Arquitectura de Sistemas de Software y Tecnologías Asociadas
Fundamentos de Arquitectura de Software

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. Arquitectura de software
2. Ingeniería de software basada en arquitecturas
3. Organización de las decisiones de diseño
4. Lenguajes de descripción arquitectural
5. Líneas de productos de software

Estructura de la presentación

Fundamentos de Arquitectura de Software

Contexto histórico

Programación/implementación vs. diseño de software (’70)

Diseño de software vs. arquitectura de software (’80 - ’90)

Revoluciona la ingeniería del software
ciclo de vida del software
estudio de factibilidad - mantenimiento

Múltiples definiciones de arquitectura de software

Sin consenso en la definición de arquitectura de software

Más de 150 definiciones en SEI-CMU site (y contando)
Gran diversidad y falta de madurez en la disciplina

Perry and Wolf. 1992
Garlan and Perry. 1995
IEEE 1471. 2000
Taylor, Medvidovic, and Dashofy. 2009
...

Foco en restricciones

Software Architecture = { Elements, Form, Rationale }

Foundations for the study of software architecture
Perry and Wolf, 1992

Énfasis en estructura interna

The structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.

Garlan and Perry, 1995

Énfasis en estructuras interna y externa

The fundamental organization of a system
embodied in its components,
their relations to each other, and to the environment,
and the principles guiding its design and evolution.

IEEE 1471 2000

Foco en distinguir diseño relevante

The set of principal design decisions governing a system.

Software Architecture: Foundations, Theory and Practice
Taylor, Medvidovic, and Dashofy. 2009

Foco en distinguir diseño relevante

The set of principal design decisions governing a system.

Software Architecture: Foundations, Theory and Practice
Taylor, Medvidovic, and Dashofy. 2009

¿Qué decisiones de diseño de software son arquitectónicas?

Decisiones de diseño de software arquitectónicas

Raison d'être

El propósito primario de la arquitectura de software es asegurar:

- objetivos de negocio
- atributos de calidad
- requerimientos de comportamiento

Documenting Software Architectures
Clements et al. 2011

Una taxonomía de vistas

El modelo de una arquitectura de software presenta vistas interrelacionadas

Tipo de abstracción

- Estática
- Dinámicas
- Estocástica
- ...

Dos enfoques relevantes

- El modelo de vistas 4+1
- Views and beyond

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
- Vista de desarrollo (estática)
- Vista de física (estática)
- Escenarios (dinámica)

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
  Conceptos claves del dominio del problema
  - soporta los requerimientos funcionales
- Vista de procesos (estática)
- Vista de desarrollo (estática)
- Vista de física (estática)
- Escenarios (dinámica)

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
  Programas que ejecutan independientemente e interactúan
  - estudio de concurrencia y distribución
  - soporta requerimientos no funcionales (performance, tolerancia a fallas, disponibilidad)
- Vista de desarrollo (estática)
- Vista de física (estática)
- Escenarios (dinámica)

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
- Vista de desarrollo (estática)
  Módulos en el ambiente de desarrollo de software
  - asignar requerimientos y trabajo
  - evaluar costos y planificar
  - analizar reuso, portabilidad, seguridad
  - establecer una linea de productos
- Vista de física (estática)
- Escenarios (dinámica)

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
- Vista de desarrollo (estática)
- Vista de física (estática)
  Red de nodos en donde se ejecuta el software
  - soportar requerimientos no-funcionales
- Escenarios (dinámica)

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
- Vista de desarrollo (estática)
- Vista de física (estática)
- Escenarios (dinámica)
  Interacciones de instancias de casos de uso
  - abstraer los requerimientos más importantes
  - descubrir elementos arquitectónicos
  - validar e ilustrar

El modelo de vistas arquitectónicas 4+1 – Vistas

The 4+1 View Model of Architecture
P.B. Kruchten. 1995

- Vista lógica (estática)
- Vista de procesos (estática)
- Vista de desarrollo (estática)
- Vista de física (estática)
- Escenarios (dinámica)

¿Cómo se relacionan las vistas?

