

## Taller avanzado de Casos de Uso y Arquitectura SOA

Cómo traducir requerimientos a componentes de software  
con UML 2.1, un contrato de servicios SOA y web services

### Contenido

PROPÓSITO .....	2
ALCANCE .....	2
REFERENCIAS .....	2
ACTUACIÓN .....	2
INTRODUCCIÓN .....	3
CONTEXTO DEL TALLER .....	4
OBJETIVOS GENERALES .....	5
OBJETIVOS ESPECÍFICOS .....	6
CONOCIMIENTOS .....	6
HABILIDADES .....	6
COMPETENCIAS .....	7
DIRIGIDO A .....	8
PRECONDICIONES .....	8
CONTENIDOS DEL TALLER .....	9
1. PRIMERA SESIÓN: FUNDAMENTOS .....	9
2. SEGUNDA SESIÓN: TRADUCIR REQUERIMIENTOS A CASOS DE USO .....	9
3. TERCERA SESIÓN: TRADUCIR CASOS DE USO A COMPONENTES DE SOFTWARE .....	10
4. CUARTA SESIÓN: ARQUITECTURA SOA A PARTIR DE LOS CASOS DE USO .....	10
5. QUINTA SESIÓN: PLAN DE PRODUCCIÓN BASADO EN CASOS DE USO .....	11
METODOLOGÍA DOCENTE .....	11
RECURSOS DIDÁCTICOS .....	12
REQUISITOS DEL AULA .....	13
MÉTODO DE EVALUACIÓN .....	14
REFERENCIAS .....	14

*El copyright de los materiales del taller pertenece a VICO OPEN MODELING, S.L. y sólo puede ser distribuido en forma física (impreso en papel) o electrónica no modificable (formato pdf) previo acuerdo suscrito con los autores.*

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvc vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 1 de 16
---	---------------------------------	----------------	--------------------

## Propósito

Definir el contenido curricular de un taller sobre la especificación de Casos de Uso. Donde mostramos los mecanismos de extensión de escenarios basados en “Aspectos”, y la “realización” de Casos de Uso a través de un “contrato de servicios” basado en arquitectura SOA y “Web Services”, con la finalidad de reunir los materiales necesarios para concertar una propuesta de formación, de acuerdo con las necesidades de un equipo de proyecto orientado al desarrollo de software.

## Alcance

Empresa: **Coordinador de formación**  
VICO: **Josep Vilalta & Rafael Rosa**

## Referencias

UML_TRAD_GuionCurso_abierto.pdf	<b>Diseño curricular</b>
UML_TRAD_GuionCurso_abierto_Profe.pdf	<b>Guión del profesor</b>
UML_TRAD_GuionCurso_abierto_Alum.pdf	<b>Guión del alumno</b>
UML_TRAD_poster.pdf	<b>Mapa del taller</b>

## Actuación

Título: **Taller avanzado de Casos de Uso y Arquitectura SOA**  
Subtítulo: **Cómo traducir requerimientos a software con UML, SOA y web services**  
Modalidad: **Formación presencial Ref.- TRAD205**  
Recursos: **Material UML-TRAD + soporte de tutoría y recursos web Campus virtual  
<http://www.aprendeuml.com>**  
Canales: **VICO & Empresas & Universidad & Administración Pública**  
Programa: **Opción A: 25 horas – 5 sesiones de 5 horas  
Opción B: 24 horas – 3 sesiones de 8 horas  
Opción C: personalizada**  
Aula: **Sede empresa / Centro concertado**  
Profesor: **Josep Vilalta Marzo - OMG Certified UML Professional (OCUP)**  
Participantes: **Profesionales empresa**  
Matrícula: **Tarifa general: 950,00 € + iva por alumno  
Pack de 10 alumnos: tarifa forfait: 6.500,00 €+ iva**  
Gastos asociados: **Desplazamiento: ..... € Dietas: ..... €/día  
Alojamiento y transportes: ..... € Aula centro concertado: ..... €**  
Formalización: **Aceptación de oferta y concertación de agenda**

## Introducción

Unified Modeling Language “UML”, es una notación patrocinada por el Object Management Group (OMG), que se ha convertido en un estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de un sistema. Desde [vico open modeling](#), como [empresa acreditada por el OMG](#), realizamos un transfer de conocimiento y experiencia sobre UML combinando distintas líneas de actuación para cada necesidad:

### Formación

- Presencial: A través de centros concertados o en la propia empresa.
- No presencial: A través de tutorías asistidas en soporte web.

