

Una Introducción al UML

El Modelo Dinámico

Autor: Geoffrey Sparks, Sparx Systems, Australia

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

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

Tabla de Contenidos

TABLA DE CONTENIDOS	2
EL MODELO DINÁMICO	3
INTRODUCCIÓN AL UML	3
LOS DIAGRAMAS DE SECUENCIA.....	3
LOS DIAGRAMAS DE ACTIVIDAD	5
LAS CARTAS DE ESTADOS	9
RESUMEN	10
LECTURA RECOMENDADA	11

El Modelo Dinámico

Este artículo describe cómo modelar los aspectos dinámicos de los sistemas de software usando la notación y la semántica del UML. Los tres temas que se cubren son los diagramas de secuencia, los diagramas de actividad y las cartas de estados. Se da una explicación de cada uno de ellos y de cómo calzan en la estructura de modelo completo.

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.

Este artículo se enfoca en el modelado del comportamiento dinámico usando la notación y la semántica del UML. La interacción dinámica y el comportamiento se dividen en tres categorías principales:

1. Las interacciones entre las instancias de los objetos en tiempo de ejecución. Se modelan usando los diagramas de secuencia o los diagramas de colaboración. Este artículo sólo discutirá los diagramas de secuencia, por el hecho de que los diagramas de colaboración y secuencias son semánticamente idénticos.
2. Las descripciones de actividades generales cubriendo el proceso de negocio y la interacción del usuario. Los diagramas de actividad y los de proceso de negocio se usan para estos propósitos.
3. Cambios de estado a lo largo del tiempo. El UML soporta las cartas de estados para modelar los cambios de estado.

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 páginas de recursos de UML del OMG (*Object Management Group*) www.omg.org/technology/uml/ y www.omg.org/technology/documents/formal.

Los diagramas de Secuencia

Propósito

Los diagramas de secuencia se usan para mostrar la interacción entre los usuarios, las pantallas y las instancias de los objetos en el sistema. Proveen un mapa secuencial del paso de los mensajes entre los objetos a lo largo del tiempo. Frecuentemente, estos diagramas se

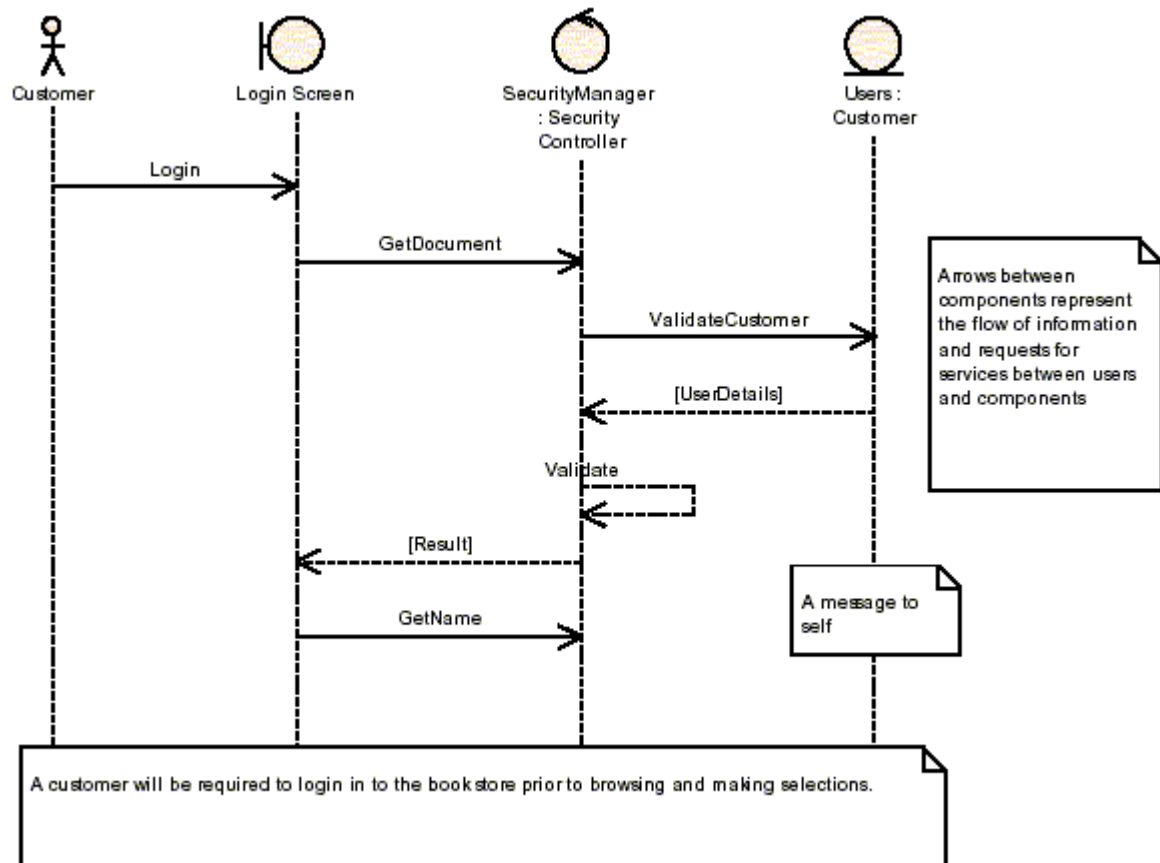
ubican bajo los casos de uso o los componentes en el modelo para ilustrar un escenario, un conjunto de pasos comunes que se siguen en respuesta a un evento externo y que genera un resultado. El modelo incluye qué inicia la actividad en el sistema, qué procesamiento y cambios ocurren internamente y qué salidas se generan. Muchas veces las instancias de los objetos se representan usando íconos especialmente estereotipados; existen íconos para objetos de interfaz (*Boundary*), controladores (*Controllers*) y entidades persistentes (*Entity*).

