

Una Estrategia de Integración de Modelos Conceptuales

Guillermo Bustos Reinoso

gbustos@ucv.cl

Escuela de Ingeniería Industrial - Universidad Católica de Valparaíso

Av. Brasil 2147 - Casilla 4059 - C. P. 2370201 - Valparaíso - Chile

Resumen

Este artículo presenta una estrategia de integración de modelos conceptuales de sistemas de información. La unidad de construcción de los modelos es la *entidad dinámica*, que posee atributos y también exhibe comportamiento. Utilizando instrumentos de modelado ya consolidados para las dimensiones estática (diagrama entidad-relacionamiento), dinámica (red de Petri) y funcional (diagrama de flujo de datos), esta estrategia relaciona claramente todos los modelos resultantes, asegurando así la consistencia global del conjunto. Esta propuesta está aún siendo usada en carácter experimental y requiere de mayores perfeccionamientos.

Introducción

El proyecto de construir soluciones, basadas en software, a problemas organizacionales se puede dividir en tres grandes etapas: 1) análisis, donde se realiza el modelado conceptual del sistema de información y se lidia con el dominio del problema; 2) diseño, donde el modelo conceptual es transformado en un modelo de implementación y se lidia con el dominio de la solución tecnológica; y 3) la implementación del modelo anterior. En este sentido, las etapas 2 y 3 habilitan la concepción que se hizo del sistema en la primera etapa, de ahí la importancia de contar con instrumentos adecuados y criterios de modelado para asegurar un modelo conceptual completo, correcto, expresivo y consistente, entre otros atributos de calidad.

Para abordar el modelado conceptual se puede optar entre diferentes estrategias como se verá en la sección 1. Entre las principales pueden nombrarse la propuesta sintetizada en el Análisis Estructurado Moderno (AEM) [Yourdon89] y la de la orientación a objetos. Sin embargo ambas estrategias no son por completo satisfactorias. En el AEM existe una subordinación de unos modelos a otros, logrando una integración asimétrica entre los modelos. Por otra parte, la orientación a objeto ha iniciado su revolución en el desarrollo de software partiendo desde la implementación, donde sus ventajas son claras, llegando posteriormente al diseño, y alcanzando también al análisis, donde la encapsulación de operaciones no es ventajosa e incluso inconveniente.

Es en esta perspectiva que se propone una estrategia de integración de los modelos, que no es orientada a objeto, pero que puede servir de base para un diseño bajo este paradigma. La unidad de construcción de los modelos es la denominada *entidad dinámica*, es decir, una entidad en el sentido estático, pero que además exhibe comportamiento sin encapsular operaciones. Con esta unidad se logra una simetría de las relaciones entre los modelos estático, dinámico y funcional, sin introducir prematuramente la encapsulación de operaciones en el modelado conceptual.

Este artículo se organiza como sigue. La sección 1 describe brevemente las estrategias de integración existentes más importantes. La sección 2 presenta la propuesta, describiendo el modelado en cada dimensión, las relaciones para los modelos de cada dimensión y algunas consideraciones sobre la extensión de la integración. Finalmente se presentan las conclusiones.

1 Estrategias de Integración

Un hito importante en lo que modelado de sistemas de información se refiere es la estrategia de integración de modelos propuesta en el AEM, cuyo foco de modelado es el proceso. Desde el punto de vista de la integración de los modelos, el AEM propone reglas de “balanceo” que relacionan los modelos funcional y dinámico por medio de un diagrama de flujo de datos (DFD) con procesos de control. El comportamiento del sistema es particionado en un conjunto de procesos de control con comportamiento y flujos de control entre ellos. Aunque estas reglas son claras, no son simétricas ya que no contemplan correlaciones estático-dinámicas, es decir, virtualmente no existen relaciones entre un diagrama entidad-relacionamiento (DER) y un diagrama de transición de estados (DTE) [Yourdon89].

Aunque el AEM permite modelar sistemas de información, ya que se basa en el análisis esencial, este tipo de análisis se está abandonando en favor de metodologías orientadas a objeto, principalmente por la dificultad en la transición al diseño orientado a objeto.