### Coaching

- Presencial: Coordinamos equipos de desarrollo de software y/o modelado de procesos de negocio, en el mismo entorno de proyecto.
- No presencial: A través de soporte web.

### Outsourcing selectivo

- Elaboramos entregables de proyecto correspondientes a las fases de captura de requerimientos, modelos de dominio, procesos de negocio y modelo de Casos de Uso.
- Implantamos metodologías, modelos de métrica y ayudamos a superar las certificaciones de calidad del software (SPICE – CMMI – ISO 9000:2000).

Nuestra misión es aumentar la capacidad de trabajo y el nivel de competencia de sus profesionales en los siguientes puntos:

1. Saber **trabajar en equipo** con unos **roles diferenciados** (Analista de Negocio, Arquitecto, Programador...).
2. Capacidad de **planificación y predictibilidad** (Plan Director de Iteraciones).
3. Capacidad de **autoformación** (Recursos de vico virtual C@mpus).
4. Saber **cooperar** y **reutilizar** el trabajo realizado dentro de un equipo de proyecto.
5. Expresión oral y escrita en base a un **vocabulario controlado** (Manuales de Estilo).

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvC vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 3 de 16
---	---------------------------------	----------------	--------------------

## Contexto del taller

Todas las metodologías asociadas a la notación UML coinciden en destacar los Casos de Uso como un artefacto central. El desarrollo de software basado en iteraciones, utiliza los elementos que configuran cada Caso de Uso para organizar un *plan de producción*. Esta relevancia no tiene correspondencia con el grado de formalismo que se supone debiera tener su especificación.

- No existe a día de hoy una plantilla normalizada que indique la manera de *especificar un flujo de eventos*.
- No hay una visión compartida para *ordenar la secuencia principal de eventos* en un Caso de Uso.
- No hay una norma para abordar la *activación de extensiones* que aportan nueva funcionalidad a un Caso de Uso.
- No disponemos, y esto quizás sea el peor problema, de unos criterios consensuados sobre *cómo aplicar una granularidad sistemática* para definir el casting de Actores y el censo de Casos de Uso principales en cada proyecto.

Sólo tenemos una certeza, ya anticipada por Cockburn hace unos años, “tu Caso de Uso no es mi Caso de Uso”.

Estas carencias provocan una *disminución del retorno de inversión* en los proyectos donde el esquema de producción está basado en Casos de Uso. Si queremos obtener aplicaciones, más fáciles de mantener, más extensibles y más reusables, no hay duda que el Caso de Uso es nuestro mejor aliado, *¿pero a qué precio?*; ¿estamos ante un problema de productividad?

Todo el esfuerzo que nos exige el modelo de Casos de Uso puede tener una *alta rentabilidad* si sabemos jugar bien:

- a. La conversión de listas interminables de requerimientos en grupos homogéneos de funciones básicas que representan todas las maneras posibles de usar un sistema a través de un *vocabulario controlado*.
- b. Averiguar los mínimos clics que un Actor ha de realizar para *obtener el mayor valor posible* del sistema.
- c. La trazabilidad entre un *contrato de servicios* (funcionalidad exigida al sistema), los elementos ordenados en una arquitectura modular que de la mejor respuesta de diseño a dicho contrato, y un código organizado y bien documentado.

- d. La **garantía de fiabilidad y calidad** basada en pruebas que certifican las precondiciones, postcondiciones y las invariantes declaradas en cada Caso de Uso.

