

Arquitecturas de Software

Diseño y Arquitectura de Software
Grado en Ingeniería de Software



Carlos E. Cuesta
carlos.cuesta@urjc.es

Arquitectura de Software

Introducción

■ Motivación

- Incremento en el tamaño y complejidad del software
- Necesidad de aprender de la experiencia
 - Reutilización de estructuras usadas en problemas similares
- Una adecuada estructura general es tan importante o más que las implementaciones concretas de las partes.

■ Definición

- *La **arquitectura de software** está compuesta por la estructura de los elementos de un programa o sistema, sus interrelaciones y los principios y reglas que gobiernan su diseño y evolución a lo largo del tiempo*

Arquitectura de Software

Introducción

- La definición incluye:
 - la descripción de los componentes con los cuales se construyen los sistemas
 - las interacciones entre esos componentes
 - Los patrones para guiar la composición
 - restricciones sobre dichos patrones
- Describe los elementos arquitectónicos básicos
 - Estructura, componente, conexión, evolución
- Componentes
 - Servidores, clientes, bases de datos, filtros, capas en un sistema jerárquico, etc.

Arquitectura de Software

Introducción

- Interacciones (conexiones o *conectores*)
 - Llamadas a procedimientos, protocolos C/S, protocolos de acceso a BD, etc.
 - Pero no todo está tan claro, siempre...
- Se recupera la perspectiva arquitectónica
 - La verdadera transición entre análisis y diseño
 - Basada en principios conocidos, pero “olvidados”

“Par ma foi! il y a plus de quarante ans que je dis de la prose sans que j'en susse rien!”

Le Bourgeois Gentilhomme (Molière)

Arquitectura de Software

Introducción

- ¿De qué se ocupa?
 - Diseño preliminar o de alto nivel.
 - Algunos autores dirían: una actividad *previa* al diseño
 - Organización a alto nivel del sistema, incluyendo aspectos como la descripción y análisis de propiedades relativas a su estructura y control global, los protocolos de comunicación y sincronización utilizados, la distribución física del sistema y sus componentes, etc.
 - Otros aspectos relacionados con el desarrollo del sistema y su evolución y adaptación al cambio:
 - composición, reconfiguración, reutilización, escalabilidad, mantenibilidad, etc.

Arquitectura de Software

Introducción

- ¿De qué no se ocupa?
 - Diseño detallado
 - Diseño de algoritmos
 - Pero claro, ¿qué es un algoritmo?
 - Diseño de estructuras de datos.
 - Pero claro, ¿qué es una estructura de datos?
- La granularidad es importante
 - No estamos hablando a un único nivel
 - Podemos hablar de estructuras grandes o pequeñas
 - ... pero hay niveles que difícilmente son “arquitectura”

Enfoques Arquitectónicos

- Enfoque prescriptivo
 - El original: Perry & Wolf, 1989
 - Indica lo que *requiere* la arquitectura, y con ello acota el espacio de soluciones, *restringiéndolo*
- Enfoque descriptivo
 - El más desarrollado, al menos inicialmente
 - Debido a Taylor, popularizado por Garlan & Shaw
 - Se indica el conjunto de elementos que la forman
 - Componentes: desde McIlroy hasta Garlan
 - Conectores: la aportación de Mary Shaw

Elementos fundamentales: enfoque descriptivo

- Componentes (i.e. elementos)
 - Naturaleza esencialmente jerárquica
 - El caso de UML
 - Componentes vs. Structured Classes (UML 2.0)
- Conectores (conexiones, *interaction components*)
 - Opcionales: definen el enfoque C&C
- Puertos (y “roles”)
 - Attachments y bindings
- Sistemas, configuraciones y arquitecturas
- Vistas arquitectónicas (perspectivas, etc.)