La orientación a objetos ya cambió la forma en que se construye el software hoy en día, sin embargo es cuestionable su utilización en el modelado conceptual de sistemas. A fines de los 80, era común encontrar metodologías de análisis y diseño de sistemas orientados a objeto (ver por ejemplo [Constantine89], [Navathe&Pillalamarri89] y [Ward89]), que utilizaban algún modelo no orientado a objeto, tal como el DFD o el DER, para identificar en él los objetos que formaban parte de los modelos de análisis y diseño. Este mapeamiento era más (a partir del DFD) o menos (a partir del DER) difícil, debido a los problemas de ortogonalidad de las unidades de construcción de los modelos: un objeto es diferente a un proceso y no es exactamente lo mismo que una entidad. Ya en los inicios de los 90, las metodologías sugerían que había que desechar el modelado no orientado a objeto y construir modelos íntegramente bajo este paradigma de implementación (ver por ejemplo [Rumbaugh91], [Coad&Yourdon91], [Shlaer&Mellor92] y [Coleman94].) Más recientemente, con el advenimiento del modelo de casos de uso (*use case*) [Jacobson92], y en particular con UML (*Unified Modeling Language*) [Booch99] y el Proceso Unificado (*Unified Software Development Process*) [Jacobson99], se está imponiendo una visión más razonable: la utilización de modelos no orientados a objeto en el inicio del modelado para posteriormente derivar el modelo de objetos de diseño.

En el caso de UML, las reglas no son definidas explícitamente, pero sí es posible establecer relaciones estático-dinámico a través del diagrama de clases, cuyas asociaciones sirven de base para las interacciones que aparecen en el diagrama de colaboraciones. En la orientación a objeto, y más específicamente en UML, el foco del modelado dinámico es el objeto. De esta forma, el

comportamiento del sistema es particionado en un conjunto de objetos que presentan comportamiento y que colaboran entre sí.

Ya se indicó lo inconveniente de introducir encapsulación de operaciones en los modelos conceptuales, ya que lo se busca es la representación de lo que hará el sistema y no cómo lo hará. Aunque esta encapsulación facilita la transición hacia el diseño orientado a objeto, sesga y dificulta el modelado conceptual al agregarle decisiones de diseño e implementación.

2 Estrategia Propuesta

Una estrategia de integración de modelos conceptuales para el análisis no debiera depender de la forma en que se llevará a cabo el diseño (por ejemplo si éste se hará o no orientado a objeto) y no debiera imponer dependencias indeseables entre los modelos (por ejemplo con relaciones asimétricas entre los modelos). A fin de resolver estas dificultades de las alternativas de integración arriba mencionadas, es que se propone una estrategia que utiliza instrumentos para el modelado conceptual del dominio del problema, pero que además facilita la transición de los modelos resultantes hacia un diseño orientado a objeto.

Del AEM se rescata la clara definición de las reglas de balanceo, y de la orientación a objeto, la utilización del objeto como unidad de comportamiento, pero sin la encapsulación de operaciones. A este objeto *semi-encapsulado* se le denomina *entidad dinámica*, *entidad* por tener sólo atributos y *dinámica* por exhibir comportamiento.

Otro principio que rige la propuesta es la utilización de instrumentos de modelado ya consolidados en las diferentes dimensiones, es decir, modelos que son conocidos y a los que sólo se le han hecho algunos ajustes para facilitar su integración. Sin embargo, habría que destacar el papel preponderante que tiene el modelo dinámico utilizado (la red de Petri) que, dado su poder de expresión, facilita el relacionamiento del mismo con los modelos de las otras dimensiones.

Esta propuesta no es una metodología, porque no establece ningún proceso para construir los modelos, sino que sólo indica cuáles son los instrumentos de modelado a utilizar y cómo los modelos resultantes se relacionan simétricamente, a fin de asegurar la consistencia global del conjunto de modelos conceptuales. En este sentido, ésta es claramente una estrategia dirigida por modelos y no por métodos [Embley92].

2.1 Modelado Dinámico

Como ya se indicó, la unidad de comportamiento de esta propuesta es la entidad dinámica. Primeramente se debe identificar estas entidades, que son las que “viven” o “transitan” por el sistema, y posteriormente se debe modelar sus respectivos ciclos de vida. Para esto existen algunos instrumentos conocidos: *Statecharts* [Harel87] y redes de Petri [Reisig82]. Se escogen estos modelos por presentar facilidades para representar concurrencia. Sin embargo, los *statecharts* presentan mayor dificultad para reorganizar sus estados, dada la fuerte estructuración que presentan,

por esto se opta por las redes de Petri del tipo condición / evento de acuerdo a la versión encontrada en [Heuser90].

