Operaciones para Conformar Esquemas Orientados a Objetos*

A.Abelló[1], F.Araque[2], C.Delgado[2], E.Garví[2], M.Oliva[3], E.Rodríguez[1], E.Ruiz[2], F.Saltor[1], J.Samos[2] y M.Torres[4]

[1] *U. Politècnica de Catalunya* [2]*U. de Granada* [3]*U. de Lleida* [4]*U. de Almería*

Palabras clave: Integración de Información, Sistemas de Información Cooperativos, Bases de Datos Federadas, Bases de Datos Orientadas a Objetos.

1 Introducción[1]

Un "Sistema de Información Cooperativo" (SIC) se construye sobre un conjunto de sistemas de información ya existentes, y, en general, con "heterogeneidad sintáctica". Una forma de superar dicha heterogeneidad es utilizar para el SIC un "Modelo de Datos Canónico" (MDC), y traducir los diferentes esquemas desde sus modelos nativos al MDC [5]. De acuerdo con [4], en este trabajo asumiremos como MDC de un SIC un modelo Orientado a Objetos (OO).

En este artículo, proponemos un conjunto de operaciones para *conformar* esquemas OO, es decir, para cambiar la *forma* de un esquema. Estas operaciones de conformación las utilizaremos en el proceso de construcción de un SIC que consta de dos etapas: primera, cada *Esquema de Exportación* [5] se transforma a una forma común (conformación) y, segunda, todos los esquemas conformados se integran mediante alguna operación de "Generalización" [1] en un *Esquema Federado*. Las operaciones que presentamos son operaciones primitivas que tienen en cuenta la forma del esquema y no aumentan su "capacidad de información" [2]. En nuestra aproximación, el "enriquecimiento semántico" tiene lugar cuando se generan los *Esquemas Componentes* [1].

Utilizamos la terminología de UML, a la que añadimos restricciones de integridad para limitar la semántica de algunas operaciones, siendo nuestros resultados aplicables a cualquier modelo OO. Las operaciones de conformación que proponemos actúan en las dimensiones de *Generalización/Especialización* (G/E) y *Agregación/Descomposición* (A/D).

En la Figura 1 se muestra el diagrama de clases con las metaclases de UML relevantes para nuestro propósito. Aunque UML permite **Asociaciones** n-arias, consideramos únicamente **Asociaciones** binarias unidireccionales.

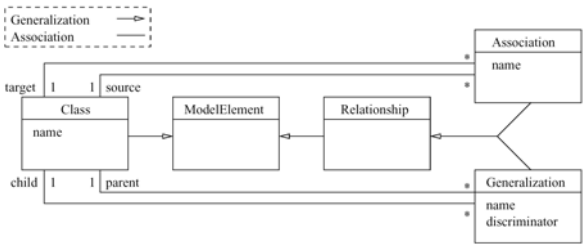


Figura 1. Subconjunto de metaclases de UML.

El resto de este artículo se ha organizado como sigue: la sección 2, representa el núcleo de este trabajo y en ella se presentan tanto las operaciones como los patrones que nos sirven de base para la conformación de esquemas; la sección 3 ofrece unas breves conclusiones y, finalmente, se citan las referencias.

* Este trabajo ha sido financiado parcialmente por el Programa de Investigación PRONTIC, bajo los proyectos TIC2000-1723-C02-(01 y 02).

[1] Utilizamos **negrita** para los términos UML, *cursiva* para los términos nuevos y entre "comillas" para los términos introducidos por otros autores.

2 Operaciones en Esquemas Orientados a Objetos

Un *patrón de interés* representa la información mínima necesaria para aplicar una operación de conformación determinada; como patrones candidatos consideramos digrafos donde los nodos representan **Clases** y los arcos **Relaciones** entre ellas (Figura 1). En un esquema aparecen múltiples ocurrencias de los patrones. Las operaciones de conformación se aplican sucesivamente a cada uno de estos patrones hasta obtener el esquema conformado; la conformación es la *composición* de esas operaciones, que es asociativa pero no conmutativa. Llamamos *primitivas* a las operaciones atómicas necesarias para *conformar* esquemas; las operaciones *derivadas* son las que se obtienen como composición de primitivas. Hemos establecido quince operaciones primitivas, y ninguna de ellas puede omitirse sin afectar al conjunto de operaciones derivadas.