Recientemente el modelo de Casos de Uso ha recibido un nuevo empuje desde dos frentes: la **“Programación orientada a aspectos”** (AOP) y la **“Arquitectura orientada a servicios”** (SOA). Ambos esquemas persiguen una mejor modularidad del código gracias a mecanismos de extensión de las Clases (Aspectos), que evitan la redundancia de funcionalidades repartidas en múltiples métodos (funciones transversales), y un estilo de diseño más alineado con la **cadena de valor de los procesos de negocio** que con la tecnología pura y dura. La fusión de ambos modelos promete dar mucha guerra y está por ver el grado de sinergia generará la alianza entre Casos de Uso, Aspectos y SOA.

De lo que sí estamos seguros, es que cualquier esquema de trabajo que nos ayude a identificar lo antes posible, desde los requerimientos si cabe, todas aquellas funciones y/o servicios candidatos a ser compartidas por múltiples Clases, será clave para lograr **una modularidad más eficiente** del sistema en discusión.

## Objetivos generales

Involucrar a los profesionales de un equipo de proyecto en las decisiones que han de tomar para establecer un **plan de producción de software** basado en la especificación de requerimientos, reglas y procesos de negocio con **artefactos UML**. Los contenidos y la metodología docente del taller, están orientados a que los participantes puedan comprender y aprender los procesos principales asociados a la especificación y certificación de **escenarios de usabilidad** de un sistema, dentro de un esquema de desarrollo de software centrado en Casos de Uso, en una **arquitectura orientada a servicios** (SOA) y su implementación basada en **“web services”**.

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvC vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 5 de 16
---	---------------------------------	----------------	--------------------

## Objetivos específicos

Examinar en profundidad los conceptos y los elementos esenciales que configuran las distintas **vistas de arquitectura** de los elementos de un sistema. Realizar una serie de prácticas **“hands-on”** en un escenario simplificado, con el propósito de **saber usar herramientas de modelado** para visualizar los conceptos explicados.

## Conocimientos

*Adquirir los fundamentos sobre la materia, definir los conceptos básicos, conocer las reglas de actuación dentro de un dominio, establecer un vocabulario controlado, etc. (Analogía: conocer el código de circulación).*

1. Explicar los fundamentos sobre la **notación** de los distintos modelos y elementos asociados: clases, objetos, casos de uso, estados, etc., aplicables en el estilo de diseño de una arquitectura orientada a servicios.
2. Presentar estrategias para establecer una **trazabilidad** desde los requerimientos a las Clases y el código.
3. Conocer qué criterios de **granularidad** son aplicables a los distintos elementos para obtener una mayor reusabilidad de los artefactos de modelado.
4. Conocer los **criterios de evaluación** para ponderar qué es una buena arquitectura y cuál es la mejor solución de diseño.

## Habilidades

*Adquirir pericia y destreza para realizar una actuación. Saber actuar dentro de un dominio conforme a unas reglas establecidas. (Analogía: saber conducir un vehículo determinado).*

1. Adquirir el conocimiento suficiente para **formalizar los requerimientos** con escenarios de Casos de Uso y contratos de servicios (funcionalidad esperada).
2. Adquirir pericia en la utilización de herramientas CASE para construir y administrar un **repositorio de artefactos de modelado** como esquema (framework) de producción.
3. Conocer los procedimientos básicos para **definir los Casos de Uso principales** y aplicar mecanismos de extensión trazables como “Aspectos”.

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvc vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 6 de 16
---	---------------------------------	----------------	--------------------

4. Utilizar el modelo de Casos de Uso en combinación con modelos de Testing para optimizar la **calidad y fiabilidad del código**.

## Competencias

Capacidad para realizar la actuación necesaria con garantía de calidad, gracias al conocimiento y habilidades adquiridas en diversas materias. (Analogía: transportar mercancías con el vehículo adecuado al destino requerido).

