Una Introducción al UML

El Modelo de Componentes

Autor: Geoffrey Sparks, Sparx Systems, Australia

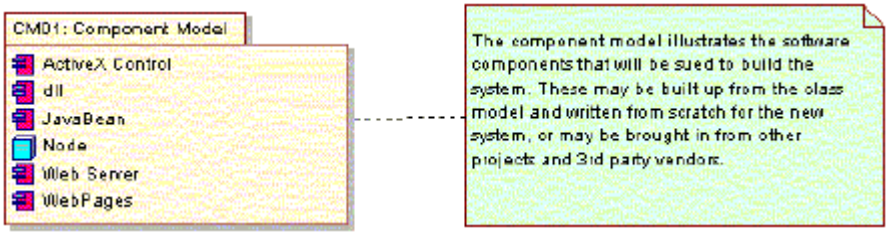
Traducción: Fernando Pinciroli (Solus S.A., Argentina) y Aleksandar Orlic (Craftware Consultores Ltda., Chile)

www.sparxsystems.com.ar - www.sparxsystems.cl

Tabla de Contenidos

El Modelo de Componentes

Este artículo describe cómo modelar los componentes de software y hardware en UML. El modelo de componentes ilustra los componentes de software que se usarán para construir el sistema. Se pueden construir a partir del modelo de clases y escribir desde cero para el nuevo sistema o se pueden importar de otros proyectos y de productos de terceros. Los componentes son agregaciones de alto nivel de las piezas de software más pequeñas y proveen un enfoque de construcción de bloques de "caja negra" para la elaboración de software.



Introducción al UML

El Lenguaje Unificado de Modelado (UML) es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software. El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos.

El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. El UML también soporta la idea de extensiones personalizadas a través elementos estereotipados.

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo.

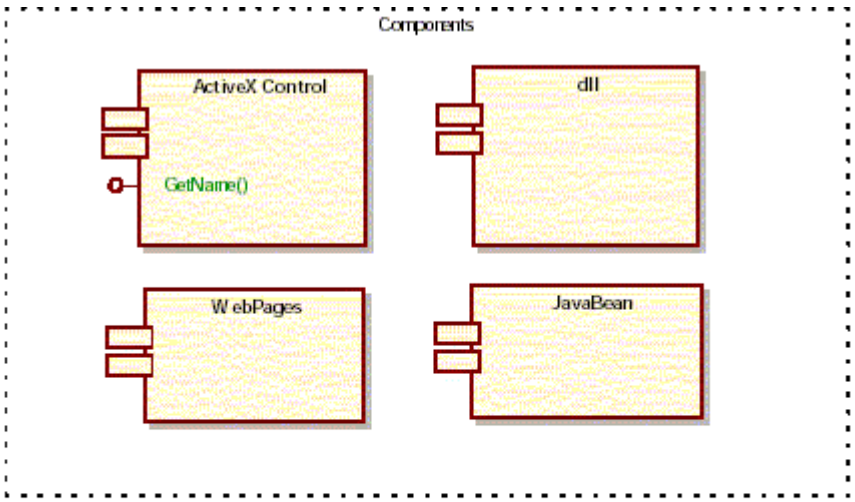
En los libros mencionados en la sección de lectura recomendada se puede encontrar más información sobre el UML y de los documentos de especificación del UML que se pueden encontrar en las paginas de recursos de UML del OMG (*Object Management Group*) www.omg.org/technology/uml/ y www.omg.org/technology/documents/formal.

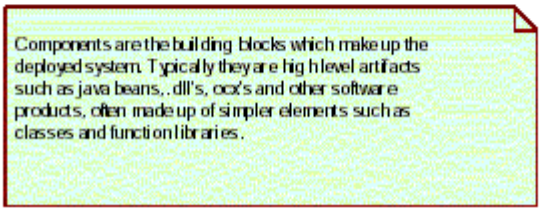
La Notación de Componentes

Un componente puede ser algo como un control Actives; tanto un componente de la interfaz de usuario como un servidor de reglas de negocio. Los componentes se representan gráficamente como muestra la figura siguiente:

El Diagrama de Componentes

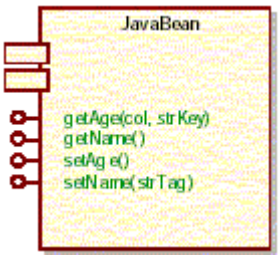
El diagrama de componentes muestra la relación entre componentes de software, sus dependencias, su comunicación su ubicación y otras condiciones.





