

Capítulo 6

Estudio de Caso para la adopción de los Modelos de Trazado

Este capítulo presenta un estudio de caso que muestra paso a paso la validación del Patrón de Trazabilidad, los Modelos de Trazado y el modelo de transformación que se definen en los capítulos anteriores de esta disertación. Para lograr esto se desarrollan los siguientes objetivos: (1) definir por medio de algoritmos declarativos las operaciones más relevantes del Patrón de Trazabilidad, (2) definir un estudio de caso para visualizar la forma de trabajar con los Modelos de Trazado, (3) definir un método para soportar el uso de los modelos de trazado y los servicios de trazabilidad que pueden soportar la transformación de modelos desde los Requisitos, a través del análisis y hasta la arquitectura de diseño. El método se valida con el estudio de caso Sistema de Subastas.

6.1 Introducción

El grupo de desarrollado crea los Modelos de Trazado con base en las características y elementos del Patrón de Trazabilidad y las necesidades de garantizar la consistencia y la completitud de los modelos de desarrollo. También, este grupo toma decisiones de modelado con información que las instancias de los Modelos de Trazado generan durante las transformaciones de modelos.

El Patrón de Trazabilidad está diseñado para establecer un marco de trabajo conceptual y metodológico que proporciona las operaciones necesarias para realizar actividades tales como la definición de Modelos de Trazado, el control de la creación de modelos fuente de la transformación, la generación de modelos destino por medio del planificador de transformaciones y la gestión de los cambios que suceden durante el proceso de desarrollo.

El Patrón de trazabilidad se ha implementado usando el marco de trabajo KERMETA con el fin de experimentar el uso de esta herramienta y su utilidad para definir metamodelos y modelos para la transformación. KERMETA es compatible con el EMF [Steinberg et al. 2008] y usa la herramienta Eclipse para editar, almacenar y visualizar modelos. Además, extiende al *Essential Meta Object Facility* (EMOF 2.0) con un lenguaje que permite definir la estructura y el comportamiento asociado con las clases definidas en el nivel del metamodelo. EMOF 2.0 es un subconjunto del MOF cuya meta es definir metamodelos simples usando conceptos simples de la orientación a objetos [EMOF 2006]. El modelo EMOF utiliza paquetes UML 2.0 e incluye capacidades adicionales al lenguaje que facilitan la generación de modelos en diferentes formatos (en el Apéndice B se hace la implementación del patrón).

Las operaciones definidas en las metaclases que conforman el Patrón de Trazabilidad son descritas en lenguaje natural de forma declarativa. Esto se hace con el fin de establecer la lógica básica de funcionamiento que el marco de trabajo ofrece para usar las metaclases de configuración. Estas operaciones posteriormente se pueden implementar en cualquier lenguaje de programación.

El método de trazabilidad propuesto tiene el objetivo de orientar el uso de los Modelos de Trazado. Está compuesto por un conjunto de pasos que se explican por medio del desarrollo del caso de estudio para el Sistema de Subastas.

Este capítulo está estructurado como se define a continuación. La Sección [6.2](#) define el método de trazabilidad que facilita el uso de los Modelos de Trazado. La Sección [6.3](#) concluye el capítulo.

6.2 Método de Trazabilidad

El método de trazabilidad es un conjunto de pasos que definen la aplicación de los Modelos de Trazado como controladores de las transformaciones de los modelos en el contexto de MDD. Además facilita la gestión de configuración en servicios como: la verificación de la consistencia y la completitud, y el manejo del cambio. La Figura 77 ilustra el flujo de cada uno de los pasos necesarios para usar los Modelos de Trazado (ver la descripción de problema en el Apéndice C).

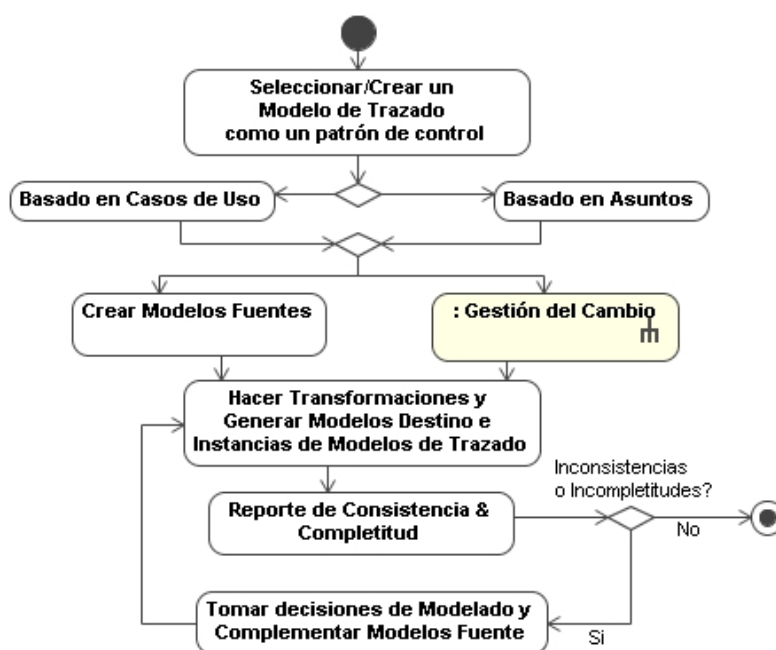


Figura 77. Método de Trazabilidad para soportar el uso de los Modelos de Trazado en la transformación de modelos.

Con el objetivo de simular la forma como el marco conceptual de los Modelos de Trazado se puede gestionar desde una herramienta de modelado. Un modelo de trazado se crea en una herramienta CASE como un artefacto de trazabilidad para el control y ejecución de la transformación.

La Figura 78(a) ilustra un prototipo general que describe el nivel de los directorios que se crean por defecto al crear un proyecto de desarrollo de software. Este facilita la definición de los Modelo de Trazado desde el Patrón de Trazabilidad, la separación de modelos de desarrollo en tres nivel de abstracción (*Requisitos*, *Análisis*, y *Diseño*), y la ejecución de las transformaciones de estos modelos para generar los modelos destino y las vistas versionadas de los Modelos de Trazado.

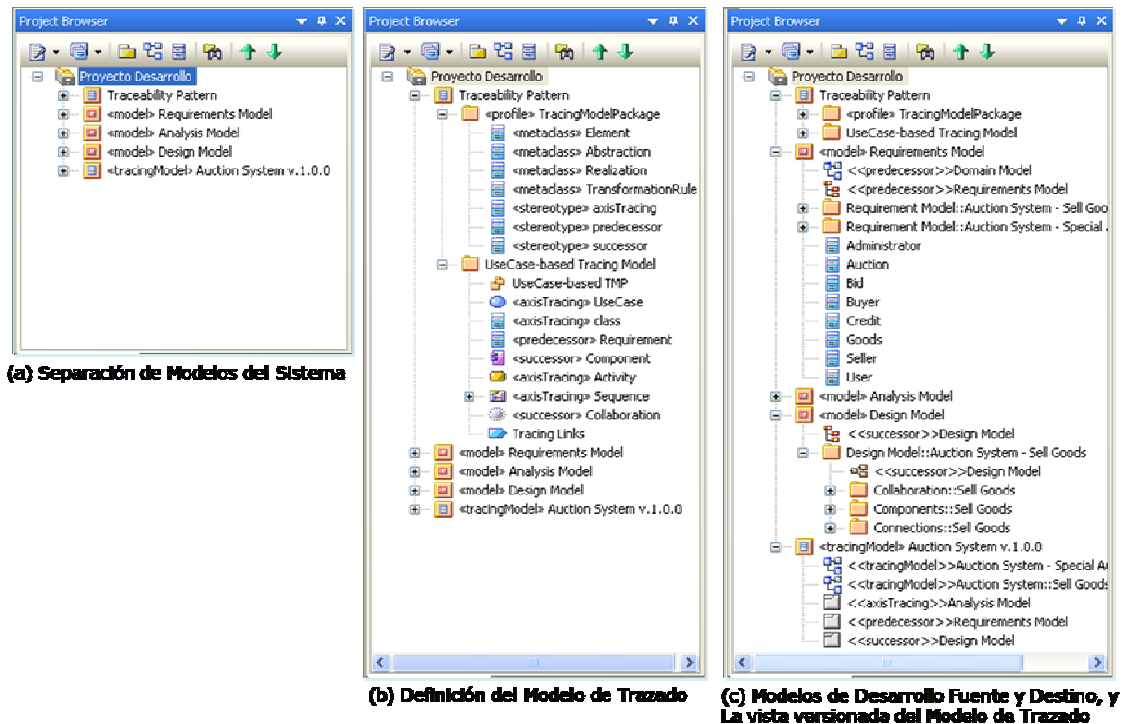


Figura 78. Prototipo inicial de los modelos de configuración del marco conceptual de los Modelos de Trazado.

El Patrón de Trazabilidad (*Traceability Pattern*). Es un directorio de definición que incluye el perfil UML en el cual se definen los elementos básicos que pueden ser usados en la definición de los Modelos de Trazado. Además, se definen los Modelos de Trazado y sus elementos que posteriormente serán usados para controlar la transformación de los modelos de desarrollo. La Figura 78(b) ilustra el detalle de la definición del perfil, el detalle de la definición del Modelo de Trazado basado en Casos de Uso el cual guiará la trazabilidad y transformación de los modelos de desarrollo.

Los Modelos de Desarrollo. Para este caso específico los modelos de desarrollo se definen o transforman para el Auction System. La Figura 78(c) ilustra los elementos de modelo definidos al nivel del Modelo de Requisitos (*Requirements Model*). De igual forma para el Modelo de Análisis (*Analysis Model*) y los el Modelo de Diseño (*Design Model*) se genera un sistema de subdirectorios y elementos de modelo que son creados manualmente o generados a partir de un proceso de transformación.

La vista versionada del Modelo de Trazado. Cuando se hace el proceso de creación y transformación de elementos de modelo se generan las vistas versionadas del Modelo de Trazado.

En este caso, la Figura 78(c) ilustra el directorio <<tracingModel>> *Auction System v.1.0.0* en el cual se crean las vistas de los elementos de modelo creados o generados durante la transformación con cada uno de los roles (<<axisTracing>>, <<predecessor>>, <<successor>>) que juegan en la instancia versionada del Modelo de Trazado base.

6.2.1 Seleccionar/Crear Modelo de Trazado como un Patrón de Control

Una vez que el proceso de desarrollo de software y el tipo de proyecto son seleccionados por el equipo de desarrollo, ellos crean o seleccionan un modelo de trazado después de atender las sugerencias definidas en la Sección 3.4.3.

Aunque el patrón facilite la creación de diferentes tipos de Modelos de Trazado para soportar las actividades de gestión de la configuración, el método de trazabilidad es probado con dos definiciones genéricas: (1) el Modelo de Trazado basado en Casos de Uso (ver la Figura 26), y (2) el Modelo de Trazado basado en Asuntos (este Modelo de Trazado se define en el Capítulo 5).

Para el Modelo de Trazado basado en Casos de Uso, algunos equipos de desarrollo prefieren representar los requisitos en casos de uso, en el nivel de Requisitos. Entonces, el cambio de la declaración de elementos en el modelo de trazado es de la siguiente forma: en el nivel de Requisitos el elemento `Requirement` es sustituido por el elemento `UseCase`, y en el elemento `UseCase` del nivel del Análisis es sustituido por el elemento `RealizationUseCase` (`Collaboration`). El vínculo de trazado se define de la siguiente forma: <<refine>> (<<axisTracing>>Analysis::UseCaseRealization::name, <<Requirement>>Requirement::UseCase::name).

De la Tabla 3 a la Tabla 7 se muestra la declaración de los elementos del modelo de trazado i.e, elementos trazables, vínculos de trazado, secuencias de traza, y secuencias implícitas. El modelo de transformación es definido en el Capítulo 4.

6.2.2 Crear Modelos Fuente

Para controlar tareas de modelado por medio de un modelo de trazado basado en casos de uso se sugiere que el equipo cree modelos de desarrollo una vez adopten una técnica o método sugerido para un proceso de desarrollo de software como el Proceso de desarrollo Unificado.

Los requisitos del Sistema de Subasta son definidos en el apéndice C. La Figura 79 muestra el modelo de requisitos para cuatro objetivos: *Enroll*, *Authentication*, *Sell Goods* and *Buy Goods*. En este nivel de abstracción cada objetivo es representado por un `Package` el cual agrupa sus

requisitos y otros elementos de modelo definidos para este nivel. Por ejemplo, actividades y clases del modelo del dominio. La Figura 80 muestra el modelo del dominio.

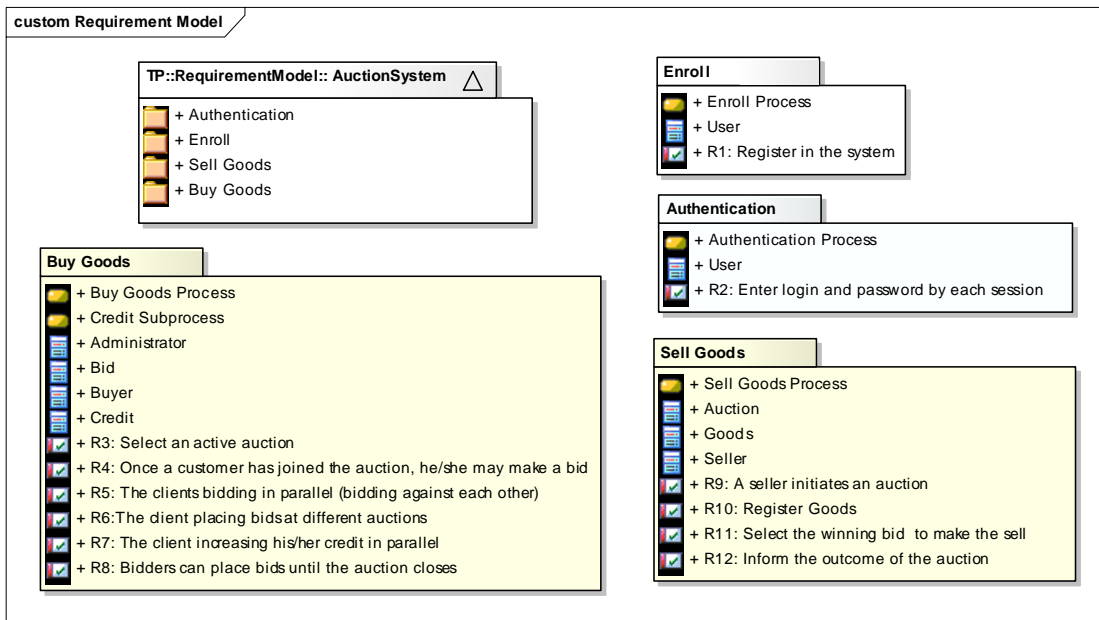


Figura 79. Una vista del Modelo de Requisitos para el Auction System.