1. Saber realizar un “Casting de Actores” para **definir la cadena de valor** que los vincula con el “contrato de servicios” de un sistema en discusión.
2. Saber cuantos artefactos (Clases, Casos de Uso, Estados, etc.), son suficientes para abordar un proyecto aplicando **criterios de granularidad** sistemáticos.
3. Saber especificar un **contrato de servicios** y estimar el esfuerzo de implementación con garantías de cumplir las exigencias de funcionalidad de los Actores.
4. Saber aplicar **mecanismos de extensión** a los Casos de Uso y a las Clases para lograr una organización del código no redundante y con las mínimas dependencias posibles.
5. Saber delimitar la arquitectura de un sistema con una **modularidad eficiente** que evite riesgos de colapso y facilite un esquema de producción en paralelo.
6. Saber especificar un modelo de funcionalidad y una arquitectura orientada a servicios (SOA) **independiente de plataforma** (PIM), con los mecanismos de composición y extensión más adecuados.
7. Saber **trabajar en equipo** con unos roles diferenciados (Analista de Negocio, Arquitecto, Programador...), y utilizar un **vocabulario controlado** para compartir conocimiento.

## Dirigido a

Los profesionales que requieran asumir las capacidades de los siguientes roles en un equipo de proyecto:

- **Gestor de Conocimiento.**- Para definir las necesidades de los usuarios y especificar, como experto de dominio, un contrato de servicios con un vocabulario controlado.
- **Analista de Negocio.**- Para mejorar su práctica profesional en la formalización de requerimientos y procesos aplicando el modelo de Casos de Uso.
- **Arquitecto y Diseñador.**- Para conseguir soluciones de diseño más modulares y aplicaciones más extensibles.
- **Ingeniero de Testing.**- Para aplicar la especificación de escenarios como materia prima en la elaboración de sus Casos de Test.
- **Programador.**- Para conocer la hoja de ruta de una trazabilidad real entre los requerimientos, Casos de Uso, Clases, y su código.
- **Ingeniero de Calidad.**- Para decidir la unidad mínima de documentación sostenible en cada proyecto de desarrollo de software.
- **Jefe de Proyecto.**- Para tener una visión global de todo el proceso de desarrollo y de las responsabilidades compartidas de un equipo de profesionales.

## Precondiciones

- Conocimiento básico sobre la orientación a objetos
- Conocimiento básico sobre los procesos de negocio
- Recomendable experiencia en la participación de proyectos de desarrollo de software
- Recomendable experiencia previa en la utilización de UML
- No es necesario una experiencia previa en la utilización de herramientas CASE
- No es necesario una experiencia previa en la aplicación de metodologías de desarrollo

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvC vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 8 de 16
---	---------------------------------	----------------	--------------------

## Contenidos del taller

### 1. Fundamentos

- 1.1. Visión general de los contenidos y objetivos del taller
- 1.2. Revisión de los recursos documentales y herramientas
- 1.3. Descripción del escenario del caso práctico
- 1.4. Conceptos clave (Especificación, Arquitectura, Patrones)
- 1.5. Casos de Uso desde la perspectiva “Aspect Oriented Software Development”
- 1.6. Conceptos clave de un “Service-Oriented Architecture”
- 1.7. Elementos que participan en el modelo de Caso de Uso
- 1.8. Categorías de Casos de Uso (principales, secundarios y auxiliares)
- 1.9. Manual de estilo para especificar los escenarios de usabilidad de un sistema
- 1.10. Mecanismos de composición y criterios de granularidad de los Casos de Uso

### 2. Traducir requerimientos a Casos de Uso

- 2.1. Estrategias de casting de Actores y definición de sus responsabilidades
- 2.2. Definir servicios con el soporte de “Grupos de Casos de Uso Relacionados”
- 2.3. Identificar y especificar Casos de Uso principales
- 2.4. Especificar la secuencia principal de un Caso de Uso
- 2.5. Trazabilidad de requerimientos funcionales con Casos de Uso principales
- 2.6. Definir puntos de activación y extensión para entrelazar Casos de Uso auxiliares
- 2.7. Establecer una granularidad sistemática con Casos de Uso secundarios
- 2.8. Prototipar interfaces gráficas de usuario y esquemas de navegación
- 2.9. Aplicar una escala de abstracción variable entre Casos de Uso y Procesos
- 2.10. Framework de Procesos de Negocio y servicios

