

Enterprise Architect y UML Básico



Diciembre 2008

Carlos Alexander Zuluaga

Agenda

- Diagrama de despliegue.
 - Introducción.
 - Modelado.
- Diagrama de estructuras compuestas.
 - Introducción.
 - Modelado.
- Modelo 4 + 1 vistas.
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- Uso de figuras.
 - Ejercicio.
- Generación de código e ingeniería inversa.
- Recursos.

Agenda

- **Diagrama de despliegue.**
 - Introducción.
 - Modelado.
- Diagrama de estructuras compuestas.
 - Introducción.
 - Modelado.
- Modelo 4 + 1 vistas.
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- Uso de figuras.
 - Ejercicio.
- Generación de código e ingeniería inversa.
- Recursos.

Diagrama de Despliegue - Introducción

- Es la representación física del sistema y muestra como está o estará desplegado.
- Para adicionar información se puede crear un metamodelo y clasificar los elementos físicos dentro de él.

Elementos:

- **Nodo:** Elemento físico que suele contar con memoria y capacidad de procesamiento.
- **Artefacto:** Elemento que puede ser ejecutado dentro de un nodo.
- **Conectores:** Representan conexiones físicas entre nodos (esto incluye las inalámbricas).
- **Dispositivo (Device):** Nodo sin capacidad de procesamiento.
- **Ambiente de Ejecución (Execution Environment):** Un nodo que ofrece un ambiente de ejecución.