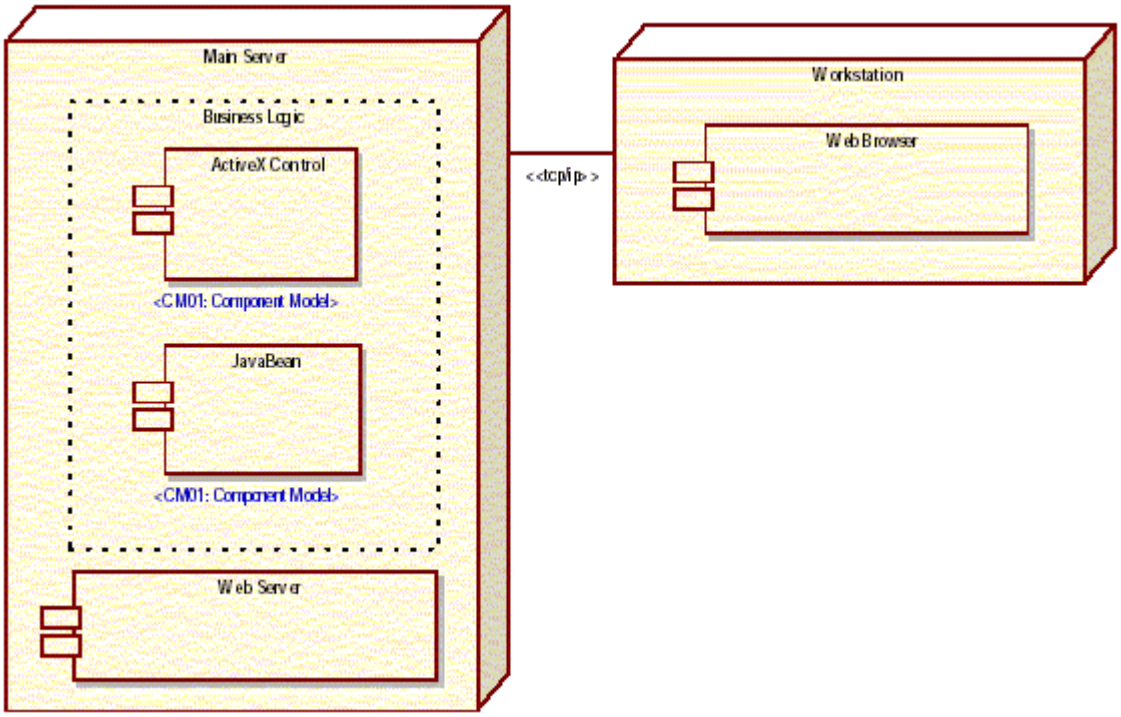
Interfaces

Los componentes también pueden exponer las interfaces. Estas son los puntos visibles de entrada o los servicios que un componente está ofreciendo y dejando disponibles a otros componentes de software y clases. Típicamente, un componente está compuesto por numerosas clases y paquetes de clases internos. También se puede crear a partir de una colección de componentes más pequeños.



