Diseño de Almacenes de Datos con UML

Sergio Luján-Mora

Resumen para la admisión de Tesis Doctorales en lenguas no oficiales

Resumen

Un almacén de datos (en inglés, *data warehouse*, DW) es un sistema de información complejo utilizado principalmente en el proceso de toma de decisiones mediante el uso de las aplicaciones de Procesamiento Analítico en Línea (*On-Line Analytical Processing*, OLAP). Hasta el momento, se han propuesto varias aproximaciones para acometer el diseño de las distintas partes de un almacén de datos, como pueden ser los esquemas conceptual y lógico del almacén de datos o los procesos de extracción, transformación y carga de datos (*Extraction-Transformation-Loading*, ETL). Sin embargo, en la actualidad no existe un método ampliamente aceptado para realizar el diseño de un almacén de datos que abarque todas sus fases e integre los distintos modelos utilizados en cada una de ellas de una forma coherente. En esta tesis, presentamos un método global para acometer el diseño de todas las fases y aspectos relevantes de los almacenes de datos, incluyendo las fuentes de datos operacionales, los procesos ETL y el propio esquema del almacén de datos.

Las principales ventajas de nuestra propuesta son: el uso de una notación estándar (el lenguaje de modelado UML) en los modelos utilizados en todas las fases de diseño, la integración de las distintas fases de diseño en un modelo simple y coherente, el uso de un mecanismo de agrupamiento (los paquetes de UML) que permite al diseñador estructurar los modelos en diferentes niveles de detalle, y la propuesta de una método para el diseño de almacenes de datos basado en principios ampliamente aceptados (basado en el método de desarrollo UP).

1. Introducción

A principios de los noventa, Inmon [7] definió el término "almacén de datos" (*Data Warehouse*, DW): "Un almacén de datos es una colección de datos orientados por temas, integrados, variables en el tiempo y no volátiles para el apoyo de la toma de decisiones". Un almacén de datos es "integrado" porque los datos que se introducen en el almacén de datos se obtienen de una variedad de fuentes de datos (sistemas heredados, bases de datos relacionales, ficheros COBOL, etc.). Los procesos de extracción, transformación y carga (*Extraction-Transformation-Loading*, ETL) son los responsables de la extracción de los datos a partir de las diversas fuentes de datos heterogéneas, su transformación (conversión, limpieza, etc.) y su carga en el almacén de datos.

Por otro lado, es ampliamente aceptado que los almacenes de datos, las bases de datos multidimensionales y las aplicaciones de Procesamiento Analítico en Línea (*On-Line Analytical Processing*, OLAP)

están basados en el modelado multidimensional. En estos últimos años, se han propuesto varias aproximaciones para acometer el diseño conceptual y lógico de los almacenes de datos y los sistemas multidimensionales, tales como [2, 5, 26, 33, 6, 32, 1], para representar las principales propiedades estructurales y dinámicas del modelado multidimensional. Sin embargo, ninguna de estas propuestas ha sido ampliamente aceptada como un modelo conceptual estándar para acometer el modelado multidimensional.

Sin embargo, dada la gran variedad de modelos utilizados en las distintas fases del diseño de los almacenes de datos, se hace absolutamente necesario desarrollar un método lo más estándar posible que proporcione guías de diseño para crear y transformar estos modelos durante la fase de desarrollo de un almacén de datos. En la actualidad, existen algunos intentos de proporcionar un método para el desarrollo de almacenes de datos. Sin embargo, desde nuestro punto de vista, ninguno de ellos cubre todas las fases y transformaciones necesarias para disponer de un método global y estándar para el diseño de almacenes de datos.

En esta tesis, presentamos un método de diseño global orientado a objetos que integra todas las fases de diseño de los almacenes de datos desde las fuentes de datos operacionales hasta la implementación, incluyendo la definición de los procesos ETL y los requisitos de usuario. Nuestro objetivo es combinar, en un conjunto de modelos relacionados, todo el análisis y diseño de los almacenes de datos desde las fuentes de datos operacionales hasta su implementación final. En nuestra aproximación, un modelo de almacén de datos está basado en la típica arquitectura de un sistema de almacén de datos [9] y se compone de cinco fases y tres niveles. Al contrario que otros métodos y propuestas para el modelado multidimensional y el diseño de almacenes de datos, nuestro método es independiente de cualquier implementación específica (relacional, multidimensional, orientado a objetos, etc.). Finalmente, nuestro método está basado en un lenguaje de modelado estándar (*Unified Modeling Language*, UML) [25] y en un método ampliamente aceptado (*Unified Process*, UP) [8], lo que evita que los diseñadores aprendan una nueva notación o lenguaje específicos para el diseño de almacenes de datos.

El resto de este resumen está estructurado como sigue. En la sección 2 se presenta un breve resumen de los métodos más relevantes presentados hasta el momento para acometer el diseño de los almacenes de datos. En la sección 3 mostramos de forma general los diagramas y mapeos que componen nuestro método, para luego en la sección 4 comentar los aspectos más significativos del método de diseño de almacenes de datos que proponemos. A continuación, en la sección 5 explicamos como aplicar nuestro método en el diseño de un almacén de datos. En la sección 6 resumimos las principales

aportaciones de esta tesis. En la sección 7 comentamos la producción científica fruto de esta tesis, que se compone de una serie de publicaciones presentadas en diferente congresos y revistas internacionales. Por último, la sección 8 presenta las principales conclusiones extraídas de la presente tesis y los trabajos futuros que se puede desarrollar a partir de ella.

2. Estado de la cuestión

En esta sección, presentamos un breve resumen de los métodos más relevantes para el diseño de los almacenes de datos.

En [10], se presentan varios casos de estudio significativos de *data marts* (almacenes de datos departamentales) a los que se aplica el esquema estrella (*star schema*) y sus variantes de copo de nieve (*snowflake*) y constelación de hechos (*fact constellation*) para realizar el modelado multidimensional. Además, propone lo que denomina una matriz de arquitectura de BUS para integrar el diseño de varios *data marts* y conseguir así un almacén de datos corporativo y global. Si bien consideramos estos trabajos como un referente fundamental en el modelado multidimensional, a nuestro juicio todos ellos adolecen de un método formal para el diseño de almacenes de datos. En [11], se presenta el ciclo de vida de un proyecto de almacén de datos, para el cual los autores proponen distintas herramientas y técnicas a seguir, sin que se proponga un método o modelo global y formal a lo largo de todo el proceso.