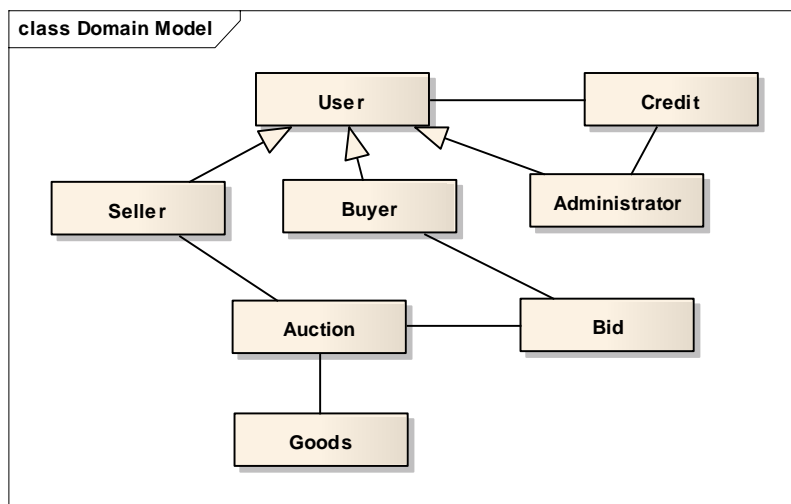


Figura 80. Modelo de Requisitos: Modelo del Dominio para Auction System.

Para facilitar el ejercicio este se desarrolla con el objetivo (subsistema) *Sell Goods* cuyo proceso de negocio se ilustra en la Figura 81.

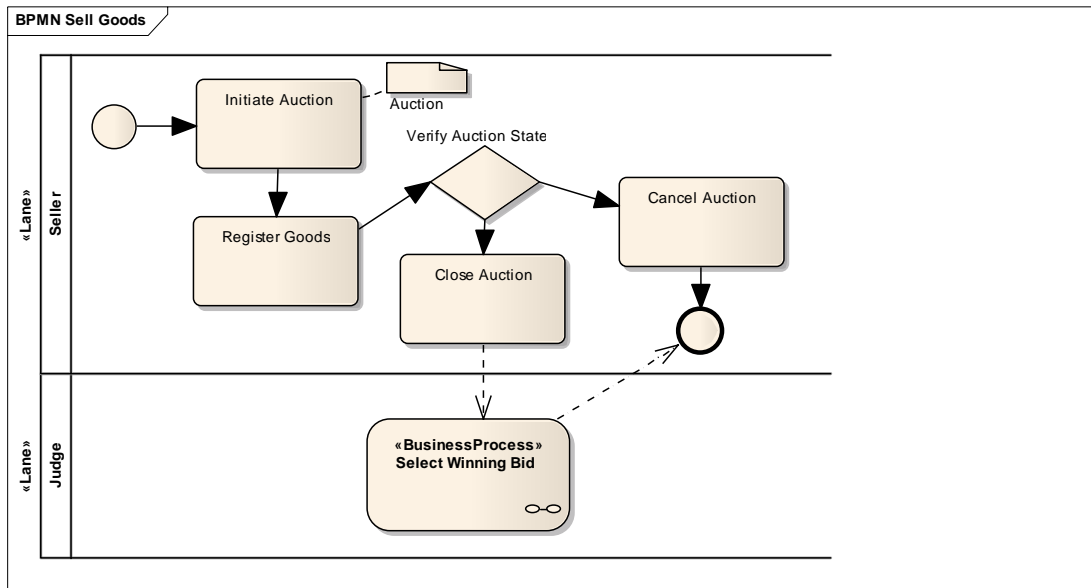


Figura 81. Modelo de Requisitos: Proceso del negocio *Sell Goods*.

6.2.3 Hacer Transformaciones de Modelos y Generar las instancias del Modelo de Trazado

En este caso, los elementos modelos son creados en el nivel de Requisitos lo cual significa que los elementos trazables serán instancias de elementos declarados como ejes de trazado y sucesores. Por ejemplo, la Figura 82 muestra los requisitos, las clases, y el proceso del negocio relacionados con el subsistema *Sell Goods* definido en el nivel de Requisitos e identificado en el modelo de trazado con el estereotipo <<predecesor>>.

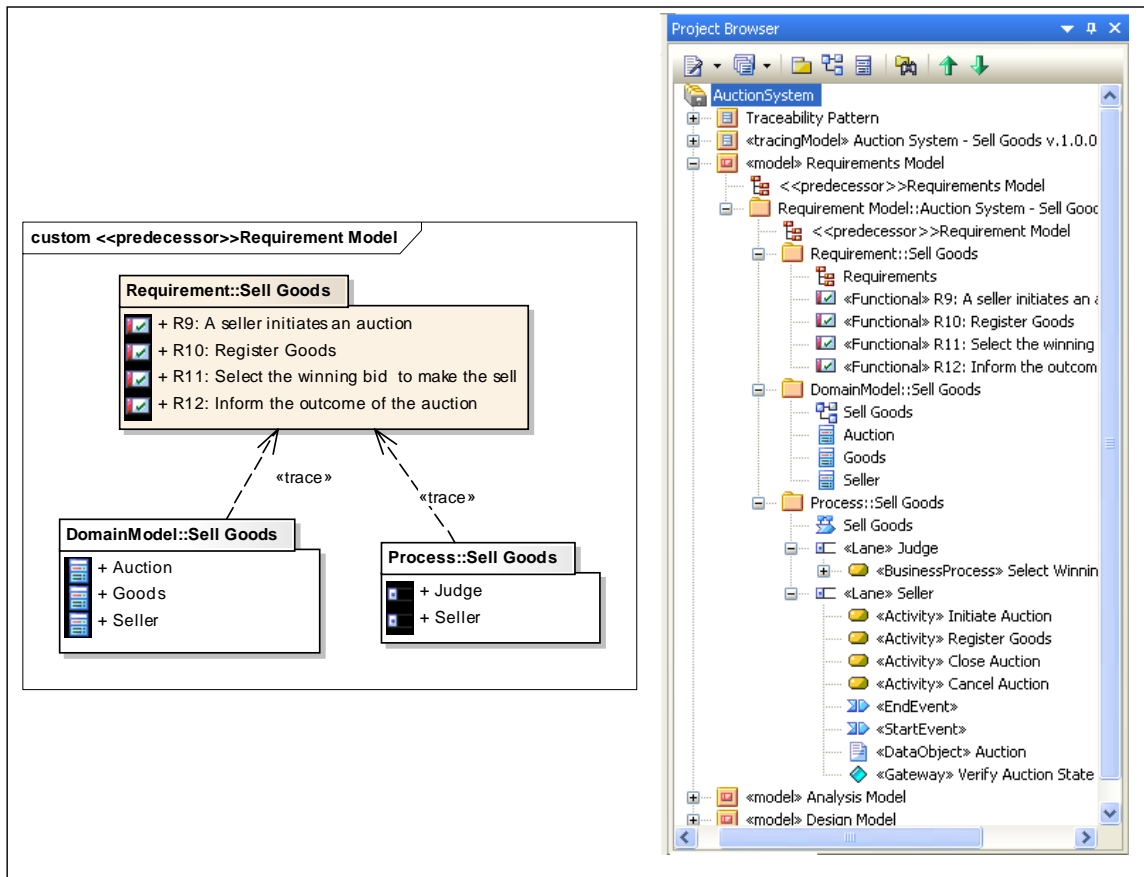


Figura 82. Definición del modelo fuente en el nivel de Requisitos. Este se muestra como una instancia de predecesores definidos para modelo de trazado del Auction System – Sell Goods.

De la Tabla 24 a la Tabla 27 se muestran las instancias de los vínculos de trazado y las secuencias de traza que proveen información al planeador de la transformación para ejecutar las reglas sobre el subsistema *Sell Goods*. Para cada vínculo de trazado se asocia una regla tipo *Root* con la cual se hace la transformación y se activa la tarea del planeador (ver definición en el Capítulo 4).

Tabla 24. Algunas instancias de los vínculos de trazado activadas por la definición del objetivo Sell Goods en el nivel de Requisitos. Además, la secuencia de reglas de transformación ejecutada por el planeador de transformación.

Tracing Link	Transformation Scheduler
<pre><<refine>> (<<axisTracing>>Analysis:: Package::Sell Goods, <<predecessor>> Requirement:: Package:: Sell Goods)</pre>	For each Requirement do <ol style="list-style-type: none"> 1. Rule 1 (Root)→ Requirement2UseCase 2. Rule 2 (Subordinate)→ Entity2Actor 3. Rule 3 (Subordinate)→ RelRequirement2RelUC 4. Rule 4 (Complementary)→ Requirement2SpeUC{basicFlow}
<pre><<refine>> (<<successor>>Design:: Package:: Sell Goods, <<axisTracing >> Analysis:: Package:: Sell Goods)</pre>	
<pre><<refine>> (<<axisTracing>>Analysis:: UseCase:: Sell Goods:: A seller initiates an auction,</pre>	

<pre><<predecessor>> Requirement:: Requirement:: Sell Goods:: A seller initiates an auction)</pre>	5. Rule 5 (Alternative)→ nRequirement2UseCase
<pre><<refine>> (<<axisTracing>>Analysis:: UseCase:: Sell Goods:: Register Goods, <<predecessor>> Requirement:: Requirement:: Sell Goods:: Register Goods)</pre>	
<pre><<refine>> (<<axisTracing>>Analysis:: UseCase:: Sell Goods:: Select the winning bid to make the sell, <<predecessor>> Requirement:: Requirement:: Sell Goods:: Select the winning bid to make the sell)</pre>	
<pre><<refine>> (<<axisTracing>>Analysis:: UseCase:: Sell Goods:: Inform the outcome of the auction, <<predecessor>> Requirement:: Requirement:: Sell Goods:: Inform the outcome of the auction) ...</pre>	

Tabla 25. Instancias de Secuencias de Traza que complementan la consistencia y completitud del proceso de negocio Sell Goods. Además, la secuencia de reglas de transformación ejecutada por el planeador de transformación.

Trace Sequence	Transformation Scheduler
<pre>Sequence 2: <<refine>> Analysis::UseCase:: Sell Goods, Requirement::Activity(Business Process):: Sell Goods) Sequence 3: <<refine>> (<<axisTracing>>Analysis::Activity:: Sell Goods, <<predecessor>>Requirement:: Activity(BusinessProcess):: Sell Goods) Implicit Traces(sequence 3): <<refine>> (<<axisTracing>>Analysis::Activity::Lane:: Sell Goods Jugde, <<axisTracing>>Requirement::ActivityBP::Swimline:: Goods Jugde) <<refine>> (<<axisTracing>>Analysis::Activity::Action:: Sell Goods Initiate Auction, <<axisTracing>>Requirement::ActivityBP::ActivityBP:: Goods Initiate Auction) ...</pre>	<pre>For each ActivityBP do 6. Rule 7 (Root)→ ActivityBP2UseCase 7. Rule 8 (Subordinate)→ ActivityBP2Activity 8. Rule 9 (Complementary)→ LaneBP2Swimline 9. Rule 10 (Complementary)→ ActivityBP2Action 10. Rule 11 (Complementary)→ ControlFlowBP2ControlFlow 11. Rule 12 (Complementary)→ ControlFlowBP2ControlFlow 12. ...</pre>

Tabla 26. Secuencia de Traza complementada con trazas implícitas para el modelo del dominio Sell Goods. Además, la secuencia de reglas de transformación ejecutada por el planeador de transformación.

Trace Sequence	Transformation Scheduler
<pre>Sequences 5 (from Sequence 1): * <<refine>> (<<axisTracing>>Analysis::Class:: Auction, <<predecessor>>Requirement:: Class(DM):: Auction) * <<refine>> (<<axisTracing>>Analysis::Class:: Goods, <<predecessor>>Requirement:: Class(DM):: Goods) * <<refine>> (<<axisTracing>>Analysis::Class:: Seller,</pre>	<pre>For each Class do 1. Rule 14 (Root)→ DomainModel2ClassModel 2. Rule 15 (Subordinate)→ Class- DM2Class-CM 3. Rule 16 (Complementary)→ Operation-DM2Operation-CM 4. Rule 17 (Complementary)→ Propertie-DM2Propertie-CM</pre>

<pre> <<predecessor>>Requirement:: Class (DM) :: Seller) <u>Implicit Traces(sequence 4):</u> * <<refine>> (<<axisTracing>>Analysis::Class::Operation: : Auction operationname, <<axisTracing>>Requirement::Class (DM) ::Oper ation:: Auction operationname) * <<refine>> (<<axisTracing>>Analysis::Class::Operation: : Goods operationname, <<axisTracing>>Requirement::Class (DM) ::Oper ation:: Goods operationname) * <<refine>> (<<axisTracing>>Analysis::Class::Operation: : Seller operationname, <<axisTracing>>Requirement::Class (DM) ::Oper ation:: Seller operationname) * <<refine>> (<<axisTracing>>Analysis::Class::Propertie: : Goods propertiename, <<axisTracing>>Requirement::Class (DM) ::Oper ation:: Goods propertiename) ... </pre>	<p>5. Rule 18 (Complementary)→ Relationship-DMBP2Relationship-CM 6. ...</p>
<pre> Sequence 9: <<realize>> (<<successor>>Design:: Component:: Sell Goods, <<axisTracing>> Analysis:: ClassModel:: Sell Goods:: Register Goods) Sequence 11: <<realize>> (<<successor>>Design:: Interface:: Auction, <<axisTracing>> Analysis:: Class:: Auction) <u>Implicit Traces(sequence 10):</u> <<realize>> (<<successor>>Design:: Interface::Operation:: Auction operationname, <<axisTracing>> Analysis:: Class::Operation:: Auction operationname) </pre>	<p>7. Rule 26 (subordinate)→ ClassModel2Component For each class 8. Rule 27 (subordinate)→ Class2Interface 9. Rule 28 (complementary)→ OperationCM2OperationI 10. ...</p>

Tabla 27. Transformación (AxisTracing2Successor) del modelo Sell Goods UseCase.

Tracing Link	Transformation Scheduler
<pre> <<realize>> (<<successor>>Design::Collaboration:: Sell Goods, <<axisTracing>>Analysis::UseCase:: Sell Goods) ... </pre>	<pre> For each UseCase do 1. Rule 20 (Root)→ UseCase2Collaboration </pre>

La Figura 83 ilustra, desde la vista de modelo de trazado, las instancias de los elementos destino que se transformaron desde el modelo de Requisitos al modelo de Análisis por medio de la regla tipo *Root* Requirement2UseCase. Estas instancias en la vista de modelo de trazado se definen con el estereotipo <<axisTracing>>.