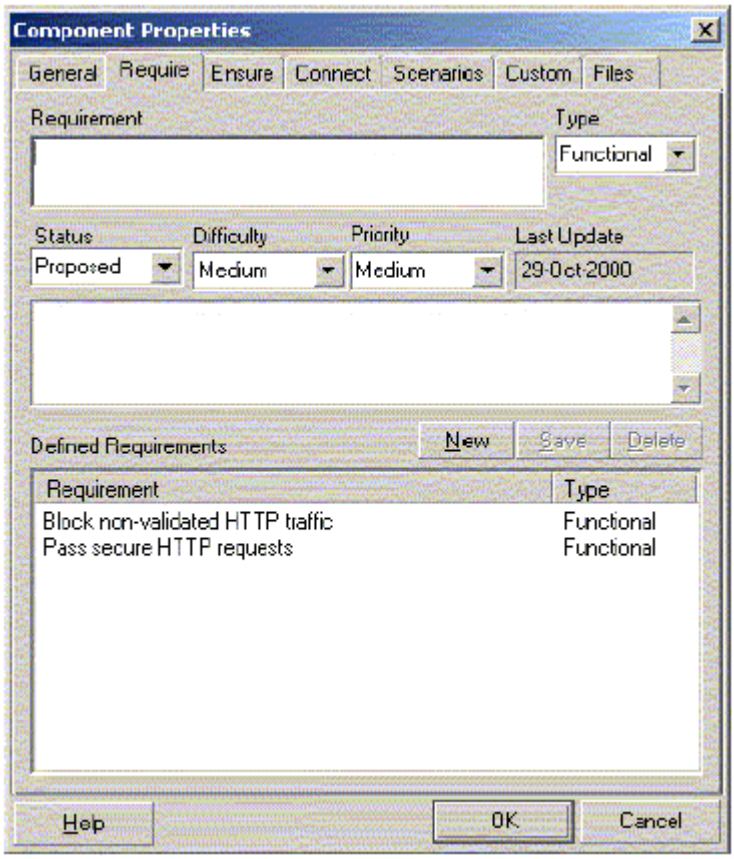
Los componentes y los Nodos

Un diagrama de despliegue muestra el despliegue físico del sistema en un ambiente de producción (o de prueba). Muestra dónde se ubican los componentes, en qué servidores, máquinas o hardware. Puede representar los enlaces de redes, el ancho de banda de la LAN, etc.



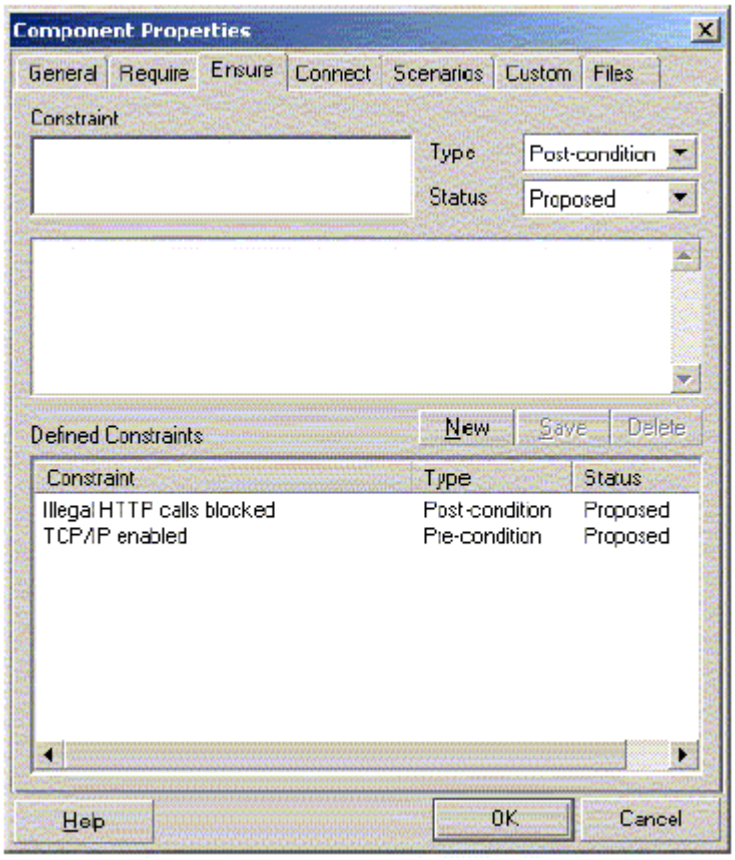
Requisitos

Los componentes pueden tener requisitos adjuntos para indicar sus obligaciones contractuales; esto es, qué servicios proveen en el modelo. Los requisitos ayudan a documentar el comportamiento funcional de los elementos de software.