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvc vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 9 de 16
---	---------------------------------	----------------	--------------------

### 3. Traducir Casos de Uso a componentes de software

- 3.1. Identificar los objetos de negocio participantes en escenarios de los Casos de Uso
- 3.2. Realizar la transición a diagramas de Clases de Análisis
- 3.3. Mecanismos de asociación y composición para una correcta granularidad
- 3.4. Realizar la transición del análisis al diseño con la ayuda de plantillas y patrones
- 3.5. Estrategia de desarrollo basada en componentes e interfaces
- 3.6. Tipos de interfaces y estrategias en los mecanismos de herencia
- 3.7. Tipos de componentes, estereotipos y módulos
- 3.8. Estrategias de “realización” de los Casos de Uso en un IDE
- 3.9. De la función a la estructura y viceversa (Casos de Uso, Clases, Aspectos, Servicios)
- 3.10. Máquinas de estados para monitorizar ciclos de vida

### 4. Arquitectura SOA a partir de los Casos de Uso

- 4.1. Tipos de arquitectura y punto de partida “Architecture Baseline”
- 4.2. Establecer una funcionalidad y arquitectura independiente de plataforma (MDA)
- 4.3. Vistas de Casos de Uso dependientes e independientes de plataforma
- 4.4. Trazabilidad de requerimientos NO funcionales con Casos de Uso de infraestructura
- 4.5. Criterios para evaluar las decisiones de arquitectura (ATAM)
- 4.6. Cómo identificar Casos de Test a partir de los Casos de Uso
- 4.7. Elementos clave a certificar en una prueba de conformidad
- 4.8. Cómo definir precondiciones, postcondiciones e invariantes
- 4.9. Diseño e implementación de tests: Criterios de fiabilidad y conformidad
- 4.10. Aplicación de patrones (Extended Use Case Test Design Pattern)

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvc vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 10 de 16
---	---------------------------------	----------------	---------------------

## 5. Plan de producción basado en Casos de Uso

- 5.1. Metodologías ágiles e itinerarios de desarrollo para sobrevivir a más de un proyecto
- 5.2. Censo de Casos de Uso del proyecto: "Contrato de servicios"
- 5.3. Construyendo sistemas con monitores y vistas de Casos de Uso
- 5.4. Controlando las Clases y sus extensiones con Aspectos y vistas de Casos de Uso
- 5.5. Optimizando la granularidad de los Casos de Uso principales con puntos de extensión
- 5.6. Arquitectura SOA y gestión de procesos de negocio (BPM) con los Casos de Uso
- 5.7. Transición del diseño a un entorno de ejecución
- 5.8. Mecanismos de integración con "Web Services"
- 5.9. Estimación de esfuerzo y esquema de cooperación del equipo de desarrollo
- 5.10. Planificación y control del proyecto con un esquema de iteraciones

## Metodología docente

Este taller está estructurado con los siguientes formatos.-

- **Clases de teoría:** Exposición de conceptos por parte del profesor con la ayuda de recursos didácticos y recursos de software.
- **Laboratorio:** Todas las entradas del temario que empiezan con el verbo "Realizar" son prácticas orientadas a la elaboración del **repertorio de entregables** de un micro-proyecto. El profesor organiza a los participantes del taller en equipos y explica como usar las plantillas de documentos, **manuales de estilo**, patrones de análisis, patrones de Casos de Uso, patrones de diseño y arquitectura. Después de cada práctica hay una discusión sobre los resultados, **criterios de evaluación** de los entregables, revisión de procedimientos aplicables y propuestas para optimizar el trabajo en equipo por parte del profesor.

