



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Heurísticas de diseño conceptual

Diseño de bases de datos

Heurísticas de diseño conceptual

- Modelado semántico = Modelado conceptual.
- Proceso de creación del esquema conceptual.
- Enfoques para el diseño del esquema conceptual.
- Identificación de entidades, relaciones y atributos.
- Refinamiento del esquema conceptual:
 - Relaciones de especialización / generalización.
 - Entidades débiles.
- Apéndice:
Metodología incremental para el diseño conceptual.



Modelado semántico



ETAPA DE DISEÑO CONCEPTUAL

Objetivos:

- Comprensión de la estructura, semántica, relaciones y restricciones de la BD.
- Descripción estable del contenido de la base de datos.
- Comunicación entre usuarios, analistas y diseñadores.

Tarea: **Modelado de los datos del sistema**

Resultados:

- Representación gráfica del modelo de datos.
- Diccionario de datos.



Modelado semántico



Consiste en estudiar los datos que se pretenden almacenar en la base de datos antes de elegir el modelo de datos concreto que se vaya a utilizar:

El modelado semántico o modelado conceptual permite **separar el análisis (¿qué?) del diseño (¿cómo?)**.

NOTA: El modelado semántico o modelado conceptual de una base de datos forma parte de lo que se suele denominar **modelado del dominio** en Ingeniería del Software (en orientación a objetos, la representación visual de las clases de objetos relevantes en el dominio de aplicación de un sistema concreto).



Modelado semántico



El modelado conceptual es **subjetivo**:

- No existe una forma única de modelar un problema.
- Para asegurar un buen diseño, hay que **revisar y refinar** el esquema obtenido.

En una etapa posterior (la etapa de diseño lógico), analizaremos las dependencias funcionales existentes y normalizaremos nuestro esquema.



Creación del esquema conceptual



Se puede utilizar el siguiente proceso de forma iterativa:

- Se identifican las entidades relevantes.
- Se representan gráficamente en un diagrama.
- Se añaden atributos y relaciones.
- Se revisa y refina el esquema conceptual obtenido hasta que se satisfagan todos los requerimientos del sistema.



Enfoques para el diseño conceptual

Podemos optar por diseñar el esquema conceptual de una base de datos de dos formas diferentes:

- **Enfoque centralizado.**
- **Enfoque de integración de vistas**
(usando una estrategia "divide y vencerás").



Enfoques para el diseño conceptual

Enfoque centralizado

Se combinan los requisitos de todos los grupos de usuarios y aplicaciones de nuestro sistema en un único conjunto de requisitos **antes** de comenzar el diseño del esquema.

NOTA: El enfoque centralizado sólo suele ser factible en proyectos de poca envergadura (proyectos de unos pocos meses de duración con un equipo de desarrollo pequeño).



Enfoques para el diseño conceptual

Enfoque integración de vistas

1. **Esquemas parciales:** Se diseña un esquema (o vista) para cada tipo de usuario y/o aplicación a partir de sus requisitos específicos.
2. **Integración de vistas:** Se combinan (integran) los distintos esquemas obtenidos para crear un esquema conceptual global (del cual cada vista individual puede considerarse un esquema externo).

NOTA: El enfoque de integración de vistas suele ser la única estrategia viable en proyectos de gran envergadura.



Identificación de entidades

¿Cómo podemos identificar las entidades que han de formar parte del esquema conceptual de nuestra base de datos? De menor a mayor dificultad, podemos...

1. Reutilizar modelos ya existentes.
2. Utilizar listas de categorías.
3. Identificar sintagmas nominales en el texto de los requerimientos de nuestro sistema.



Identificación de entidades



1. Patrones de diseño: Reutilización de modelos ya existentes.

La estrategia más sencilla consiste en recurrir a modelos de datos especificados en forma de patrones de diseño:

- Los patrones de diseño describen **soluciones elegantes** a problemas que se repiten a menudo en la práctica.
- Estas soluciones nos pueden servir de **punto de partida** en el diseño de nuestro esquema conceptual.