Los patrones de interés y las primitivas pueden representarse en un digrafo G = (P, O), donde P es el conjunto de patrones y O es el conjunto de operaciones (Figura 2). Un arco o_{ij} de p_i a p_j representa la operación que convierte el patrón p_i en el patrón p_j. Distinguimos tres subgrafos correspondientes a las operaciones sobre G/E, A/D y entre dimensiones. El arco reflexivo sobre p_6 refleja la existencia de tres operaciones que actúan sobre el mismo patrón, dando lugar a tres formas diferentes (a, b, c) que pueden verse como casos especiales de él.

G es un grafo simétrico, ya que para cada arco o_{ij} existe un arco o_{ji} que representa la operación *opuesta*, y que puede ser o no su *inversa* matemática. Cuando o_{ij} reduce la capacidad de información, o_{ji} no puede recuperarla y produce un esquema diferente al inicial; sin embargo los patrones tienen la misma forma.

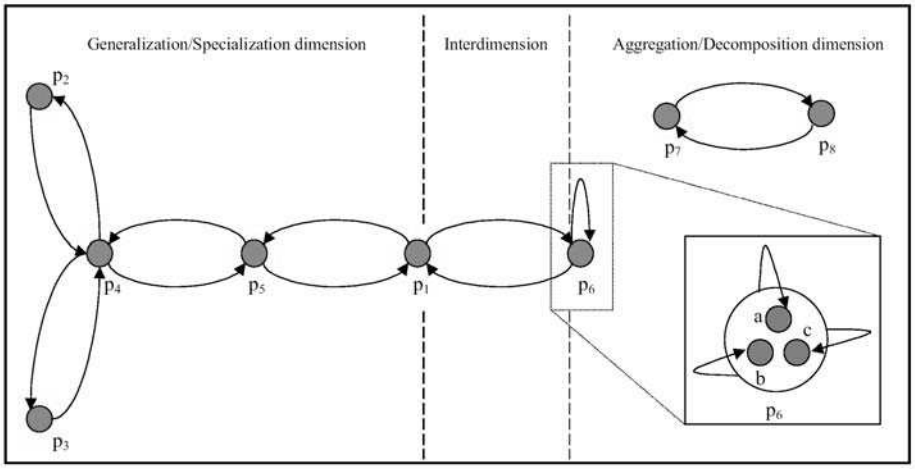


Figura 2. Grafo de transición entre patrones.

Las matrices representadas en la Figura 3 muestran las operaciones primitivas que se aplican sobre los patrones de interés en los tres subgrafos; los cr_i son los criterios de los **discriminadores** y los n_i la cardinalidad de las **Asociaciones**; las celdas vacías en las matrices representan operaciones derivadas.

2.1 Operaciones en la Dimensión de Generalización/Especialización

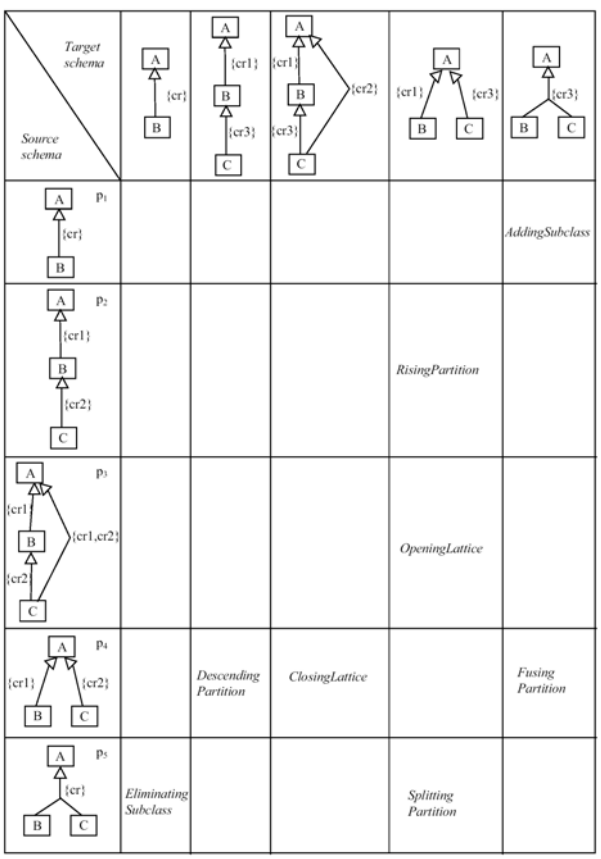
Hay ocho operaciones primitivas para la conformación de esquemas que se aplican exclusivamente a la dimensión G/E (Figura 3a); en [3] se pueden ver ejemplos detallados.