Notación

La notación que se emplea es típicamente un conjunto de actores y objetos desplegados horizontalmente, teniendo cada uno de ellos una línea de vida vertical. Los mensajes (usualmente las llamadas de los métodos, pero también pueden representar los mensajes que se pasan usando servicios de colas de mensajes y otros eventos) se dibujan de un objeto hacia otro con una flecha indicando el sentido del flujo.

Un diagrama de secuencia es la representación de los mensajes que se pasan entre las instancias de los objetos, por lo que generalmente estos mensajes se transformarán en las operaciones de las clases durante el proceso de diseño del sistema. Los diagramas de secuencia iniciales indican qué comportamiento público se necesita para que los objetos cumplan con el trabajo y cooperen, y durante el diseño, los diagramas de secuencia se usan para mostrar qué operaciones y responsabilidades reales se asignan a qué clases.

El diagrama de ejemplo a continuación muestra algunas características de los diagramas de secuencia:



Tenga en cuenta el uso de los íconos estereotipados para mostrar los objetos particulares: por ejemplo la interfaz de usuario (*Login Screen*) se muestra con un estereotipo interfaz (*Boundary*) y el usuario como un estereotipo entidad (*Entity*). Ellos ayudan a diferenciar visualmente los roles de los objetos durante el análisis.

Los Diagramas de Actividad

Propósito

Los diagramas de actividad se usan para mostrar cómo se construyen los diferentes flujos de trabajo o los procesos dentro de un sistema, cómo se inician, los variados caminos alternativos que se pueden tomar desde el inicio hasta el fin y dónde puede ocurrir el procesamiento paralelo durante la ejecución.

Un diagrama de actividades generalmente no modela el comportamiento exacto de un sistema de software (como lo hace un diagrama de secuencia), sino los procesos y los flujos a un muy alto nivel. Las actividades generalmente serán realizadas por uno o más casos de

uso; la actividad describe el proceso que se desarrolla y tanto el caso de uso como un actor usará el sistema para realizar toda o parte de una actividad.