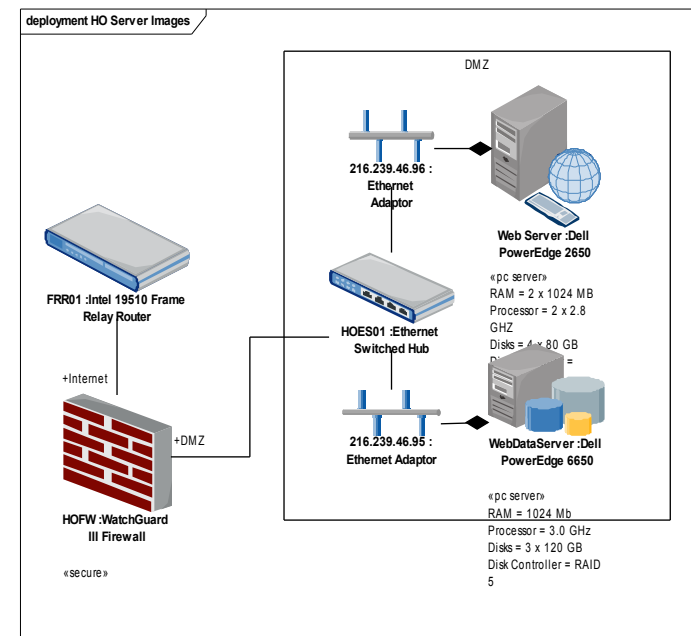


Diagrama de Despliegue - Modelado

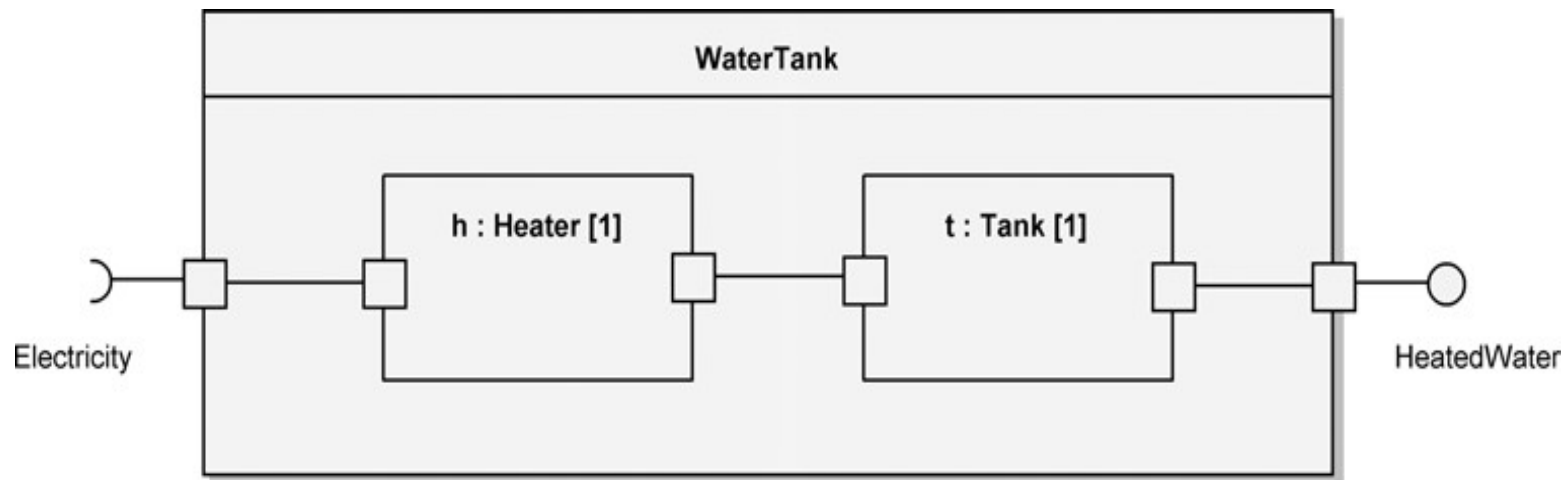
- Para crear un diagrama de despliegue: Clic derecho sobre el paquete → *Add* → *Add Diagram* → *UML Structural* → *Deployment*.
- Una buena guía para elaborar diagramas es el modelo de ejemplo de Enterprise Architect.
- Para clasificarlo como una instancia:
 - Clic derecho sobre el elemento → *Advanced* → *Convert to Instance*.
 - Clic derecho de nuevo sobre el objeto → *Advanced* → *Instance Classifier*.
 - Seleccionar la clase a la que pertenece.
 - Para asignar los valores: Clic derecho sobre el elemento → *Advanced* → *Set Run State* → Asignar los atributos parametrizados.

Agenda

- Diagrama de despliegue.
 - Introducción.
 - Modelado.
- **Diagrama de estructuras compuestas.**
 - Introducción.
 - Modelado.
- Modelo 4 + 1 vistas.
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- Uso de figuras.
 - Ejercicio.
- Generación de código e ingeniería inversa.
- Recursos.

Diagramas de Estructuras Compuestas - Introducción

- Proveen una forma para estructurar elementos en un diagrama e incluir la definición de su estructura interna.
- Muestran la colaboración interna de los elementos.
- Elementos básicos:
 - Parte.
 - Puerto.
 - Interfase.
 - Conector.



Diagramas de Estructuras Compuestas - Modelado

- Para agregar un nuevo diagrama de Estructuras Compuestas: Clic derecho sobre el paquete → *Add* → *Add Diagram* → *UML Structural* → *Composite Structure*.
- Para definir la estructura interna de una clase existente una buena opción es convertirla en una instancia, aunque la clase base no estará ligada automáticamente con su instancia.
- Otra forma de hacerlo es ocultar las operaciones y atributos del diagrama.
- Exponer y consumir interfaces a través del elemento *Expose Interface*.
- Ensamblar interfaces a través del elemento *Assembly*.
 - Dirigirlas desde el consumidor hasta el proveedor.

Agenda

- Diagrama de despliegue.
 - Introducción.
 - Modelado.
- Diagrama de estructuras compuestas.
 - Introducción.
 - Modelado.
- **Modelo 4 + 1 vistas.**
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- Uso de figuras.
 - Ejercicio.
- Generación de código e ingeniería inversa.
- Recursos.

La Arquitectura de Software

- Definiciones:
 - *“Estructura de estructuras”*.
 - *“Propiedades visibles externamente”*.
 - *“Diseño de alto nivel”*.
 - *“Componentes y conectores”*.
- Un arquitecto debe tener en cuenta múltiples influencias internas y externas:
 - Usuarios.
 - Gerentes.
 - Desarrolladores.
 - Experiencias exitosas.
 - Entorno técnico.
 - Seguridad.
 - Legislación.