El “estándar” IEEE 1471

- ANSI/IEEE Std RP-1471-2000
 - Desarrollado entre 1996 y 2000 (Rich Hilliard)
- Sus aportaciones son las siguientes
 - Metamodelo conceptual
 - Asociación con los *concerns* de un *stakeholder*
 - Define las arquitecturas como *multi-vista*
 - Distingue *vista* de *punto de vista*
 - Noción de *conformidad arquitectónica*
 - Énfasis inicial en el *architectural rationale*
- Definición estándar basada en la anterior

Arquitectura: Definiciones

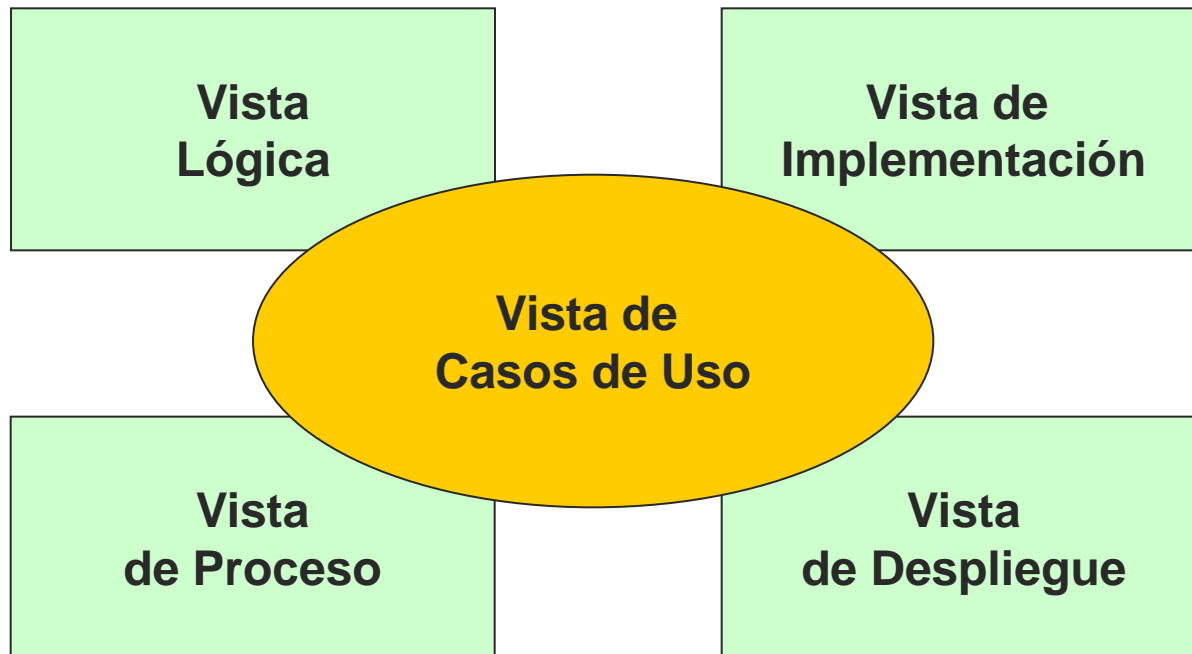
- Perry & Wolf, 1992
 - Architecture = Elements + Forms + Rationale
- Bass, Clemens & Kazman, 2003
 - The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them
- UML 1.3
 - Architecture is the organizational structure of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.
- IEEE Std. 610.12-1990
 - Architecture is the organizational structure of a system

Definición: IEEE 1471

- La definición oficial es:
 - The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution
 - Claramente basada en Perry & Garlan, 1995
 - Pero “suavizada”
- Ni siquiera requiere *estructura*, que era prácticamente el consenso previo
- Su gran novedad son las *vistas*

El Modelo de “4+1 Vistas”