Con esta clase de red de Petri se modela el ciclo de vida de la entidad desde su creación hasta eventual eliminación. Para facilitar la integración con los otros modelos, la red se debe reorganizar de la siguiente manera:

- Los estados de la entidad dinámica son divididos en 2 tipos: esenciales y auxiliares. Los estados esenciales son los que interesa registrar y mantener en el sistema, los estados auxiliares sólo sirven como pasos intermedios para alcanzar los esenciales.
- Se distingue a los agentes que envían eventos a la entidad dinámica o que reciben eventos desde ésta. Estos agentes pueden presentar estados y eventos propios representados en el ciclo de vida de la entidad, sin embargo no es necesario describir el comportamiento interno, porque sólo interesa su interacción con la entidad que está siendo modelada. Estos agentes sirven además de definición explícita de los límites del sistema cuando son externos (agentes no externos corresponden a otras entidades dinámicas).
- La distinción entre estados esenciales, estados auxiliares y agentes se representa por medio de pistas o carriles (*swimlanes*), que dividen verticalmente cada parte de la red.

La figura 1 presenta un ejemplo de una porción de red de Petri que modela el ciclo de vida de proyectos de obras públicas [Alfaro99]. Los estados que aparecen en el carril izquierdo son *Proyecto Licitado*, *Proyecto con Contrato* y *Proyecto en Ejecución* son estados importantes y registrables y por lo tanto esenciales. Esto no ocurre con estados tales como *Proyecto con Propuestas*, *Proyecto con Licitación Rechazada* o *Proyecto Adjudicado*, que sólo representan estados intermedios entre los esenciales y se representan en el segundo carril. Como agentes externos aparecen (en el tercer y cuarto carril) el *Departamento de Inversiones* y la *Unidad Técnica*, que tratan, respectivamente, de los aspectos financiero y técnico del proyecto. Las áreas demarcadas corresponden a actividades que serán explicadas más adelante.

Los agentes pueden ser otras entidades dinámicas dentro del mismo sistema, cuyos ciclos de vida también se modelan. En este caso, cada una de las entidades aparecerá como agente de la otra en su respectivo ciclo de vida. Por ejemplo, la entidad *Proyecto* podría relacionarse con la entidad *Cuenta Presupuestaria* y ésta podría ser agente en el ciclo de vida de *Proyecto* y viceversa. Incluso sería posible colocar ambos ciclos de vida en una sola red de Petri, pero esto puede complicar innecesariamente los modelos.

2.2 Modelado Estático

Dado que la entidad dinámica es una entidad, en el sentido estructural del modelo entidad-relacionamiento, es éste justamente el instrumento a utilizar en esta dimensión. El modelado se conduce de la forma tradicional, como es indicada por ejemplo en [Batini92]. La notación de estos autores es también la utilizada en esta propuesta. Debe privilegiarse la construcción de jerarquías de tipos de entidades, dado que esto facilita la relación con los otros modelos del sistema.

2.4 Modelos Dinámico vs. Estático

Los estados que deben registrarse (representados como los estados esenciales en los ciclos de vida) deben asociarse a entidades del DER. Estas entidades pueden organizarse en jerarquías de tipos bajo la entidad cuyo ciclo de vida ha sido modelado. Las instancias que pertenecen a una entidad pueden estar en uno o más estados esenciales diferentes, por lo que es posible distinguir subconjuntos de instancias dentro de la entidad, de acuerdo al estado esencial en que se encuentren. Así por ejemplo, en la figura 2(a), los estados esenciales en la porción de ciclo de vida de *Proyecto*, se relacionan en forma biunívoca con las subentidades de la jerarquía del DER.