La Figura 84 complementa la vista de la instancia de modelo de trazado ilustrando como sería el vínculo de trazado entre el UseCase::Sell Goods y la Activity::Sell Goods, la cual a su vez agrupa otro conjunto de elementos que serán trazados durante el proceso de refinamiento.

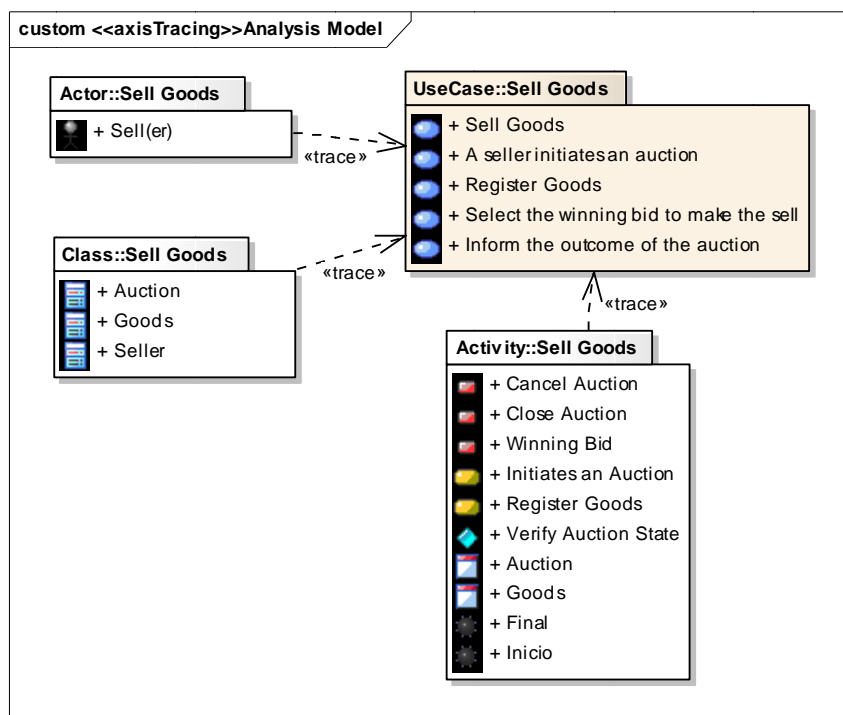


Figura 83. Una vista del modelo de trazado una vez es generado el Modelo de Análisis del subsistema *Sell Goods*.

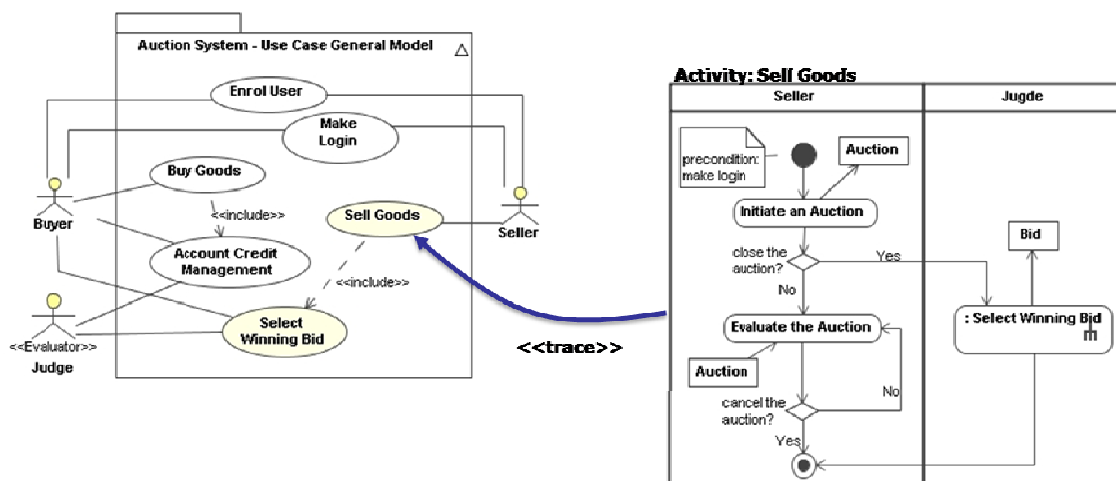


Figura 84. Detalle del elemento *Activity::Sell Goods* que traza al *UseCase::Sell Goods* en el nivel del Análisis.

La Figura 85 ilustra una alternativa de transformación de los modelos destino del subsistema *Sell Goods* en el nivel de Análisis, por medio de la regla tipo *Alternative nRequirement2UseCase*. Esta regla activa regla tipo *Complementary nRequirement2SpecUC{basicFlow}* donde cada requisito se transforma en un paso del flujo básico del caso de uso *Sell Goods*.

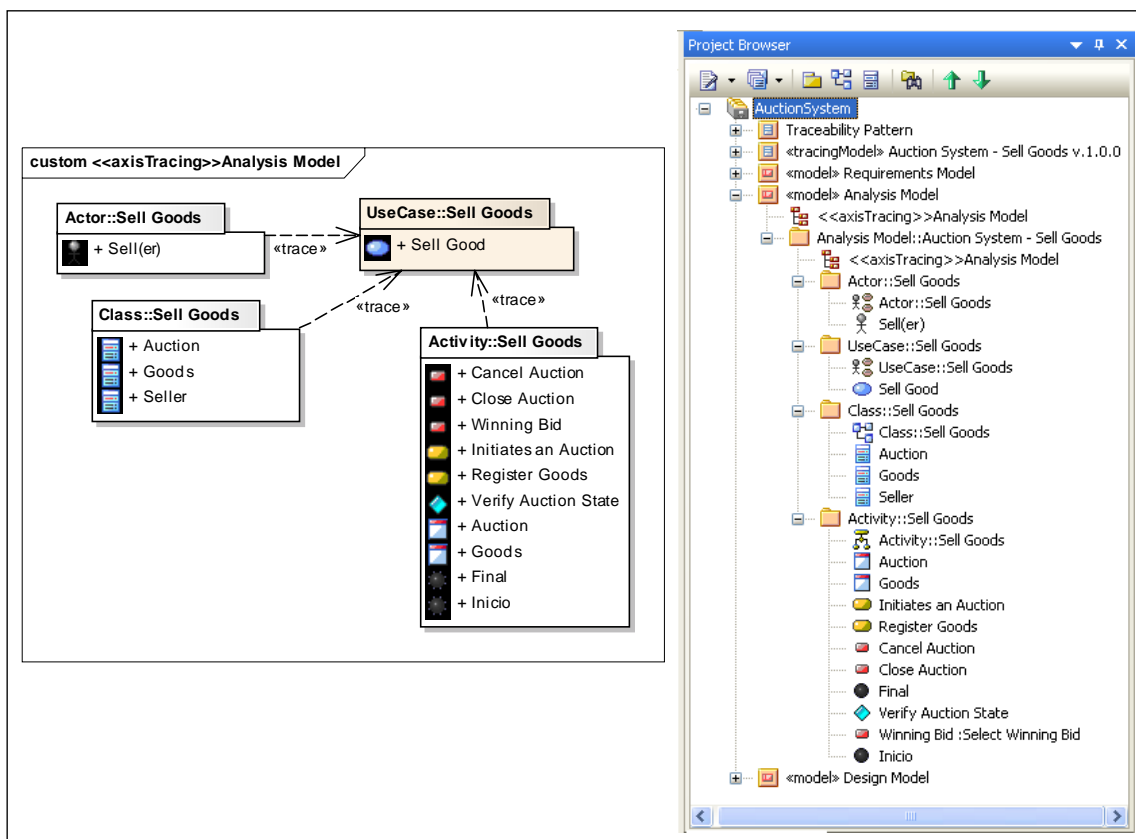


Figura 85. Alternativa de Transformación. Una vista del modelo de trazado una vez es generado el Modelo de Análisis del objetivo *Sell Goods*.

La Figura 86 ilustra el modelo de Diseño para el subsistema *Sell Goods* desde la vista de modelo de trazado cuyos elementos se identifican con el rol <<successor>>. Las instancias se logran al ejecutar la regla tipo Root UseCase2Collaboration y la cadena de reglas asociada al vínculo de trazado <<realize>> (Design::Collaboration, Analysis::UseCase).

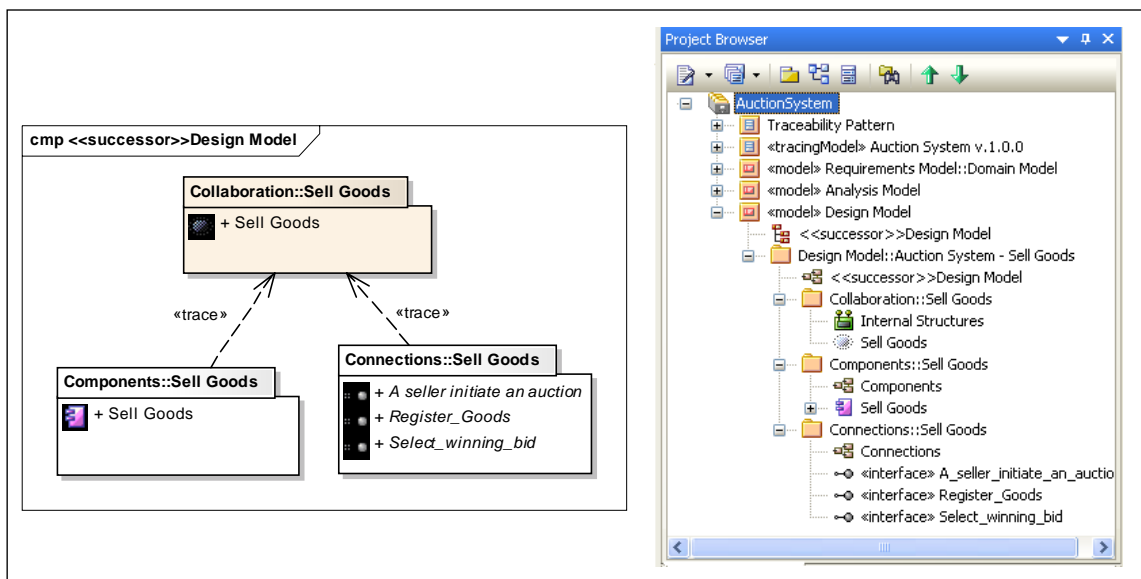


Figura 86. Modelo de Diseño generado por la transformación del objetivo *Sell Goods* desde el nivel de los Requisitos.

La Figura 87 ilustra una vista general de la instancia del modelo de trazado para el *Auction System – Sell Goods, version 1.0.0*.

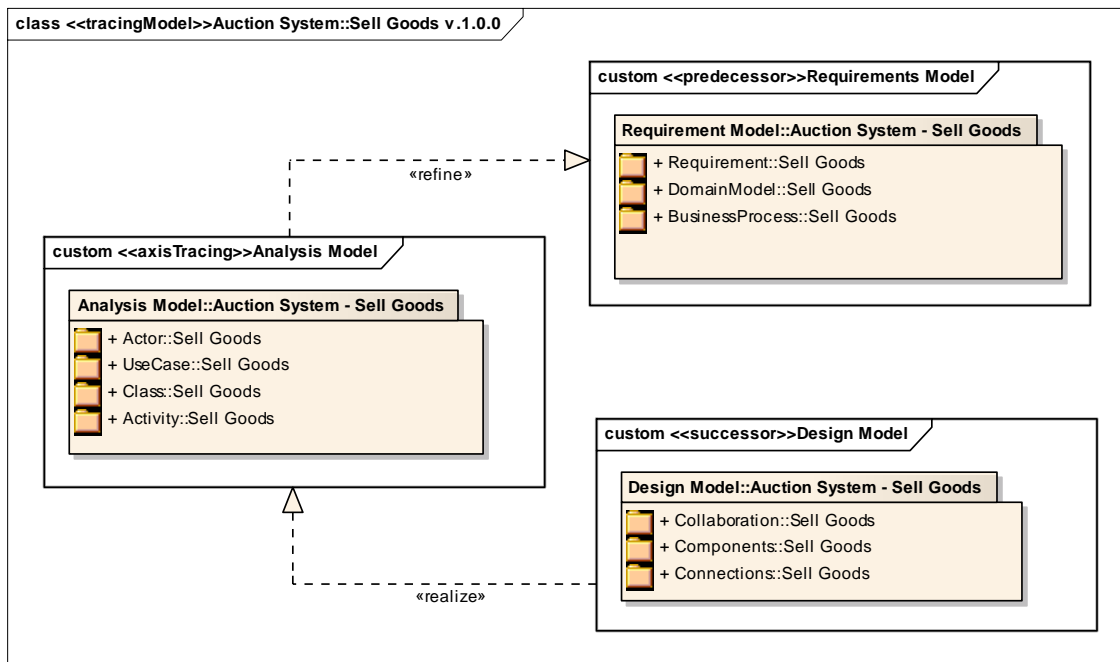


Figura 87. Una vista de modelo de trazado para el sistema *Auction System, subsistema Sell Goods, version 1.0.0*.

6.2.4 Reporte de Consistencia y Completitud

Una vez generados los modelos destino y las vistas de modelo de trazado, se genera el reporte de consistencia y completitud. La Figura 88 muestra un ejemplo del reporte de consistencia y completitud durante la transformación de los elementos de modelo del subsistema Sell Goods. En este caso se reportan dos inconsistencias que comúnmente se producen por transformaciones que no tienen instancia del modelo fuente sino que se logró por medio las secuencias de trazado implícitas al modelo.

Elemento de Modelo	Consistencia/Completitud	Estado
Requirement Model::Requirement:: Sell Good:: R9: A seller initiates an auction	- Requirement:: Domain Model::Class: Sell Good Auction - Requirement:: Domain Model::Class:: Sell Good Seller - Requirement:: Activity(Business Process)::Lane:: Sell Good Seller - Requirement::Activity(Business Process):: Activity::Sell Goods initiates an auction - Requirement::Activity(Business Process):: dataObject:: Sell Goods Auction ...	Consistency
Requirement Model::Requirement:: Sell Good:: R10: Register Goods	- Requirement:: Domain Model:: Class:: Sell Good Goods - Requirement:: Activity(Business Process)::Lane:: Sell Good Seller - Requirement::Activity(Business Process):: Activity::Sell Goods Register Goods	Consistency
Requirement Model::Requirement:: Sell Good:: R11: Select a winning bid to make the sell	- Requirement:: Domain Model:: Class: Sell Good Auction - Requirement:: Activity(Business Process)::Lane:: Sell Good Judge - Requirement::Activity(Business Process):: Activity::Sell Goods Select a winning bid to make the sell.	Consistency
	- Requirement::Activity(Business Process):: Gateway::Sell Goods Verify Auction State. - Requirement::Activity(Business Process):: Activity::Sell Goods Cancel Auction. - Requirement::Activity(Business Process):: Activity::Sell Goods Closs Auction.	Warning - Inconsistent
Requirement Model::Requirement:: Sell Good:: R12: Inform the outcome of the auction	Requirement:: Domain Model:: Class::Sell Good Auction	Consistency
	- Requirement::Activity(Business Process):: Activity::Sell Goods Inform the outcome of the auction.	Warning - Inconsistent

Figura 88. Reporte de Consistencia y Completitud obtenido como producto de la transformación de los modelos para el subsistema *Sell Goods*.