En [5], los autores proponen el *Dimensional-Fact Model* (DFM) como una notación propia para acometer el modelado conceptual de los almacenes de datos. Además, proponen un marco metodológico para definir un esquema conceptual a partir de los esquemas Entidad-Relación (ER) que representan las fuentes de datos operacionales. Analizando los aspectos puramente metodológicos, esta propuesta está únicamente enfocada al diseño conceptual y lógico del almacén de datos, ya que no considera un aspecto tan relevante como es el diseño de los procesos ETL. Además, los autores presuponen una implementación relacional del almacén de datos y que se dispondrá de los esquemas ER de todas las fuentes de datos operacionales, lo que por desgracia no sucede en muchas ocasiones.

En [2], los autores presentan el *Multidimensional Model* (MD), un modelo lógico para acometer el diseño de un almacén de datos y un método para construirlo a partir de los esquemas ER de las fuentes de datos operacionales. Aunque los pasos del método están definidos de una forma coherente y lógica, el diseño del almacén de datos está basado únicamente en las fuentes de datos operacionales, lo que a nuestro juicio es insuficiente puesto que estos sistemas son

eminentemente de consulta, por lo que también se tiene que tener en cuenta en su diseño los requisitos de usuario.

En [24] se propone de nuevo cómo construir el esquema estrella (y sus diferentes variantes) a partir de los esquemas conceptuales de las fuentes de datos operacionales de que disponga la empresa. Una vez más, presupone que las fuentes de datos están definidas mediante esquemas ER. Se diferencia de otras propuestas en que no propone su propia notación gráfica para el diseño conceptual del almacén de datos, sino que emplea ER.

Más recientemente, cabe destacar el trabajo [4], donde se propone otro método para el diseño de almacenes de datos. Este método está basado en el modelo MD IDEA y propone una serie de procesos que cubren el diseño conceptual, lógico y físico de un almacén de datos. Una de las principales ventajas con respecto a las propuestas anteriores es el hecho de que para extraer el esquema conceptual se tenga en cuenta, además de las fuentes de datos operacionales, los requisitos de usuario. Sin embargo, este método está enfocado al modelado de datos y no abarca otros aspectos del diseño de los almacenes de datos como puede ser el diseño de los procesos ETL.

Finalmente, en [3] se evalúan diversos métodos de diseño de almacenes de datos y se propone un nuevo método que destaca por tener en cuenta la gestión de los metadatos. Sin embargo, carece de un modelo que permita reflejar y documentar el diseño del almacén de datos, ya que únicamente enumera una serie de actividades que se tienen que realizar para la construcción del almacén de datos.

Por tanto y, según lo comentado anteriormente, consideramos que en la actualidad no existe un método estándar, formal y riguroso ampliamente aceptado que cubra de una forma integrada todas las fases de diseño de los almacenes de datos desde el modelado de datos, pasando por el diseño de los procesos ETL hasta la implementación final del almacén de datos.

3. Diseño de un almacén de datos

La arquitectura de un almacén de datos se suele representar como varias capas a través de las cuales circulan los datos, de modo que los datos de una capa se obtienen a partir de los datos de la capa previa [9]. A partir de esta arquitectura, consideramos que el desarrollo de un almacén de datos se puede estructurar en un marco integrado por cinco etapas y tres niveles que definen los diferentes diagramas empleados para modelar un almacén de datos, tal como se resume en la Figura 1.

- **Etapas**: distinguimos cinco etapas en la definición de un almacén de datos:

- Origen (*Source*): define los orígenes de datos del almacén de datos, como los sistemas de Procesamiento de Transacciones en Línea (*OnLine Transaction Processing*, OLTP), las fuentes de datos externas (datos sindicados, datos censales), etc.
- Integración (*Integration*): define el mapeo entre los orígenes de datos y el propio almacén de datos.
- Almacén de datos (*Data Warehouse*): define la estructura del almacén de datos.
- Adaptación (*Customization*): define el mapeo entre el almacén de datos y las estructuras empleadas por el cliente.
- Cliente (*Client*): define las estructuras concretas que son empleadas por los clientes para acceder al almacén de datos, como *data marts* o aplicaciones OLAP.

- **Niveles**: cada etapa se analiza desde tres niveles o perspectivas que se crean en el siguiente orden:
  - Conceptual: define el almacén de datos desde un punto de vista conceptual, es decir, desde el mayor nivel de abstracción y contiene únicamente los objetos y relaciones más importantes.
  - Lógico: abarca aspectos lógicos del diseño del almacén de datos, como la definición de las tablas y claves, la definición de los procesos ETL, etc.
  - Físico: define los aspectos físicos del almacén de datos, como el almacenamiento de las estructuras lógicas en diferentes discos o la configuración de los servidores de bases de datos que mantienen el almacén de datos.
- **Diagramas**: cada etapa o nivel necesita formalismos de modelado diferentes. Por lo tanto, nuestra aproximación se compone de 15 diagramas (5 etapas y 3 niveles), pero el diseñador del almacén de datos no necesita definir todos los diagramas en cada proyecto de almacén de datos. En nuestra aproximación, usamos UML [25] como lenguaje de modelado, porque su potencia expresiva es la suficiente para el modelado de todos los diagramas de nuestra aproximación. Pero como UML es un lenguaje de modelado general, necesitamos usar los mecanismos de extensión de UML para adaptarlo al dominio específico de los almacenes de datos.

En la Figura 1, mostramos los quince diagramas que forman nuestra aproximación. Para cada uno de ellos, se indica su nombre, el

	Source (S)	Integration	Data Warehouse (DW)	Customization	Client (C)
Conceptual	**SCS** Class diagram *Standard UML*	**DM** Class diagram *Data Mapping Profile*	**DWCS** Class diagram *Standard UML* *Multidimensional Profile*	**DM** Class diagram *Data Mapping Profile*	**CCS** Class diagram *Standard UML* *Multidimensional Profile*
Logical	**SLS** Class diagram *Different data modeling profiles*	**ETL Process** Class diagram *ETL Profile*	**DWLS** Class diagram *Different data modeling profiles*	**Exporting Process** Class diagram *ETL Profile*	**CLS** Class diagram *Different data modeling profiles*
Physical	**SPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**DWPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**CPS** Comp. & deploy. diagrams *Database Deployment Profile*

LEGEND: CS: Conceptual Schema, LS: Logical Schema, PS: Physical Schema, Comp. & deploy: Component and deployment

Figura 1: Diagramas de diseño de un almacén de datos

diagrama de UML que empleamos y la notación empleada (estándar o una extensión de UML mediante un perfil).