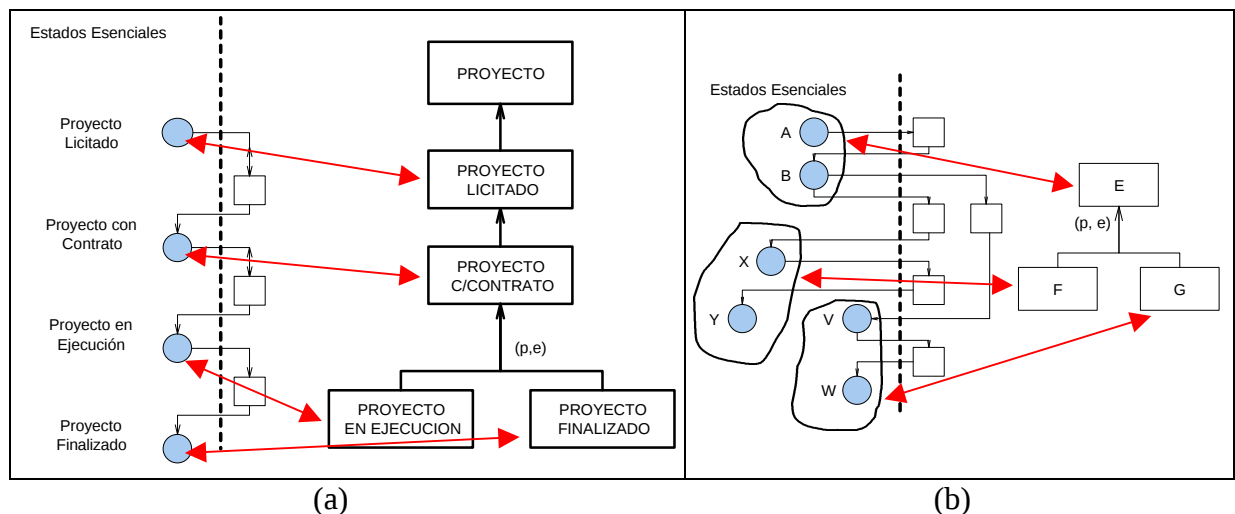


Figura 2 – Relaciones dinámico-estáticas: (a) Ejemplo específico de la entidad *Proyecto* usando clasificación dinámica; (b) Ejemplo genérico usando subciclos de vida.

Además, existen relaciones de concurrencia y exclusividad entre los estados esenciales del ciclo de vida, que deben ser consistentes con la ubicación de las subentidades dentro de las jerarquías y las relaciones de totalidad y exclusión entre ellas. En el ejemplo de la figura 2(a), un *Proyecto* que está en el estado esencial *Proyecto Licitado* no deja de serlo una vez que pasa al estado *Proyecto con Contrato*, lo cual se expresa en la jerarquía indicando que el *Proyecto C/Contrato* es una subentidad de *Proyecto Licitado*. En el mismo ejemplo, es posible ver que los estados *Proyecto en Ejecución* y *Proyecto Finalizado* son excluyentes entre sí, lo cual se expresa con la exclusividad de las subentidades respectivas en la jerarquía.

Esto significa que una determinada instancia de una entidad del tope de una jerarquía puede cambiar de subentidad durante su vida. En este caso, las subentidades se interpretan como roles de la entidad superior y al proceso de cambio de subentidades se le denomina clasificación dinámica [Odell98].

Otra forma de relacionar los modelos dinámico y estático, que evita el uso de roles y clasificación dinámica, es la que se logra al establecer subciclos dentro de un ciclo de vida [Bustos96]. Un subciclo se produce debido a una bifurcación en algún estado que determina 2 o más caminos alternativos que no se relacionan entre sí, con lo cual, una vez que una instancia entra en un sub-

ciclo, no puede salir más de él. Se puede asociar entonces una subentidad a cada subciclo. Como la identificación de subciclos es recursiva, es posible generar de esta manera jerarquías con varios niveles. La figura 2(b) muestra un ejemplo genérico de esta forma de relacionar los modelos. Al subciclo que comienza con el estado esencial *X*, se le asocia la subentidad *F*. Lo mismo ocurre con los estados *V* y *W* y la subentidad *G*. Al ciclo principal, con los estados *A* y *B*, se le asocia a la entidad superior *E*.

Es importante destacar que todas las relaciones indicadas son biunívocas, es decir, ningún modelo, ya sea dinámico o estático, determina o se deriva del otro, sino más bien ambos reflejan aspectos de las entidades del dominio del problema, por lo tanto son mutuamente dependientes.