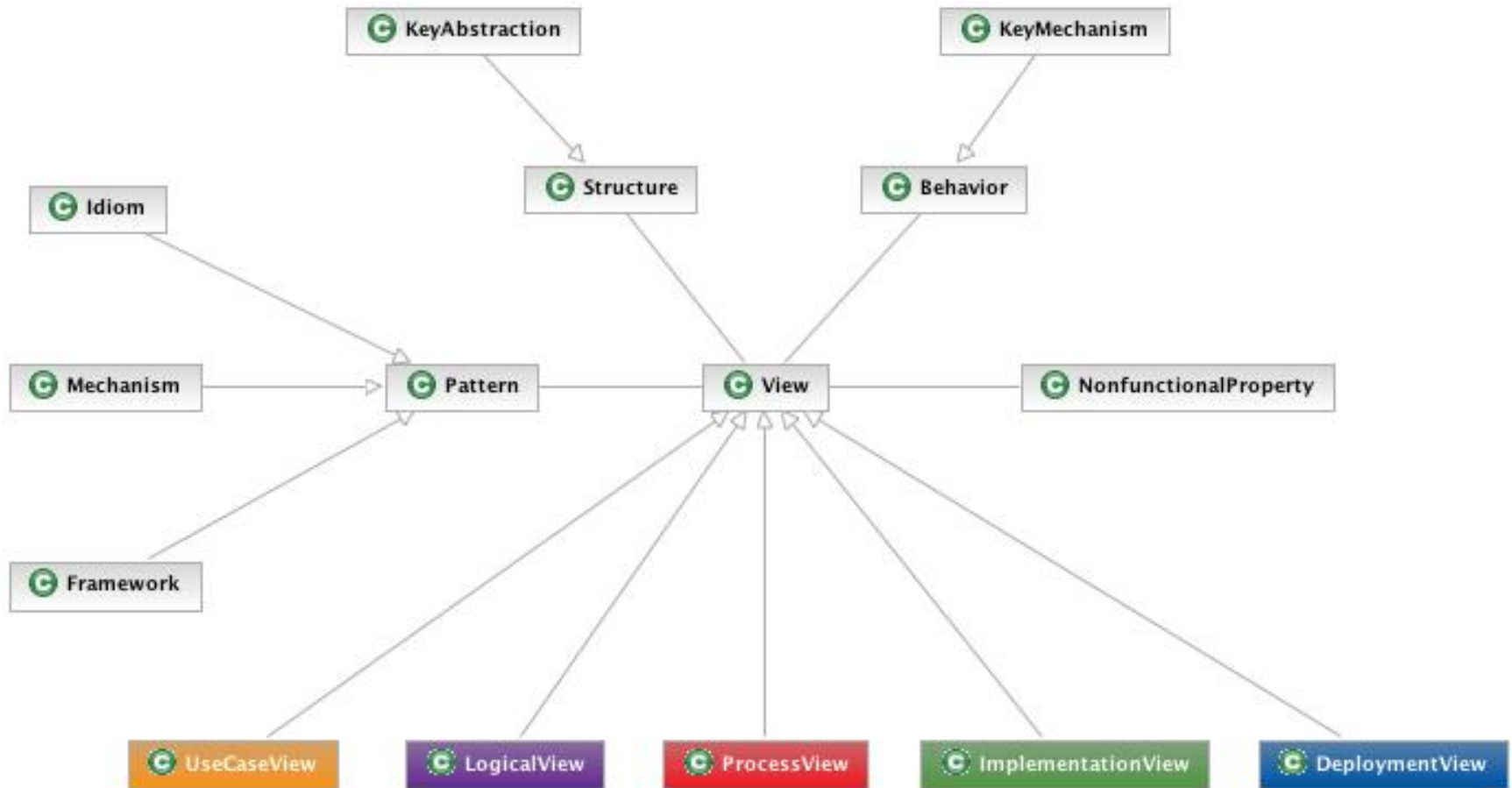
- Elaborado por Philippe Kruchten (*circa* 1995)