La principal ventaja de nuestra aproximación es que siempre empleamos la misma notación (basada en UML) para el diseño de los diferentes diagramas y las correspondientes transformaciones entre los diagramas de una forma integrada.

En las siguientes secciones, se explica con detalle los principales diagramas de nuestra aproximación.

3.1. Modelado conceptual del almacén de datos

Nuestro perfil de UML para el diseño conceptual de almacenes de datos según el modelado multidimensional permite representar las principales propiedades multidimensionales a un nivel conceptual, como son las relaciones muchos-a-muchos entre hechos y dimensiones, las dimensiones degeneradas, las jerarquías múltiples y de camino alternativo, etc. Nuestro perfil de UML se encuentra definido formalmente mediante reglas expresadas con *Object Constraint Language* (OCL) [25], que definen el correcto uso de los nuevos elementos de modelado, con lo que se evita un uso arbitrario del perfil.

Además, nuestro perfil también incluye el uso de los paquetes de UML. Gracias a ello, cuando se modelan almacenes de datos grandes y complejos, no estamos restringidos a diagramas de clases planos. Nuestra propuesta establece el proceso de diseño en tres niveles (la Figura 2 muestra un resumen de nuestra propuesta):

Nivel 1 : Definición del modelo. Un paquete representa un esquema estrella de un modelo multidimensional. En este nivel, una dependencia entre dos paquetes indica que los esquemas estrella comparten al menos una dimensión.

Nivel 2 : Definición de un esquema estrella. Un paquete representa un hecho o una dimensión de un esquema estrella. En este nivel, una dependencia entre dos paquetes de dimensión indica que las dimensiones comparten al menos un nivel en sus correspondientes jerarquías.

Nivel 3 : Definición de un hecho o dimensión. Se compone de un conjunto de clases que representan los niveles jerárquicos en un paquete de dimensión o el esquema estrella completo en el caso de un paquete de hecho.

En la Tabla 1 mostramos los seis estereotipos más representativos de nuestro perfil de UML para el diseño conceptual de un almacén de datos.

Concepto MD (Estereotipo)	Descripción	Icono
`StarPackage`	Paquetes de este estereotipo representan esquemas estrellas, compuestos de hechos y dimensiones; se emplea en el nivel 1	
`FactPackage`	Paquetes de este estereotipo representan hechos, compuestos de medidas y relacionados con las dimensions; se emplea en el nivel 2	
`DimensionPackage`	Paquetes de este estereotipo representan dimensiones, compuestas de jerarquías; se emplea en el nivel 2	
`Fact`	Clases de este estereotipo representan hechos, compuestos de medidas; se emplea en el nivel 3	
`Dimension`	Clases de este estereotipo representan dimensiones, compuestas de jerarquías; se emplea en el nivel 3	
`Base`	Clases de este estereotipo representan niveles de jerarquía en una dimensión; se emplea en el nivel 3	

Cuadro 1: Conceptos multidimensionales y su representación en UML

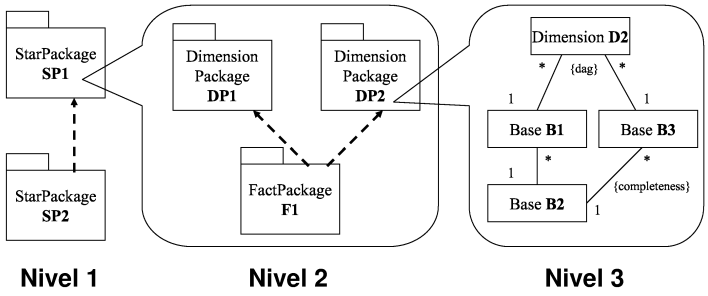

Figura 2: Los tres niveles de un modelo multidimensional representados mediante paquetes de UML

3.2. Diagrama de mapeo de datos

El diagrama de mapeo de datos (*Data Mapping*) es un nuevo tipo de diagrama adaptado para representar el flujo de datos, con varios niveles de detalle en un almacén de datos.

Para capturar las interconexiones entre los distintos elementos de diseño, en términos de los datos, empleamos la noción de *mapeo*. Un mapeo se define mediante tres elementos lógicos:

- El proveedor: una entidad (esquema, tabla o atributo) responsable de generar los datos que posteriormente se propagan.
- El consumidor: que recibe los datos del proveedor.
- El emparejamiento: que define la forma en la cual el mapeo se realiza, incluyendo cualquier tipo de transformación o filtrado.

Los mapeos se pueden definir con distintos niveles de granularidad: al nivel de esquema, tabla o atributo. En nuestra propuesta, el mapeo se establece a nivel de tabla/atributo entre las fuentes de datos (el SCS) y el almacén de datos (el DWCS), y entre el almacén de datos (el DWCS) y las estructuras empleadas por los clientes (el CCS).

Como un diagrama de mapeo de datos puede ser muy complejo, nuestra propuesta permite organizarlo en diferentes niveles gracias al uso de los paquetes de UML. Nuestra propuesta se compone de cuatro niveles (ver la Figura 3):

Nivel de base de datos (o **Nivel 0**). En este nivel, cada esquema del almacén de datos (por ejemplo, esquema de las fuentes de datos a nivel conceptual en el SCS, esquema conceptual del almacén de datos en el DWCS, etc.) se representa mediante un

paquete. Los mapeos entre los diferentes esquemas se modelan en un único paquete de mapeo, que encapsula todos los detalles.

Nivel de flujo de datos (o **Nivel 1**). Este nivel describe las relaciones de datos a nivel individual entre las fuentes de datos hacia los respectivos destinos en el almacén de datos.

Nivel de tabla (o **Nivel 2**). Mientras que el diagrama de mapeo en el nivel 1 describe las relaciones entre las fuentes y los destinos de datos mediante un único paquete, el diagrama de mapeo de datos en el nivel de tabla detalla todas las transformaciones intermedias que tienen lugar durante ese flujo.

Nivel de atributo (o **Nivel 3**). En este nivel, el diagrama de mapeo de datos captura los mapeos existentes a nivel de atributo.