- **Actividades no presenciales:** Utilización del [Foro del Curso](#) <sup>1</sup> para publicar consultas, trabajos y recursos compartidos. Soporte de tutoría no presencial por parte del profesor.

Las definiciones y explicaciones sin más nunca son suficientes para una comprensión real de la materia de estudio. Las actividades del laboratorio están orientadas a que el alumno se enfrente a un caso práctico y colabore en su definición y solución dentro de un equipo. Nuestro propósito es facilitar un **proceso de aprendizaje** que refuerce los nuevos conceptos adquiridos y **evite los bloqueos más habituales** en la especificación de artefactos de modelado con UML.

Todo el mundo entiende y recuerda mejor aquello que ha descubierto por si mismo. Uno de los valores más atractivos del taller, consiste en **aportar una experiencia real** que ayude al alumno a desarrollar unas habilidades prácticas para diseñar e implementar un plan de producción de software basado en Casos de Uso con una metodología ágil y sostenible y dentro de un **esquema de cooperación** y conocimiento compartido.

## Recursos didácticos

### Diario de a bordo.-

- [WebLog del curso](#) con un Foro de consultas abierto a los alumnos. <sup>2</sup>

### Materiales docentes.-

- **Presentaciones:** Material docente de un temario ordenado en transparencias.
- **Fichas de conocimiento:** Resumen de los conceptos básicos de un tema ordenados en fichas hipervinculadas.
- **Guiones de prácticas:** Procedimientos y recursos de referencia ordenados para realizar las prácticas de laboratorio y facilitar el cumplimiento de su propósito didáctico.
- **Entregables:** Repertorio de documentación correspondiente al caso práctico y soluciones de las prácticas de laboratorio del curso.
- **Plantillas:** Documentos genéricos para utilizar como punto de partida en la elaboración de un entregable de proyecto.

<sup>1</sup> Esta referencia es presentada como ejemplo de utilización de un foro y otros servicios web en un curso.

<sup>2</sup> Esta referencia de WebLog es presentada como ejemplo de utilización de los servicios web en un curso.

- **Manuales de Estilo:** Alertas, recordatorios y orientaciones sobre la elaboración de artefactos de modelado y entregables de proyecto.
- **Recursos prácticos** para ampliar conocimientos a través del autoestudio.<sup>3</sup>
- **Mapa de recursos del taller en la web** con presentaciones, pósters, plantillas de entregables, patrones, herramientas CASE freeware, etc.<sup>4</sup>



## Requisitos del aula

- Red local de PC's con WinXP; =>256 Mb RAM; => 2 Gb libres de disco
- Impresora de red
- Pizarra grande
- Proyector
- Microsoft Office 2003 en adelante, OpenOffice
- Adobe Acrobat Reader 5.0 en adelante
- Enterprise Architect 7.0 en adelante

<sup>3</sup> Esta referencia de Centro de Recursos es presentada como ejemplo de utilización junto con otros servicios web en un curso.

<sup>4</sup> Muestra del mapa de recursos de un taller. Está sujeto a novedades y actualizaciones.

## Método de evaluación

VICO expedirá un certificado de asistencia y aprovechamiento del taller a los alumnos con un porcentaje de asistencia superior al 75 % de las sesiones y que hayan realizado todas las prácticas de manera presencial, o no presencial, a través del **Foro del taller** en el campus virtual.

## Referencias

[Adolph et al. 2003] Steve Adolph, Paul Bramble. **Patterns for Effective Use Cases**, Addison-Wesley, Boston, 2003.

[Ambler, 2005] Scott W. Ambler. **The Elements of UML 2.0 Style**, Cambridge University Press, Cambridge, 2005.

[Armour et al. 2001] Frank Armour, Granville Miller. **Advanced Use Case Modeling – Software Systems**, Addison-Wesley, Boston, 2001.

**AspectJ Homepage.** [URL:http://www.eclipse.org/aspectj/](http://www.eclipse.org/aspectj/)

[Bittner et al. 2003] Kurt Bittner, Ian Spence. **Use Case Modeling**, Addison-Wesley, Boston, 2003.