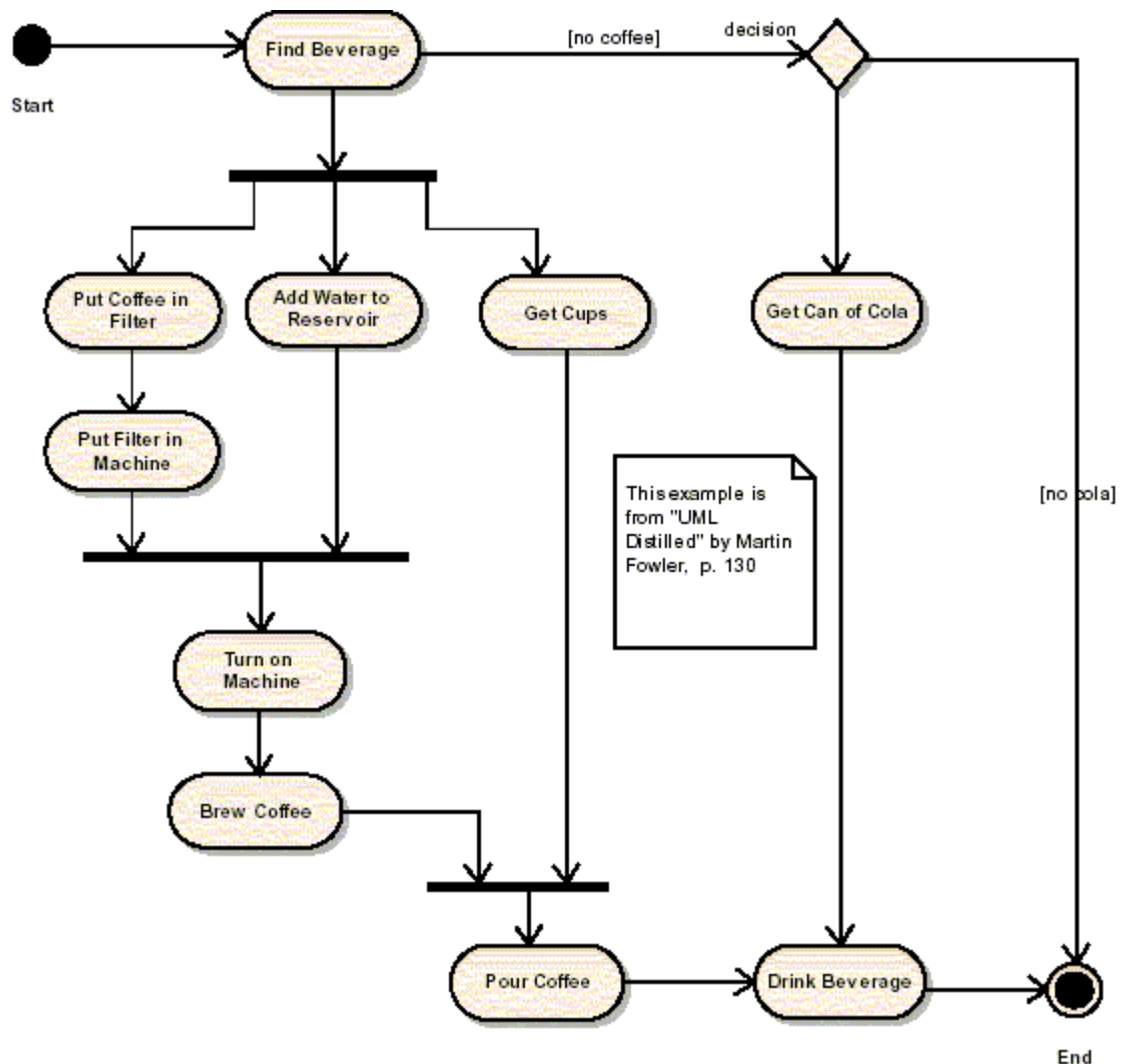
Notación

La Notación del Diagrama de Actividad

La notación estándar del UML emplea un rectángulo con las esquinas redondeadas para representar una actividad. Las actividades pueden ser acompañadas por flujos de procesos o eventos. Adicionalmente, el nodo de decisión puede modelar el comportamiento divergente basado en una condición. Típicamente se definen los nodos de Inicio y de Fin para una representación completa de la actividad. También se pueden definir los puntos de sincronización para ilustrar cómo se puede llevar a cabo el procesamiento en paralelo y cómo se sincroniza en un punto antes de proceder con las actividades siguientes.

El ejemplo de más abajo muestra la mayoría de estos puntos; describe el proceso vinculado a la adquisición de una bebida de una máquina expendedora. Los rectángulos redondeados son las actividades, los diamantes son los puntos de decisión y las barras negras horizontales son los puntos de sincronización.