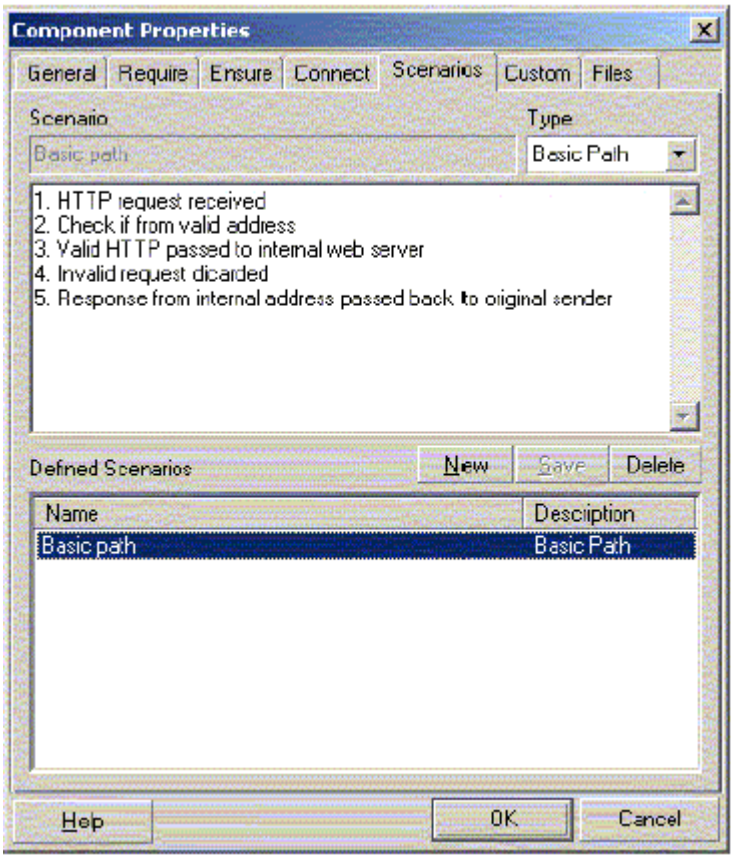
Restricciones

Los componentes pueden restricciones asignadas que indican el entorno en el que operan. Las pre-condiciones especifican lo que debe ser verdadero antes de que un componente pueda realizar alguna función; las post-condiciones indican lo que debe ser verdadero después de que un componente haya realizado algún trabajo y los invariantes especifican lo que debe permanecer verdadero durante la vida del componente.



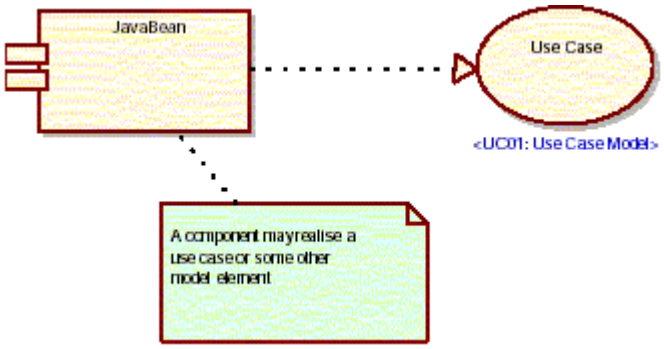
Escenarios

Los escenarios son descripciones textuales y procedimentales de las acciones de un objeto a lo largo del tiempo y describen la forma en la que un componente trabaja. Se pueden crear múltiples escenarios para describir tanto el camino básico (una ejecución perfecta) como las excepciones, errores y otras condiciones.



Trazabilidad

Puede indicar la trazabilidad por medio de vínculos de realización. Un componente puede implementar otro elemento del modelo (por ejemplo un caso de uso) o un componente puede ser implementado por otro elemento (por ejemplo un paquete de clases). Al emplear las relaciones de realización desde y hacia los componentes, se pueden seguir las dependencias entre los elementos del modelo y la trazabilidad desde los requisitos iniciales hasta la implementación final.