AddingSubclass (celda c_{15}), añade una nueva subclase a una **Partición** (conjunto de **Generalizaciones** que comparten el mismo padre y **discriminador**) mediante unión, intersección, diferencia, etc. de instancias de **Clases** existentes, sin cambiar el **discriminador** de la **Partición**. La operación *EliminatingSubclass* (celda c_{51}) elimina una subclase de una especialización y puede producir una pérdida de información.

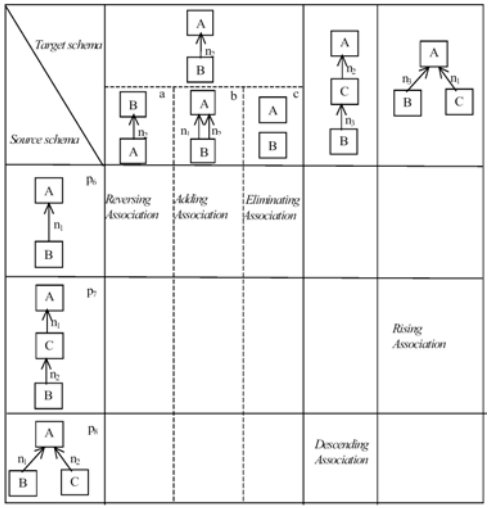
RisingPartition (celda c_{24}) deriva una nueva **Partición** de una superclase aplicando la propiedad transitiva, convirtiendo una subclase indirecta de una superclase en una directa. La operación *DescendingPartition* (celda c_{42}) hace la transformación opuesta.

OpeningLattice (celda c_{34}) transforma un patrón en semiretículo en otro en árbol eliminando parte de un camino entre dos **Clases** y suprimiendo la herencia múltiple; esto no supone una pérdida de información porque cuando hay dos caminos de especialización entre dos **Clases**, ambos implican el mismo conjunto de funciones. La transformación inversa la lleva a cabo la operación *ClosingLattice* (celda c_{43}).

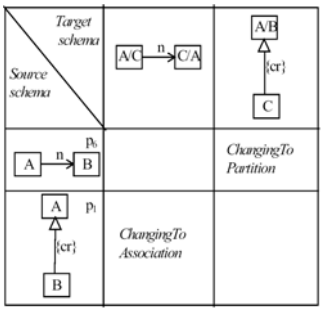
FusingPartition (celda c_{45}) une las **Generalizaciones** de dos **Particiones** de una superclase en una sola **Partición** que tiene como **discriminador** la unión de los **discriminadores** de las **Particiones** fuente. Su inversa, *SplittingPartition* (celda c_{54}) separa subclases de una **Partición** en dos **Particiones**.



a. Dimensión *Generalización/Especialización*.



b. Dimensión *Agregación/Descomposición*.



c. Cambio de Dimensión.

Figura 3. Patrones y Operaciones

2.2 Operaciones en la Dimensión de Agregación/Descomposición

La matriz asociada a la dimensión A/D (Figura 3b) contiene tres patrones (p_6, p_7, p_8) y muestra operaciones genéricas de conformación cuya aplicación dependerá del tipo de **Asociación**. Al patrón p_6 se le pueden aplicar tres operaciones y, para cada una de ellas, se muestra la forma de los posibles esquemas destino. Las formas producidas por *AddingAssociation* y *EliminatingAssociation* no son patrones sino casos especiales de p_6.