En la parte más izquierda de la Figura 3, una única relación entre el DWCS y el SCS (representada mediante un único paquete llamado Data Mapping) y estos tres elementos de diseño constituyen el diagrama de mapeo de datos a nivel de base de datos (o Nivel 0). Suponiendo que existan tres tablas en el almacén de datos que se quieren poblar con datos, el paquete Data Mapping abstrae el hecho de que existen tres escenarios (Mapeo 1, Mapeo 2 y Mapeo 3), uno para cada una de las tablas. En el nivel de flujo de datos (o Nivel 1), se modelan mediante un paquete las relaciones de flujo de datos existentes entre las fuentes y los destinos de datos en el contexto de cada escenario. Si se explora con más detalle uno de estos escenarios, por ejemplo el llamado Mapeo 1, podemos observar las particularidades del mapeo: los datos de Fuente 1 se transforman en dos pasos (sufren dos transformaciones), como se muestra en la Figura 3. Se puede ver que existe un almacenamiento temporal denominado Intermedio, que almacena los datos generados por la primera transformación (Paso 1), antes de dirigirse a la segunda transformación (Paso 2). Por último, en la parte inferior derecha de la figura, se muestra como se realiza el mapeo a nivel de atributo entre Fuente 1 e Intermedio. De este modo, en el caso de que se esté modelando un almacén de datos grande y complejo, nuestra propuesta permite ocultar los detalles de la transformación de los atributos en el nivel 3.

Para representar los mapeos que proponemos, hemos desarrollado una extensión de UML mediante un perfil. Brevemente, los elementos de modelado que empleamos para realizar los mapeos en cada nivel son:

- Los diagramas de base de datos y de flujo de datos (niveles 0 y 1) emplean una notación estándar de UML. Más concretamente, en estos diagramas empleamos (a) los paquetes para modelar

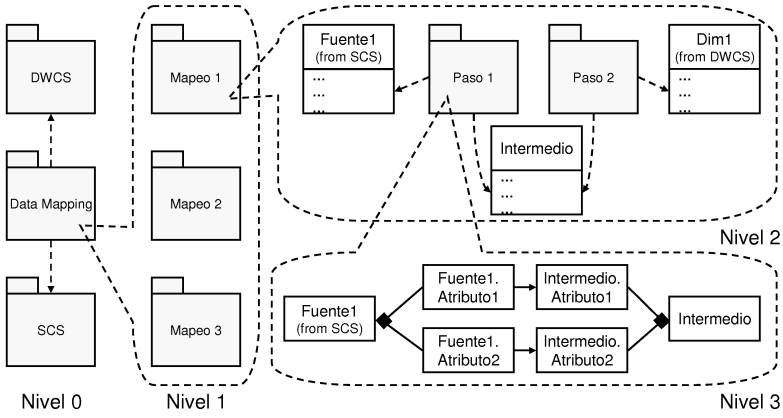

Figura 3: Niveles de mapeo de datos

las relaciones de datos y (b) dependencias entre los elementos involucrados. Las dependencias indican que los paquetes de mapeo son sensibles a los cambios en las fuentes y los destinos de datos.

- El diagrama de nivel de tabla (nivel 2) extiende UML con tres estereotipos: (a) ≪Mapping≫, empleado con paquetes para encapsular las relaciones entre los distintos almacenamientos de datos y (b) ≪Input≫ y ≪Output≫ para definir los distintos roles de proveedores y consumidores en un mapeo.

- El diagrama a nivel de atributo (nivel 3) emplea los estereotipos: ≪Map≫, ≪MapObj≫, ≪Domain≫, ≪Range≫, ≪Input≫, ≪Output≫ e ≪Intermediate≫.

3.3. Diseño de los procesos ETL

Hemos propuesto una extensión de UML que permite el modelado conceptual de los procesos ETL. Nuestra extensión proporciona los mecanismos necesarios para especificar las operaciones típicas de los procesos ETL, como la integración de distintas fuentes de datos, la transformación de los atributos, la generación de claves substitutas (*surrogate keys*), etc. Un proceso ETL se define combinando los distintos mecanismos que proporcionamos.

En nuestra propuesta, hemos definido un conjunto reducido pero potente de mecanismos ETL, con el fin de reducir la complejidad de nuestra propuesta y facilitar su uso.

Mecanismo ETL (Estereotipo)	Descripción	Icono
`Aggregation`	Agrega los datos (SUM, AVG, MAX/MIN, COUNT, etc.) en base a algún criterio	
`Conversion`	Cambia los tipos de datos, el formato o calcula nuevos datos (atributos derivados) a partir de los datos existentes	
`Filter`	Filtra los datos no deseados y verifica la calidad de los datos en base a restricciones	
`Incorrect`	Redirige los registros incorrectos o descartados a un destino separado para su posterior verificación; sólo se puede usar con `Filter`, `Loader` y `Wrapper`	
`Join`	Une dos fuentes de datos relacionadas entre sí a través de uno o varios atributos	
`Loader`	Carga los datos en el destino de un proceso ETL (en un hecho o dimensión del almacén de datos)	
`Log`	Controla y registra la actividad de otro mecanismo ETL, con el fin de auditar las transformaciones realizadas	
`Merge`	Integra los datos provenientes de dos o más fuentes de datos con atributos compatibles	
`Surrogate`	Genera una clave substituta única, que se emplea para reemplazar la clave empleada en las fuentes de datos	
`Wrapper`	Transforma una fuente de datos nativa en una fuente de datos basada en registros	

Cuadro 2: Mecanismos ETL y su representación en UML

En la Tabla 2 mostramos un resumen de los mecanismos de nuestra propuesta, en la que los mecanismos ETL se relacionan entre sí por medio de dependencias de UML. Además, a cada mecanismo se le puede añadir una nota de UML para explicar el funcionamiento del mecanismo y definir el mapeo entre los atributos en el origen y en el destino.

3.4. Diseño físico

En el diseño físico, empleamos los diagramas de componentes y de despliegue para modelar el nivel físico del almacén de datos. Para ello, proponemos los siguientes cinco diagramas, que se corresponden con las cinco etapas presentadas en la Figura 1:

- *Source Physical Schema* (SPS): define la estructura física de los orígenes de datos que alimentan el almacén de datos.
- *Integration Transportation Diagram* (ITD): define la estructura física de los procesos ETL empleados en la carga de datos en el almacén de datos desde los orígenes de datos. Se emplea para establecer la relación entre el diagrama anterior y el siguiente.
- *Data Warehouse Physical Schema* (DWPS): define la estructura física del almacén de datos.
- *Customization Transportation Diagram* (CTD): define los procesos de exportación desde el almacén de datos hacia las estructuras empleadas por los clientes. Se emplea para establecer la relación entre el diagrama anterior y el siguiente.
- *Client Physical Schema* (CPS): define la estructura física de las estructuras concretas que son empleadas por los clientes para acceder al almacén de datos.