[Cockburn, 2001] Alistair Cockburn. **Writing Effective Use Cases**, Addison-Wesley, Boston, 2001.

**DAJ Homepage.** [URL:http://sourceforge.net/projects/daj/](http://sourceforge.net/projects/daj/)

[Denney, 2005] Richard Denney. **Succeeding with Use Cases – Working Smart to Deliver Quality**, Addison-Wesley, Boston, 2005.

[Jacobson et al. 2005] Ivar Jacobson, Pan-Wei Ng. **Aspect-Oriented Software Development with Use Cases**, Addison-Wesley, Boston, 2005.

[Jacobson, 2003] Ivar Jacobson, “**Use cases and Aspects - Working together**”. Invited Talk at AOSD 2003, Boston, USA.

[Kulak et al. 2000] Daryl Kulak, Eamonn Guiney. **Use Cases – Requirements in Context**, Addison-Wesley, Boston, 2000.

Dir.: L:\_sincroMentor\_TRAD CD Borrador\_Proyectos\_CarteraProjectesRecents\_proyecto vvC vico virtual campus\_Catálogo de Talleres\_CursosNuevoFormato\CU_SOA_TRAD_GuionCurso_abierto.	Fecha act.: 29/10/2007 17:18	Revisión: 8	Página: 14 de 16
---	---------------------------------	----------------	---------------------

## Mapping Use Case Level Aspects to ObjectTeams/Java.

<http://trese.cs.utwente.nl/workshops/oopsla-early-aspects-2004/Papers/HerrmannEtAl.pdf>

[Newcomer et al. 2005] Eric Newcomer, Greg Lemow. **Understanding SOA with Web Services**, Addison-Wesley, Boston, 2005.

[Rosenberg et al. 1999] Doug Rosenberg, Kendall Scott. **Use Case Driven Object Modeling with UML**, Addison-Wesley, Boston, 1999.

[Rosenberg et al. 2001] Doug Rosenberg, Kendall Scott. **Applying Use Case Driven Object Modeling with UML – An Annotated e-Commerce Example**, Addison-Wesley, Boston, 2001.

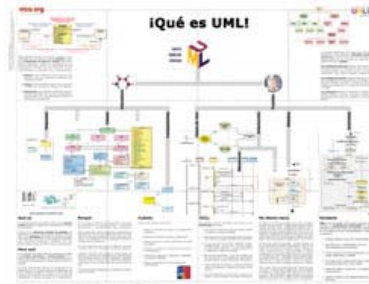
[Schneider et al. 1998] Geri Schneider, Jason P. Winters. **Applying Use Cases – A Practical Guide**, Addison-Wesley, Boston, 1998.

## Use Case Modularity using Aspects.

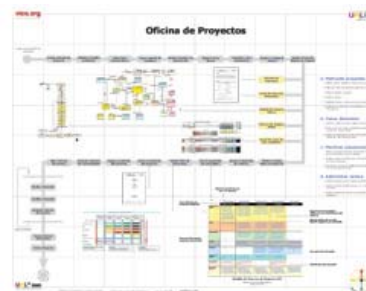
<http://www.ccs.neu.edu/research/demeter/posters/ecoop2003/PosterAbstract.doc>

[Vilalta, 2005] Josep Vilalta. **Patrón ZenO y HOX de Casos de Uso & Patrón ERPAC de objetos de negocio esenciales**, <http://www.vico.org>

Fundamentos de la notación visual UML.



Flujo de trabajo de la metodología UML-TRAD.



Repertorio de entregables predefinido por la plantilla 3PA, una vez ajustada la escala del proyecto objetivo y nuestro nivel de documentación sostenible para su elaboración.



Diseño de un "Contrato de servicios" basado en grupos relacionados de Casos de Uso candidatos y un "Plan de Producción" basado en un esquema de iteraciones para certificar su progresiva realización.