2.5 Modelos Dinámico vs. Funcional

La transición entre estados esenciales en un ciclo de vida puede significar numerosos eventos y estados auxiliares intermedios. Usando el concepto de actividad de una red de Petri abstracta de tipo canal / actividad, es posible agrupar eventos y estados auxiliares en una actividad que lleve a cabo una transición entre estados esenciales. A esta actividad se le puede asociar un proceso en un DFD [Heuser90], de tal manera que este proceso represente una visión funcional de la transición. Es necesario destacar que sólo los estados auxiliares del ciclo de vida y los eventos directamente conectados a ellos son los que constituyen las actividades, y no los eventos y estados que puedan pertenecer a los agentes.

En la figura 1 se identifican 3 actividades. Por ejemplo, la actividad de la parte superior agrupa eventos y estados que permiten que el proyecto pueda alcanzar el estado esencial *Proyecto Licitado*.

Para evaluar la calidad del agrupamiento es posible adaptar el concepto de cohesión propuesto en el diseño estructurado (ver [Yourdon&Constantine78] y [Page-Jones80]). En este contexto, la cohesión se puede definir como el grado de fuerza existente entre los estados auxiliares y eventos. El espectro original de grados de cohesión es también aplicable (en orden decreciente de intensidad): funcional, secuencial, comunicacional, procedimental, temporal, lógica y coincidental. Al usar como criterio de agrupamiento la cohesión, se debe privilegiar actividades que presenten las mayores cohesiones posibles –idealmente funcionales–, evitando las cohesiones menores del espectro –como la lógica o la coincidental por ejemplo–.

Como las actividades pertenecen a la entidad, la interacción con los agentes externos (representados en los carriles) queda en el límite de los procesos. Estos agentes externos pueden entenderse como terminadores en el DFD.

En la figura 3 se puede ver un ejemplo que asocia, en forma biunívoca, las actividades del ciclo de vida de la figura 1, a los procesos del DFD; y los agentes externos en el ciclo de vida, a los terminadores del DFD.