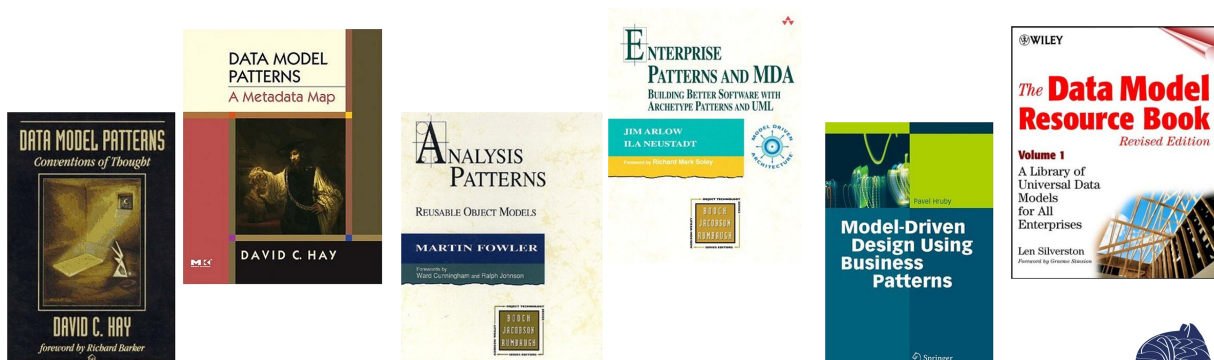


Identificación de entidades



1. Patrones de diseño: Reutilización de modelos ya existentes.

Existen catálogos de patrones de diseño útiles para el modelado de datos desde distintos puntos de vista...



Identificación de entidades



2. Listas de categorías:

Podemos comenzar nuestro diseño conceptual construyendo una lista de entidades candidatas a partir de una lista de categorías comunes que, usualmente, merece la pena tener en cuenta.



Identificación de entidades



2. Listas de categorías: Categorías comunes (1/3)

- Transacciones económicas (suelen resultar críticas porque involucran el uso de dinero): ventas, pagos, reservas...
- Detalles de las transacciones (las transacciones suelen involucrar varios artículos, por lo que deberemos considerar la creación de entidades débiles asociadas a las transacciones en las que se recojan los detalles particulares de cada transacción).
- Productos y/o servicios relacionados con las transacciones (o los detalles de las transacciones): artículo, libro, película, vuelo, asiento, comida...
- El lugar donde se realiza o registra la transacción: caja, cuenta...



Identificación de entidades



2. Listas de categorías: Categorías comunes (2/3)

- Los roles de las personas u organizaciones relacionadas con las transacciones (los actores de los casos de uso): empleado, cliente...
- El lugar donde se presta el servicio asociado a la transacción: sucursal, almacén, aeropuerto, avión...
- Eventos destacados (usualmente ligados a un momento y un lugar que hemos de recordar): venta, vuelo, juego, partido...
- Objetos físicos tangibles (cuyo seguimiento hemos de realizar de algún modo): lote, avión, cheque, tarjeta de crédito...



Identificación de entidades



2. Listas de categorías: Categorías comunes (3/3)

- Contenedores de objetos (reales o abstractos) y objetos incluidos en dichos contenedores: almacén/artículo, tablero/casilla, avión/pasajero...
- Documentos de trabajo, financieros o legales: facturas, albaranes, pedidos, contratos, libros de contabilidad...
- Calendarios, manuales y documentos a los que hay que hacer referencia para realizar determinadas tareas (porque contienen datos que cambian con el tiempo): listas de tarifas vigentes, impuestos...