6.2.5 Tomar decisiones de Modelado y Complementar modelos fuente

Basado en los modelos destino de la transformación, que se lograron por la ejecución de las reglas *Alternative* durante las transformaciones de los modelos fuente, y en el reporte de consistencia y completitud, el equipo de desarrollo toma decisiones de modelado según objetivos de sistema y la arquitectura deseada. Los modelos destino resultantes facilitan la medición del esfuerzo que un equipo de desarrollo debe hacer para complementar los modelos (esto se define en el Capítulo 4 y 5).

Una vez que el equipo de desarrollo analiza la información reportada después de transformaciones de los modelos, ellos pueden proceder a realizar cambios que son considerados necesarios en los modelos de la fuente o en los modelos destino. Los cambios puede ser de diferentes tipos: por nombre, por nuevo elemento, por elemento eliminado, etc. Esto se trata en el Capítulo 7 de Gestión del Cambio.

6.3 Conclusiones

En este capítulo se define la especificación de las operaciones del Patrón de Trazabilidad que facilitan el uso del marco conceptual. La forma como se definen en lenguaje natural de forma declarativa busca ayudar a comprender de forma simple como se gestionan los Modelos de Trazado además de otras operaciones abstractas que pueden ser usadas durante la transformación de modelos.

También se define el prototipo de apoyo a la gestión de los Modelos de Trazado, el cual guía el desarrollo de estos en el marco conceptual sobre plataformas como el IDE Eclipse.

El método de trazabilidad se crea para guiar a los desarrolladores en el uso de los Modelos de Trazado y la forma como estos se convierten en la base de la transformación de los modelos de desarrollo. Además, con esto se logra crear una conciencia para realizar la gestión de la configuración con esta nueva teoría de trazabilidad.

Capítulo 7

Gestión del Cambio soportada en Modelos de Trazado

Este capítulo define un enfoque para gestionar el cambio desde los modelos de trazado. Para conseguir esto, se definen los siguientes objetivos: (1) seleccionar un conjunto de características que guíen la gestión del cambio, (2) definir la forma como se debe controlar la propagación del cambio, (3) establecer las métricas necesarias para medir el impacto del cambio en actividades del desarrollo dirigido por modelos.

Para gestionar el cambio, previo a la creación de los modelos de la fuente se ha definido o seleccionado el modelo de trazado que controla la transformación de dichos modelos. Estos proveen los elementos trazables, vínculos de trazado y reglas de transformación que determinan la ejecución de la transformación y facilitan la evaluación del impacto del cambio y su propagación en los modelos destino.

El cambio se define a partir de un conjunto de características que determinan las acciones de propagación y la creación de métricas que soportan la medición y valoración del cambio. De esta forma, los equipos de desarrollo pueden evaluar el impacto del cambio en los modelos destino previo o posterior a la ejecución de las transformaciones que definen la propagación. Estas métricas serán aplicadas a cambios realizados sobre los modelos del sistema de subastas logrando cuantificar características tales como el nivel de exactitud de los modelos destino y el costo de modelos alternativos generados desde la transformación.

7.1 Introducción

La Gestión del Cambio (*Change Management*) es una práctica de control que da soporte al proceso de desarrollo cuando los artefactos de software son modificados durante el proceso de desarrollo o en la etapa de producción. La gestión del cambio, también llamada “Gestión de la Configuración”, es una práctica importante en la administración de un proyecto de desarrollo de software. Esta se encarga de la identificación, organización, y control de la evolución de los requisitos del software cuando cambian debido a la variabilidad de las reglas del negocio, la re-ingeniería del software, o las decisiones que soportan la arquitectura del sistema [Pressman 2006].

Los cambios se realizan por la ocurrencia de eventos de modificación internos o externos en las diferentes etapas del proceso de desarrollo. Durante la especificación de los requisitos los eventos de cambio se pueden generar por mejoras o correcciones en las necesidades del cliente u otros stakeholders, en la arquitectura del sistema, etc. Durante el diseño se pueden hacer mejoras o correcciones a los modelos, las interfases, los componentes, el diseño de la base de datos, etc. En la implementación se pueden mejorar o corregir requisitos o modelos de diseño, estructuras de datos, porciones de código, planes de pruebas, etc. [Bonher 2002].

La gestión del cambio se soporta en la trazabilidad de los requisitos y elementos de modelo o artefactos que los representan a través de su ciclo de vida de desarrollo. Dicha gestión incluye actividades que facilitan el control de los cambios en el sistema. Algunas de estas actividades son: formalización del manejo del cambio, análisis del impacto y propagación del cambio, estimación del costo del cambio, y control de versiones.

La formalización del cambio se hace por medio de plantillas que registran la información detallada del cambio. En estas se registra información básica para realizar el proceso del cambio con información general de la solicitud (solicitante, fecha solicitud, descripción general, estado del cambio, etc.), información detallada del cambio, información del análisis del impacto del cambio, etc. La formalización del cambio es un contrato entre los desarrolladores y el cliente

El análisis de impacto del cambio se define como “el proceso de identificar las consecuencias potenciales de un cambio, y estimar qué necesita ser modificado para lograr el cambio” [Bohner and Arnold 1996]. Un cambio puede afectar varios requisitos o elementos de modelo, por lo tanto el análisis del impacto del cambio debe tener la información suficiente para saber qué entidades cambiarán, qué partes del sistema serán afectadas, como evolucionará el sistema, y como se mantendrá consistente. Además, debe conocerse qué áreas del negocio, fases del proceso y disciplinas del desarrollo (p.ej., diseño, pruebas, etc.) son afectadas De esta forma es posible estimar

los costos y la complejidad de los cambios y ayudar a decidir si la implantación de esos cambios llevan a una nueva versión (*upgrade*) del sistema [Mens et al. 2005]. Para determinar el impacto del cambio es importante identificar los módulos con mayor variabilidad y los componentes más problemáticos del sistema, determinar elementos simples y complejos que impacta el cambio, y analizar las relaciones de dependencias, composiciones, y fusiones entre elementos de modelo [Pressman 2006]. El alcance de un cambio muchas veces requiere de mucho esfuerzo, por eso es importante que el impacto refleje la actualización de la agenda del manejo del proyecto y en tiempo, dinero y otros recursos involucrados en las actividades que implican realizar el cambio.

La estimación del costo de un cambio se formula a partir de variables tales como el tipo de cambio, los artefactos afectados, el tiempo estimado, los recursos humanos y operativos (p.ej., documento de trazabilidad, modelos del sistema, herramientas de desarrollo, plataforma de desarrollo, tamaño del código, etc.) que soportan el sistema. Una buena negociación o compensación (*trade-off*) de conflictos entre los *stakeholders* requiere un buen uso de técnicas de estimación del esfuerzo del cambio, por esto diferentes métodos o técnicas se han implementado. Por ejemplo, el método de estimación desde modelos creados en UML el cual toma la información desde la verificación de la consistencia de los modelos teniendo en cuenta actividades tales como detectar y clasificar automáticamente los cambios por medio de las diferentes versiones de modelos, verificar consistencia de diagramas de cambio, realizar el análisis de impacto para determinar los efectos secundarios potenciales en el diseño, etc. [Briand et al. 2003]. Usualmente, el equipo de desarrollo o el equipo de calidad de las empresas de desarrollo usan o adaptan métodos de estimación del esfuerzo de desarrollo basado en casos de uso [Schneider and Winters 2001], [Carroll 2005] u otros como los puntos de función [Parthasarathy 2007], [Koteswara Rao et al. 2008].

No obstante, la estimación del cambio desde el enfoque del desarrollo dirigido por modelos debe tener en cuenta otras características y acciones relacionadas a las reglas de transformación y las decisiones de modelado que toman el equipo de desarrollo durante la transformación de modelos. Por lo tanto, en este capítulo se define la gestión del cambio desde los Modelos de Trazado y las métricas que permiten conocer el nivel de exactitud de la transformación y el costo ocasionado por la generación de modelos alternativos. Para realizar la estimación del cambio, se identifica el modelo de trazado que ha controlado la transformación de los modelos en el proyecto de software; este provee los elementos trazables, vínculos de trazado, y reglas de transformación que se toman como insumo para la propagación del cambio y la definición de métricas que ayudarán a la estimación del esfuerzo. La gestión del cambio se hace con el caso de estudio Sistema de Subasta.

Éste capítulo está estructurado como se define a continuación. La Sección [7.2](#) define las características para manejar el cambio desde los Modelos de Trazado. La Sección [7.3](#) describe la propagación del cambio de forma controlada. La Sección [7.4](#) define métricas para estimar el esfuerzo de los equipos de desarrollo al realizar cambios. La Sección [7.5](#) analiza los trabajos relacionados y los compara con nuestro trabajo. La Sección [7.6](#) concluye el capítulo.

7.2 Caracterización del Cambio

El cambio es un requisito que genera una actualización del sistema en desarrollo o en producción. Es provocado por eventos internos o externos al sistema por lo que debe ser controlado y cuantificado con el fin de mantener productos de software confiables y disminuir los conflictos que puedan presentarse entre el equipo de desarrollo y los usuarios o patrocinadores del proyecto.

El cambio sólo puede ser gestionado con confiabilidad a partir del rastro que se registra durante la evolución de los requisitos en el proceso de desarrollo, ya que esta es la fuente de información a partir de la cual los requisitos de cambio se deben analizar para determinar su impacto en el sistema. En esta disertación, las instancias de Modelos de Trazado se convierten en el artefacto de trazabilidad que facilita la información para la gestión del cambio en el contexto MDD.

7.2.1 Definición del Cambio

Los Modelos de Trazado proporcionan características de modificabilidad tales como la identificación de los elementos trazables por roles de trazado, el control de las trazas horizontales y verticales para la ejecución de la transformación y la propagación de los cambios. Las instancias de los Modelos de Trazado son vistas versionadas en un sistema específico (nombrada: vista del modelo de trazado) e independientes de los modelos de desarrollo fuente o destino de la transformación (como se explicó detalladamente en el Capítulo 3). Estas instancias proveen la información necesaria para analizar el impacto del cambio y coordinar su propagación automática en otros niveles de abstracción.

Por ejemplo, la Figura 88 ilustra la vista general del modelo de trazado para el “*Auction System*”, subsistema “*Sell Goods version 1.0.0.*”, sobre el cual se hace la gestión de los cambios que afectan directamente este subsistema (por ser la última versión).

Para un Modelo de Trazado un requisito de cambio es un elemento de modelo estructurado representado en un cuadro (rectángulo) que usa el estereotipo <<change>> y tiene características propias que determinan y controlan la actualización de sistema. Estas características se expresan en

términos de atributos tales como identidad, descripción del cambio, instancia del modelo de trazado sobre la cual se realizará el análisis del impacto del cambio, modelo de la fuente donde se hace el cambio, elementos de la fuente directamente afectado por el cambio, operación de cambio, elementos del destino generado o modificados por el cambio, fecha en la cual se realiza el cambio, y un estado que indica si el cambio se propaga o no. En la Figura 89 se describe cada una de las características.

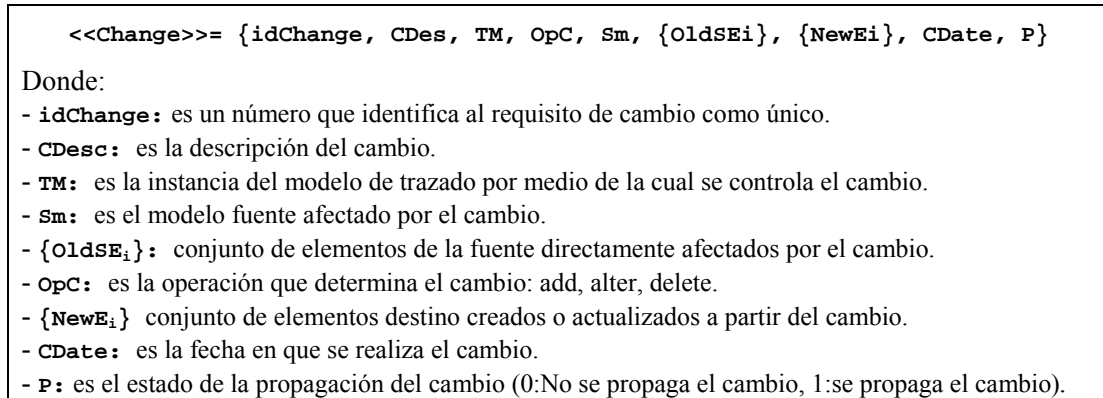


Figura 89. Características del elemento de modelo por medio del cual se gestiona un requisito de cambio.

La Figura 90 ilustra la definición genérica de un requisito de cambio en la vista del modelo de trazado donde se relaciona con el elemento modelo de la fuente directamente afectado por el cambio por medio de un vínculo de trazado <<trace>>.

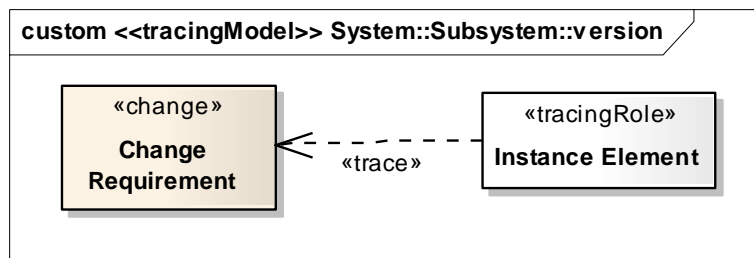


Figura 90. Registro de un cambio en la vista del modelo de trazado.

Los grupos de calidad de las empresas de desarrollo de software usan plantillas para formalizar las solicitudes de cambio de tal forma que los requisitos (o eventos) de cambio se definan de forma estándar y se facilite su control. Por esta razón, las características del requisito de cambio definidas anteriormente son complementadas por otras características administrativas que ayudan a la gestión de la configuración.

