

Propuestas para el Mejoramiento de la Calidad del Diagrama de Clases de UML

CLAUDIA JIMÉNEZ RAMÍREZ

UNIVERSIDAD NACIONAL DE COLOMBIA. Facultad de Minas.

Escuela de Sistemas.

csjimene@perseus.unalmed.edu.co

Recibido para revisión 23 de Feb de 2004, aceptado 27 de Abr de 2004, versión final recibida Jun de 2004

Resumen: La elección de un buen modelo conceptual en el análisis de un sistema, es definitiva para la correcta especificación de los requisitos planteados en un dominio de aplicación en particular; pues, con toda seguridad, conducirán al desarrollo o evolución de sistemas informáticos de mayor satisfacción para los usuarios finales. El presente trabajo es fruto del análisis comparativo de varios modelos conceptuales usados en la ingeniería del software, específicamente los llamados modelos estructurales, como son: el Modelo Entidad - Relación extendido por Richard Barker, el Modelo de Objetos Semánticos de Hammer y McLeod, el Diagrama de Configuración de Clases de OO-Method y el Diagrama de Clases de UML. Para determinar las bondades de cada uno de estos modelos, se emplearon los siguientes criterios de calidad: riqueza semántica, simplicidad, rigor y suficiencia. En el análisis comparativo, primero se identificaron los conceptos que permiten representar cada modelo y, luego, mediante ejemplos se analizó la conveniencia, o no, de la inclusión de cada concepto. La comparación de los modelos permitió descubrir lo que ellos aportan y presentar una serie de propuestas para el mejoramiento de la calidad del Diagrama de Clases de UML.

Palabras Clave: Ingeniería de software, Modelos Conceptuales, Modelamiento de Datos, Modelo Estructural, Dominio del Problema.

Abstract: The election of a good conceptual model in the analysis of a system, is definitive for the correct specification of the requirements raised in a domain of application; since, with complete certainty, they will lead to the development or evolution of computer science systems, for greater satisfaction for the end users. The present work is result of the comparative analysis of several used conceptual models in the engineering of software, more specifically the calls structural models, as they are: Entity-Relation model extended by Richard Barker, the Model of Semantic Objects of Hammer and McLeod, the Diagram of Configuration of Classes of OO-Method and the Diagram of Classes of UML. In order to determine kindness of each one of these models, the following criteria of quality were used: semantic wealth, simplicity, rigor and sufficiency. In the comparative analysis, first the concepts were identified that allow to represent each model and, soon, by means of examples was analyzed the convenience of the inclusion of each concept. The comparison of the models allowed to discover what they contribute and display a series of proposals for the improvement of the quality of the Diagram of Classes of UML.

Keywords: Software engineering, Conceptual Models, Data Modeling, Structural Model, Domain of Problem

1 INTRODUCCIÓN

Tradicionalmente, en bases de datos, se ha utilizado el modelo Entidad-Relación como modelo estructural. Sin embargo, otros autores debido a su orientación, o buscando una alternativa mejor, han propuesto otros modelos de esa misma categoría para representar un dominio particular.

De la gama de modelos propuestos en los últimos años, el Diagrama de Clases de UML (Unified Modeling Language), fue acogido como el modelo estándar

por la OMG (Object Management Group) y avalado por empresas mundialmente reconocidas como Oracle Corporation y Microsoft (Rational, 1999). Por ello, se quiso comparar este modelo con su antecesor, el modelo Entidad- Relación, y con otros propuestos para detectar posibles diferencias significativas; tanto semánticas, como sintácticas que incidieran sobre la calidad del diagrama de clases de UML.

En la comparación de los modelos, se usó un ejemplo de la vida real: el Sistema de Gestión Académica de la Universidad Nacional, Sede Medellín pues este ofrece

cierta complejidad e incluye muchos aspectos que se deben tener en cuenta en el desarrollo de otros sistemas informáticos, de diversa índole.

2 EL PAPEL DE UN MODELO CONCEPTUAL

Los modelos conceptuales se construyen con el objetivo de obtener una representación de la realidad que capture la mayor semántica, tanto sobre las propiedades estáticas, como dinámicas del universo del discurso para poderle transmitir ese conocimiento al computador y, así, éste pueda darnos el apoyo que le solicitamos. Como el computador requiere de una especificación completa y libre de ambigüedades o imprecisiones para llevar a cabo una tarea, es necesario entregarle una visión de esa realidad procurando que no haya pérdidas significativas que incidan en la calidad del sistema informático.