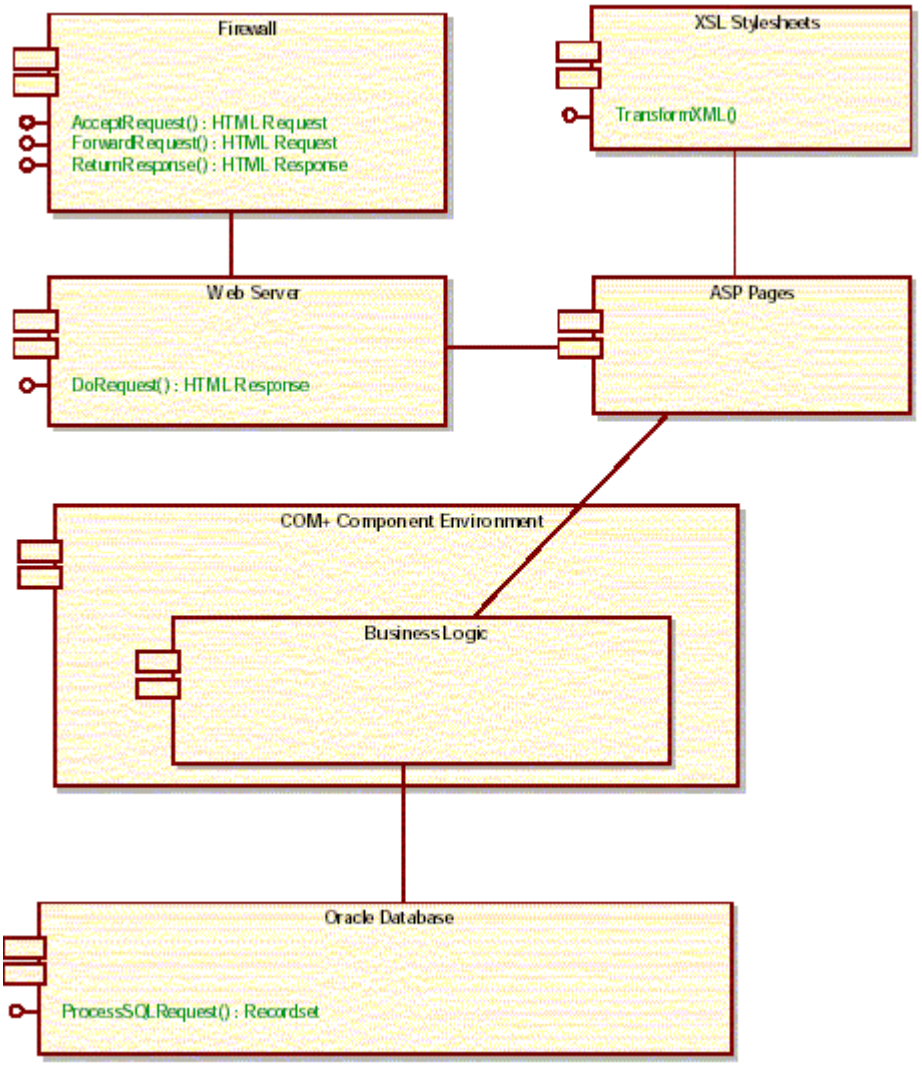


Un Ejemplo

El ejemplo siguiente muestra cómo se pueden relacionar los componentes para proveer una vista conceptual/lógica de la construcción de un sistema. Este ejemplo representa los elementos del servidor y la seguridad de una tienda de libros en línea. Se incluyen elementos tales como el servidor WEB, el firewall, las páginas ASP, etc.