El SPS, DWPS y CPS emplean los diagramas de componentes y de despliegue de UML, mientras que el ITD y el CTD emplean únicamente el diagrama de despliegue. Los cinco diagramas propuestos emplean una extensión de UML que hemos llamado *Database Deployment Profile* y que está formada por una serie de estereotipos, valores etiquetados y restricciones. Por falta de espacio, no incluimos en este artículo la definición formal de esta extensión.

4. Método de diseño de un almacén de datos

El método de diseño de almacenes de datos que proponemos, llamado *Data Warehouse Engineering Process* (DWEP), se basa en el

Proceso Unificado de Desarrollo de Software (*Unified Software Development Process*), también conocido como UP [8]. El UP es una estándar de la industria del software creado por los autores de UML (Grady Booch, Ivar Jacobson y James Rumbaugh) que define cuales son los artefactos, roles y prácticas principales que se tienen que realizar en un proyecto de software. Mientras que el UML define un lenguaje visual de modelado, el UP especifica cómo desarrollan productos software mediante UML. Por tanto, ambos se complementan y cada uno por separado no proporciona todo su potencia.

El UP define un proceso de ingeniería del software genérico que tiene que instanciarse para una organización, proyecto o dominio concretos. El método que nosotros proponemos es nuestra instanciación de UP para el desarrollo de almacenes de datos.

Tal como establece UP, el ciclo de vida de un proyecto se divide en cuatro fases (*Inception*, *Elaboration*, *Construction*, y *Transition*) y cinco actividades (*Requirements*, *Analysis*, *Design*, *Implementation*, y *Test*). Además, hemos añadido dos actividades adicionales a las cinco que establece UP: *Maintenance* y *Post-development review*. Durante el desarrollo de un proyecto, el interés se desplaza entre las distintas interacciones.

En la Figura 4 mostramos de forma resumida las fases y actividades de nuestra propuesta.

5. Aplicación del método

Un buen método no se compone únicamente de una notación gráfica, sino que también debe incluir una forma de usarlo. Los pasos que el diseñador debería seguir para la construcción de un almacén de datos mediante nuestro método son:

- **Análisis**:
  - Determinar requisitos iniciales: se define el alcance del almacén de datos mediante entrevistas con los usuarios finales; se revisan informes ya existentes y se recopilan los requisitos iniciales de los usuarios.
  - Definir reglas de negocio: se definen diversas reglas de negocio que se aplicarán en la construcción del almacén de datos (por ejemplo, la definición de medidas derivadas como "beneficio neto" o "porcentaje de devolución de producto").
  - Identificar fuentes de datos operacionales y externas: se definen las fuentes de datos, tanto operacionales como externas (datos económicos, censos de población, etc.), que

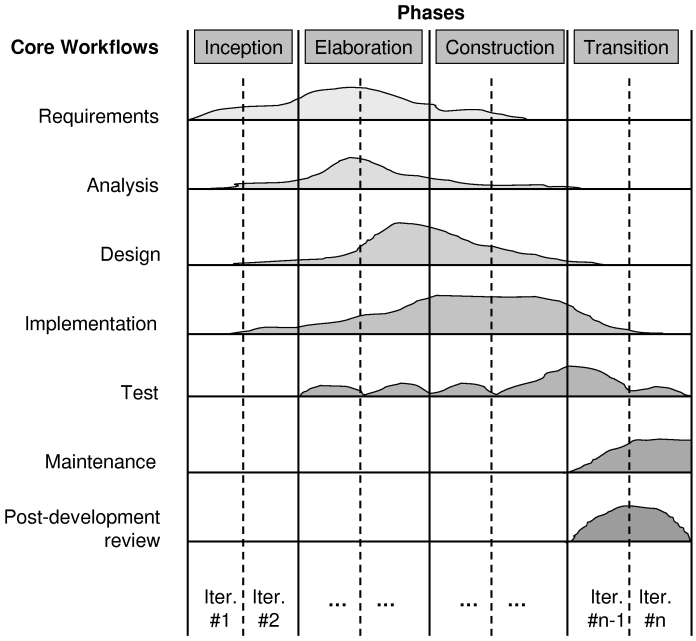

Figura 4: Fases y actividades de DWEP

alimentarán el almacén de datos. Para ello se tienen en cuenta las necesidades expresadas por los usuarios finales.

- **Diseño**:
  - Construir esquema conceptual: al final de esta actividad se obtiene el `DWCS`. Para acometer esta actividad existen dos estrategias "extremas", que se han representado mediante las dos transiciones con una línea discontinua: *top-down* (definir el almacén de datos según los requisitos de los usuarios finales), o *bottom-up* (definir el almacén de datos en base a los datos disponibles en las fuentes de datos).
  - Definir procesos ETL: se definen los procesos ETL como un mapeo entre las fuentes de datos (`SCS`) y el almacén de datos (`DWCS`); las reglas de negocio se aplican para calcular atributos derivados, definir transformaciones de atributos, etc. Esta actividad y la anterior definen un ciclo, ya que al crear los procesos ETL se puede detectar algún fallo en el `DWCS` (por ejemplo, que un atributo de una dimensión no exista en el `SCS`), por lo que será necesario modificar el `DWCS`.
  - Definir *data marts*: a partir de los requisitos iniciales de usuario y del `DWCS` se definen distintos `CCS`, que pueden implementarse como *data marts* reales o virtuales.
  - Definir informes: las consultas iniciales se definen mediante las clases cubo.
- **Implementación**:
  - Definir almacenamiento: se define el tipo de almacenamiento empleado para el almacén de datos (relacional, MD, OO, etc.) y se crea el correspondiente esquema lógico (el `DWLS`).
  - Definir procesos exportación: se define el mapeo entre el `DWCS` y el `DWLS`. Este mapeo puede definirse de forma manual o automática mediante una serie de algoritmos de exportación.
  - Implementar informes: los informes solicitados por los usuarios se implementan en la herramienta de consulta empleada (generalmente una aplicación OLAP).