Por lo anterior, la construcción de un modelo conceptual es el proceso que más esfuerzo requiere de todo el proceso de desarrollo de software; pudiendo considerarse las fases siguientes, como meras fases de transformación o refinamiento de ese modelo.

El ser humano cuando piensa realizar una obra, debe elaborar previamente un modelo, aunque sea sólo en su mente, que le permita concretar y plasmar sus ideas, o las de los demás. Con la comprensión de la realidad, a través de uno o varios modelos, se puede establecer la capacidad de acción o de intervención en ella para definir un mundo futuro; que se espera sea mejor que sin ese conocimiento.

El analista de software que, en muchos casos, no tiene un conocimiento profundo del dominio de la aplicación, debe comprender y capturar lo necesario de ese dominio usando el modelo verbal del usuario final o del experto y, a partir de ahí, debe construir los modelos conceptuales; donde queda especificado el "qué" del espacio del problema, sin importar todavía el "cómo" se llegará a la solución. Esta propiedad, permite que un modelo conceptual sea independiente de cualquier tipo de modelo lógico o de su implementación.

Los modelos conceptuales, también, tienen la valiosa utilidad de ser un medio de comunicación y validación de los requerimientos con el usuario final o experto. Actividad imprescindible para detectar carencias, errores o malas interpretaciones que conduzca a las correcciones en una etapa temprana del proceso de desarrollo o evolución; implicando, obviamente, costos menores.

Por todo lo descrito, los modelos conceptuales determinan el producto final y por lo tanto, tienen el valor de fundamentar, o restringir, los modelos lógico y físico. Los modelos conceptuales sirven entonces de guía en las etapas de diseño y desarrollo de las aplicaciones, más aún, si los encargados de estas dos últimas tareas no

han sido partícipes del análisis del sistema.

Por último, no se puede dejar de mencionar otra valiosa utilidad de los modelos conceptuales: los modelos que se tengan de los sistemas "viejos" sirven de guía imprescindible para la reingeniería o la evolución de los sistemas informáticos. De ahí que, su utilidad persista más allá de la aplicación.

3 CRITERIOS DE CALIDAD DE LOS MODELOS CONCEPTUALES

Con los planteamientos hechos sobre los posibles usos de un modelo conceptual, queda resaltada su importancia. No obstante, para considerar que un modelo es "bueno", éste debe caracterizarse por cumplir con ciertos criterios de calidad. Los criterios elegidos para el estudio, son:

- **Riqueza semántica o expresividad:** un modelo es rico semánticamente, si ofrece una amplia gama de conceptos para cubrir la mayor cantidad de aspectos relevantes del dominio del problema.
- **Comprendibilidad:** un modelo debe ser comprensible para facilitar la comunicación entre los usuarios finales y los modeladores. El modelo comprensible permite resaltar los detalles importantes del sistema para plantear la solución y, así, poder transferirle ese conocimiento al computador.
- **Formalidad o Rigor:** un modelo es formal, si cada concepto tiene una interpretación única, precisa y bien definida. Esta cualidad determina la capacidad de traducción o transformación de un modelo a otro, de más bajo nivel de abstracción.
- **Simplicidad:** un modelo simple es legible o comprensible. Infortunadamente esta cualidad casi siempre está en conflicto con la expresividad. Por eso, siempre se busca un punto de equilibrio entre ambas cualidades; buscando la parsimonia en el modelo.
- **Suficiencia o Minimalidad:** Un modelo es mínimo o suficiente, si cada concepto modelado no puede ser deducido de otros conceptos ya incluidos en el modelo. Si se tiene un modelo suficiente; se incrementa la probabilidad de tener un modelo de la solución correcta; ya que se minimizará la posibilidad de especificar contradicciones. Esta cualidad incide positivamente, también, en la simplicidad y comprensibilidad del modelo.

4 LOS MODELOS CONCEPTUALES BAJO ESTUDIO

Dentro de los modelos conceptuales, se suele denominar "estructural" o "estático", a aquel modelo que permita

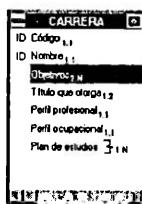


Figura 1: Objeto Semántico

representar la estructura de los objetos del dominio del problema y sus relaciones con los demás objetos. Por lo tanto, un modelo estructural sirve para representar la anatomía del sistema, principalmente. Los modelos elegidos, dentro de esta categoría, para el análisis comparativo son: el diagrama Entidad - Relación extendido por Barker (1991), el Diagrama de Clases de UML (Rational, 1999) y (Fowler y Scott, 1997), el Modelo de Objetos Semánticos de Hammer y Dennis (1995) y Kroenke (1996) y, por último, el Diagrama de Configuración de Clases de OO-Method (Pastor López, 1998), (Pastor, Insfran y Pelechano, 1998).

