

Arquitectura de Sistemas de Software y Tecnologías Asociadas

Desarrollo de software basado en arquitectura

Alejandro Sánchez
asanchez@unsl.edu.ar

Departamento de Informática
Universidad Nacional de San Luis

Maestría en Informática
Universidad Nacional de San Juan
8-9 Nov 2013

Estructura de la presentación

Fundamentos de Arquitectura de Software

- 1 Attribute-Driven Design (ADD)
- 2 Scenario Based Architecture Analysis Method (SAAM)
- 3 Architecture Tradeoff Analysis Method (ATAM)
- 4 Architecture Based Design Method (ABD Method)

Estructura de la presentación

Desarrollo de software basado en arquitectura

- 1 Attribute-Driven Design (ADD)
- 2 Scenario Based Architecture Analysis Method (SAAM)
- 3 Architecture Tradeoff Analysis Method (ATAM)
- 4 Architecture Based Design Method (ABD Method)

Principales características

Proceso de diseño

- basado en req. de atributos de calidad
- descomposición recursiva de elementos según req. controladores
 - tácticas
 - patrones
- Ciclo
 - **Plan.** Selección de tipos de elementos según req. de atributos de calidad y restricciones de diseño
 - **Do.** Satisfacción de atributos de calidad y req. funcionales instanciando elementos
 - **Check.** Análisis para determinar si los req. son satisfechos

Entrada/Salida ADD

- Entradas (requerimientos priorizadas por los stakeholders)
 - Requerimientos funcionales
 - Requerimientos de atributos de calidad
 - Restricciones de diseño
- Utiliza vistas
 - módulos
 - componentes y conectores
 - asignación
- Salida (diseño) en términos de
 - Elemento de software
 - Responsabilidad
 - Role
 - Propiedad
 - Relación

Pasos

- 1 Confirmar suficiente información sobre requerimientos
- 2 Elegir un elemento para descomponer
- 3 Identificar controladores arquitectónicos candidatos
- 4 Elegir el concepto de diseño que satisface los controladores
- 5 Instanciar los elementos arquitectónicos y asignar responsabilidades
- 6 Definir las interfaces para los elementos instanciados
- 7 Verificar y refinar los requerimientos, y convertirlos en restricciones para los elementos instanciados
- 8 Iterar a partir de 2 si necesario

Paso 1

Confirmar suficiente información sobre requerimientos

- Los requerimientos deben estar priorizados por los stakeholders
- Los requerimientos de atributos de calidad deben estar expresados en formato estímulo-respuesta
 - origen del estímulo
 - estímulo
 - artefacto
 - ambiente
 - respuesta
 - valor de métrica para respuesta

Paso 2

Elegir un elemento para descomponer

Se elige según algún criterio

- conocimiento de la arquitectura
- riesgos y dificultad
- criterios del negocio
- criterios de organización

Paso 3

Identificar controladores arquitectónicos candidatos

- Priorizar los requerimientos de acuerdo a su impacto en la arquitectura
- Seleccionar controladores arquitectónicos candidatos (5 o 6)

Paso 4

Elegir el concepto de diseño que satisface los controladores

- 1 Identificar las preocupaciones de diseño en base a los controladores arquitectónicos candidatos
- 2 Para cada preocupación, identificar los patrones/tácticas que la tratan
- 3 Seleccionar los patrones/tácticas más apropiados y registrar la justificación
- 4 Combinar los patrones/tácticas en un nuevo patrón
- 5 Describir el patrón documentando las vistas necesarias
- 6 Evaluar y resolver las inconsistencias

Paso 5

Instanciar los elementos arquitectónicos y asignar responsabilidades

Responsabilidades se derivan de

- Requerimientos funcionales asociados a los controladores arquitectónicos candidatos
- Requerimientos funcionales asociados con el elemento padre

Paso 6

Definir las interfaces (provistas/requeridas)
para los elementos instanciados

- 1 Ejercitar los requerimientos funcionales de los elementos instanciados
- 2 Observar la información producida/consumida considerando distintas vistas
- 3 Documentar las interfaces para cada elemento

Paso 7

Verificar y refinar los requerimientos, y convertirlos en restricciones para los elementos instanciados

- 1 Verificar que los requerimientos del elemento padre han sido asignados a elementos hijos
- 2 Traducir responsabilidades asignadas a elementos hijos en requerimientos funcionales
- 3 Refinar requerimientos de atributos de calidad en elementos hijos