El modelo de vistas arquitectónicas 4+1 – Relaciones

- Vista lógica $\rightarrow$ vista procesos
  Conceptos claves en procesos

El modelo de vistas arquitectónicas 4+1 – Relaciones

- Vista lógica $\rightarrow$ vista procesos
  Conceptos claves en procesos
- Vista lógica $\rightarrow$ vista desarrollo
  Conceptos claves a módulos de desarrollo/subsistemas.

El modelo de vistas arquitectónicas 4+1 – Relaciones

- Vista lógica $\rightarrow$ vista procesos
  Conceptos claves en procesos
- Vista lógica $\rightarrow$ vista desarrollo
  Conceptos claves a módulos de desarrollo/subsistemas.
- Vista procesos $\rightarrow$ vista física
  Procesos en hardware disponible
  Asumen distintas configuraciones,
  por ej. desarrollo, testing, producción

Views and Beyond

Documenting Software Architectures
Clements et al. 2011

- View.
  - principio fundamental para organizar la documentación
  - vistas diferentes exponen diferentes atributos de calidad
  - vistas diferentes suportan objetivo y stakeholders diferentes
- Beyond. información adicional arquitectónica

Views and Beyond

Documenting Software Architectures
Clements et al. 2011

- View.
  - principio fundamental para organizar la documentación
  - vistas diferentes exponen diferentes atributos de calidad
  - vistas diferentes suportan objetivo y stakeholders diferentes
- Beyond. información adicional arquitectónica

"La filosofía es que un documento arquitectónico debe ser útil a las personas que dependen de ella para hacer su trabajo"

Views and Beyond

Documenting Software Architectures
Clements et al. 2011

- View.
  - principio fundamental para organizar la documentación
  - vistas diferentes exponen diferentes atributos de calidad
  - vistas diferentes suportan objetivo y stakeholders diferentes
- Beyond. información adicional arquitectónica

"Documentar una arquitectura es una cuestión de documentar las vistas relevantes y luego agregar documentación que aplica a mas de una vista."

Views and Beyond - Vistas

- Vista de módulos
- Vista de componentes y conectores
- Vista de asignación

Views and Beyond - Módulos

Muestra las principales unidades de implementación
y las relaciones entre estas unidades.

- Elementos: módulos
- Relaciones:
  - es parte de
  - depende de
  - es una

- base para construir y analizar
- herramienta de comunicación entre stakeholders

Views and Beyond - Módulos

Muestra las principales unidades de implementación
y las relaciones entre estas unidades.

- Elementos: módulos
- Relaciones:
  - es parte de
  - depende de
  - es una

- base para construir y analizar
- herramienta de comunicación entre stakeholders

¿Con qué vista del modelo 4+1 se superpone?

Views and Beyond - Componentes y conectores

Muestra elementos con presencia en tiempo de ejecución: componentes (proceso, objetos, repositorios de datos, ...), y conectores (protocolos de interacción, flujo de información, ...).

- Elementos:
  - components (ports)
  - connectors (roles)
- Relaciones:
  - attachments
  - interface delegation

Análisis de propiedades:

- performance
- confiabilidad
- seguridad
- ...

Views and Beyond - Componentes y conectores

Muestra elementos con presencia en tiempo de ejecución: componentes (proceso, objetos, repositorios de datos, ...), y conectores (protocolos de interacción, flujo de información, ...).

- Elementos:
  - components (ports)
  - connectors (roles)
- Relaciones:
  - attachments
  - interface delegation

Análisis de propiedades:

- performance
- confiabilidad
- seguridad
- ...

¿Con qué vista del modelo 4+1 se superpone?

Views and Beyond - Asignación

Presenta un mapa entre
elementos de software (módulos, componentes o conectores)
y los elementos que no son software
(hardware, sistemas archivo, equipos desarrollo).

- Hardware - performance
- Sistema archivo - administración sistema
- Equipo desarrollo - administración proyecto

Views and Beyond - Asignación

Presenta un mapa entre
elementos de software (módulos, componentes o conectores)
y los elementos que no son software
(hardware, sistemas archivo, equipos desarrollo).