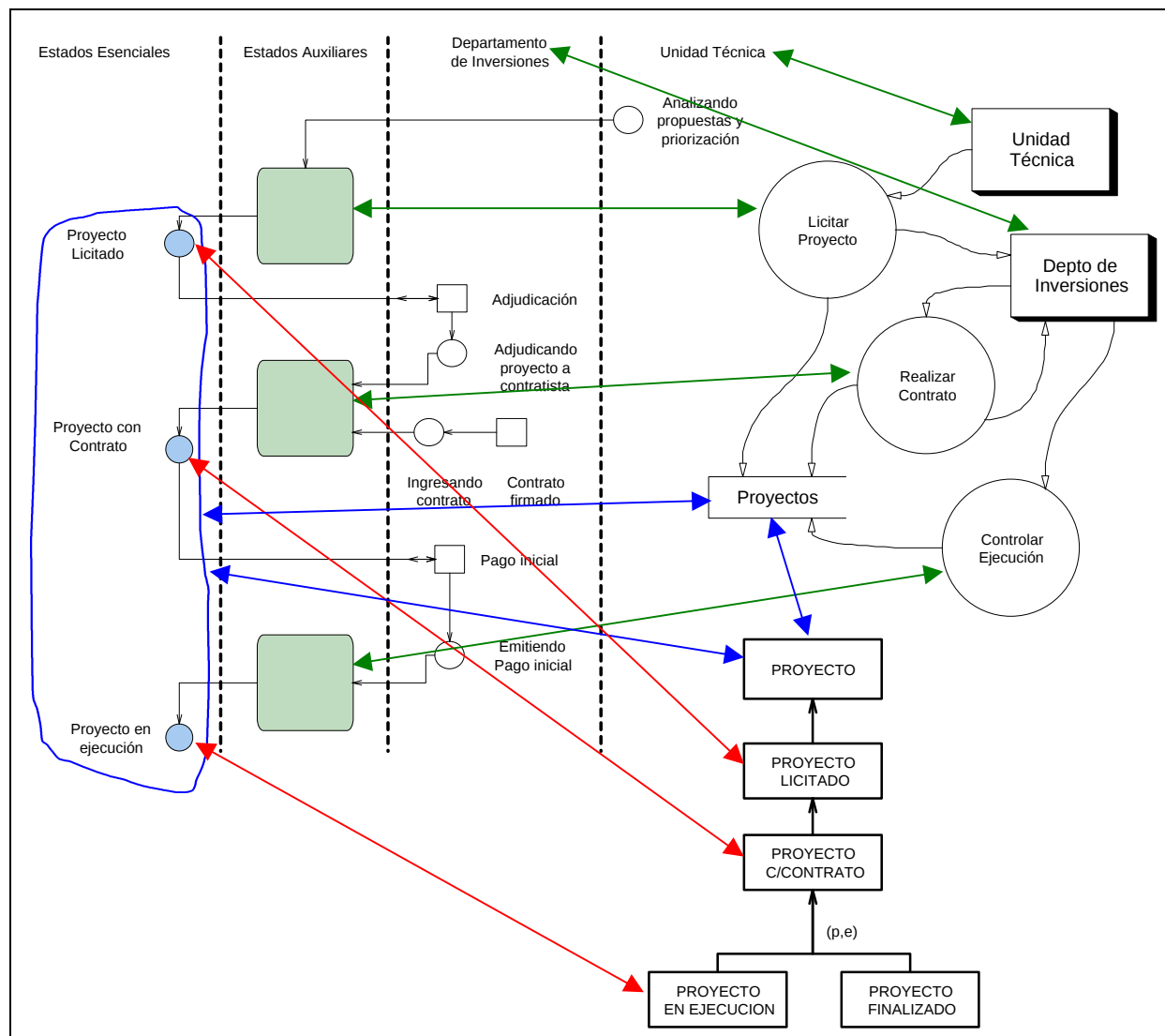


Figura 3 – Ejemplo de relaciones tridimensionales para *Proyecto*.

2.6 Modelos Estático vs. Funcional

La relación entre los modelos estático y funcional es conocida del AEM: las entidades del DER se corresponden biunívocamente con los depósitos de datos del DFD. Con fines de simplificación, sólo se representa como depósito a la entidad del tope de una jerarquía (ver figura 3).

2.7 Modelos Dinámico vs. Estático vs. Funcional

A modo de síntesis, la figura 3 muestra las relaciones entre los modelos de las 3 dimensiones. Se puede apreciar la interconexión simétrica entre los modelos, producto de la interdependencia de sus componentes. Por ejemplo, el depósito *Proyectos* que aparece en el DFD, tiene su

correspondiente en la entidad del tope de la jerarquía del DER, y en el ciclo de vida de la entidad dinámica, representada por el conjunto de estados esenciales.

Adicionalmente:

- Los accesos a los depósitos de datos del DFD aparecen correlacionados con los arcos de entrada y salida de los estados esenciales. Cuando un proceso actualiza un depósito, puede ocurrir que una instancia de la entidad alcance un estado esencial.
- Los flujos de datos desde / hacia los terminadores del DFD se correlacionan con los arcos de entrada / salida que se conectan a los agentes externos en el ciclo de vida.
- Los procesos del DFD no necesitan ser especificados, pues su descripción está dada por la porción de red que es representada por la actividad correspondiente en el ciclo de vida (ver figura 1).
- Las entidades que interactúan con otras como agentes en sus respectivos ciclos de vida, pudieran aparecer conectadas por un relacionamiento en el DER.

La tabla 1 muestra las principales correlaciones entre los modelos dinámico, estático y funcional, desde la perspectiva dinámica.

Tabla 1 – Elementos correlacionados de las 3 dimensiones.

DINÁMICO	ESTÁTICO	FUNCIONAL
entidad dinámica	entidad	depósito
estado esencial	subentidad	_____
actividad	_____	proceso
agente	entidad o...	...o terminador
evento desde / hacia un agente	_____	puede ser flujo de datos de salida / entrada a un terminador

2.8 Extensión de la Integración

Es importante destacar que, aunque los modelos resultantes en las diferentes dimensiones aparecen relacionados, esto no significa que los modelos queden circunscritos a los elementos comunes o correlacionados, sino que pueden extenderse para cubrir más elementos. Es decir, la extensión de la integración puede ser incompleta.