Para lograr esto se ha diseñado una plantilla que identifica datos acerca de la administración del cambio tales como los responsables del cambio, las características básicas del requisito de cambio y los datos operativos del cambio tales como la vista del modelo de trazado, estimación de su propagación y estimación del esfuerzo de transformación. La Figura 91 ilustra la plantilla con cinco secciones claramente definidas: sección 1, *Descripción General de la Solicitud del Cambio*, define atributos administrativos. Estos son necesarios para establecer información que puede afectar la administración del proyecto (p.ej., recursos, costos, etc.); sección 2, *Descripción del Cambio*, define el detalle del requisito del cambio, donde se distingue el requisito, sus restricciones, reglas de negocio y otros aspectos que determinan el cambio; sección 3, *Evaluación del Cambio*, describe el estado del sistema, el modelo de trazado sobre el cual se hará la evaluación del impacto del cambio, el tipo de cambio, los elementos de modelo directamente afectados, y la operación de cambio; sección 4, *Impacto de la Propagación de Cambio*, describe el análisis de impacto que se hace desde la vista del modelo de trazado del sistema; sección 5, *Estimación del Esfuerzo/Costos*, muestra la estimación del esfuerzo de propagación y ejecución de la transformación de los modelos.

1. Descripción General de la Solicitud del Cambio			
Control No.			
Proyecto/Sistema			
Solicitante			
Desarrollador Evaluador			
Fecha de Solicitud			
Fecha de Evaluación			
Estado del Cambio	En estudio:	Rechazado:	Aceptado:
2. Descripción del Cambio			
Descripción detallada y precisa del requisito cambio:			
Restricciones asociadas al cambio:			
Reglas de Negocio asociadas al cambio:			
Otros Aspectos Relevantes al cambio:			
3. Evaluación del Cambio			
Estado del Proyecto/Sistema	En Desarrollo:	En Producción:	
Vista actual del modelo de trazado	Tipo:		
	Nombre:		
	Versión:		
Elemento(s) de modelo directamente afectado			
Operación de Cambio (Adición/Modificación/Eliminación)			
Tipo de Cambio			
4. Impacto de la Propagación del Cambio (ítems de configuración)			
Trazas Activas			

Reglas Ejecutadas			
Total nuevos elementos de modelo creados			
Total elemento de modelo existentes afectados			
Total de Relaciones afectadas			
Medición del Impacto (Alto/medio/bajo)			
Nueva Versión			
5. Estimación de Costos			
Tiempo Total		Costo Total	
<u>Observaciones Generales:</u>			

Figura 91. Solicitud de Gestión del Cambio basada en Modelos de Trazado.

En los siguientes párrafos se describen las secciones 1, 2 y 3 de la plantilla y por medio de un ejemplo se muestra su uso.

Sección 1. Solicitud del Cambio: un cambio está asociado a un sistema o proyecto de software en desarrollo o en producción. Es solicitado y descrito formalmente por un usuario o desarrollador del proyecto en una fecha determinada. El responsable de la gestión del cambio en el equipo de desarrollo debe evaluar la validez y factibilidad de la solicitud; así esta toma uno de los siguientes tres estados: “*En Estudio*”, “*Rechazada*”, “*Aceptada*”. La Figura 92 muestra el ejemplo de la solicitud de un cambio para el Sistema de Subastas descrito en el Apéndice C.

1. Descripción General de la Solicitud del Cambio			
Control No.	<i>1</i>		
Proyecto/Sistema	<i>Sistema de Subastas</i>		
Solicitante	<i>Ana Smith</i>		
Desarrollador Evaluador	<i>Marta Tabares</i>		
Fecha de Solicitud	<i>15-Noviembre-2008</i>		
Fecha de Evaluación	<i>16-Noviembre-2008</i>		
Estado del Cambio	En estudio: <i>X</i>	Rechazado:	Aceptado:

Figura 92. Un ejemplo de la descripción general de la Solicitud del Cambio para el Sistema de Subasta.

Sección 2. Descripción del Cambio. El solicitante debe hacer una descripción clara y concisa del requisito de cambio separando sus restricciones, reglas de negocio y otros aspectos relevantes o complementarios que involucre el cambio. Así, el evaluador podrá realizar un análisis más certero de la validez del cambio en las vistas de modelos de trazado involucrados en el cambio.

Por ejemplo, en la Figura 93 se muestra el requisito cambio “*Los clientes ofrecen Paquetes de bienes los cuales son grupos de bienes que están registrados en el sistema y se puedan vender como un solo bien. Cuando el vendedor coloca el Paquete de bienes a la venta, los bienes que lo componen no deben tener ofertas pendientes y no se pueden vender como bienes individuales*”. El

desarrollador hace un análisis semántico de la descripción y define que esto es un requisito funcional con dos restricciones.

2. Descripción del Cambio	
<u>Descripción detallada y precisa del Requisito cambio:</u> <i>Los clientes ofrecen “Paquetes de bienes” los cuales son grupos de bienes que están registrados en el sistema y se puedan vender como un solo bien.</i>	
<u>Restricciones asociadas al cambio:</u> <ul style="list-style-type: none"> - <i>Cuando el vendedor coloca el “Paquete de Bienes” a la venta, los bienes que lo componen no deben tener ofertas pendientes.</i> - <i>Los bienes que forme parte del “Paquete de Bienes” no se pueden vender como bienes individuales.</i> 	
Reglas de Negocio asociadas al cambio: -----	
Otros Aspectos Relevantes al cambio: -----	

Figura 93. Un ejemplo de la descripción de un requisito de Cambio en la Solicitud de Cambio.

Sección 3. Evaluación del Cambio. A partir de la descripción del requisito de cambio, el evaluador analiza aspectos que determinarán la factibilidad del cambio. Por ejemplo el estado del sistema, la vista del modelo de trazado identificada para analizar el cambio, los elementos de modelo directamente afectados por el cambio, etc.

En la Figura 94 se ilustran los valores que toman las diferentes propiedades que describen la evaluación del requisito de cambio usado como ejemplo. Aunque para este caso específico se generan tres tipos de cambio diferente, solo se ilustra a modo de ejemplo la gestión con el requisito funcional.

3. Evaluación del Cambio		
Estado del Proyecto/Sistema	En Desarrollo: <i>X</i>	En Producción:
Vista actual del modelo de trazado	Tipo: <i>Basado en Casos de Uso</i>	
	Nombre: <i>Auction System::Sell Goods</i>	
	Versión: <i>1.0.0</i>	
Elemento(s) de modelo directamente afectado por el cambio	<i>Sell Goods:: Goods Package</i>	
Operación de Cambio (Adicionar/Modificar/Eliminar)	<i>Adicionar</i>	
Tipo de Cambio	<i>Requisito Funcional</i>	

Figura 94. Un ejemplo de la Evaluación del Cambio hecha para un requisito de cambio en la Solicitud de Cambio.

Identificar el estado del sistema es una propiedad informativa que permite identificar la vista del Modelo de Trazado línea base por medio del cual es o fue controlado el desarrollo del sistema, es decir la huella de la última versión generada en desarrollo o la última liberación en producción.

Los cambios realizados en sistemas en desarrollo se pueden hacer en cualquier etapa del ciclo de vida por lo tanto es importante identificar las áreas del negocio afectadas con el cambio y los equipos de desarrollo que se deben involucrar para estimar el cambio. Cuando el sistema se encuentra en producción el requisito de cambio genera mayor impacto en el sistema ya que las operaciones y la información se deben mantenerse en tiempo real y es indispensable mantener su integridad sin generar fallas de ejecución o información errada. En este caso las copias de seguridad, las pruebas y la liberación del producto y nueva versión son algunos factores que se deben manejar cuidadosamente y de acuerdo a políticas establecidas en producción.

El desarrollador busca la vista del modelo de trazado que se puede afectar por el cambio. Esta se puede buscar el tipo de modelo, el nombre y por la versión. En un repositorio de modelos de trazabilidad se van generando las diferentes versiones a medida que se modifica el sistema, así el evaluador del cambio puede identificar la última versión de la vista actual del modelo de trazado.

Un requisito de cambio puede afectar la definición original de los requisitos del usuario, la especificación de un caso de uso del sistema, la estructura o las operaciones de elementos de modelo, las relaciones entre los elementos, etc. (por lo general para un subsistema específico). También puede ser un requisito totalmente nuevo en el sistema que afecte o no a los subsistemas existentes. De acuerdo con el tipo de modelo de trazado usado durante el desarrollo (en este caso es basado en casos de uso), el evaluador identifica los elementos de modelo y el nivel de abstracción donde debe realizar el cambio.

7.2.2 Operaciones de Cambio

Con base en la información básica del cambio, el equipo de desarrollo determina la operación de cambio que se debe realizar en el sistema. Estas operaciones pueden ser *Adicionar*, *Modificar* o *Eliminar*. Cada operación de cambio se asocia a uno o varios tipos de cambios que guían a los desarrolladores a realizar una acción específica en el sistema. La Tabla 28 muestra los tipos de cambio que se asocian a cada operación.

Tabla 28. Tipos de Cambio asociados a cada operación.

Operación de Cambio	Tipo de Cambio
Adicionar	- Adicionar un nuevo elemento de modelo en el nivel de Requisitos.

	- Adicionar un nuevo elemento de modelo en los modelos de Análisis o de Diseño.
	- Adicionar un nuevo vínculo de trazado en la vista actual del modelo de trazado.
Modificar	- Modificar el nombre de un elemento de modelo.
	- Modificar la especificación de un elemento de modelo existente
	- Modificar la fuente o el destino de un vínculo de trazado en la vista actual del modelo de trazado.
Eliminar	- Eliminar un elemento activo en un modelo del sistema.
	- Eliminar vínculos de trazado de la vista del modelo de trazado actual.

Operación Adicionar. Esta operación se refiere a la acción que define un nuevo elemento modelo en un sistema de modelos versionados. El requisito de cambio se debe aplicar en el modelo de Requisitos o en su defecto en el modelo de Análisis donde son definidos los elementos ejes del trazado (es una decisión del desarrollador). Esto es con el fin de que el cambio sea propagado consecuentemente en los modelos destino a fin de conservar la completitud y la consistencia de los modelos, no obstante los Modelos de Trazado están diseñados para hacer la propagación desde cualquier modelo y cualquier elemento de modelo.

Para este tipo de operación siempre se hace la propagación del cambio por lo tanto la propiedad de propagación $\langle P \rangle$ en la especificación del requisito de cambio es igual a “1”. En los casos de adición la propiedad $\langle \{OldSEi\} \rangle$ toma el valor de “nulo” ya que el cambio es un nuevo requisito o elemento de modelo y no existen otros elementos de modelo que cambien. Para esta operación los tipos de cambio adicionar un nuevo elemento de modelo en el nivel de Requisitos, adicionar un nuevo elemento de modelo en los modelos de Análisis o de Diseño, y adiciona un nuevo vínculo de trazado en la vista actual del modelo de trazado, se describe a continuación:

☞ **Adicionar un nuevo elemento de modelo en el nivel de Requisitos.** Un nuevo objetivo (asunto o subsistema) y sus requisitos son creados en el modelo de Requisitos de la versión actual del sistema. Por ejemplo, la Figura 8 ilustra la definición del requisito de cambio para el *Auction System* que crea el nuevo subsistema “*Special Auction*” con el requisito “*A seller offers Goods’ Packages*”.

Este requisito de cambio hace que el equipo de desarrollo cree los nuevos elementos en el modelo de Requisitos el cual se convierte en el modelo fuente de la propagación. Así, el planificador de transformaciones identifica vínculos de trazado asociados a dichos elementos para controlan su transformación.

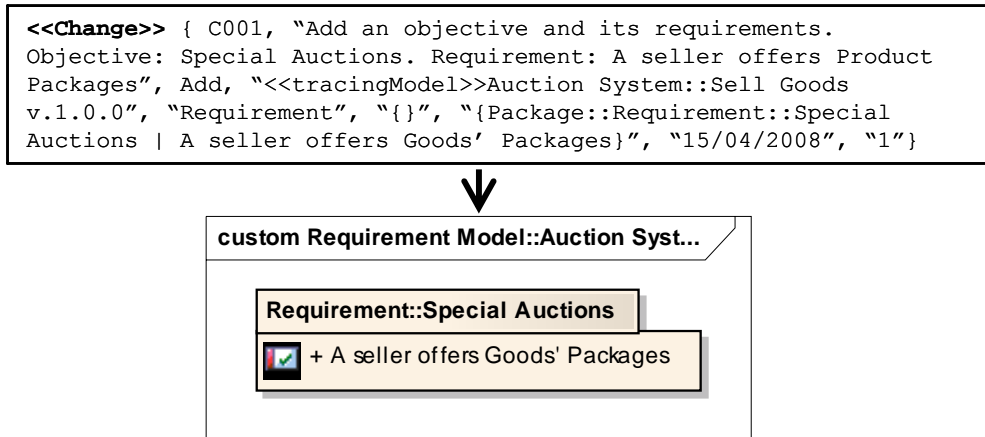


Figura 95. Ejemplo del requisito de cambio (C001) para adicionar el nuevo objetivo “*Special Auction*” y su requisito “*A seller offers Goods Packages*” en el *Auction System*.

La identificación se refiere a una búsqueda (realizada por la operación `retrieveTracingLink(traceableElement)`) de los vínculos de trazado del Modelo de Trazado basado en Casos de Uso cuyos elementos predecesores (`targetTrace`) sean del tipo `Package` y `Requirement`. La Tabla 29 muestra algunos de los vínculos de trazado que van a garantizar la consistencia del requisito con el modelo del dominio y el proceso de negocio cuyas instancias se obtienen una vez se ejecuten las reglas de transformación asociadas a cada vínculo. Los vínculos de trazado `<<refine>>` ó `<<realize>>` garantizan la propagación del cambio y la completitud de los modelos en otros niveles de abstracción.

Tabla 29. Vínculos de trazado básicos en el nivel de Requisitos para el requisito de cambio C001.

TracingLinks	sourceTrace	targetTrace	TransformationRule
<code><<trace>></code>	<code><<predecessor>>Requirement::Package::DomainModel</code>	<code><<predecessor>>Requirement::Package::Requirement</code>	- RPackage2DMPackage - DMPackage2RPackage
<code><<trace>></code>	<code><<predecessor>>Requirement::Package::BusinessProcess</code>	<code><<predecessor>>Requirement::Package::Requirements</code>	- RPackage2CDPackage - BPPackage2RPackage
<code><<refine>></code>	<code><<axisTracing>>Analysis::Package::UseCase</code>	<code><<predecessor>>Requirement::Package::Requirements</code>	- RPackage2UCPackage - UCPackage2RPackage
<code><<realize>></code>	<code><<successor>>Design::Package::Collaboration</code>	<code><<axisTracing>>Analysis::Package::UseCase</code>	- RPackage2UCPackage - UCPackage2RPackage
...