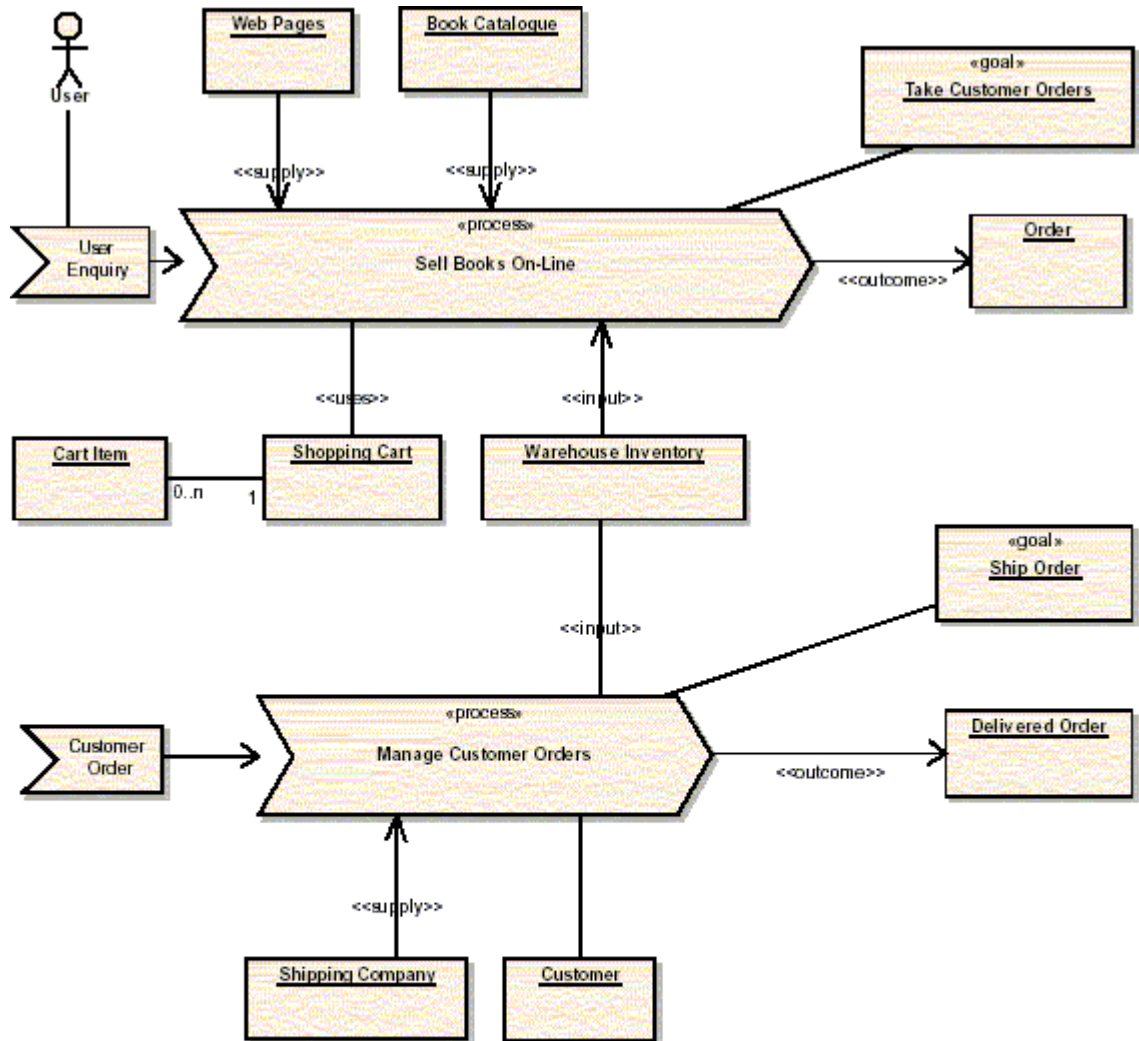
Tenga en cuenta que si Ud. estuviera construyendo el software para la máquina expendedora, serían relevantes solamente algunas de las actividades, aunque el diagrama completo provee una buena imagen de lo que abarca el proceso total de obtener una bebida. Sería necesario un análisis más profundo para determinar qué partes de este modelo se implementarían o soportaría el software.