- Hardware - performance
- Sistema archivo - administración sistema
- Equipo desarrollo - administración proyecto

¿Con qué vista del modelo 4+1 se superpone?

Views and Beyond - Mas allá de las vistas

Documentar una vista puede requerir

- indicar las interfaces de los elementos
- indicar el comportamiento de los elementos y su comportamiento emergente
- explicar las decisiones de diseño

Views and Beyond - Mas allá de las vistas

Documentar una vista puede requerir

- indicar las interfaces de los elementos
- indicar el comportamiento de los elementos y su comportamiento emergente
- explicar las decisiones de diseño

¿Con qué vista del modelo 4+1 se superpone?

Views and Beyond - Relaciones entre vistas

- módulos – componentes y conectores
- módulos – asignación
- asignación (hardware) – componentes y conectores
- asignación (hardware) – asignación (sistema ficheros)
- asignación (sistema ficheros) – componentes y conectores
- asignación (equipo desarrollo) – módulos

Enfoques diferentes

¿Qué diferencia hay entre documentar y modelar?

Enfoques diferentes

¿Qué diferencia hay entre documentar y modelar?

- Documentar
  Registra decisiones de diseño

Documenting Software Architectures
Clements et al. 2011

Enfoques diferentes

¿Qué diferencia hay entre documentar y modelar?

- **Documentar**
  Registra decisiones de diseño

  Documenting Software Architectures
  Clements et al. 2011

- **Modelar**
  Permite razonar (calcular) sobre las abstracciones (analogía construcción)

  Foundations for the study of software architecture
  Perry and Wolf, 1992

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él
- animar y razonar utilizando el modelo

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él
- animar y razonar utilizando el modelo
- actualizar el modelo cuantas veces sea necesario

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él
- animar y razonar utilizando el modelo
- actualizar el modelo cuantas veces sea necesario
- calcular una solución e implementarla

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él
- animar y razonar utilizando el modelo
- actualizar el modelo cuantas veces sea necesario
- calcular una solución e implementarla

¿Algún argumento en favor del enfoque documental?

Argumento en favor de modelar

Estrategia de resolución de problemas de la escuela de Física

- entender el problema
- construir un modelo matemático de él
- animar y razonar utilizando el modelo
- actualizar el modelo cuantas veces sea necesario
- calcular una solución e implementarla

¿Algún argumento en favor del enfoque documental?

¿Hay alguna relación entre modelar y los métodos ágiles?

Estructura de la presentación

Fundamentos de Arquitectura de Software

Relación con el ciclo de vida del sistema

Actividades relacionadas a la arquitectura de software
se realizan en todas las fases del ciclo de vida del sistema.

La arquitectura de software
no es una fase del desarrollo de un sistema,
ni el producto de una fase.

- Requerimientos
- Diseño
- Implementación
- Análisis y testing
- Evolución y mantenimiento

Requerimientos – enfoque previo

Enfoque requerimientos:
El completo entendimiento de los requerimientos debe preceder cualquier trabajo hacia una solución.

"The central part of this paper sketches an approach to problem analysis and structuring that aims to avoid the magnetic attraction of solution-orientation"

Jackson, 2000

Requerimientos – enfoque previo

Enfoque requerimientos:
El completo entendimiento de los requerimientos debe preceder cualquier trabajo hacia una solución.

"The central part of this paper sketches an approach to problem analysis and structuring that aims to avoid the magnetic attraction of solution-orientation"

Jackson, 2000

¿Se ajusta este enfoque a
las limitaciones humanas de razonamiento abstracto?

Requerimientos – enfoque moderno aceptado

Motivaciones

- La arquitectura provee vocabulario para discutir
- La percepción de que y como funciona,
  afecta la percepción de necesidades
- Sistemas existentes ayudan a imaginar que podría funcionar, y permiten evaluar costos y tiempos en etapas tempranas

Enfoque requerimientos:
El análisis de requerimientos y
las consideraciones de diseño arquitectónicas
son natural y apropiadamente trabajadas
de manera cooperativa y simultanea.

Diseño – asociado a otras actividades

La actividad de diseño es por definición
la que crea la arquitectura de software.