5 APORTES DE LOS MODELOS

A pesar de la existencia de varios conceptos comunes en los modelos, se observó que los estudiados ofrecían elementos que podrían enriquecer al Diagrama de Clases de UML. En ciertos casos, la inclusión de algunos conceptos nuevos para el lenguaje UML podría ofrecer, no sólo una ganancia semántica y aumentar el rigor, sino también reducir significativamente su complejidad. A continuación, se presentan los conceptos que se pueden considerar aportes de cada uno de ellos.

5.1 Conceptos faltantes en el Modelo de Clases de UML

- **Cardinalidad mínima y máxima de los atributos:** El Modelo de Objetos Semánticos permite representar la cardinalidad mínima y máxima de un atributo; admitiendo que un atributo de un objeto puede poseer múltiples valores. Para expresar la cardinalidad mínima y máxima, el Modelo de Objetos Semánticos usa subíndices m, n colocados como sufijos a los nombres de los atributos; tal como lo ilustra la Figura 1. Si el Diagrama de Clases de UML, incluyera el concepto de cardinalidad mínima y máxima de los atributos, no sólo mejoraría su riqueza semántica, al permitir especificar restricciones sobre otros aspectos de la realidad que son necesarios transmitir al modelo físico, sino también permitiría su simplificación porque evitaría tener

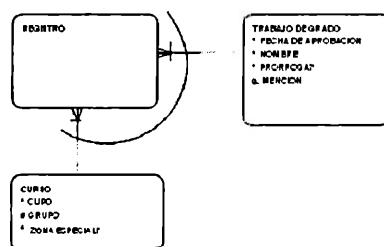


Figura 2: Relación Exclusiva en el diagrama E-R

que ascender un atributo multivaluado a la categoría de clase y estableciendo una relación $m : 1$ con la entidad que lo posciba, o especificando varios atributos con el mismo significado, en una clase.

• **Relación exclusiva:** es un concepto incluido en el Modelo Entidad - Relación para definir un tipo de relación especial existente entre una entidad con otras dos, o más, entidades. Se caracteriza porque cuando se establece la relación con una de las entidades, inmediatamente se excluyen las relaciones con las demás entidades participantes en esa relación. Por ejemplo, la relación entre un REGISTRO ACADÉMICO con un CURSO y un TRABAJO DE GRADO, es una relación exclusiva porque, unas veces el registro académico se elabora para un curso y otras veces, para un trabajo de grado; pero no para ambos, en forma simultánea. La notación empleada, en este modelo, para representar una relación exclusiva es un arco que intercepta las líneas de las relaciones participantes como lo ilustra la Figura 2. El arco, como adorno de la relación de exclusividad, agrega precisión al modelo y advierte a los diseñadores para que tomen medidas especiales para impedir que se viole esta regla. Sin el arco, parece que un registro puede relacionarse, al tiempo, con un curso y con un trabajo de grado.

• **Relación intransferible.** Este calificativo dado a una relación en el modelo Entidad - Relación, permite especificar que ésta, una vez establecida, nunca cambia o se transfiere a otro objeto de su misma clase. Un ejemplo de este tipo de relación, es la existente entre un estudiante y un préstamo condonable; puesto que no es posible que, después de concedérsele un préstamo a un estudiante, luego pase a figurar a nombre de otro. Obsérvese, entonces, que este concepto es necesario incorporarlo al lenguaje de UML para que permita expresar este tipo de restricción en una relación. La sintaxis usada para representar una relación intransferible es un rombo que se coloca sobre la línea de la relación.

- **Atributo objeto-valorado.** Un atributo es objeto-valorado cuando él, a su vez, contiene atributos. El Diagrama de Clases admite la representación de este tipo de atributo pero no lo distingue de los demás, ni hace observaciones especiales sobre éste. Además, el modelo fuerza a representar el atributo objeto-valorado como una clase separada para poder especificar sus atributos.

El modelo de Objetos Semánticos, por su lado, ofrece el término de atributo compuesto que se puede considerar sinónimo de objeto-valorado y la notación que usa para diferenciarlo es un corchete colocado a la derecha del atributo; que puede expandirse hasta mostrar los atributos atómicos que constituyen la composición. La Figura 1, muestra el plan de estudios como un atributo compuesto de la carrera.