Más específicamente, esto significa que:

- Un DER no puede incluir sólo las jerarquías de las entidades dinámicas y los eventuales relacionamientos entre ellas, sino que debe extenderse para incluir más relacionamientos e incluso otras entidades que no sean dinámicas, es decir que no posean ciclos de vida. Además este modelo debe refinarse y, si se ha optado por usar roles, definir los atributos y relacionamientos específicos para las subentidades.
- Un DFD tampoco puede tener como depósitos sólo a las entidades dinámicas, sino que debe incluir a todas las entidades del DER, lo cual significa que deben existir nuevos procesos que manipulen estos depósitos.

- Un ciclo de vida puede tener estados auxiliares y eventos conectados que no pertenecen a ninguna actividad y no aparecer, consecuentemente, como procesos en el DFD. Aunque esto pueda parecer inconveniente, el agrupar todos los elementos puede tener como consecuencia ciclos de vida que presenten muchas actividades que agrupan muy pocos eventos y estados auxiliares.

No está claro todavía, si es necesario que todas las entidades del DER sean dinámicas, es decir, que haya que construir ciclos de vida para toda entidad identificada para el problema. En términos de consistencia, y de manera análoga a la orientación a objeto, esto debiera hacerse, sin embargo en términos prácticos no se revela interesante tener demasiados ciclos de vida, sobretodo cuando éstos pueden ser muy simples.

Conclusiones

Se ha presentado una estrategia de integración de modelos conceptuales de sistemas de información. Esta estrategia utiliza como unidad de construcción a la entidad dinámica, para la cual se construyen ciclos de vida usando redes de Petri, modelos de procesos que manipulan las entidades dinámicas como depósitos de datos usando DFD, y modelos de entidades que muestran sus jerarquías de tipos usando DER.

Los aportes específicos de la propuesta están dados por:

- *El modelado dinámico:* Usando red de Petri que incluye la división de estados esenciales y auxiliares, la definición de agentes y la representación mediante carriles (*swimlanes*).
- *Las relaciones dinámico-estáticas:* Sobre la base de la clasificación dinámica o alternativamente mediante la identificación de subciclos de vida.
- *La utilización del criterio de cohesión:* Con esto se definen las actividades en las relaciones dinámico-funcionales.

Algunas características de la estrategia propuesta son:

- *La integración de diversos elementos:* Con esto se configura un conjunto simétrico y articulado de relaciones entre los modelos conceptuales de las 3 dimensiones.
- *La carencia de nuevos instrumentos de modelado:* Se opta por definir nuevas relaciones entre modelos conocidos, facilitando con esto la comprensión y aplicación de la estrategia.
- *La opcionalidad del modelado funcional:* El DFD sirve para destacar elementos relacionados de los modelos dinámico y estático. Así el DFD entonces es en realidad sólo una visión funcional del sistema que se encuentra modelado con los ciclos de vida y el DER. De esta manera, la necesidad de contar con modelos funcionales dependerá de si se está lidiando con dominios de problemas que requieran un rediseño de procesos organizacionales, por ejemplo.
- *La utilización de modelos complementarios:* Los modelos descritos pueden complementarse con otros modelos de documentación, tales como un Diccionario de Datos para definir los flujos de datos, depósitos y entidades, y las Especificaciones de Proceso usando pre y post-condiciones, por ejemplo.

En términos generales, la propuesta ofrece:

- *Una forma alternativa de modelado conceptual*: Se centra en el concepto de entidades dinámicas como aglutinadoras de procesos, es decir que, mediante el análisis de las entidades participantes del sistema, se pueda definir los procesos que las manipulan.
- *Una mejor articulación de las relaciones entre los modelos*: Logrando una adecuada simetría en la integración del conjunto de modelos conceptuales.
- *Un énfasis mayor en el modelo dinámico*: Otorgándole un rol equivalente, o incluso superior, a los modelos de las otras dimensiones.

Hasta la fecha, la estrategia propuesta ha sido aplicada en 2 sistemas de diferente envergadura. Un conjunto de 10 entidades dinámicas con ciclos de vida grandes, complejos e interrelacionados fue modelado para representar los sistemas académico, administrativo y financiero de una universidad, en la perspectiva del control de gestión [Bustos&Steinfort99]. En este caso no se construyó visión funcional. Otro sistema de menor envergadura, algunos de cuyos modelos son utilizados como ejemplos en este artículo, trata del control de proyectos públicos [Alfaro99]. En este caso sí se utilizó DFD para documentar funcionalmente. En ambos sistemas, la estrategia resultó efectiva y no presentó problemas importantes.