La Arquitectura de Software

- ¿Cómo se crea una arquitectura?
 - Entradas:
 - Requerimientos no funcionales.
 - Casos de uso arquitectónicamente significativos.
 - Influencias externas.
 - Objetivos del negocio.
 - Elementos que la constituyen:
 - Patrón arquitectónico.
 - Patrones de diseño.
 - Lineamientos.
 - Documentación.
- ¿Cómo es una buena arquitectura?
 - Debe ser el producto de un arquitecto o un grupo de arquitectos con un líder identificado.
 - Cubre los requerimientos no funcionales.
 - Debe darse a conocer.
 - Debe ser validada y medida en términos cuantitativos.
 - *No debe depender de productos comerciales.*
 - ¡RACIONALIZAR DECISIONES!

Modelo 4+1 Vistas en Enterprise Architect

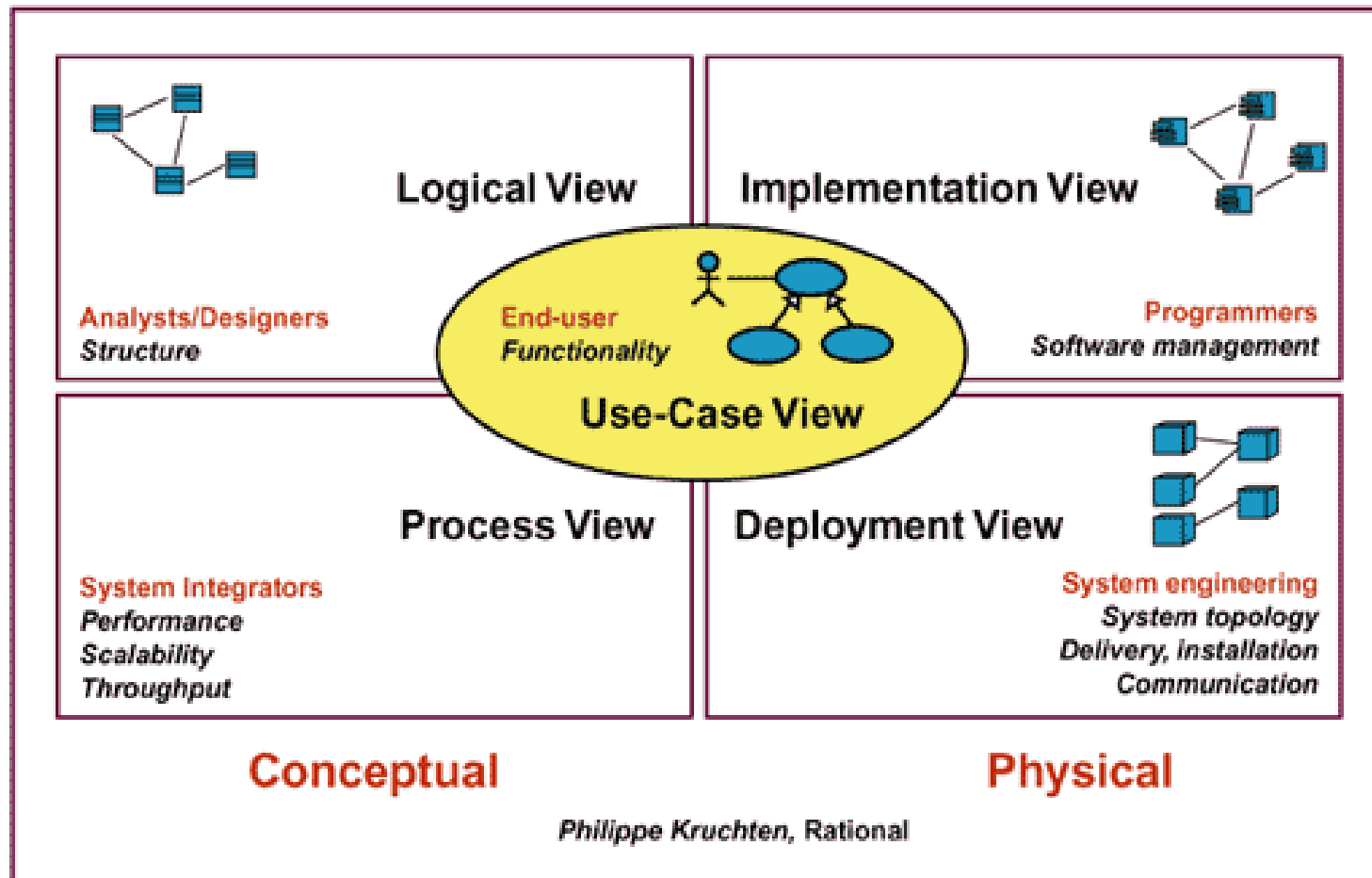


Imagen tomada de: <http://www.ibm.com/developerworks/rational/library/797.html>

Las Vistas

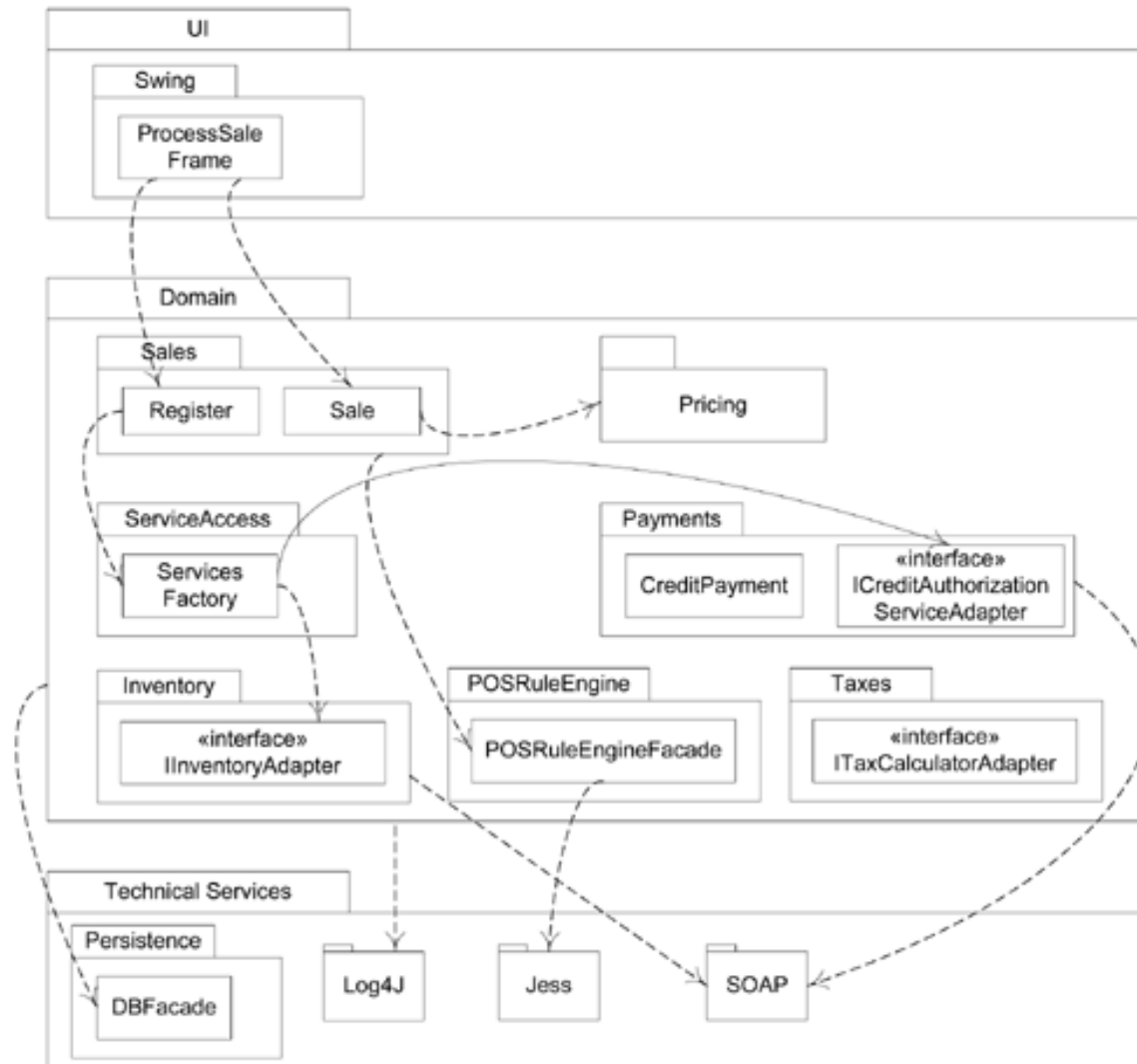
- **Vista Lógica:** Se enfoca en la funcionalidad de la aplicación en términos de elementos estructurales. Los diagramas más adecuados para modelarla son:
 - Clases.
 - Estructuras compuestas.
 - Objetos.
 - Estados.
 - Paquetes.