Identificación de entidades



3. Análisis lingüístico: Identificación de sintagmas nominales.

Las entidades corresponden a objetos que existen en el "mundo" y que son distinguibles unos de otros, por lo que deben aparecer como *sustantivos* o sintagmas nominales en los requerimientos del sistema:

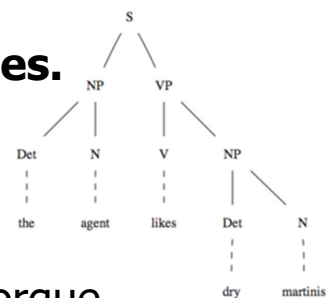
Los sintagmas nominales que aparecen en la descripción textual de los requerimientos se deben considerar como entidades candidatas (o bien como atributos de otras entidades).



Identificación de entidades



3. Análisis lingüístico: Identificación de sintagmas nominales.



Debemos ser especialmente cuidadosos porque establecer una correspondencia directa entre nombres y entidades no es posible:

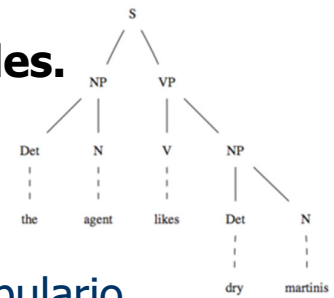
- Las palabras pueden tener varios significados.
- Distintas palabras pueden hacer referencia a lo mismo.



Identificación de entidades



3. Análisis lingüístico: Identificación de sintagmas nominales.



- Es conveniente mantener siempre el **vocabulario del usuario** del sistema para hacer referencia a las entidades que identifiquemos (por ejemplo, película en lugar de producto en un videoclub).
- La imprecisión del lenguaje natural hace recomendable que combinemos el análisis lingüístico de nuestra especificación con las técnicas anteriores (patrones de diseño y listas de categorías).



Identificación de entidades



3. Análisis lingüístico: Identificación de sintagmas nominales.

¿Dónde pueden aparecer los sintagmas nominales que hacen referencia a las entidades?

- En la lista de requerimientos funcionales.
- En la especificación de los casos de uso.
- En los documentos adicionales que adjuntamos al documento de especificación del sistema (vg. modelos de informes).



Identificación de entidades



3. Análisis lingüístico: Identificación de sintagmas nominales.

¿Entidad o atributo?

Si no podemos interpretar X como un simple número (o cadena de texto) en el dominio de nuestro sistema, probablemente X sea una entidad y no un mero atributo.

p.ej. Para una compañía aérea, un *aeropuerto* no es sólo el destino de algunos de sus *vuelos*, por lo que, probablemente, sea una entidad independiente de *vuelo* (para una agencia de viajes, posiblemente no lo sea).



Identificación de relaciones



- Las relaciones entre entidades hay que plasmarlas siempre que tengamos que preservar el conocimiento de la relación por un período de tiempo determinado (sean segundos o años).
- Debemos evitar añadir “demasiadas” relaciones a nuestro esquema y eliminar las relaciones redundantes de nuestro esquema.

NOTA: Son redundantes aquellas relaciones que se podrían reconstruir a partir de otras que ya están presentes en él.



Identificación de relaciones



- Usualmente, nombraremos nuestra relación con un sintagma verbal que nos permita leer nuestro diagrama con facilidad, de forma que la secuencia **NombreEntidad1-VerboRelación-NombreEntidad2** tenga sentido.
- Opcionalmente, especificaremos los roles asociados a los extremos de una relación (especialmente, en las relaciones involutivas).



Identificación de relaciones



- Las relaciones, usualmente, aparecen como **verbos** o sintagmas verbales en la especificación textual de los requerimientos del sistema.
- También podemos utilizar listas de relaciones comunes (igual que hicimos con las listas de categorías para las entidades).

RECORDATORIO: Pueden existir múltiples relaciones entre dos conjuntos de entidades dados (p.ej. un vuelo parte de un aeropuerto y tiene como destino otro aeropuerto).