Como trabajos futuros se requiere formalizar las relaciones mediante un análisis exhaustivo de las relaciones dinámico-estáticas, definir criterios para la extensión de la integración y aplicar la estrategia a otros sistemas para resolver algunos problemas menores derivados de su aplicación.

Bibliografía

- [Alfaro99] Alfaro, R. **Propuesta de una Estrategia de Integración de Modelos Conceptuales de Sistemas de Información**. Informe Final de Proyecto 2. Valparaíso: Escuela de Ingeniería Industrial – Universidad Católica de Valparaíso, 1999.
- [Batini92] Batini, C. et al. **Conceptual Database Design: An Entity-Relationship Approach**. Redwood City: Benjamin/Cummings, 1992.
- [Booch99] Booch, G. et al. **The Unified Modeling Language User Guide**. Reading: Addison-Wesley, 1999.
- [Bustos96] Bustos, G. **Uma Proposta de Modelagem Conceitual de Sistemas Dirigida por Comportamento**. Tesis para optar al grado de Doctor en Ciencia de Computación. Porto Alegre: CPGCC, 1996. (en portugués)
- [Bustos&Steinfort99] Bustos, G.; Steinfort, B. **Especificación de Requerimientos Bajo un Enfoque Integrado**. Trabajo presentado en XXII Taller de Ingeniería de Sistemas, Santiago, Jul. 1999.

- [Coad&Yourdon91] Coad, P.; Yourdon, E. **Object-Oriented Analysis**. Englewood-Cliffs: Prentice-Hall, 1991.
- [Coleman94] Coleman, D. et al. **Object-Oriented Development: The Fusion Method**. Englewood Cliffs: Prentice-Hall, 1994.
- [Constantine89] Constantine, L. Object-Oriented and Structured Methods: Toward Integration. **American Programmer**, New York, v.2, n.7/8, Aug. 1989.
- [Embley92] Embley, D. et al. **Object-Oriented System Analysis: A Model-Driven Approach**. Englewood Cliffs: Prentice-Hall, 1992.
- [Harel87] Harel, D. Statecharts: A Visual Formalism for Complex Systems. **Science of Computer Programming**, Amsterdam, v.8, n.3, p.231-274, June 1987.
- [Heuser90] Heuser, C. **Modelagem Conceitual de Sistemas: Redes de Petri**. Buenos Aires: Kapelusz, 1990.
- [Jacobson92] Jacobson, I. et al. **Object-Oriented Software Engineering: A Use Case Driven Approach**. Wokingham: Addison-Wesley, 1992.
- [Jacobson99] Jacobson, I. et al. **The Unified Software Development Process**. Reading: Addison-Wesley, 1999.
- [Navathe&Pillalamarri89] Navathe, S.; Pillalamarri, M. OOER: Toward Making the E-R Approach Object-Oriented. In: Batini, C. (ed.). **Entity-Relationship Approach**. Amsterdam: North-Holland, 1989. p.185-206.
- [Odell98] Odell, J. **Advanced Object-Oriented Analysis and Design Using UML**. New York: SIGS Books, 1998.
- [Page-Jones80] Page-Jones, M. **The Practical Guide to Structured Systems Design**. New York: Yourdon Press, 1980.
- [Reisig82] Reisig, W. **Petri Nets: An Introduction**. Berlin: Springer-Verlag, 1982.
- [Rumbaugh91] Rumbaugh, J. et al. **Object-Oriented Modeling and Design**. Englewood Cliffs: Prentice-Hall, 1991.
- [Shlaer&Mellor92] Shlaer, S.; Mellor, S. **Object Life Cycles: Modeling the World in States**. Englewood Cliffs: Prentice-Hall, 1992.
- [Ward89] Ward, P. How to Integrate Object Orientation with Structured Analysis and Design. **IEEE Software**, Los Alamitos, v.6, n.3, Mar. 1989.
- [Yourdon89] Yourdon, E. **Modern Structured Analysis**. Englewood-Cliffs. Prentice-Hall, 1989.
- [Yourdon&Constantine78] Yourdon, E.; Constantine, L. **Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design**. New York: Yourdon Press, 1978.