- En la fase de diseño hay énfasis preocupaciones arquitectónicas
- El diseño arquitectónico no es exclusivo de la fase de diseño
- La actividad de diseño se asociada a otras actividades
- Requiere un rico repertorio de técnicas de diseño

Implementación

La arquitectura no está completa
al comenzar actividades de implementación,
las cuales agregan y modifican la arquitectura.

- Se mantiene la consistencia entre código y arquitectura
- Desarrollo basado en arquitecturas facilita enfoques de:
  - generación de código
  - reutilización de código

Análisis y testing

Las actividades de análisis y testing
son conducidas normalmente luego que el código está escrito.

Cuanto antes se detecta un error
menor es el costo de su corrección.

Una representación rigurosa permite anticipar análisis
preferentemete con asistencia de herramientas

- Consistencia, correctitud, requerimientos no funcionales
- Consistencia con requerimientos
- La arquitectura determina estrategias viables de análisis y testing
- Puede compararse con modelos obtenidos con ingeniería reversa

Evolución y mantenimiento

El mayor riesgo en la evolución es la degradación de calidad.
(Cambios, típicamente ad hoc, hechos de la forma más expeditiva.)

Un enfoque-basado en arquitecturas
es una base solida para soportar la evolución.

Un modelo arquitectónico consistente, fiel a la implementación:

- ayuda a entender y analizar un cambio
- ofrece una base para decidir si un cambio es razonable

Primero se modifica el modelo arquitectónico,
luego se implementa el cambio.

Estructura de la presentación

Fundamentos de Arquitectura de Software

Patrones y estilos arquitectónicos elevan el nivel de abstracción

Software Architecture Volume 1: A System of Patterns
F.Buschmann et al. 1996
Software Architecture: Perspectives on an Emerging Discipline
D. Garlan and M. Shaw. 1996

Estilo arquitectónico

"... es una especialización de tipos de elementos y relaciones, junto con un conjunto de restricciones sobre como deben ser usados."

Software architecture in practice
Bass, Clements, and Kazman. 2003

Patrón arquitectónico

"... describe un problema de diseño recurrente particular
que emerge en contextos específicos de diseño, y
presenta un esquema genérico y bien probado para su solución.
El esquema de solución es especificado describiendo
sus componentes constituyentes, sus responsabilidades y relaciones,
y las formas en que estos colaboran."

Software Architecture Volume 1: A System of Patterns
F.Buschmann et al. 1996

Estilo vs. patrón arquitectónico

Ambos

- Enfoques arquitectónicos observados
- Proveen un vocabulario elevando abstracción en comunicaciones
- Paquete de decisiones de diseño
  - tipos de elementos y relaciones
  - propiedades y restricciones sobre su estructura e interacciones
- Resultan en un balance de atributos de calidad conocido

Los patrones incluyen un problema recurrente/contexto de aplicación, los estilos generalmente no.

Tácticas

Una táctica identifica una única decisión de diseño
que apunta a lograr un atributo de calidad específico.

Software architecture in practice
Bass, Clements, and Kazman. 2003

Ejemplos en tolerancia a fallos

- Recuperación de fallos
  - Redundancia activa
  - Redundancia pasiva
  - Repuesto

Estructura de la presentación

Fundamentos de Arquitectura de Software

Caracterización

Los lenguajes de descripción arquitectónica (ADL) modelan las arquitecturas en términos de componentes y conectores que se combinan en configuraciones de acuerdo a sus interfaces.

- Componente.
  - Encapsula funcionalidad y/o datos
  - Restringe acceso a través de interfaz
  - Tiene dependencias explícitas de su contexto requerido
- Conector. Elemento arquitectónico a cargo de la tarea de efectuar y regular las comunicaciones entre componentes.
- Configuración. Conjunto de asociaciones específicas entre los componentes y conectores de la arquitectura de un sistema.