En la Figura 5 mostramos los principales pasos de nuestro método mediante un diagrama de actividades de UML. El diagrama se ha dividido en dos calles (*swimlanes*) según quién guía las actividades descritas: Usuarios finales del DW (los usuarios finales "orientan" el

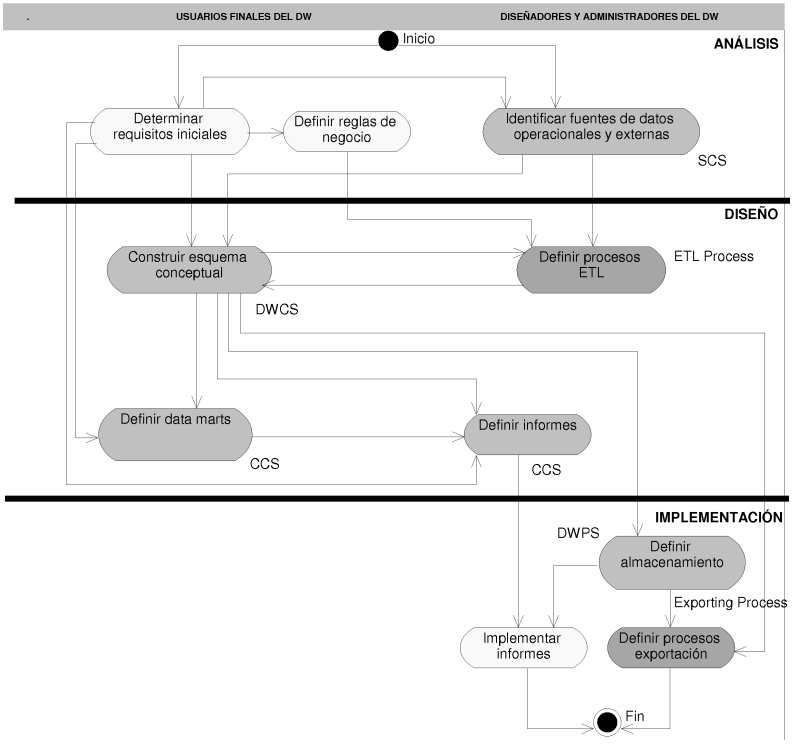

Figura 5: Diagrama de actividad con los principales pasos de aplicación del método

trabajo de los diseñadores y administradores del almacén de datos) y Diseñadores y administradores del DW (no necesitan de la participación de los usuarios finales, ya que disponen de toda la información necesaria para realizar su labor). Las actividades donde se aplican los modelos presentados en este artículo aparecen sombreadas: con un color claro aquellas actividades donde se crean los esquemas (se indica en una esquina el esquema creado) y con un color oscuro aquellas actividades donde se crean los mapeos entre esquemas. Además, las actividades se han dividido en tres grupos según la fase de creación del almacén de datos en la que participan: análisis, diseño e implementación. Por último, las transiciones definen un orden secuencial de las actividades y también indican el empleo de información procedente de otra actividad.

6. Principales aportaciones

Esta tesis contiene las siguientes aportaciones novedosas:

- La definición del *UML Profile for Multidimensional Modeling*, una extensión de UML mediante un perfil que permite modelar las principales propiedades multidimensionales de un almacén de datos a nivel conceptual.
- La definición de una extensión de UML que permite emplear los atributos como elementos de modelado de primer nivel.
- La definición del *Data Mapping Diagram*, un nuevo tipo de diagrama que permite reflejar el flujo de datos en un almacén de datos con varios niveles de detalle.
- La definición del *ETL Profile*, una extensión de UML para el modelado de procesos ETL
- La definición del *Database Deployment Profile*, una extensión de UML que permite modelar diferentes aspectos del nivel físico de un almacén de datos.
- El desarrollo de un add-in para Rational Rose que permite emplear nuestra propuesta con esta herramienta CASE.

7. Producción científica

Durante el desarrollo de esta tesis, los resultados obtenidos se han presentado en diferentes foros científicos: conferencias, revistas y capítulos de libros.

Todas las publicaciones realizadas pasaron un proceso de revisión por dos o más revisores cualificados que evaluaron la aportación de los trabajos y su calidad técnica.

A continuación se resume la producción científica desarrollada:

- Conferencia nacional (1): ADTO'01.
- Conferencia internacional (12): ICEIS'01, XMLDM'02, PHD-OOS'02, BNCOD'02, UML'02, ER'02, DMDW'03, ER'03, ICEIS'04, ADVIS'04, ER'04, DOLAP'04.
- Revistas internacionales (3): IJCIS'02, JDM'04, JDM'06.
- Capítulos de libros (1): IDEA'03.

Las principales publicaciones son:

- UML'02, donde se presentó la primera parte del *UML Profile for Multidimensional Modeling* que comprendía la definición de las principales propiedades del modelado multidimensional.
- ER'02, donde se presentó la segunda parte del *UML Profile for Multidimensional Modeling* que comprendía el empleo de los paquetes de UML para el modelado en tres niveles.
- ER'03, donde se presentó el *ETL Profile* para el modelado conceptual de los procesos ETL en un almacén de datos.
- JDM'04, donde se mostró cómo emplear los diagramas de clases, de estados y de interacción de UML para el modelado multidimensional.
- ER'04, donde se presentó el *Data Mapping Diagram* y una extensión de UML para el empleo de los atributos como elementos de modelado de primer nivel.
- JDM'06, donde se presenta el *Database Deployment Profile* para el modelado físico de los almacenes de datos.

A continuación se incluye la referencia bibliográfica completa de cada una de las publicaciones realizadas:

- S. Luján-Mora and E. Medina. Reducing Inconsistency in Data Warehouses. In *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01)*, pages 199–206, Setúbal, Portugal, July 7 - 10 2001. ICEIS Press
- J. Trujillo, S. Luján-Mora, and E. Medina. Utilización de UML para el modelado multidimensional. In *I Taller de Almacenes de Datos y Tecnología OLAP (ADTO 2001), VI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2001)*, pages 12–17, Almagro, Spain, November 22 2001
- S. Luján-Mora, E. Medina, and J. Trujillo. A Web-Oriented Approach to Manage Multidimensional Models through XML Schemas and XSLT. In *Proceedings of the XML-Based Data Management and Multimedia Engineering (EDBT 2002 Workshops)*, volume 2490 of *Lecture Notes in Computer Science*, pages 29–44, Prague, Czech Republic, March 24 2002. Springer-Verlag
- S. Luján-Mora. Multidimensional Modeling using UML and XML. In *Proceedings of the 12th Workshop for PhD Students in Object-Oriented Systems (PhDOOS 2002)*, volume 2548 of *Lecture Notes in Computer Science*, pages 48–49, Málaga, Spain, June 10 - 14 2002. Springer-Verlag