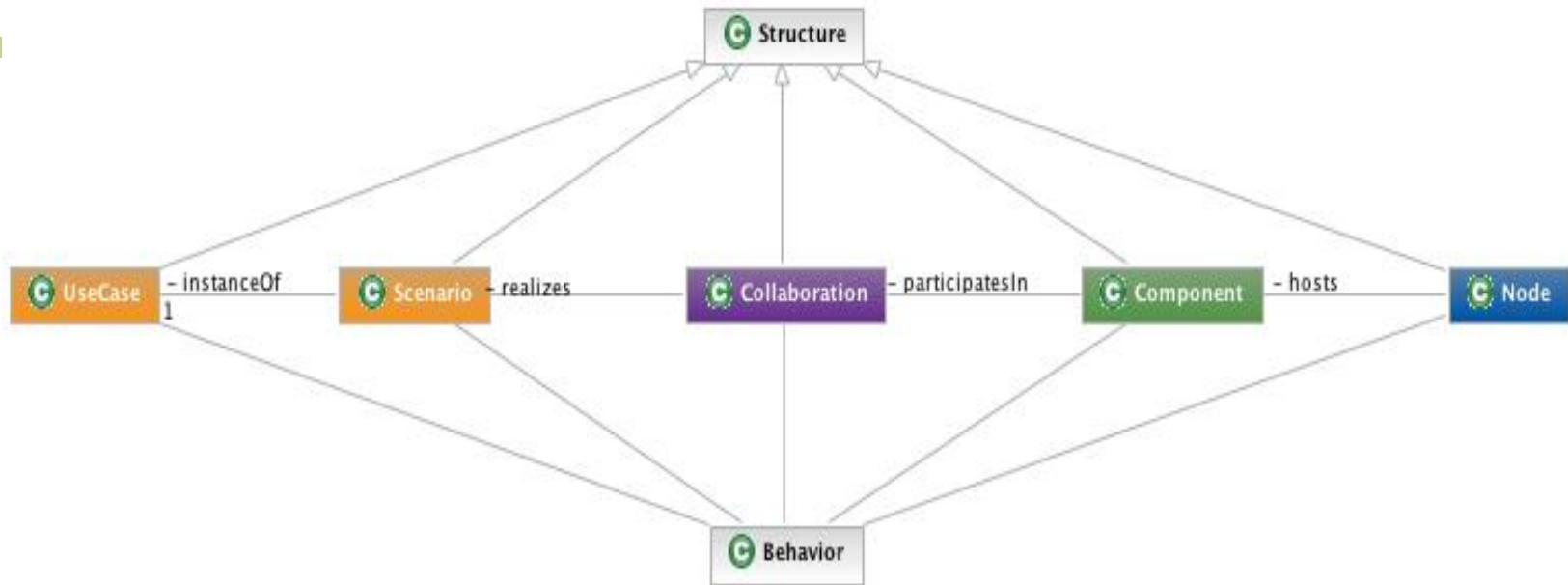
Vistas en Rational Rose

- Inspiradas en el Modelo 4+1 de Kruchten
 - Vista de Casos de Uso
 - Diagramas de Casos de Uso
 - Diagramas de Clases
 - Diagramas de Colaboración, Secuencia, Estados, Actividad
 - Vista Lógica
 - Admite los mismos diagramas que la Vista anterior
 - Modelo de Análisis
 - Modelo de Diseño
 - Vista de Componentes
 - Diagramas de Componentes
 - Modelo de Implementación
 - Vista de Despliegue
 - Diagrama de Despliegue

Vistas en UML – Metamodelo



Trazabilidad en UML



- Los escenarios son instancias de los casos de uso
 - Plantean dos representaciones de la misma abstracción
- Una colaboración *realiza* un escenario
- Los componentes participan en colaboraciones
- Los nodos alojan a los componentes

Arquitectura de Software

Estilos Arquitectónicos

- Indican:
 - Los tipos de componentes y conectores involucrados.
 - Patrones y restricciones de interconexión o composición entre ellos
 - *Invariantes* del estilo (restricciones)
- Asociados a cada estilo hay una serie de propiedades que lo caracterizan
 - Determinan sus ventajas e inconvenientes
 - Condicionan la elección de uno u otro estilo.

[Bibliografía]

- M. Shaw & D. Garlan.
Software Architecture: Perspectives on an Emerging Discipline, 1996.
- L.Bass, P.Clements & R.Kazman.
Software Architecture in Practice, 2003.
- N.Rozanski & E.Woods.
Software Systems Architecture, 2005.
- R.Taylor, N.Medvidovic & E.Dashofy.
Software Architecture: Foundations, Theory & Practice, 2009.