Una vez se identifican los vínculos de trazado que determinan la transformación del nuevo elemento, se crean las instancias de dichos vínculos que el planificador tomará como base para la transformación del elemento “*Special Auctions*”. El requisito de cambio se crea como un elemento de la vista del modelo de trazado y se relaciona con el nuevo elemento por medio del vínculo de trazado `<<trace>>`. La Figura 96(a) ilustra esta relación que garantiza la

consistencia entre el cambio *C001* y el elemento de modelo directamente afectado. La Figura 96(b) muestra la propagación del cambio a partir de la transformación del requisito de cambio en elementos de modelo tales como el paquete `DomainModel::Special Auctions` con su clase `SpecialAuctions`, el proceso del negocio `BussinessProcess::Special Auctions` con su actividad “A seller offers Goods’ Packages”, y el elemento agregado a la actividad `lane-pool:: SpecialAuctions`. Estos elementos se crean por medio de las instancias de los vínculos de trazado cuyas reglas de transformación son ejecutadas durante el proceso de transformación. De igual forma, se crean otros modelos destino en otros niveles de abstracción.

- Adicionar un nuevo elemento de modelo en los modelos de Análisis o de Diseño.** En los modelos del análisis o el diseño a menudo se crean elementos de modelos tipo *collaboration*, *class*, *activity*, *sequence*, *component*, etc., y sus instancias de forma autónoma representan requisitos del usuario para los diferentes subsistemas del sistema solución. Sin embargo, es posible que estos elementos se creen en modelos o diagramas generales de modelado tales como diagramas de clases sin estar asociados a un subsistema específico entonces es importante considerar los siguientes casos.

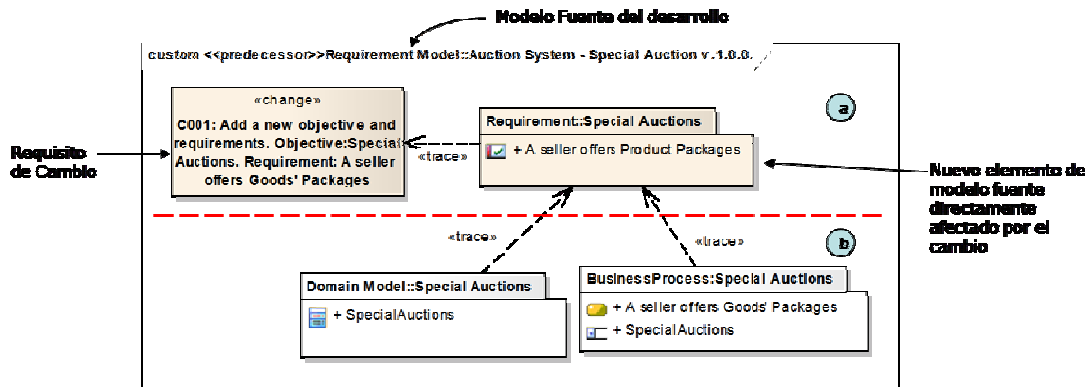


Figura 96. La nueva vista del modelo de trazado con el nuevo requisito de cambio ejecutado.

- a.** El nuevo elemento de modelo no es relacionado con elementos del mismo tipo ni a algún subsistema existente, se crea en un modelo general. Por ejemplo, la Figura 97(a) muestra la nueva clase *BrowserAuction* (tipo *interface*) la cual se crea en el modelo de clases del nivel del Análisis del sistema *Auction System*. Esta clase no es asociada con otros elementos del mismo tipo en este modelo. Esto no indica que esté bien o mal modelado pero el Modelo de Trazado provee los elementos trazables y los vínculos de trazado que propagarán este elemento para mantener la consistencia y la completitud de los modelos, no obstante se reportará como

una advertencia de inconsistencia o incompletitud para que el equipo de desarrollo revise esta situación.

b. El nuevo elemento es asociado con otro elemento del mismo tipo, por lo tanto el nuevo elemento debe ser adoptado en el subsistema al que pertenece el elemento al cual se relaciona. Por ejemplo, la Figura 97(b) en el modelo de clases se crea la clase *CanceledAuctions* en el modelo de análisis con el atributo `stampPrice` y una relación de generalización con la clase *Auction*.

El planificador de la transformación identifica los vínculos de trazado relacionados en el elemento `<<axisTracing>>Analysis::Class` para ejecutar las transformaciones correspondientes, propagar el cambio y generar una nueva vista de modelo de trazado. Este incluye un vínculo de trazado `<<trace>>` entre el cambio y los nuevos elementos de modelo con el fin de conservar el modelo consistente.

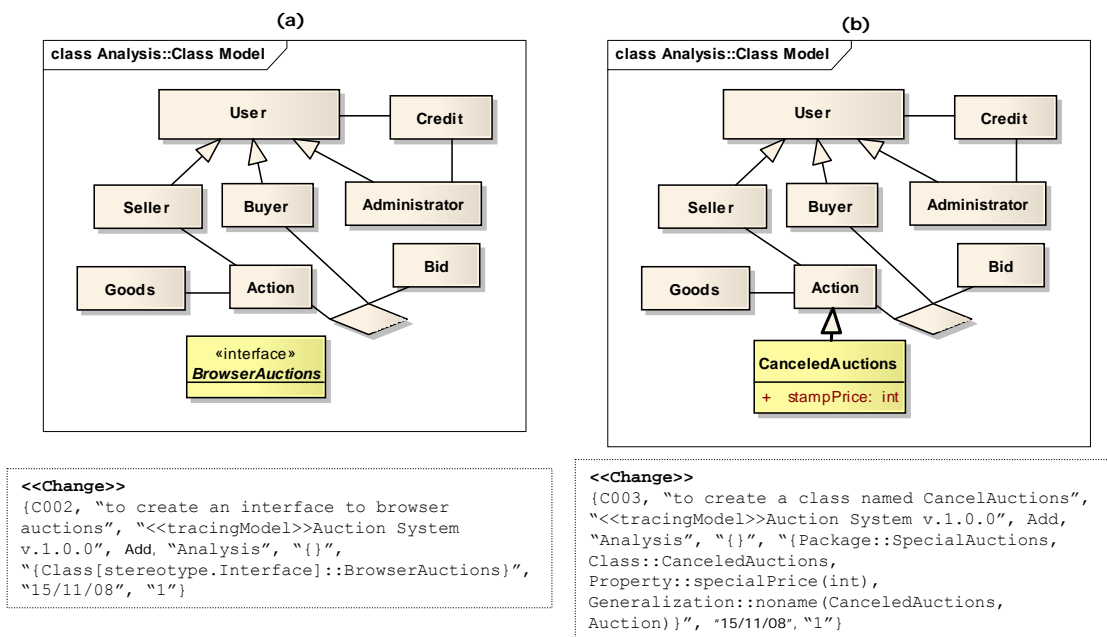


Figura 97. Dos casos de requisitos de cambio para el Modelo de Clases en el Modelo de Análisis.

☞ **Adiciona un nuevo vínculo de trazado en la vista actual del modelo de trazado.** En algunas ocasiones los vínculos de trazado debe ser a adicionados manualmente ya que algunos elementos de modelo no son creados en el subsistema que los usa. Por ejemplo, cuando fue creado el paquete `Requirement::SpecialAuctions`, por consistencia la clase *SpecialAuctions* fue creada en el modelo del dominio, sin embargo el desarrollador considera que la clase *Auction* también debe trazar este objetivo del sistema.

En este caso el desarrollador tiene dos alternativas para tomar la decisión. Por ejemplo, en la Figura 98(a) se adiciona la clase *Auction* en el paquete `DomainModel::Special Auction` del modelo de Requisitos convirtiéndose en una nueva instancia de traza para la próxima transformación. En la Figura 98(b) se adiciona un vínculo de trazado `<<trace>>` entre la clase *Auction* y el paquete `Requirement::Special Auctions` en la vista de modelo de trazado. Esta opción es la única forma de modificar las vistas de modelo de trazado.

Operación Modificar. Esta operación se refiere a la acción que modifica elementos del sistema actual. Los cambios considerados en esta operación se describen en los siguientes párrafos.

🌀 **Modificar el nombre de un elemento de modelo.** El nombre de un elemento de modelo se puede cambiar en cualquier momento durante el proceso de desarrollo. Este tipo de cambios es muy importante controlarlo en un desarrollo dirigido por modelos ya que el nombre del elemento es el valor base de la transformación y por consecuencia de otras actividades de trazabilidad como la propagación del cambio.

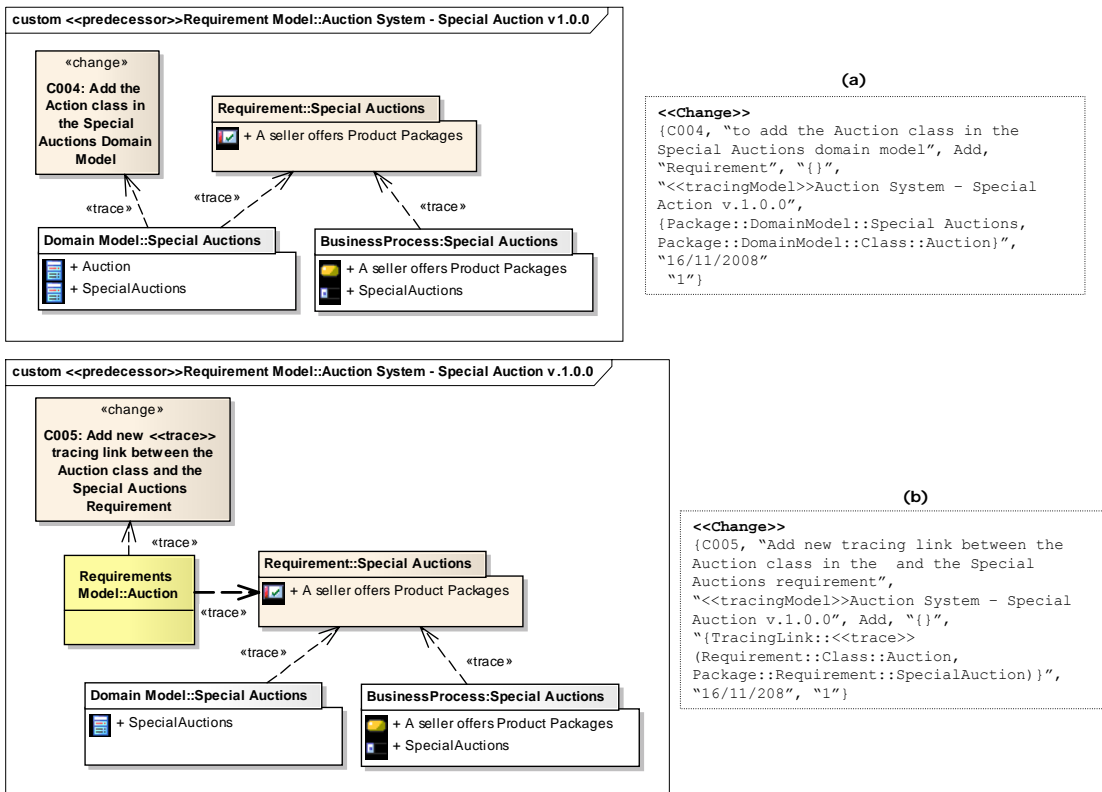


Figura 98. Dos formas de cambio para adicionar un nuevo vínculo de trazado `<<trace>>` en una vista de modelo trazado.

La Figura 99 muestra un ejemplo donde el requisito de cambio *C010* cambia el nombre del caso de uso *A seller initiates an auction* por el nuevo nombre *Sell Goods*.

```
<<Change>>
{ C010, "Change the use case name from "A seller initiates an
auction" to new name "Initiate an auction" ",
"<<tracingModel>>Auction System v.1.0.0", Alter, "Analysis",
"{Package::UseCase::Sell Goods | A seller initiates an auction
Packages}", "{Package::UseCase::Sell Goods | Sell Goods ",
"20/11/2008", "0" }
```

Figura 99. Operación Modificar: el nombre del caso de uso “A seller initiates an auction” por el nuevo nombre “Sell Auction”.

Este tipo del cambio se puede hacer en cualquier nivel de abstracción por lo tanto es necesario tener en cuenta dos posibles caminos: por ejemplo en la Figura 100 se cambia sólo el nombre del elemento modelo directamente afectado y la instancia del vínculo de trazado, sin ser propagado en elementos destino (Propagación = 0). Una vez hecho el cambio y las transformaciones, una nueva vista de modelo de trazado se genera con un nuevo número de versión. Otros tipos de cambios también pueden generar una nueva versión, sin embargo este tipo de cambio siempre crea una nueva versión. Esto significa que la consistencia y la completitud se mantienen por las instancias de los vínculos de trazado.

El otro camino es cambiar el nombre del elemento de modelo y en la transformación se propaga el cambio de nombre a otros modelos destino ($P = I$). Así, otros vínculos de trazado o secuencias de traza que tengan este nombre `UseCase::Sell Goods | A seller initiates an auction` también son cambiadas. La vista de modelo de trazado incluye un vínculo de trazado `<<trace>>` entre el requisito de cambio y el elemento de modelo afectado directamente por el cambio.

☞ **Modificar la especificación de un elemento de modelo existente.** Los elementos de modelo que agregan otros elementos de modelo como son las clases, las secuencias, etc., se pueden modificar por adición o borrado de los elementos que lo componen en su estructura o comportamiento. Por ejemplo, la Figura 101 muestra la adición del atributo `typeAuction` en la clase *Auction*.

En este caso esta modificación puede alterar el conjunto de elementos involucrados en los vínculos de trazado asociados al tipo de elemento `Class` tanto en las trazas verticales como horizontales (la traza vertical y horizontal se explica en el Capítulo 3), y en especial a las trazas implícitas de los elementos de modelo que agrupan o están compuestos por otros elementos de modelo (las trazas implícitas se explican en el Capítulo 3).