Paso 8

Iterar a partir de 2 si necesario

Se detiene cuando se pueden mostrar satisfechos
los requerimientos

Estructura de la presentación

Desarrollo de software basado en arquitectura

- 1 Attribute-Driven Design (ADD)
- 2 Scenario Based Architecture Analysis Method (SAAM)**
- 3 Architecture Tradeoff Analysis Method (ATAM)
- 4 Architecture Based Design Method (ABD Method)

Objetivos

Scenario-based architecture analysis method (SAAM)

- Verificar que la arquitectura satisface los requerimientos de atributos de calidad
 - modificabilidad (SAAM)
 - flexibilidad (SAAMCS)
 - reuso de conocimiento del dominio (ESAAMI)
 - evolución y reuso (SAMMER)
- Evaluar riesgos
- Enfocar puntos problemáticos
- Comparar arquitecturas

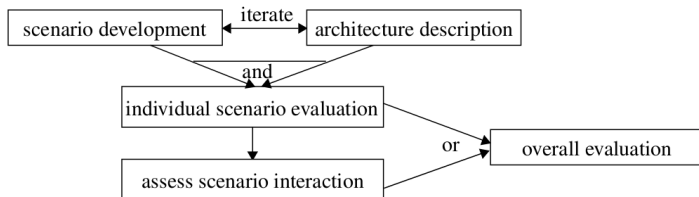
Basado en escenarios

Los **atributos de calidad** (modificabilidad, portabilidad, seguridad) resultan difíciles de medir (magnitud escalar)

Solución: utilizar escenarios

- **Escenarios de uso:**
usos concretos del sistema (instancias de casos de uso)
- **Escenarios de calidad (de cambio):**
cambios esperados en el sistema

Actividades

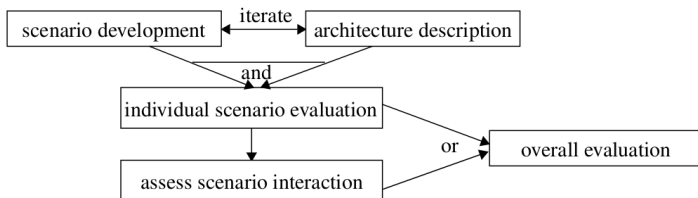


Kazman et al, 1996

Desarrollo de escenarios.

- Entrada
 - descripción de arquitectura
 - declaración de requerimientos (funcionales y no funcionales)
- Desarrollo de escenarios (todos los stakeholders)
 - escenarios de uso
 - escenarios de calidad (cambio)

Actividades

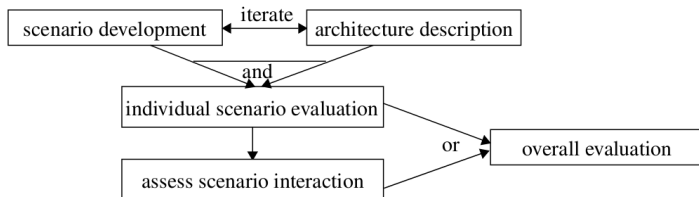


Kazman et al, 1996

Descripción de arquitectura.

- Notación comprendida por todos los stakeholders
- Vistas
 - Lógica (funcionalidad)
 - Componentes y conectores (estática y dinámica)
 - Asignación

Actividades



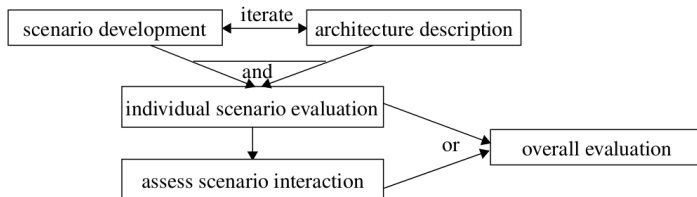
Kazman et al, 1996

Evaluación individual de escenarios.