La conciencia de la existencia de atributos objeto-valorados evita la representación de relaciones "compuesto por" o de agregación, que se caracteriza regularmente por ser una relación inclusiva y estricta. Una relación se define, en OO-Method, como inclusiva si el acceso a la clase componente, siempre es a través de la clase compuesta y se considera estricta, si el objeto componente no puede vivir independientemente de la vida del agregado.

Por lo anterior, la representación de atributos objeto-valorados, evita la descomposición poco natural de los objetos, aumentando la simplicidad del modelo. Sin embargo, cuando se admite la representación atributos de tipo objeto, se tiene también que permitir representar sus relaciones relevantes con otros objetos; tal como la relación existente entre el plan de estudios, atributo de la carrera, con las asignaturas. Luego, el diagrama de Clases de UML, debería admitir la posibilidad de representar relaciones con atributos, además de las relaciones entre clases, para aumentar el rigor del modelo.

- **Unicidad de los atributos.** Un atributo, o un conjunto de atributos, se califica como único cuando éste debe tomar un valor diferente para cada instancia de una clase. Esta restricción evita la duplicidad de información, que haga pensar que existen más objetos de los que realmente hay, en ese dominio particular.

El modelo Entidad - Relación permite la especificación de un identificador único para cada entidad, mientras en el Modelo de Objetos Semánticos se pueden especificar varios identificadores únicos; tal como lo muestra la Figura 1 donde aparecen como identificadores únicos, el código y el nombre para una carrera. Pero en este último modelo, falta rigor porque no es claro si los atributos son únicos en

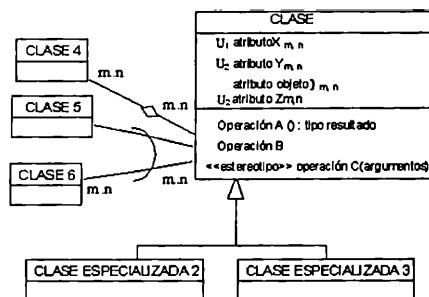


Figura 3: Diagrama de Clases de UML, mejorado

forma conjunta, o en forma separada. Por lo tanto, se propone utilizar un subíndice distinto para cada atributo o conjunto de atributos únicos.

Adicionalmente, se sugiere reemplazar la visibilidad de un atributo que se clasifica en el Diagrama de Clases UML como pública, privada, protegida o de implementación, que es un detalle de implementación en un lenguaje de programación orientado por objetos, por la unicidad y, así, se evita recargar el diagrama de clases con adornos innecesarios.

El Diagrama de Clases de UML, con todas estas propuestas presentadas para mejorar su calidad, tendría una notación similar a la que aparece en la Figura 3.

6 CONCLUSIONES

El análisis comparativo de los modelos conceptuales, conduce a plantear las conclusiones siguientes: Un mismo fenómeno puede ser representado por modelos diferentes; pero cada modelo omite diferentes aspectos del fenómeno, representa con diferente precisión los conceptos incluidos o refina para mayor rigor y expresividad algunos conceptos; El Diagrama de Clases, aún es susceptible de mejorar; Un buen modelo conceptual, que intenta representar fielmente un dominio del problema; dependerá no sólo de las cualidades del modelo elegido sino, también, de las habilidades de los modeladores para la buena utilización de las herramientas conceptuales.

REFERENCIAS

- Barker, R. (1991), *Case Method: Entity-Relationship Modelling*, Oracle Press.
- Fowler, M. y Scott, K. (1997), *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley.

- Hammer, M. y Dennis, M. (1995), 'Database description with sdm: A semantic database model', pp. pp. 351-386.
- Kroenke, D. M. (1996), *Procesamiento de Bases de datos*, quinta edición edn, Prentice Hall Hispanoamericana.
- Pastor López, O. (1998), 'Oo-method. un método orientado a objetos para la producción automática de código', *Notas de curso doctoral* . Second Printing.
- Pastor, O., Insfran, E. y Pelechano, V. (1998), 'Oo-method: An oo software production environment combining conventional and formal methods', *V Jornadas Iberoamericanas de Informática* .
- Rational (1999), *Unified Modeling Language For Real-Time System Design*, Rational Software Corporation.
*<http://www.rational.com/pst/tech-papers/uml-rt.html>