- **Procesos:** Aspectos no funcionales de la aplicación como concurrencia, escalabilidad. Muestra las principales abstracciones de la aplicación en términos de procesos de sistema operativo e hilos de esos procesos. Los diagramas recomendados para modelarla son:
 - Secuencia.
 - Comunicación.
 - Timing.
 - Clases.

Vista Lógica

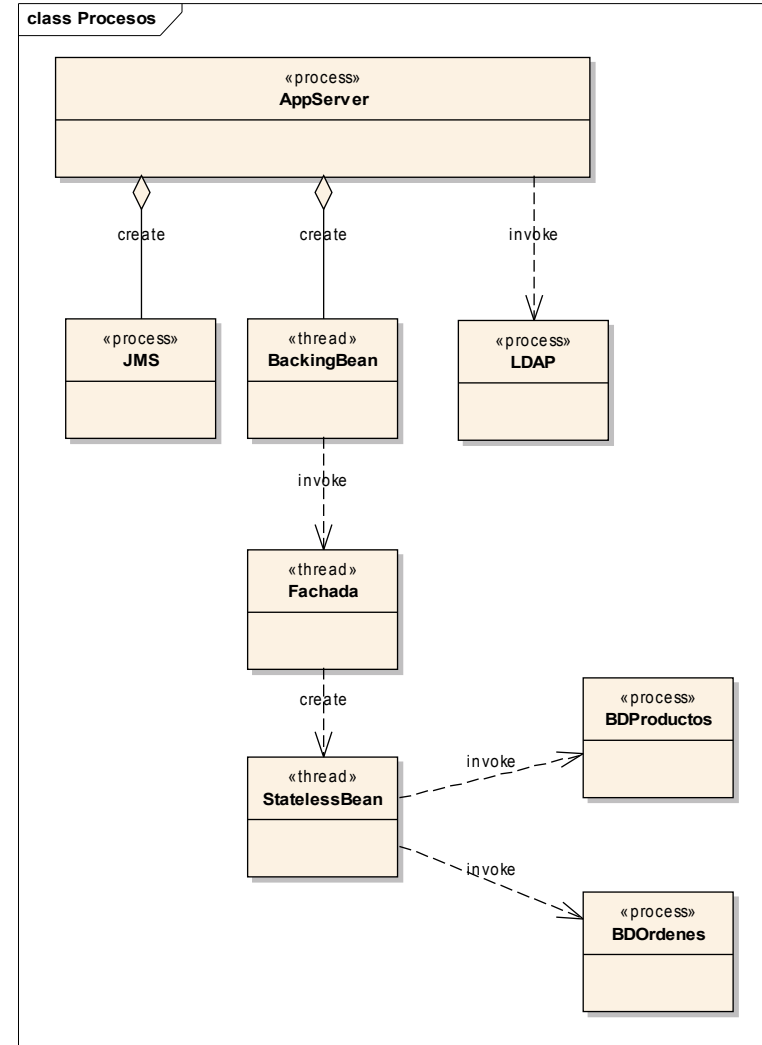
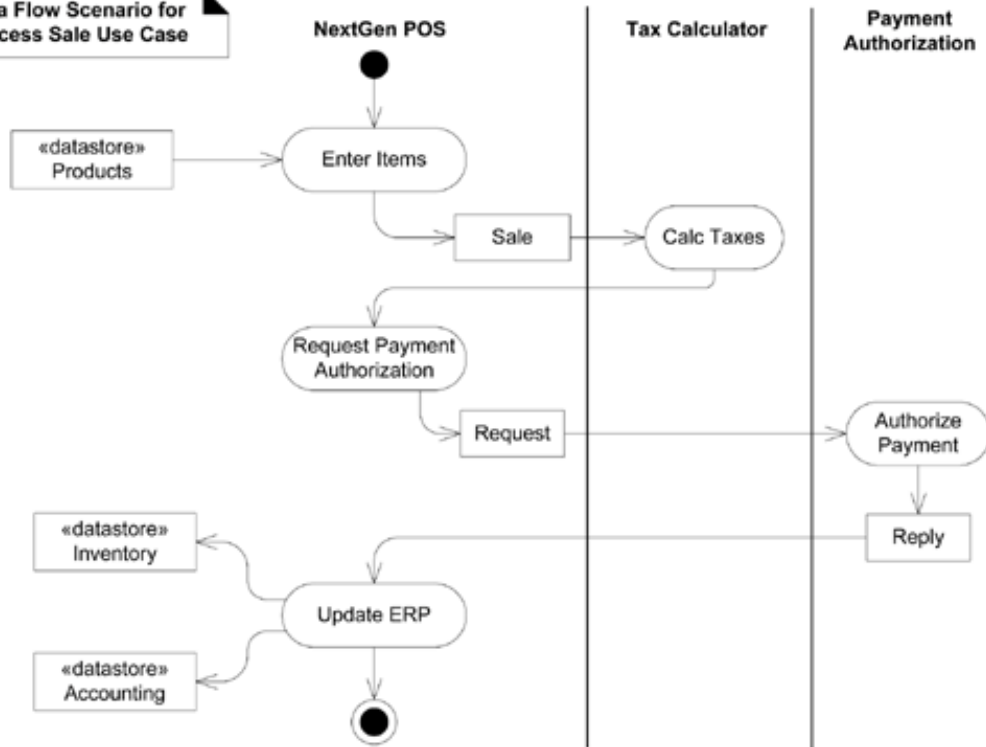
... other technical memos ...