- Evaluar que elementos se ven afectados por cada escenario
 - **directo**: no requiere modificación
 - **indirecto**: requiere modificación
- Se establece un costo asociado

Un elemento con alta interacción
puede indicar diseño pobre

Actividades

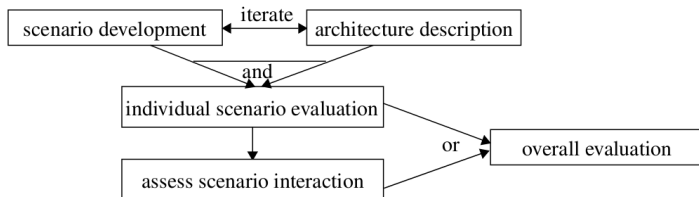


Kazman et al, 1996

Evaluación de interacción de escenarios.

Los escenarios interactúan si afectan (al menos)
a un elemento en común.

Actividades



Kazman et al, 1996

Evaluación global.

En caso de comparar múltiples arquitecturas

- se asigna un peso a cada escenario de acuerdo a su importancia relativa
- se aplica el peso para obtener una evaluación global

Se comparan las arquitecturas

Estructura de la presentación

Desarrollo de software basado en arquitectura

- 1 Attribute-Driven Design (ADD)
- 2 Scenario Based Architecture Analysis Method (SAAM)
- 3 Architecture Tradeoff Analysis Method (ATAM)**
- 4 Architecture Based Design Method (ABD Method)

Objetivos

Architecture Tradeoff Analysis Method (ATAM)

- Considerar múltiples atributos de calidad
 - modificabilidad
 - disponibilidad
 - seguridad
 - performance
 - ...
- Localizar y analizar puntos de tradeoff (alto riesgo)
- Asegurar que las preguntas correctas si hacen lo antes posible

Características principales

- Basado en escenarios (uso y calidad) y experiencia con SAAM
- Sigue principio de indivisibilidad entre análisis y diseño
- Utiliza modelos de atributos
 - Manifestaciones medibles/observables de un atributo
 - Estímulos relevantes para los cuales la arquitectura debe responder
 - Características de la arquitectura que afectan las manifestaciones
- Utiliza ABAS (Attribute-base architectural styles)
Estilo arquitectónico + marco de razonamiento

(Marco razonamiento se basa en modelos específicos de atributos)

Pasos

- 1 Planeamiento / intercambio de información
- 2 Brainstorming de escenarios
- 3 Presentación de arquitectura
- 4 Chequeo de cobertura de escenarios
- 5 Asignar prioridad y grupo a escenarios
- 6 Mapeo de escenarios de alta prioridad sobre la arquitectura
- 7 Ejecución de análisis específico de atributos
- 8 Identificar puntos de tradeoff
- 9 Consolidar resultados y desarrollar un plan de acción

Paso 1

Planeamiento / intercambio de información

- Describir método a los stakeholders
(que esperar del método y que se espera de ellos)
- Registrar los objetivos principales de calidad de los stakeholders
- Presentar arquitectura (inicial)
- Presentar escenarios (inicial)

Paso 2

Brainstorming de escenarios

- Discutir escenarios de
 - uso
 - cambios
 - fallas
- Agregar los escenarios de calidad
- Establecer necesidad de mejorar el conocimiento de la arquitectura

Paso 3

Presentación de arquitectura

- Presentar arquitectura en detalle
- Mapear los escenarios de uso más importantes a la arquitectura
- Identificar y explorar ramificaciones de estilos arquitectónicos

Paso 4

Chequeo de cobertura de escenarios

Utilizar marco de razonamiento de ABAS para realizar preguntas especificas de atributos de calidad (especial cuidado con condiciones límite)

Paso 5

Asignación de prioridad y grupo a escenarios

- Votar (stakeholders) escenarios más importantes
- Agrupar escenarios
- Determinar punto de corte (10-15 escenarios, flexibilidad)

Paso 6

Mapeo de escenarios de alta prioridad sobre la arquitectura

Mostrar para los escenarios de alta prioridad

- como afecta la arquitectura (modificabilidad)
- como la arquitectura responde (performance, seguridad, disponibilidad)

Paso 7

Ejecución de análisis específico de atributos

- Construir modelos para cada atributo de calidad
- Detectar puntos sensibles manipulando los parámetros del modelo (alta correlación entre parámetro y atributo de calidad medible)
- Se registran alternativas arquitectónicas para un plan de acción si la arquitectura se evalúa inadecuada

Disponibilidad, Performance
Cantidades de elementos redundantes activos y pasivos