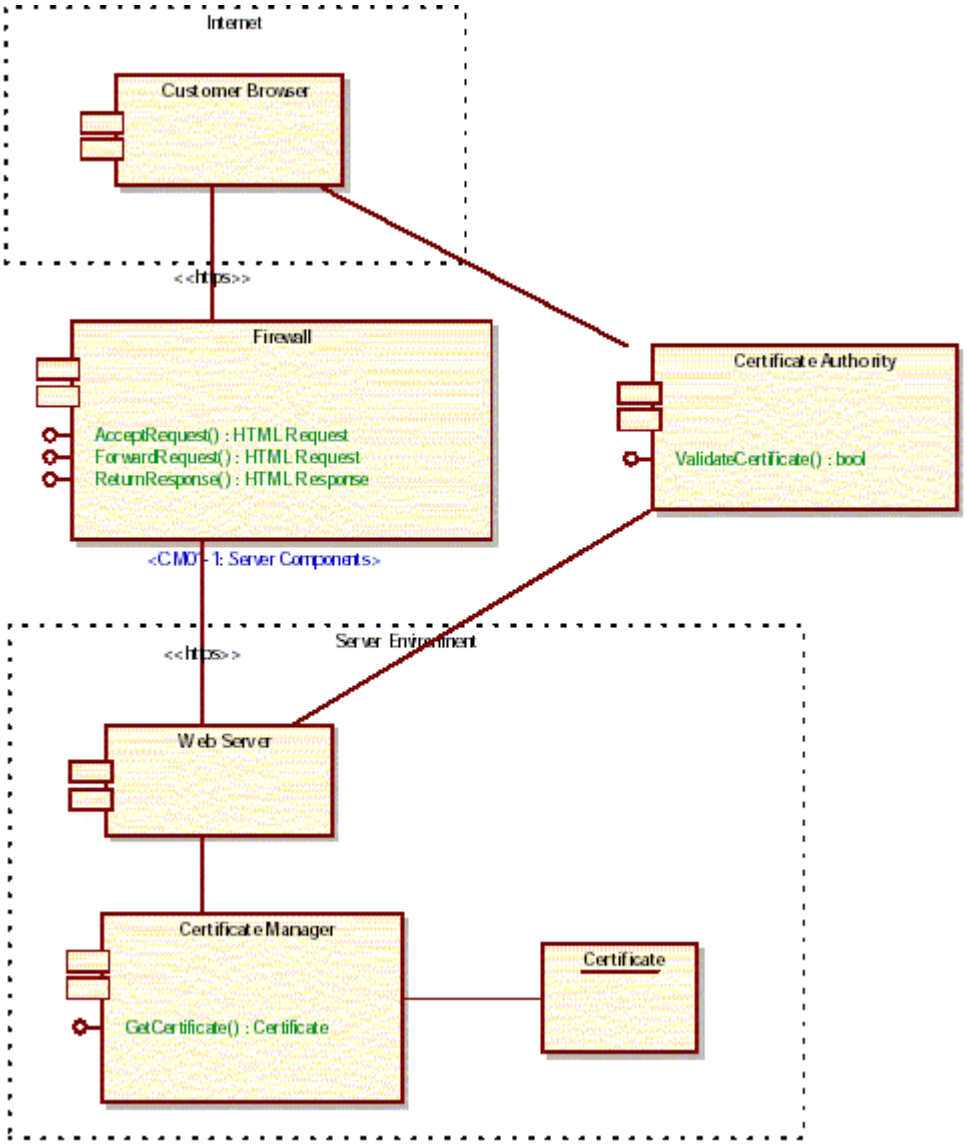
Los Componentes de Servidor

Este diagrama ilustra la organización de los componentes del lado del servidor principal que se requerirá construir para una tienda de libros en línea. Estos componentes son una mezcla de los ítems construidos a medida y adquiridos que se ensamblarán para proveer la funcionalidad requerida.



Los Componentes de Seguridad

El diagrama de componentes de la seguridad muestra cómo trabaja en conjunto el software de seguridad, tal como la Autoridad Certificadora (*Certificate Authority*), el navegador (*Browser*), el servidor WEB y otros elementos del modelo para asegurar la provisión de la seguridad en el sistema propuesto.


Internet
Customer Browser
<<https>>
Firewall
AcceptRequest() : HTMLRequest
ForwardRequest() : HTMLRequest
ReturnResponse() : HTMLResponse
Certificate Authority
ValidateCertificate() : bool
<CMD - 1: Server Components>
<<https>>
Server Environment
Web Server
Certificate Manager
GetCertificate() : Certificate
Certificate

Lectura Recomendada

Sinan Si Alhir, **UML in a NutShel**.
ISBN: 1-56592-448-7. Publisher: O'Reilly & Associates, Inc

Doug Rosenberg with Kendall Scott , Component Driven Object Modeling with UML
ISBN: 0-201-43289-7. Publisher: Addison-Wesley

Geri Scheider, Jason P. Winters, **Applying Componentes**
ISBN: 0-201-30981-5. Publisher: Addison-Wesley

Ivar Jacobson, Martin Griss, Patrik Jonsson, **Software Reuse**
ISBN: 0-201-92476-5. Publisher: Addison-Wesley

Hans-Erik Eriksson, Magnus Penker, **Business Modeling with UML**
ISBN: 0-471-29551-5. Publisher: John Wiley & Son, Inc

Peter Herzum, Oliver Sims, **Business Component Factory**
ISBN: 0-471-32760-3 Publisher: John Wiley & Son, Inc