Logical View



Vista de Procesos

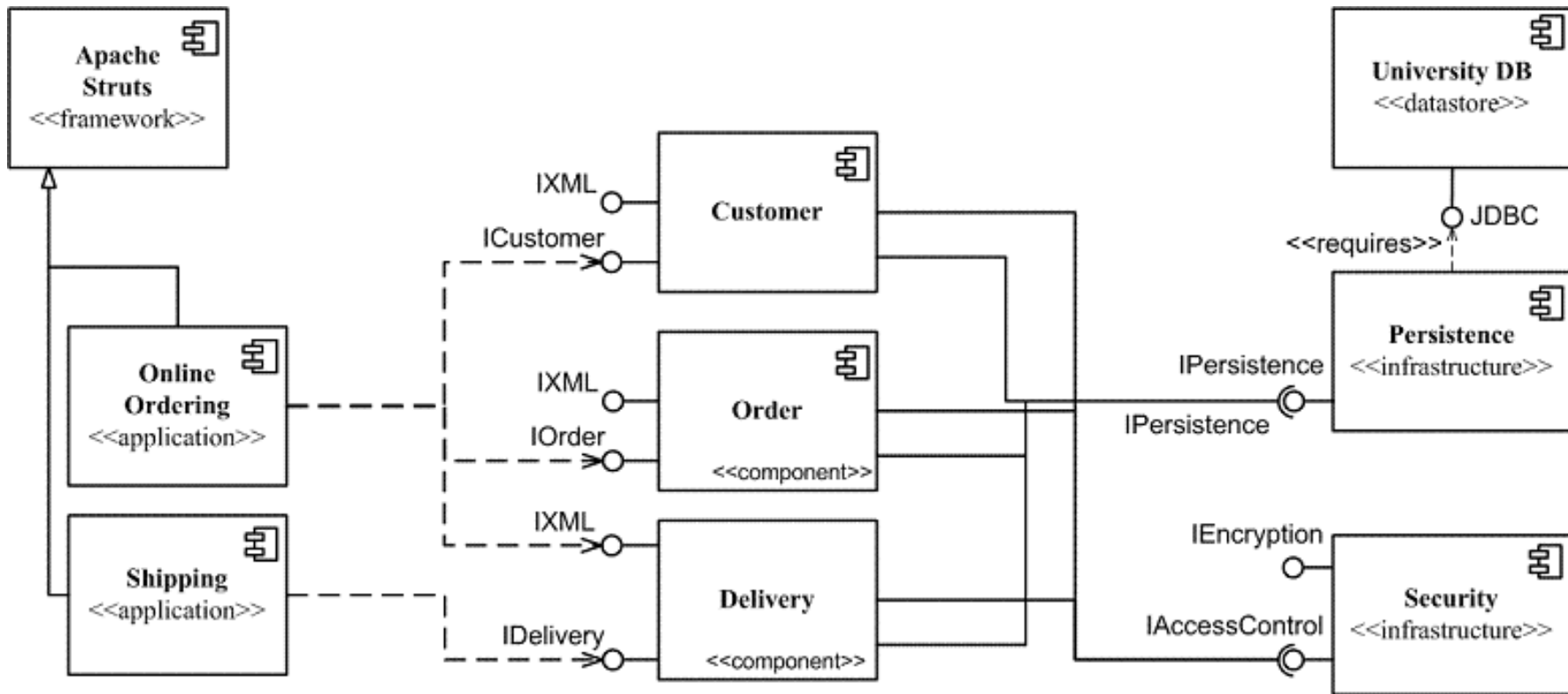
Data Flow Scenario for Process Sale Use Case