Paso 8

Identificar puntos de tradeoff

Elementos con múltiples puntos sensibles

Disponibilidad, Performance

Cantidades de elementos redundantes activos y pasivos

Paso 9

Consolidar resultados y desarrollar un plan de acción

- Pedir más información de soporte
- Establecer un conjunto de recomendaciones (plan) para mejorar la arquitectura

Estructura de la presentación

Desarrollo de software basado en arquitectura

- 1 Attribute-Driven Design (ADD)
- 2 Scenario Based Architecture Analysis Method (SAAM)
- 3 Architecture Tradeoff Analysis Method (ATAM)
- 4 **Architecture Based Design Method (ABD Method)**

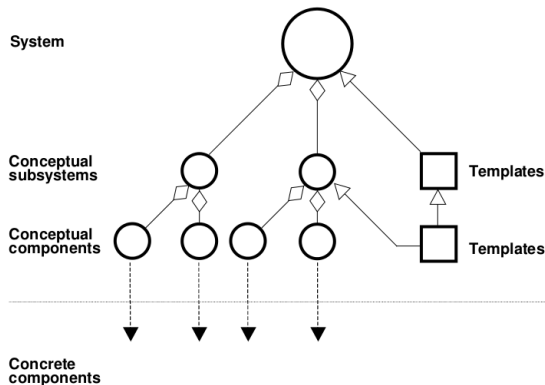
Objetivo

Diseñar la arquitectura de una linea de productos de software

Principales características

- Depende de determinar los conductores arquitectónicos
 - requerimientos de negocio
 - requerimientos de atributos de calidad
 - requerimientos funcionales
- Se fundamenta en
 - efectuar descomposición funcional
 - aplicar estilos arquitectónicos (requerimientos de calidad y de negocio)
 - usar plantillas de software (que significa ser un tipo de elemento)
- Tiene naturaleza recursiva (descomposición jerárquica)
- Utiliza vistas lógicas, de concurrencia, y de despliegue

Términos – Elementos de diseño



Bachmann et al., 2000

Elemento de diseño

- implementa responsabilidades
- tiene interfaz conceptual

Componente concreto tiene asignados:

- clase
- proceso
- thread

Términos – Requerimientos

- Funcionalidad

- Abstractos : requerimientos funcionales
- Concretos : casos de uso

- Calidad

- Abstractos: requerimientos de atributos de calidad
- Concretos: escenarios de calidad
 - escenarios de cambio
 - escenarios de confiabilidad
 - escenarios de performance
 - escenarios de interoperabilidad

Características variables y comunes

- ABD Method se preocupa de las variaciones de granularidad que impactan en la arquitectura conceptual
- Capturar y representar esta fuera del alcance de ABD Method
- Las características comunes son los puntos fijos en la línea de productos
- Variación
 - Función - la variación se logra por elecciones estructurales
 - Ambiente o Plataforma - la variación se trata como un requerimiento de calidad

Plantillas de software

- Enumeran las responsabilidades de los elementos de diseño
 - Patrones de interacción con servicios
 - Patrones de interacción con la infraestructura
 - Responsabilidades
- Propósitos
 - Reutilización de sus componentes (patrones de comportamiento)
 - Ayuda en la integración
 - Base para construir el esqueleto del sistema

El esqueleto es una implementación de la infraestructura del sistema sin funcionalidad de aplicación.

- agregar incrementalmente la funcionalidad
- materializa el soporte necesario para las variaciones en funcionalidad

Drivers (controladores) arquitectónicos

Los drivers arquitectónicos dan forma a la arquitectura

Requerimientos

- Funcionales
no necesitan ser detallados
para comenzar las actividades de diseño
- Calidad
performance, seguridad, interoperabilidad, ...
- Negocio
 - objetivos del negocio
 - background de la organización

ABD Method en pocas palabras

- Entrada

- Requerimientos funcionales (abstractos y concretos)
- Requerimientos de calidad y del negocio (abstractos y concretos)
- Restricciones

- Proceso recursivo

- Elegir estilo arquitectónico de acuerdo a los requerimientos de calidad y del negocio
- Asignar responsabilidades (requerimientos funcionales) a los elementos componentes
- Estudiar vistas (lógica, de concurrencia, y de despliegue) y asignar nuevas responsabilidades correspondientemente
- Actualizar plantillas de software
 - Agregar para los nuevos tipos de elemento
 - Determinar donde se implementaran los items de la plantilla
- Los requerimientos concretos se usan para verificar las decisiones tomadas utilizando los abstractos