- E. Medina, S. Luján-Mora, and J. Trujillo. Handling Conceptual Multidimensional Models using XML through DTDs. In *Proceedings of 19th British National Conference on Databases (BNCOD 2002)*, volume 2405 of *Lecture Notes in Computer Science*, pages 66–69, Sheffield, UK, July 17 - 19 2002. Springer-Verlag

- S. Luján-Mora, J. Trujillo, and I. Song. Extending UML for Multidimensional Modeling. In *Proceedings of the 5th International Conference on the Unified Modeling Language (UML'02)*, volume 2460 of *Lecture Notes in Computer Science*, pages 290–304, Dresden, Germany, September 30 - October 4 2002. Springer-Verlag

- S. Luján-Mora, J. Trujillo, and I. Song. Multidimensional Modeling with UML Package Diagrams. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER'02)*, volume 2503 of *Lecture Notes in Computer Science*, pages 199–213, Tampere, Finland, October 7 - 11 2002. Springer-Verlag

- J. Trujillo and S. Luján-Mora. Automatically Generating Structural and Dynamic Information of OLAP Applications from Object-Oriented Conceptual Models. *International Journal of Computer & Information Science*, 3(4):227–236, December 2002

- S. Luján-Mora and J. Trujillo. A Comprehensive Method for Data Warehouse Design. In *Proceedings of the 5th International Workshop on Design and Management of Data Warehouses (DMDW'03)*, pages 1.1–1.14, Berlin, Germany, September 8 2003

- J. Trujillo and S. Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. In *Proceedings of the 22nd International Conference on Conceptual Modeling (ER'03)*, volume 2813 of *Lecture Notes in Computer Science*, pages 307–320, Chicago, USA, October 13 - 16 2003. Springer-Verlag

- J. Trujillo, S. Luján-Mora, and I. Song. *Advanced Topics in Database Research*, volume 2, chapter Applying UML for designing multidimensional databases and OLAP applications, pages 13–36. Idea Group Publishing, 2003

- J. Trujillo, S. Luján-Mora, and I. Song. Applying UML and XML for designing and interchanging information for data warehouses and OLAP applications. *Journal of Database Management*, 15(1):41–72, January-March 2004

- S. Luján-Mora, J. Trujillo, and P. Vassiliadis. Advantages of UML for Multidimensional Modeling. In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)*, pages 298–305, Porto, Portugal, April 14 - 17 2004. ICEIS Press
- S. Luján-Mora and J. Trujillo. A Data Warehouse Engineering Process. In *Proceedings of the 3rd Biennial International Conference on Advances in Information Systems (ADVIS'04)*, volume 3261 of *Lecture Notes in Computer Science*, pages 14–23, Izmir, Turkey, October 20 - 22 2004. Springer-Verlag
- S. Luján-Mora, P. Vassiliadis, and J. Trujillo. Data Mapping Diagrams for Data Warehouse Design with UML. In *Proceedings of the 23rd International Conference on Conceptual Modeling (ER'04)*, volume 3288 of *Lecture Notes in Computer Science*, pages 191–204, Shanghai, China, November 8 - 12 2004. Springer-Verlag
- S. Luján-Mora and J. Trujillo. Modeling the Physical Design of Data Warehouses from a UML Specification. In *Proceedings of the ACM Seventh International Workshop on Data Warehousing and OLAP (DOLAP 2004)*, pages 48–57, Washington D.C., USA, November 12 - 13 2004. ACM
- S. Luján-Mora and J. Trujillo. Physical Modeling of Data Warehouses by using UML Component and Deployment Diagrams: design and implementation issues. *Journal of Database Management*, 17(1), January-March 2006. Accepted to be published

8. Conclusiones y trabajos futuros

En este resumen hemos presentado un método basado en UML que permite modelar de forma integrada las distintas partes de un almacén de datos. La principal aportación de nuestro método es proporcionar un marco global que permite modelar todos los aspectos fundamentales de los almacenes de datos como son los esquemas conceptual y lógico, los procesos ETL, etc. Además, gracias el empleo de los paquetes de UML, nuestro método es escalable y permite abordar el diseño de almacenes de datos complejos. El aprendizaje de nuestro método se simplifica gracias al empleo de un lenguaje de modelado estándar como es UML. Por último, hemos proporcionado una serie de pasos que guían la aplicación de nuestro método.

Las principales ventajas que aporta nuestra propuesta son:

- Integridad del diseño de un almacén de datos, al abarcar con una serie de modelos el diseño completo de un almacén de datos.

- Trazabilidad del diseño de un almacén de datos, desde el modelo conceptual hasta el modelo físico.
- Reducción del coste de desarrollo, al abordar en fases iniciales aspectos de la implementación que pueden incurrir en un aumento del coste del proyecto de almacén de datos si se modifican en fases posteriores.
- Diferentes niveles de abstracción, al proporcionar varios niveles de detalles sobre el mismo diagrama.

Referencias

[1] A. Abelló, J. Samos, and F. Saltor. YAM2 (Yet Another Multidimensional Model): An Extension of UML. In *International Database Engineering & Applications Symposium (IDEAS'02)*, pages 172–181, Edmonton, Canada, July 17 - 19 2002. IEEE Computer Society.

[2] L. Cabibbo and R. Torlone. A Logical Approach to Multidimensional Databases. In *Proceedings of the 6th International Conference on Extending Database Technology (EDBT'98)*, volume 1377 of *Lecture Notes in Computer Science*, pages 183–197, Valencia, Spain, March 23 - 27 1998. Springer-Verlag.

[3] L. Carneiro and A. Brayner. X-META: A Methodology for Data Warehouse Design with Metadata Management. In *Proceedings of 4th International Workshop on the Design and Management of Data Warehouses (DMDW'02)*, pages 13–22, Toronto, Canada, May 27 2002.

[4] J.M. Cavero, M. Piattini, and E. Marcos. MIDEA: A Multidimensional Data Warehouse Methodology. In *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01)*, pages 138–144, Setubal, Portugal, July 7 - 10 2001. ICEIS Press.

[5] M. Golfarelli and S. Rizzi. A Methodological Framework for Data Warehouse Design. In *Proceedings of the ACM 1st International Workshop on Data Warehousing and OLAP (DOLAP'98)*, pages 3–9, Bethesda, USA, November 7 1998. ACM.

[6] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual Data Warehouse Modeling. In *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses (DMDW'00)*, pages 6.1–6.11, Stockholm, Sweden, June 5 - 6 2000.

[7] W.H. Inmon. *Building the Data Warehouse*. QED Press/John Wiley, 1992. (Last edition: 3rd edition, John Wiley & Sons, 2002).