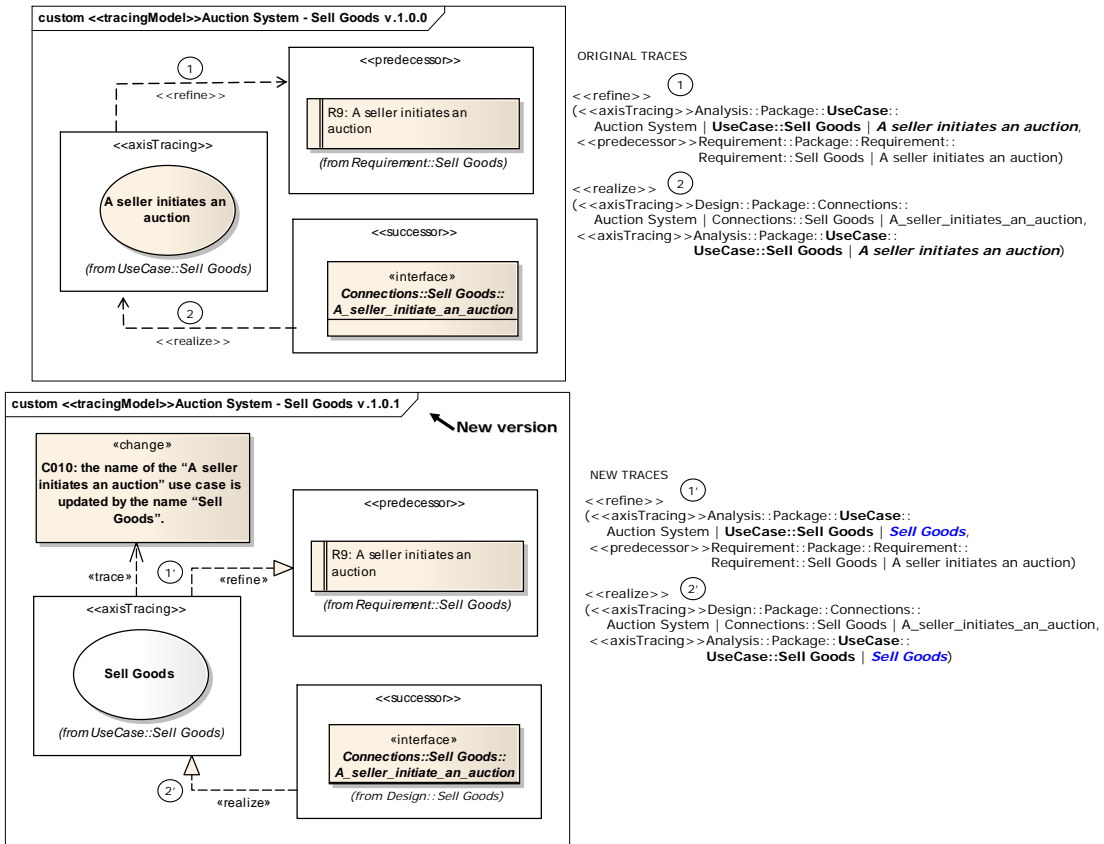


Figura 100. Requisito de Cambio: Cambiar el nombre del caso de uso “A seller initiates an auction” en la vista de modelo de trazado actual.

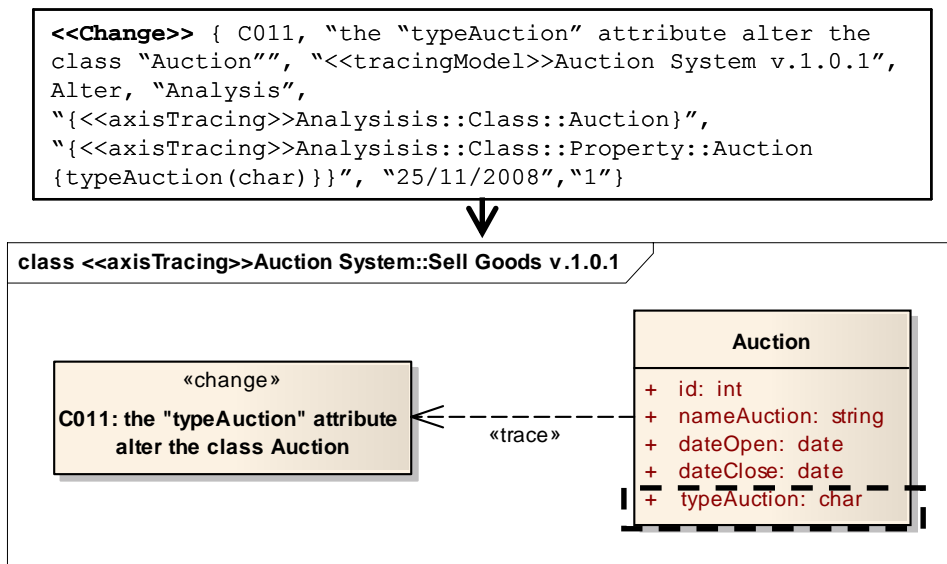


Figura 101. Modificación de la clase “Auction” con la adición del nuevo atributo “typeAuction”.

☞ **Modificar la fuente o el destino de un vínculo de trazado en la vista actual del modelo de trazado.** Cuando se crean las instancias de los vínculos de trazado, las trazas se generan de forma estándar pero muchas de ellas no son válidas para hacer futuras transformaciones de un sistema (por decisión del desarrollador). Entonces, en la vista de modelo de trazado actual se puede cambiar la fuente o el destino de un vínculo de trazado si, y sólo si, la nueva instancia es válida para los vínculos de trazado o las secuencia de traza definidas para el modelo de trazado usado como la plantilla. La Figura 102 ilustra la descripción del cambio *C011* y los vínculos de trazado `<<trace>>` entre la clase *Seller* y el caso de uso *Sell Goods*, que soportan la transformación de estos elementos. Además muestra otras trazas que son candidatas a ser cambiadas. La traza (1) cambia a la traza (1') donde el nuevo elemento de modelo destino es el actor *Sell(er)* (este es un cambio válido ya que es controlado por una secuencia de traza). Las trazas (2') y (3') son vínculos de trazado candidatos a ser borrados.

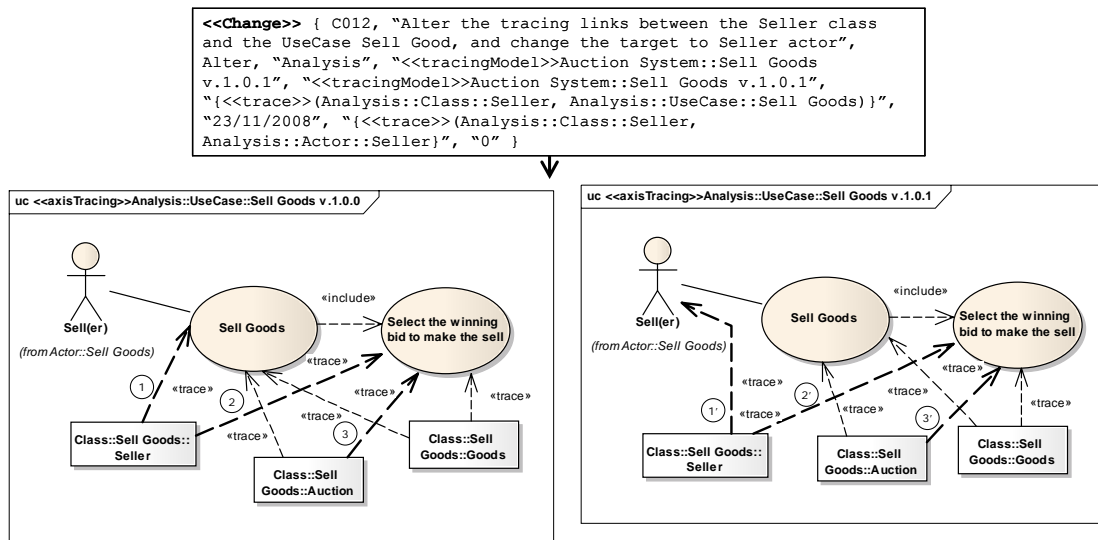


Figura 102. Operación Modificar: Borrar el vínculo de trazado entre la clase Seller y el caso de uso "Sell Goods" de la vista de modelo de trazado actual.

Operación de Eliminar. Esta operación se refiere a la acción de borrar elementos de modelo del sistema actual. Antes de ejecutarse esta operación sobre los modelos del sistema, se hace una evaluación del impacto en la última versión de la vista del modelo de trazado y genera un reporte de advertencia indicando los elementos que serán afectados por el cambio. Dos casos de cambios se consideran para esta operación.

☞ **Eliminar un elemento activo en un modelo del sistema.** Cuando se elimina un elemento activo en los modelos del sistema, el modelo de trazado garantiza su consistencia y

completitud una vez se haga la propagación de dicho cambio. En este tipo de eliminación pueden tratarse dos casos.

Caso 1. El elemento directamente afectado por el cambio es localizado en una sola vista de modelo de trazado, facilitando la propagación de la eliminación del elemento generando. La Figura 103 ilustra la descripción para este tipo de cambio.

```
<<Change>>
{ C012, "Delete the use case Sell Goods", "<<tracingModel>>Auction
System::Sell Goods v.1.0.1", Delete, "Analysis",
"{<<trace>>(Analysis::UseCase::Sell Goods)", "{}", "23/11/08", "0"}
```

Figura 103. Operación Eliminar: Un caso de uso es borrado del sistema actual en una vista específica de modelo de trazado.

Caso 2. El elemento directamente afectado por el cambio puede estar en muchas vistas de modelo de trazado. Esto puede ocurrir con algunas clases u otros elementos que el equipo de desarrollo modele en diagramas generales (esto dependerá totalmente del sistema de modelado adoptado). La Figura 104 ilustra la descripción de este tipo de cambio.

```
<<Change>>
{ C012, "Delete the use case Sell Goods", "<<tracingModel>>Auction
System v.1.0.1", Delete, "Analysis",
"{<<trace>>(Analysis::UseCase::Sell Goods)", "{}", "23/11/08", "0"}
```

Figura 104. Operación Delete: Un caso de uso es borrado del sistema actual en diferentes vistas de modelo de trazado.

∞ **Eliminar vínculos de trazado de la vista del modelo de trazado actual.** Una instancia de un vínculo de trazado se puede borrar de la vista actual si, y solo si, el elemento fuente de la traza mantiene el mismo tipo de vínculo de trazado con al menos otro elemento destino. De otra forma tanto el vínculo de trazado como la instancia de la traza son borrados. La Figura 105 ilustra la descripción del cambio para el vínculo de trazado entre dos.

```
<<Change>> { C013, Delete,
"Delete tracing links between the Seller class and the "Select the
winn..." use case",
"<<tracingModel>>Auction System v.1.0.1",
"<<tracingModel>>Auction System v.1.0.1",
"{<<trace>>(Class::Seller, UseCase::Select the winning bid to make the
sell)", "{}", "0" }
```

Figura 105. Operación Delete para eliminar vínculos de trazado en la vista actual del modelo de trazado.

7.3 Estimación del Impacto y Propagación del Cambio

La estimación del impacto y la propagación de cambio son hechos seguidos de la identificación del cambio. Cuando el cambio se realiza este se debe propagar incrementalmente en los modelos destino afectados por el cambio. La propagación de cambio se hace sobre los modelos de desarrollo y se controla por medio de las instancias de los vínculos de trazado y las secuencias de traza de la vista del modelo de trazado actual. La actualización de la versión de los modelos dependerá del sistema de versiones adoptado por el equipo de desarrollo.

Para lograr una gestión del cambio como se describió en la sección anterior, una vez se define el cambio debe analizarse el impacto que este ocasiona en otros modelos del sistema por medio de los siguientes pasos: identificación de las instancias de los vínculos de trazado relacionadas con el elemento directamente afectado por el cambio, propagación del cambio y control de versiones.

Identificación de las instancias de los vínculos de trazado relacionadas con ese elemento.

- ☞ Si la operación de cambio OpC es *Adicionar*, normalmente el nuevo elemento no se identifica en ninguna instancia de vínculos de trazado, entonces el sistema identifica todos los vínculos de trazado asociados al tipo de elemento afectado (como se muestra en la Tabla 29). A partir de estos el sistema crea las instancias con los nuevos elementos para hacer la propagación desde la transformación de modelos. La Tabla 30 muestra este caso para el cambio *C001*.

Tabla 30. Vínculos de trazado que participan en la propagación del cambio para una operación de Adicionar al nivel de los Requisitos.

Requisitos de Cambio	Instancias de vínculos de trazado
<pre><<Change>> { C001, "Add an objective and its requirements. Objective: Special Auctions. Requirement: A seller offers Goods' Packages", Add, "<<tracingModel>>Auction System v.1.0.0", "Requirement", "{}", "{(Package:: Special Auctions), (Requirement:: A seller offers Goods's Packages)}", "15/04/2008", "1"}</pre>	<ol style="list-style-type: none"> 1. <<refine>> (<<axisTracing>>Analysis::Package::UseCase::Special Auctions, <<predecessor>>Requirement::Package::Requirement::Special Auctions) 2. <<refine>> (<<axisTracing>>Analysis::Package::UseCase::UseCase::Special Auctions A seller offers Product Packages, <<predecessor>>Requirement::Package::Requirement:: Requirement::Special Auctions A seller offers Product Packages) 3. <<refine>> (<<successor>>Design::Package::Collaboration::Special Auctions, <<axisTracing >> Analysis::Package::UseCase::Special Auctions) 4. <<realize>> (<<successor>>Design::Package::Collaboration:: Collaboration::Special Auctions A seller offers Product Packages, <<axisTracing>>Analysis::Package::UseCase::UseCase::Special Auctions A seller offers Goods' Packages) 5. <<trace>>

	<pre> (<<predecessor>>Requirement::Package::Class (DM) : : DomainModel::Special Auctions SpecialAuction /* package and class take the requirement package name */ <<predecessor>>Requirement::Package:: Requirement::Requirement::Special Auctions A seller offers Goods' Packages) 6. <<trace>> (<<predecessor>>Requirement::Package::BusinessPr ocess::BusinessProcess::Special Auctions, <<predecessor>> Requirement::Package::Requirement:: Requirement::Special Auctions A seller offers Goods' Packages) ... </pre>
--	--

La Tabla 31 muestra la propagación del cambio para el cambio C002. Cuando la adición se hace en los niveles de Análisis y Diseño los vínculos de trazado que pueden ser impactados por el cambio se buscan usando el comodín “*” en algunas partes de las cadenas descriptiva del vínculo. Por ejemplo, en la instancia de modelo de trazado se puede buscar los vínculos que tengan la palabra “BrowserAuctions”. Esta búsqueda se hace por medio de la operación retrieveTracingLink() proporcionada por la metaclass TracingLink.

Tabla 31. Vínculos de Trazado que participan en la propagación del cambio en una operación de adición en los niveles de Análisis y Diseño.

