusión.
<b>Entradas</b>	Especificación de la arquitectura, requerimientos no funcionales.	Especificación de la arquitectura.	Especificación de la arquitectura, requerimientos no funcionales relacionados con la facilidad de uso.	Especificación de la arquitectura.
<b>No. de pasos</b>	5	9	4	4
<b>Salidas</b>	Dependiendo de la meta de evaluación se generan los resultados.	Hallazgos encontrados, pasos específicos a seguir y recomendaciones.	Grado de facilidad de uso que soporta la arquitectura evaluada.	Modificaciones recomendadas a la arquitectura y mapa de supervivencia.
<b>Personas involucradas</b>	Arquitecto y equipo de desarrollo.	Arquitecto, equipo de desarrollo y administradores del proyecto.	Arquitecto, ingenieros de requerimientos ó ingenieros responsables por la facilidad de uso.	Arquitecto, diseñador principal, propietarios del sistema, usuarios.
<b>Duración</b>	No especificado	7 días	No especificado	No especificado
<b>Etapas de maduración</b>	Maduro	Maduro	En proceso de maduración	Maduro
<b>Validación del método</b>	Sistemas de control embebido, sistemas médicos, telecomunicaciones, sistemas administrativos.	Sistemas basados en Web, aplicaciones financieras y sistemas en tiempo real.	Algunos casos de estudio que incluyen principalmente sistemas Web.	Sistemas comerciales y de gobierno.

Tabla 1. Cuadro comparativo entre los métodos de evaluación ALMA, PASA, SALUTA y SNA

**Referencias**

- [1] Bengtsson, PerOlof, Nico Lassing and Jan Bosch, Vliet, Hans van. "Architecture-Level Modifiability Analysis (Alma)." *The Journal of Systems and Software* 69 (2004): 129-147.
- [2] Williams, Lloyd G. and Connie U. Smith. "PASA<sup>SM</sup>: A Method for the Performance Assessment of Software Architecture." Paper presented at the *Proceedings of the 3rd Workshop on Software Performance*, Rome, Italy 2002.
- [3] IEEE Standard Glossary of Software Engineering Terminology 610.12-1990. In IEEE Standards Software Engineering, 1999 Edition, Volume One: Customer and Terminology Standards. IEEE Press, 1999.
- [4] Folmer, Eelke, Jilles van Gorp and Jan Bosch. "Scenario-Based Assessment of Software Architecture Usability." Paper presented at the *Proceedings of Workshop on Bridging the Gaps Between Software Engineering and Human-Computer Interaction, ICSE*, Portland 2003.
- [5] Eelke Folmer, Jilles van Gorp and Jan Bosch. *Investigating the Relationship Between Usability and Software Architecture, Software process improvement and practice*: Wiley, 2003.
- [6] Mead, Nancy R., Robert J. Ellison, Richard C. Linger, Thomas Longstaff and John McHugh. "Survivable Network Analysis Method." CMU/SEI, 2000.