La Notación de Proceso de Negocio

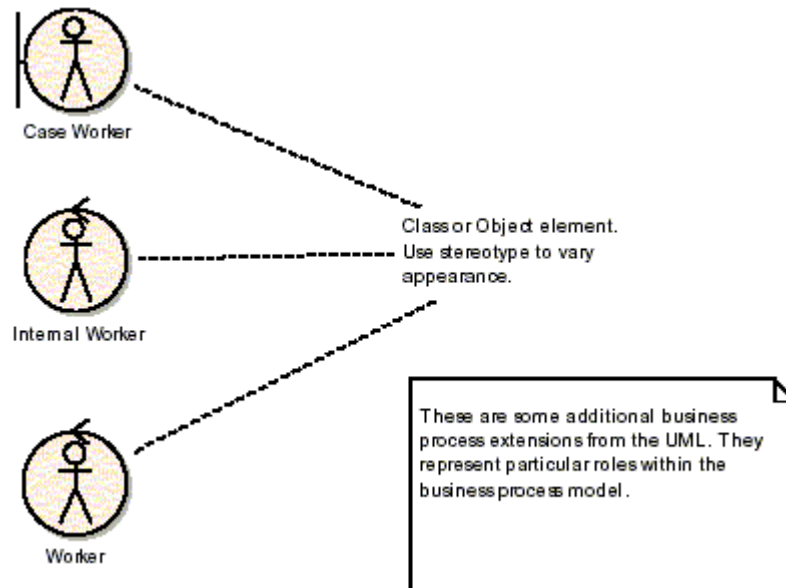
Adicionalmente a como se describe en el UML la notación estándar de los diagramas de Actividad, se definieron varias extensiones para retratar más apropiadamente los procesos de negocio. En particular, las extensiones propuestas por Hans-Erik Eriksson y Magnus Penker (ver las lecturas recomendadas) proveen una manera de enfocar este aspecto del análisis de sistemas.

Estas extensiones proveen más estereotipos, tales como el proceso de negocio (representado como una flecha rectangular alargada) y los objetos de objetivos, entrada y salida. El ejemplo de más abajo muestra cómo se pueden combinar estos objetos para producir una vista de alto nivel de la actividad de negocio.



The business process model lets you model the main business activities. A 'Process' can cut across many departments or divisions of a larger entity. It describes something a business does as part of its normal activity; focusing mainly on the inputs, outputs, goals and key events that drive the process.

El UML también define algunos íconos estereotipados adicionales para utilizar en el modelado de procesos de negocio. El ejemplo de más abajo muestra algunos de ellos. Vea cómo estos íconos reutilizan los íconos de interfaz (*Boundary*), controlador (*Controller*) y entidad (*Entity*) de los diagramas de secuencias.



Las Cartas de Estados

Propósito

Las cartas de estados son el tercer modelo dinámico que usa el UML para capturar los cambios del sistema a lo largo del tiempo. Cualquier objeto que no mantenga constantes las variables de instancia durante el tiempo de ejecución tiene algún estado potencial. El estado actual de un objeto depende de los valores de sus variables de instancia. Típicamente, las cartas de estados se asocian con clases particulares (muchas veces una clase puede tener una o más cartas de estados para describir completamente sus estados potenciales).