Las Vistas

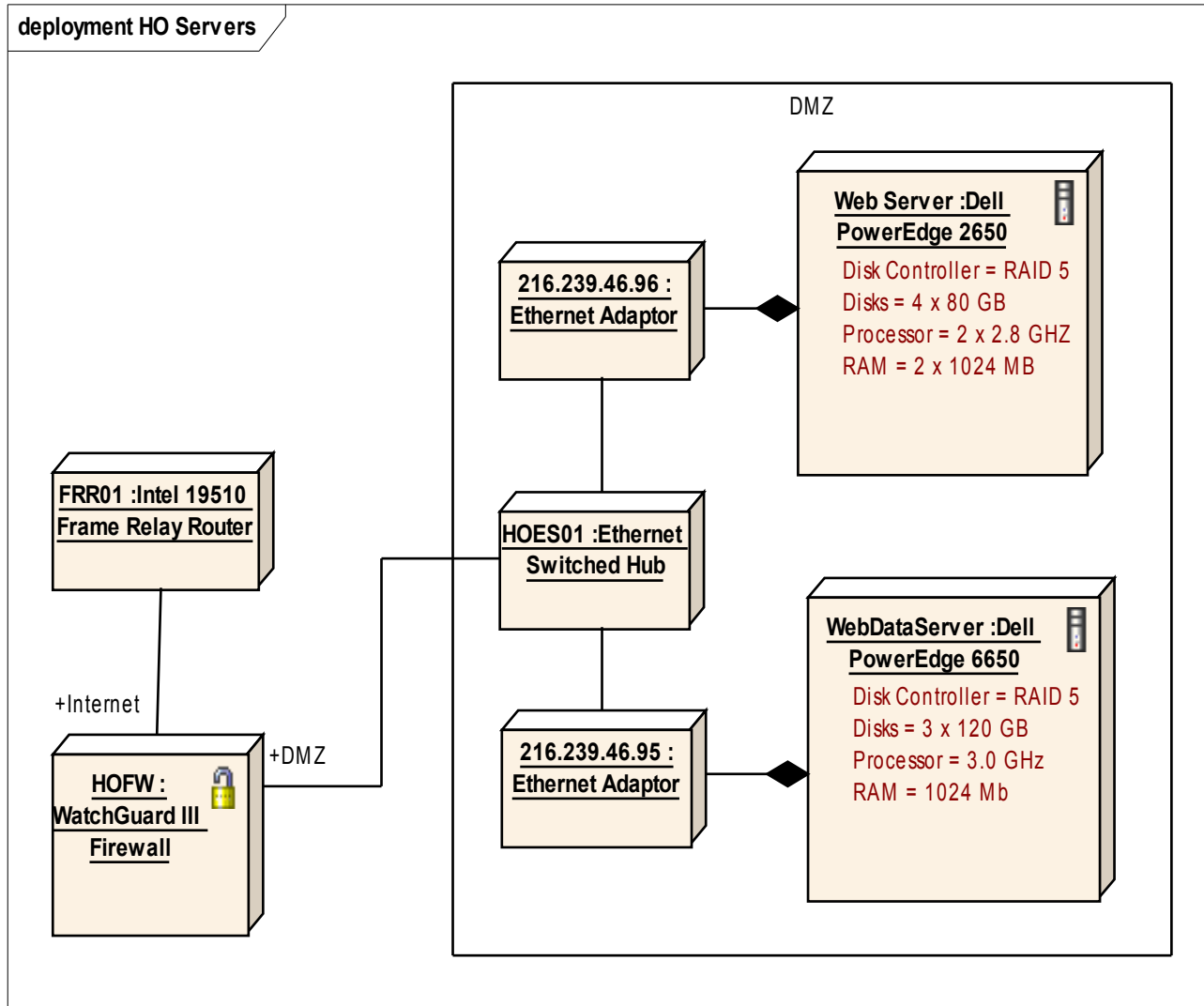
- **Vista de Desarrollo:** Muestra los componentes usados para ensamblar y entregar el sistema construido. Esta vista se debe mostrar en forma de *caja negra* y de *caja blanca*. Los diagramas recomendados para modelarla son:
 - Componentes.
 - Estructuras compuestas.
- **Vista de despliegue:** Muestra los elementos físicos del sistema como recursos computacionales, redes, conexiones físicas, dispositivos, etc. El diagrama usado para esta vista es el de despliegue.