Identificación de relaciones



Categorías comunes

- A es una transacción relacionada con otra transacción B
p.ej. pago/venta, cancelación/reserva...
- A es un detalle de una transacción B
p.ej. factura / línea de factura
- A es un producto/servicio asociado a una transacción (o detalle de transacción) B.
p.ej. vuelo/reserva, artículo/pedido...
- A es un rol asociado a una transacción B
p.ej. cliente/pago, pasajero/billete...
- A es una parte (física o lógica) de B
p.ej. asiento/sala, casilla/tablero, profesor/departamento...
- A utiliza/gestiona/posee B
p.ej. cajero/caja, piloto/avión...



Identificación de atributos



- En nuestro esquema conceptual debemos incluir únicamente aquellos atributos que aparezcan referenciados en la especificación y, además, sean estrictamente necesarios para dar soporte a la funcionalidad de nuestro sistema.
- ¿Dónde aparecen mencionados los atributos necesarios?
 - En la lista de requerimientos funcionales.
 - En la especificación de los casos de uso.
 - En los documentos adicionales que adjuntamos al documento de especificación del sistema.



Identificación de atributos



- Los atributos puede que no se incluyan explícitamente en el diagrama asociado al esquema conceptual de una base de datos (para facilitar su interpretación cuando éste es complejo) pero **SIEMPRE** deberán aparecer en el diccionario de datos asociado al diagrama.
- En el esquema conceptual se pueden incluir **atributos derivados** (que, en UML, se marcan con el prefijo /), si bien éstos no se almacenarán físicamente en la base de datos (salvo que nos encontremos con problemas de rendimiento al llegar a la etapa de diseño físico).



Identificación de atributos



- Es importante identificar el **dominio** de los atributos (el conjunto de valores permitido para cada atributo), así como indicar si el atributo puede tomar un valor nulo o no.
- Cualquier restricción reseñable deberá figurar en el diccionario de datos asociado al modelo conceptual de nuestra base de datos: claves primarias y alternativas, relaciones entre atributos...



Identificación de atributos



¿Atributo o entidad? (redux)

¿Debería ser la dirección de un cliente un atributo de la entidad *cliente* o una entidad independiente relacionada con el cliente? Depende del uso que le vayamos a dar a los datos asociados a la dirección y de la semántica particular de nuestro problema:

- Si un cliente puede tener varias direcciones asociadas, podríamos definir la dirección como un atributo multivaluado, pero generalmente se opta por crear una entidad independiente para plasmar este hecho (aplicando "preventivamente" una restricción particular del modelo relacional que no existe en otros modelos).
- Si la estructura del atributo compuesto dirección (calle, localidad, provincia...) nos interesa utilizarla para otras cosas (como campañas de publicidad), probablemente también consideremos la dirección del cliente como una entidad independiente.



28

Refinamiento del esquema



Especialización / Generalización

Crearemos relaciones de especialización/generalización si:

- **Regla del 100%:** El 100% de los atributos del supertipo son aplicables al subtipo.
- **Regla "es-un":** Todas las entidades del subtipo sean miembros del supertipo (una comprobación que puede realizarse en lenguaje natural: comprobar si decir "*entidad_{sub} es una entidad_{super}*" tiene sentido o no).

¡OJO! Usando sólo los dos criterios anteriores, podríamos llegar a la conclusión de que podemos dividir a los clientes de una empresa en hombres y mujeres pero, ¿tendría sentido? Posiblemente no, aunque dependerá de la situación.



29

Refinamiento del esquema



Especialización / Generalización

¿Cuándo especializar?

Crearemos subtipos en nuestro esquema cuando:

- El subtipo tiene atributos adicionales que son de interés.
- Las entidades del subtipo se relacionan con otras entidades de forma diferente a como lo hacen las entidades del supertipo.



Refinamiento del esquema



Especialización / Generalización

¿Cuándo generalizar?

Incluiremos un supertipo en nuestro esquema si:

- Los subtipos verifican las dos reglas (100% y "es-un").
- Los potenciales subtipos representan variaciones de un concepto similar.
- Todos los subtipos incluyen el mismo atributo (que pasará a ser un atributo del supertipo).
- Existen relaciones comunes a los distintos subtipos: los subtipos se relacionan con los mismos conjuntos de entidades y estas relaciones tienen el mismo significado para los distintos subtipos.