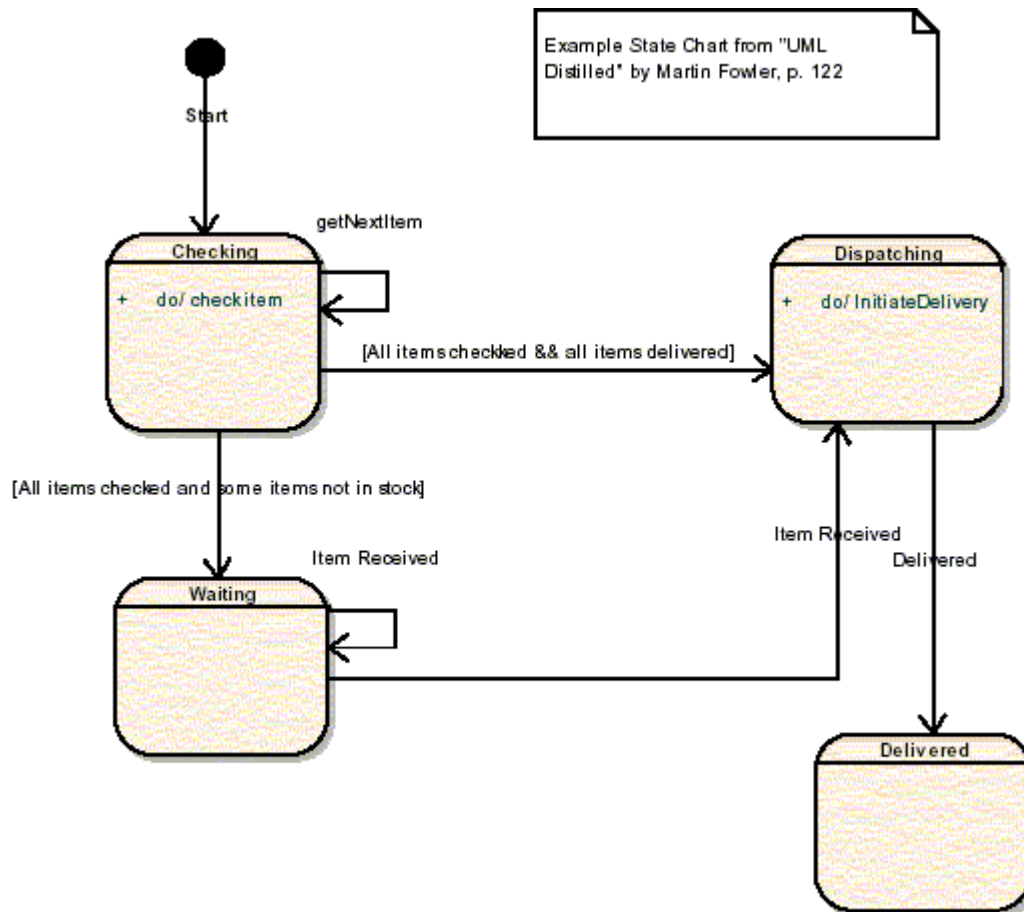
En el UML, un estado se representa como un rectángulo redondeado, con compartimientos opcionales para atributos, eventos y actividades internas. Los flujos de estados y reglas de transición se dibujan entre los estados, usualmente con condiciones de guarda y reglas regulando cómo y cuándo un objeto puede tener una transición de un estado a otro.

El uso cuidadoso de las cartas de estados ayudará a revelar las variables que se requieren para mantener completamente los estados de un objeto y las precondiciones necesarias para realizar la transición (actualizar las variables de instancia) a otro estado.

Los estados usualmente se nombran de acuerdo a su condición; por ejemplo “Controlando”, “Esperando” y “Despachando” son todas las condiciones activas en las que puede estar un objeto mientras espera la transición a otro estado o finaliza su ciclo completamente.

Los nodos de Inicio y Fin, que se representan como círculos rellenos y vacíos, se emplean para representar el inicio y el fin de todas las transiciones.

Notación de Ejemplo



Resumen

El UML provee un buen soporte para el modelado dinámico de los sistemas de software, desde la fase de análisis preliminar (diagramas de actividades y modelado de los procesos de negocio) a través de la funcionalidad de los Casos de Uso (Diagramas de Secuencia) hasta el comportamiento de las clases (Cartas de Estados). Cada tipo de diagrama ayuda a capturar la información sobre el sistema como un todo y refinar en profundidad los detalles del diseño y la implementación necesarios para completar el sistema de software.

Lectura Recomendada

Sinan Si Alhir, **UML in a NutShel**.

ISBN: 1-56592-448-7. Publisher: O'Reilly & Associates, Inc

Doug Rosenberg with Kendall Scott , Dynamic Driven Object Modeling with UML

ISBN:0-201-43289-7. Publisher: Addison-Wesley

Geri Scheider, Jason P. Winters, **Applying Dinamics**

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