Vista de Desarrollo



Copyright 2005 Scott W. Ambler

Vista de Despliegue



Las Vistas

- **Casos de Uso:** Es la vista principal para capturar escenarios sobre los que debe ser probada la arquitectura.
- Estos casos de uso son los arquitectónicamente significativos que no necesariamente serán los casos de uso del sistema.
- Se capturan en el diagrama de casos de uso.
- ¡Lo importante es la descripción!

Agenda

- Diagrama de despliegue.
 - Introducción.
 - Modelado.
- Diagrama de estructuras compuestas.
 - Introducción.
 - Modelado.
- Modelo 4 + 1 vistas.
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- **Uso de figuras.**
 - Ejercicio.
- Generación de código e ingeniería inversa.
- Recursos.

Uso de Figuras

- Para importar la imágenes:
 - Ir a la dirección <http://www.sparxsystems.com/resources/image_library.html> y descargar la librería de imágenes.
 - *Tools* → *Import Reference Data*.
 - Seleccionar el archivo *ImageLibrary.xml*.
 - Seleccionar los elementos que contiene y presionar *Import*.
 - Cerrar la ventana.
- Asignar una imagen a un elemento:
 - Clic derecho sobre el elemento.
 - *Appearance* → *Select Alternate Image*.
 - Seleccionar la imagen deseada.
 - OK.

Agenda

- Diagrama de despliegue.
 - Introducción.
 - Modelado.
- Diagrama de estructuras compuestas.
 - Introducción.
 - Modelado.
- Modelo 4 + 1 vistas.
 - La arquitectura de software.
 - Modelado en Enterprise Architect.
- Uso de figuras.
 - Ejercicio.
- **Generación de código.**
- Recursos.

Generación de Código

- Vista de código fuente: *View* → *Source Code*.
- Para generar el código fuente de un grupo de clases:
 - Clic derecho sobre el diagrama para el que se desea generar el código.
 - *Code Engineering* → *Generate Source Code*.
 - Seleccionar las clases y presionar *Generate*.
 - Para cada clase seleccionar el directorio destino o chequear la opción *Auto Generate Files*.
- El código generado se puede ver en la vista de código fuente o en el archivo físico.
- Se puede generar el código fuente de una clase haciendo clic derecho sobre ella y *Generate Code*.
- Ajustar las colecciones para relaciones X a muchos.
 - *Tools* → *Options* → *Source Code Engineering*. (Use expresiones del tipo *List<#TYPE#>*)
 - En la relación seleccionar el *Target Role* e ingresar el tipo de dato en el campo *Member Type*.
- Para editar las plantillas generadoras de código: *Settings* → *Code Generation Templates*.

Recursos

- http://www.sparxsystems.com/downloads/whitepapers/FCGSS_US_WP_Applying_4+1_w_UML2.pdf
- www.booch.com/architecture/blog/artifacts/Software%20Ar
- <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>