Refinamiento del esquema



Entidades débiles

- En algunos casos, la existencia de una entidad débil resulta evidente.
- En ocasiones, no obstante, no queda del todo claro si una entidad es débil o no:

**En caso de duda,
haremos que la entidad NO sea débil.**



Refinamiento del esquema



Entidades débiles

Algunas pistas que nos pueden indicar la existencia de una entidad débil:

- El ciclo de vida de la entidad débil siempre está acotado por el ciclo de vida de la entidad fuerte de la que depende (dependencia existencial).
- Existe una relación obvia entre un todo (la entidad fuerte) y sus partes (las entidades débiles).
- Algunos atributos de la entidad fuerte se propagan a las entidades débiles que dependen de ella (p.ej. el destinatario de los artículos incluidos en un pedido).



Refinamiento del esquema



Entidades débiles

¿Qué beneficios tiene identificar las entidades débiles?

- Se muestra explícitamente en el esquema conceptual la dependencia existencial de la entidad débil (que no puede existir de forma independiente a la entidad fuerte de la que depende).
- Sirve para que tengamos en cuenta que determinadas operaciones aplicadas a una entidad fuerte (copia o borrado) han de propagarse a las entidades débiles que dependen de ella.



Apéndice: Metodologías



Existen distintas metodologías, más o menos formales, para el diseño conceptual de bases de datos, p.ej.

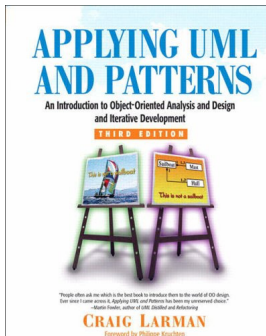
Metodología incremental
basada en el uso de primitivas de diseño conceptual
(descendentes y ascendentes)
y estrategias para el diseño de esquemas
(descendente, ascendente, centrífuga, mixta)



Carlo Batini, Stefano Ceri & Shamkant B. Navathe:
"Diseño conceptual de bases de datos"
Addison-Wesley / Díaz de Santos, 1991.
ISBN 0-201-60120-6



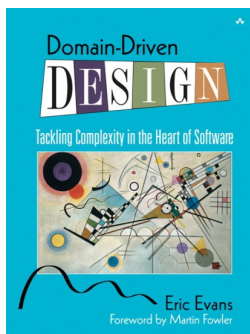
Bibliografía: Diseño conceptual



Craig Larman:

“Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development”, Prentice Hall PTR, 3ª edición, 2004, ISBN 0131489062

En castellano como: “UML y Patrones”
Prentice Hall, 2ª edición, 2004, ISBN 8420534382



Eric Evans:

“Domain-driven design: Tackling complexity in the heart of software”
Addison-Wesley, 2004, ISBN 0321125215



Bibliografía: Patrones de diseño



- David C. Hay: “Data Model Patterns: Conventions of thought”. Dorset House Publishing, 1996. ISBN 0-932633-29-3.
- David C. Hay: “Data Model Patterns: A Metadata Map”. Morgan Kaufmann, 2006. ISBN 0-12-088798-3.
- Jim Arlow & Ila Neustadt: “Enterprise Patterns and MDA. Building better software with archetype patterns and UML.” Addison-Wesley, 2003. ISBN 0-321-11230-X.
- Martin Fowler: “Analysis Patterns: Reusable object models.” Addison-Wesley, 1996. ISBN 0-201-89542-0.
- Pavel Hruby: “Model-Driven Design using Business Patterns.” Springer, 2006. ISBN 3-540-30154-2.
- Len Silverston: “The Data Model Resource Book, Volume 1: A Library of Universal Data Models for All Enterprises”. Wiley, 2001. ISBN 0-471-38023-7.