[8] I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software Development Process*. Object Technology Series. Addison-Wesley, 1999.

[9] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, 2 edition, 2003.

[10] R. Kimball. *The Data Warehouse Toolkit*. John Wiley & Sons, 1996. (Last edition: 2nd edition, John Wiley & Sons, 2002).

[11] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons, 1998.

[12] S. Luján-Mora. Multidimensional Modeling using UML and XML. In *Proceedings of the 12th Workshop for PhD Students in Object-Oriented Systems (PhDOOS 2002)*, volume 2548 of *Lecture Notes in Computer Science*, pages 48–49, Málaga, Spain, June 10 - 14 2002. Springer-Verlag.

[13] S. Luján-Mora and E. Medina. Reducing Inconsistency in Data Warehouses. In *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01)*, pages 199–206, Setúbal, Portugal, July 7 - 10 2001. ICEIS Press.

[14] S. Luján-Mora, E. Medina, and J. Trujillo. A Web-Oriented Approach to Manage Multidimensional Models through XML Schemas and XSLT. In *Proceedings of the XML-Based Data Management and Multimedia Engineering (EDBT 2002 Workshops)*, volume 2490 of *Lecture Notes in Computer Science*, pages 29–44, Prague, Czech Republic, March 24 2002. Springer-Verlag.

[15] S. Luján-Mora and J. Trujillo. A Comprehensive Method for Data Warehouse Design. In *Proceedings of the 5th International Workshop on Design and Management of Data Warehouses (DMDW'03)*, pages 1.1–1.14, Berlin, Germany, September 8 2003.

[16] S. Luján-Mora and J. Trujillo. A Data Warehouse Engineering Process. In *Proceedings of the 3rd Biennial International Conference on Advances in Information Systems (ADVIS'04)*, volume 3261 of *Lecture Notes in Computer Science*, pages 14–23, Izmir, Turkey, October 20 - 22 2004. Springer-Verlag.

[17] S. Luján-Mora and J. Trujillo. Modeling the Physical Design of Data Warehouses from a UML Specification. In *Proceedings of the ACM Seventh International Workshop on Data Warehousing and OLAP (DOLAP 2004)*, pages 48–57, Washington D.C., USA, November 12 - 13 2004. ACM.

[18] S. Luján-Mora and J. Trujillo. Physical Modeling of Data Warehouses by using UML Component and Deployment Diagrams: design and implementation issues. *Journal of Database Management*, 17(1), January-March 2006. Accepted to be published.

[19] S. Luján-Mora, J. Trujillo, and I. Song. Extending UML for Multidimensional Modeling. In *Proceedings of the 5th International Conference on the Unified Modeling Language (UML'02)*, volume 2460 of *Lecture Notes in Computer Science*, pages 290–304, Dresden, Germany, September 30 - October 4 2002. Springer-Verlag.

[20] S. Luján-Mora, J. Trujillo, and I. Song. Multidimensional Modeling with UML Package Diagrams. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER'02)*, volume 2503 of *Lecture Notes in Computer Science*, pages 199–213, Tampere, Finland, October 7 - 11 2002. Springer-Verlag.

[21] S. Luján-Mora, J. Trujillo, and P. Vassiliadis. Advantages of UML for Multidimensional Modeling. In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)*, pages 298–305, Porto, Portugal, April 14 - 17 2004. ICEIS Press.

[22] S. Luján-Mora, P. Vassiliadis, and J. Trujillo. Data Mapping Diagrams for Data Warehouse Design with UML. In *Proceedings of the 23rd International Conference on Conceptual Modeling (ER'04)*, volume 3288 of *Lecture Notes in Computer Science*, pages 191–204, Shanghai, China, November 8 - 12 2004. Springer-Verlag.

[23] E. Medina, S. Luján-Mora, and J. Trujillo. Handling Conceptual Multidimensional Models using XML through DTDs. In *Proceedings of 19th British National Conference on Databases (BNCOD 2002)*, volume 2405 of *Lecture Notes in Computer Science*, pages 66–69, Sheffield, UK, July 17 - 19 2002. Springer-Verlag.

[24] D.L. Moody and M.A.R. Kortink. From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. In *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses (DMDW'01)*, pages 5.1–5.12, Stockholm, Sweden, June 5 - 6 2000.

[25] Object Management Group (OMG). Unified Modeling Language (UML) Specification 1.5. Internet: http://www.omg.org/cgi-bin/doc?formal/03-03-01, March 2003.

[26] C. Sapia, M. Blaschka, G. Höfling, and B. Dinter. Extending the E/R Model for the Multidimensional Paradigm. In *Proceedings of the 1st International Workshop on Data Warehouse and Data Mining (DWDM'98)*, volume 1552 of *Lecture Notes in Computer Science*, pages 105–116, Singapore, November 19 - 20 1998. Springer-Verlag.

[27] J. Trujillo and S. Luján-Mora. Automatically Generating Structural and Dynamic Information of OLAP Applications from Object-Oriented Conceptual Models. *International Journal of Computer & Information Science*, 3(4):227–236, December 2002.

[28] J. Trujillo and S. Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. In *Proceedings of the 22nd International Conference on Conceptual Modeling (ER'03)*, volume 2813 of *Lecture Notes in Computer Science*, pages 307–320, Chicago, USA, October 13 - 16 2003. Springer-Verlag.

[29] J. Trujillo, S. Luján-Mora, and E. Medina. Utilización de UML para el modelado multidimensional. In *I Taller de Almacenes de Datos y Tecnología OLAP (ADTO 2001), VI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2001)*, pages 12–17, Almagro, Spain, November 22 2001.

[30] J. Trujillo, S. Luján-Mora, and I. Song. *Advanced Topics in Database Research*, volume 2, chapter Applying UML for designing multidimensional databases and OLAP applications, pages 13–36. Idea Group Publishing, 2003.

[31] J. Trujillo, S. Luján-Mora, and I. Song. Applying UML and XML for designing and interchanging information for data warehouses and OLAP applications. *Journal of Database Management*, 15(1):41–72, January-March 2004.

[32] J. Trujillo, M. Palomar, J. Gómez, and I. Song. Designing Data Warehouses with OO Conceptual Models. *IEEE Computer, special issue on Data Warehouses*, 34(12):66–75, December 2001.

[33] N. Tryfona, F. Busborg, and J.G. Christiansen. starER: A Conceptual Model for Data Warehouse Design. In *Proceedings of the ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP'99)*, pages 3–8, Kansas City, USA, November 6 1999. ACM.