Soporte de estilos/patrones arquitectónicos

(re)configuración de una instancia de un patrón/estilo

- Tipos de elementos
- Restricciones
  - Satisfacción por construcción.
    Operaciones de reconfiguración específicas del patrón y restricciones implícitas
  - Satisfacción por verificación.
    Operaciones genéricas de reconfiguración y restricciones explicitas a verificar

Semántica

Lenguajes y su semántica

- Grafos
  - ADR (Bruni et al. 94)
  - Archery (Sanchez et al. 2011)
- Álgebras de proceso
  - Darwin (Magee and Kramer 1996)
  - Wright (Allen et al. 1998)
  - Archery (Sanchez et al. 2011)

UML puede utilizarse para modelar configuraciones de componentes y conectores (arquitecturas), pero carece de estas abstracciones; no es ADL. No tiene semántica formal.

ACME (Garlan et al. 1997) es un lenguaje de intercambio de especificaciones arquitectónicas estructurales. No tiene semántica formal.

Estructura de la presentación

Fundamentos de Arquitectura de Software

Arquitectura de línea de productos

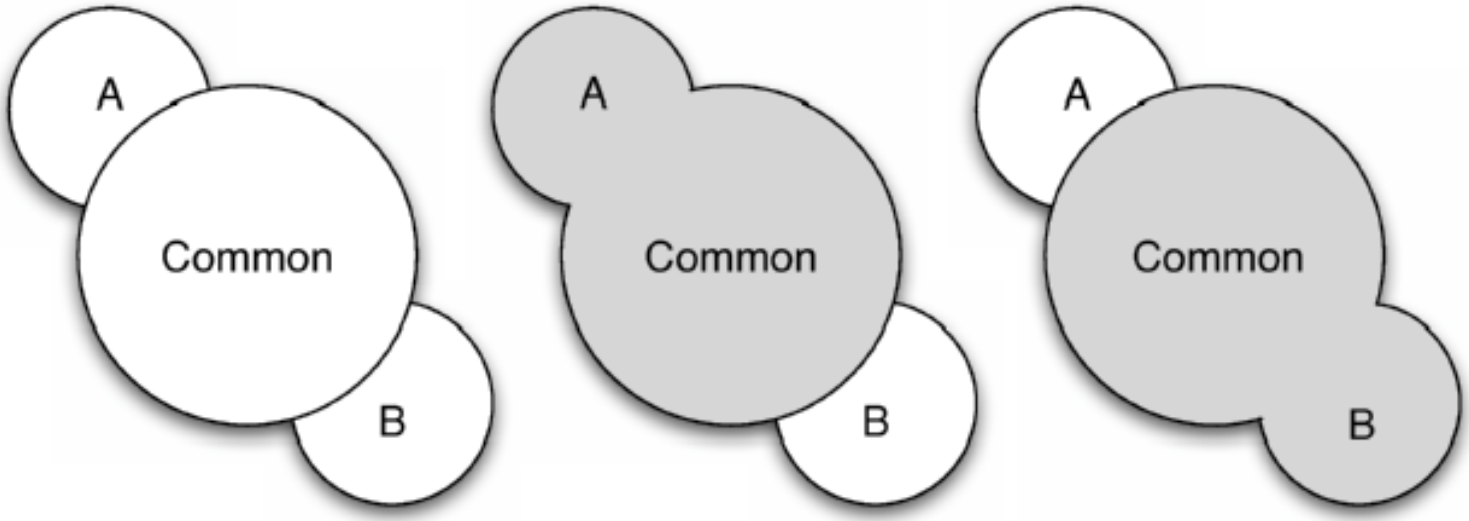
La arquitectura de software posibilita administrar familias de productos con una cantidad significativa de componentes y estructura en común en líneas de productos de software.

La arquitectura de una línea de productos
captura las arquitecturas de la familia de productos simultaneamente.

Software Architecture: Foundations, Theory and Practice
Taylor, Medvidovic, and Dashofy. 2009

Emplea puntos de variación en la arquitectura que indican donde las decisiones de diseño divergen de un producto a otro.

Ejemplo línea de productos



- Common: caract. comunes
- A: caract. especificas A
- B: caract. especificas B

- Product A = Common + A
- Product B = Common + B

Software Architecture: Foundations, Theory and Practice
Taylor, Medvidovic, and Dashofy. 2009