Requisitos de Cambio	Instancias de vínculos de trazado
<pre> <<Change>> {C002, "to create an interface to browser auctions", Add, "<<tracingModel>>Auction System v.1.0.0", "Analysis", "{}", "{Analysis::Package::Class: :ClassModel::BrowserAuction s}", "1"} </pre>	<pre> Call retrieveTracingLink(Analysis::Class::browserAuction) Select namelink.souceTrace="*BrowserAuctions*" or namelink.targetTrace="*BrowserAuctions*" Si encuentra al menos una instancia entonces verificar modelo y elementos encontrados con el fin de activar los vínculos de trazado necesarios para mantener la consistencia y la completitud del nuevo elemento. sino Crear nuevas instancias de vínculos de trazado relacionadas con el tipo de elemento adicionado 1. <<trace>> (<<axisTracing>>Analysis::Package::ClassModel::B rowserAuction, <<axisTracing>>Analysis::Package::UseCase::Brows erAuctions) 2. <<trace>> (<<axisTracing>>Analysis::Package::Class::ClassM odel::BrowserAuction BrowserAuction, <<axisTracing>>Analysis::Package::UseCase:: UseCase::BrowserAuctions BrowserAuction) 3. <<refine>> (<<axisTracing>>Analysis::Package::UseCase::UseC ase::BrowserAuctions BrowserAuction, <<predecessor>> Requirement::Package::Requirement::Requirement:: BrowserAuction BrowserAuction) 4. <<realize>> (<<successor>>Design::Package::Collaboration::Co </pre>

	<pre> llaboration::BrowserAuction BrowserAuctions, <<axisTracing>> Analysis::Package::UseCase:: UseCase::BrowserAuction BrowserAuctions) ... </pre>
--	---

- Si la operación de cambio OpC es Modificar o Borrar y el cambio debe ser propagado ($P=1$) entonces se identifican los vínculos de trazado relacionados con cada elemento de modelo registrado en $\{OldSEi\}$ son seleccionados con el fin de ser actualizado o eliminado en una nueva versión de la vista de modelo de trazado. La Tabla 32 muestra de forma general de búsqueda y la acción para la propagación de un cambio en las instancias de vínculos de trazado que se usarán para actualizar los elementos afectados por el cambio en los modelos de desarrollo. En la Tabla 33 se muestra la identificación de instancias para la propagación de operación de eliminación.

Tabla 32. Identificación de instancias vínculos de trazado para una operación de Modificar.

Requisitos de Cambio	Instancias de vínculos de trazado
<pre> <<Change>> { C010, "Change the use case name from "A seller initiates an auction" to new name "Initiate an auction" ", Alter, "<<tracingModel>>Auction System v.1.0.0", "Analysis", "{Package::UseCase::Sell Goods A seller initiates an auction Packages}", "{Package::UseCase::Sell Goods Sell Goods", "1" } </pre>	<p>Call <i>retrieveTracingLink(Package::UseCase::Sell Goods A seller initiates an auction)</i> Si encuentra al menos una instancia Por cada instancia de vínculo de trazado y secuencia de trazado encontrados con el nombre de "A seller initiates an auction" se cambian por Sell Goods</p>

Tabla 33. Identificación de instancias de vínculos de trazado para una operación de Eliminar.

Requisitos de Cambio	Instancias de vínculos de trazado
<pre> <<Change>> { C012, Delete, "Delete the use case Sell Goods", "<<tracingModel>>Auction System v.1.0.1", "Analysis", "{<<trace>>(Class::Selle r, UseCase::Select the winning bid to make the sell}", "{}", "1" } </pre>	<p>Call <i>retrieveTracingLink(Package::UseCase::Sell Goods)</i> Si encuentra al menos una instancia Las instancias de vínculo de trazado y secuencia de trazado encontrados con el nombre "Sell Goods" serán marcadas para la eliminación y se genera un reporte de posibles inconsistencias con la eliminación del elemento de modelo.</p>

Propagación del Cambio. Las instancias de los vínculos de trazado que han sido creadas, actualizadas o marcadas para ser borradas determinan la propagación del cambio en los modelos de

desarrollo por medio de las reglas que ejecuta el planificador de transformación (como se explicó en el Capítulo 4). Además, hacen que el cambio sea incremental y no invasivo afectando sólo los elementos de modelo que tienen alguna relación el elemento de modelo alterado por el cambio.

Control de Versiones. Crear una nueva instancia (vista) de modelo de trazado requiere el soporte de un sistema versiones que facilite el control del cambio. El equipo de desarrollo debe hacer uso del sistema que comúnmente utiliza teniendo en cuenta los siguientes criterios para el control de versiones:

- La primera vista del modelo de trazado se considera la línea base sobre la cual se valoran y hacen los cambios durante el desarrollo de la aplicación o subsistema del proyecto de software.
- Cambios a vistas línea base de modelo de trazado se deben identificar en las siguientes categorías:
 - Fuertes, los cuales generan una nueva versión para todo el sistema haciendo que todas las vistas de modelo de trazado adopten el nuevo número de versión (esto puede verse como una nueva liberación del sistema). Esto podría ser porque el cambio afecta la mayoría de vistas de modelo de trazado de diferentes subsistemas. Por ejemplo en el sistema de subastas sería así: *Auction System v.1.0.0.* cambia a *Auction System v.2.0.0.* Esta categoría genera una nueva línea base.
 - Medios, los cuales sólo cambian la versión de las vistas de modelo de trazado para algunos subsistemas. Por ejemplo, en el sistema de subastas la versión de los subsistemas *Auction System::Sell Goods v.1.0.0.* y *Auction System::Buy Goods v.1.1.0.* cambian a *Auction System::Sell Goods v.1.1.0.* y *Auction System::Buy Goods v.1.1.0.*
 - Las vistas de modelo de trazado de otros subsistemas pueden no cambiar de versión ya que no son afectadas por el cambio.
 - Débiles, los cuales sólo afectan una vista de modelo de trazado de un subsistema específico. Por ejemplo, en el sistema de subastas la versión del subsistema *Auction System::Sell Goods v.1.0.0.*, cambia a la versión *Auction System::Sell Goods v.1.0.1.*

Una vez el planificador de transformación haga las debidas operaciones y genere los reportes de consistencia y completitud, la solicitud de gestión del cambio se puede complementar automáticamente con el estado básico del cambio y su propagación. Por ejemplo, la Figura 106 muestra un ejemplo de la información obtenida cuando se realiza el cambio *C001* y conjuntamente

se estaría mostrando los nuevos elementos de de los modelos afectados por el cambio y la nueva vista de modelo de trazabilidad

4. Impacto de la Propagación del Cambio (ítems de configuración)	
Trazas Activas	10 – ver lista anexa (¡Error! No se encuentra el origen de la referencia.)
Reglas Ejecutadas	40 (en promedio) – ver lista en el Capítulo 4 (Tabla 5)
Total nuevos elementos de modelo creados	10
Total elemento de modelo existentes afectados	0
Total de Relaciones afectadas	0
Medición del Impacto (Alto/medio/bajo)	---
Nueva Versión	---

Figura 106. Un ejemplo de la información que provee el impacto de la propagación del cambio en la Solicitud de Cambio.

7.4 Estimación de realización del Cambio

La estimación del esfuerzo necesario para realizar un cambio se define con base en la estimación del esfuerzo de transformación definido en el capítulo anterior adicionando otros factores relacionados a los recursos humanos, operativos y técnicos. Durante la transformación de los modelos en el contexto de MDD se estandarizada la generación de modelos de forma automática o semi-automática y consecuentemente algunas actividades del proceso de desarrollo. Esto puede llevar a eliminar algunos aspectos que ahora se consideran importantes para la estimación del esfuerzo de desarrollo y mantenimiento.

Por ejemplo, la estimación del esfuerzo basada en casos de uso clasifica los actores y casos de uso como simple, promedio y complejo y de acuerdo a un peso asignado estima su implementación; estos se soportan en los factores técnicos y factores de experiencia que ayudarán a lograr una estimación más exacta del desarrollo o cambio a realizar. Esta valoración es subjetiva ya que depende del tipo de proyecto y la experiencia del grupo de desarrollo.

La transformación de modelos controlada por los Modelos de Trazado ayuda a disminuir factores de riesgo asociados a la subjetividad en la toma de decisiones de modelado. Por ejemplo, si la transformación se controla por el Modelo de Trazado basado en Casos de Uso se tienen en cuenta las siguientes variantes de estimación al modelo tradicional de estimación del esfuerzo basado en casos de uso:

- ☞ Si el actor es generado por transformación, el esfuerzo se asocia a las # de reglas ejecutadas para un vínculo de trazado específico. De lo contrario se define el esfuerzo de crearlo por el método de estimación por casos de uso [Schneider and Winters 2001], [Carroll 2005].

- ☞ Si el caso de uso es generado por transformación el esfuerzo se asocia a las # de reglas ejecutadas para un vínculo de trazado específico. De lo contrario se define el esfuerzo de crearlo por el método de estimación por casos de uso
- ☞ Si se generan modelos alternativos durante el proceso de transformación, se define el esfuerzo de crearlo por el método de estimación por casos de uso con los “factores de experiencia ponderados”.
- ☞ De igual forma los totales del esfuerzo de transformación se deben computar con el factor de riesgo asociado a la transformación.

En el caso que la transformación de los modelos sea controlada por elementos diferentes a los casos de uso, podría cambiar por métodos más comunes tales como los puntos de función [Albrecht 1979] o COCOMO [Boehm et al. 2000], adaptándolo a la ejecución de las reglas de transformación.

7.5 Trabajos Relacionados

Zhao analiza el impacto del cambio desde la programación orientada por aspectos usando la técnica de *program slicing*. El análisis de impacto del cambio es la tarea por cual los programadores pueden evaluar el grado del cambio, es decir, el componente de software que afectará el cambio, o será afectado por el cambio. El análisis de impacto de cambio proporciona técnicas para direccionar el problema identificando el efecto de réplica probable de cambios de software y usando esta información para re-ingeniar de nuevo el diseño de sistema de software. Al construir un programa orientado por aspectos, es posible formular dos preguntas tales como ¿cuáles declaraciones podrían afectar una declaración en un programa orientado por aspectos? y ¿qué declaraciones podrían ser afectadas por una declaración en un programa orientado por aspectos? El proceso del análisis de impacto de cambio de un programa orientado por aspecto es como encontrar, hacia delante y hacia atrás, algunos pedazos del programa comenzando desde una declaración cambiada [Zhao 2002]. La gestión del cambio desde los Modelos de trazado resuelve las preguntas que hace Zhao ya que tiene los vínculos de trazado para hacer la verificación del impacto.

Cleland-Huang et al. proponen el manejo del cambio a partir de eventos identificables. Define el impacto del cambio C introducido a un artefacto a desde dos puntos de vista. Un “cambio propuesto” implica que el análisis de impacto se realice para determinar cómo el cambio C impacta el sistema existente, mientras que un “cambio implantado” implica que todos los artefactos impactados y sus vínculos relacionados deban ser actualizados para reflejar el cambio. Un cambio propuesto no necesariamente resulta en un cambio implantado. Además, define artefactos trazables como aquellos elementos que están débilmente acoplados pero son vinculados a través de un evento

de servicio el cual crea un ambiente donde un cambio es manejado más eficientemente y los artefactos y sus vínculos son mantenidos en un estado restaurable. Un artefacto es crítico si este reside como un artefacto intermedio entre múltiples caminos de trazabilidad de requisitos, porque el fracaso de mantenerlo causaría el fracaso de mantener otros artefactos en niveles inferiores en el mismo camino de trazabilidad. Un artefacto está en estado consistente cuando su estado y el estado de sus vínculos exactamente representan el estado actual de la configuración del sistema [Cleland-Huang 2003]. Los Modelos de Trazado no diferencian entre tipos de cambios para conocer el tipo de propagación. Los tipos de cambio se limitan los eventos que pueden ocurrir por cada operación de cambio. Estos eventos determina la propagación del cambio incremental.

Berg, presenta un enfoque donde el análisis de impacto se basa en la trazabilidad de las dependencias entre los elementos de los artefactos de software. Para lograr esto propone un grafo de representación de de dependencias junto con una definición formal de casos específicos de dependencias tales como enmarañadas, esparcidas y transversales [Berg et al. 2006].

Kurtev et al. clasifican los casos de cambio y los analizan en función de la traza, las reglas de transformación a ser ejecutadas, y los elementos afectados en los modelos fuente y destino. Los cambios son incrementales y siempre son de la fuente al destino [Kurtev et al. 2007].

Goknil et al. presentan un análisis formal del impacto del cambio basado en las relaciones entre los requisitos. Para lograr esto proponen un metamodelo que facilita la creación de instancias de relaciones, stakeholders, y requisitos. La propuesta provee reglas de impacto del cambio y la distinción entre elementos impactados de forma directa o indirecta y elementos candidatos impactados [Goknil 2008].

Las empresas de desarrollo de software, que siguen la metodología de RUP, comúnmente definen su proceso de gestión del cambio y establecen plantillas específicas para formalizar los escenarios de cambio. En las plantillas se registra información referente a la iteración donde se realiza el cambio, las disciplinas afectadas, los participantes del cambio (cliente y grupo de desarrolladores), el contexto funcional con casos de uso afectados, los riesgos asociados a los cambios, y elementos de configuración afectados, tales como documentos y modelos de desarrollo. A partir de los modelos o las matrices existentes de trazabilidad, los desarrolladores evaluarán el impacto del cambio (costo-esfuerzo-beneficio) de tal forma que puedan diagnosticar y presupuestar de las actividades que se deberán realizar sobre modelos de desarrollo, documentos y productos que se liberarán.

7.6 Conclusiones del Capítulo

Este capítulo define un método para gestión del cambio controlada por los Modelos de Trazado. Estos garantizan la propagación del cambio en diferentes niveles de abstracción. Esto se logró a partir de la definición de un conjunto de características que determinen la gestión del cambio. Estas características identifican el cambio, le asocian una operación de cambio, identifica la última versión versionada del Modelo de Trazado sobre la cual se hace el cambio, los elementos de modelo del momento del cambio tanto anteriores como nuevos, etc.

La propagación del cambio, se logra identificando las instancias de los vínculos de trazado asociados al elemento directamente afectado por el cambio. Estos vínculos asociados a las secuencias de trazas, proveen las reglas de transformación para actualizar los modelos existentes o crear los no existentes. Esto evita la propagación invasivas o la falta de propagación por desconocimiento de las trazas existentes.

Las métricas de estimación del esfuerzo del cambio están relacionadas al esfuerzo de transformación de los modelos de desarrollo (definido en el Capítulo 4). Estas métricas se asocian a las métricas obtenidas por métodos tradicionales para la estimación del esfuerzo ya sea por punto de función o por casos de uso, ya que son los más usados en las empresas de desarrollo. Es claro que en este capítulo no se desarrolla una nueva teoría de estimación del esfuerzo de desarrollo sino que se identifican los diferentes elementos que proporcional los Modelos de Trazado para agilizar la estimación de los modelos actuales.