ReversingAssociation (celda c^a_{11}) cambia la dirección de una **Asociación**. Esta operación no se puede aplicar a **Agregaciones** porque, además de la forma del esquema, cambiaría el significado; cuando se aplica la capacidad de información del esquema fuente se conserva, y ella misma es su inversa.

AddingAssociation (celda c^b_{11}) añade una nueva **Asociación** al esquema destino que se obtiene mediante unión, diferencia, complemento, etc. de **Asociaciones** entre las **Clases** fuente, conservando la capacidad de información del esquema fuente. *EliminatingAssociation* (celda c^c_{11}) realiza la transformación opuesta, eliminando una **Asociación** del esquema fuente, pudiendo perder capacidad de información.

RisingAssociation (celda c_{23}) obtiene una nueva **Asociación** entre dos **Clases** aplicando la propiedad transitiva. Que se conserve o no la capacidad de información del esquema fuente depende de las propiedades de las **Asociaciones** que intervienen en la operación, por lo que el esquema destino puede tener menor capacidad de información. *DescendingAssociation* (celda c_{32}) es su opuesta. Aunque puede parecer que esta operación se deriva de la composición de *ReversingAssociation*, *RisingAssociation* y *ReversingAssociation*, esto no es así porque *ReversingAssociation* no se puede aplicar a una **Agregación**. El esquema destino resultante de *DescendingAssociation* no siempre conserva la capacidad de información.

2.3 Operaciones entre Dimensiones

Las operaciones primitivas de conformación que trabajan tanto en la dimensión de G/E como en A/D son las que aparecen en la matriz de la Figura 3c.

ChangingToPartition (celda c_{12}) transforma una **Asociación** en una **Partición**, de manera que se añaden una o más subclases al esquema destino. El contenido y número de esas subclases dependen, respectivamente, del conjunto de objetos de la **Clase** que se está especializando y de la **Asociación** que se está transformando en una **Partición**. La **Clase** que se va a especializar no tiene porqué ser ninguna de las dos **Clases** relacionadas a través de la **Asociación**. Teniendo en cuenta que no puede añadirse información, las subclases solamente incorporan aquellas **Asociaciones** heredadas de sus superclases; el esquema destino conserva la capacidad de información. *ChangingToAssociation* (celda c_{21}) realiza la transformación opuesta convirtiendo una **Partición** del esquema fuente en una **Asociación** (cuya dirección puede elegirse a voluntad) que ignora todas las subclases que participaban en la **Partición**; el esquema destino puede tener menor capacidad de información.

3 Conclusiones

Las operaciones de conformación cambian la forma de un esquema pero no añaden nueva información. El uso de diferentes dimensiones OO nos permite clasificar las operaciones en tres grupos, cada uno con una matriz asociada: operaciones en la dimensión de *Generalización/Especialización*, en la dimensión de *Agregación/Descomposición*, y aquellas que cambian de dimensión. Cada operación produce una correspondencia entre sus esquemas fuente y destino que pertenece a la dimensión de *Derivación*, sobre la cuál estamos trabajando actualmente. El resultado de esta investigación será la implementación de una herramienta de soporte a la integración de esquemas.

Referencias

[1] García-Solaco, M., Saltor, F., Castellanos, M.: Semantic Heterogeneity in Multidatabase Systems. In O. Bukhres and A. Elmagarmid (eds.): Object Oriented Multidatabase Systems. Prentice-Hall (1996).

[2] Miller, R.J.; Ioannidis, Y.E.; Ramakrishnan, R.: The use of information capacity in schema integration and translation. In: VLDB'93, Morgan Kaugmann (1993) 120-133.

[3] Rodríguez, E., Abelló, A., Oliva, M., Saltor, F., Delgado, C., Garví, E., Samos, J.: On Operations along the Generalization/Specialization Dimension. In: EFIS 2001. IOS Press (2001) 70-83.

[4] Saltor, F., Castellanos, M., García-Solaco, M.: Suitability of Data Models as Canonical Models for Federated DBs. ACM SIGMOD Record 20 (1991) 44-48.

[5] Sheth, A.P., Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 22 (1990) 183-236 .