

ORGANIZACIÓN DEL CONOCIMIENTO MEDIANTE PATRONES DE DISEÑO

- Autores:** José Manuel Burgos
jmburbos@fi.upm.es
- Javier Galve
jgalve@fi.upm.es
- Julio García Miguel Sutil
msutil@fi.upm.es
Facultad de Informática Universidad Politécnica de Madrid Campus de Montegancedo
- Resumen** La búsqueda de modelos más expresivos e implementaciones más efectivas en la representación y gestión del conocimiento ha cobrado auge con el crecimiento de Internet y la difusión generalizada de los sistemas de comercio electrónico. La presente propuesta trata de aplicar la tecnología de los patrones de diseño a la representación del conocimiento mediante grafos conceptuales. Se propone un modelo ontológico descrito mediante patrones que facilite la definición y modelización del conocimiento. Asimismo, se enuncian pautas para el diseño e implementación de un sistema de conocimiento basado en dicho modelo.
- Palabras Clave:** Patrones de Diseño, Representación del Conocimiento, Grafos Conceptuales.
- Abstract:** The seek for more expressive models and effective implementations of knowledge representation and knowledge management has increased with the growth of Internet and widespread of e-commerce systems. The current proposal is an intent to apply pattern technology to knowledge representation by conceptual graphs. An ontologic model is stated described in terms of design patterns to facilitate the definition and modelling of knowledge. Besides, design and implementation guidelines to develop a knowledge-based system relying on the model are provided.
- Keywords:** Design Patterns, Knowledge Representation, Conceptual Graphs

1. Introducción

Desde hace algún tiempo, venimos trabajando en la organización de los contenidos de un primer curso universitario de programación¹. A partir del concepto de *esquema de conocimiento* de la psicología cognitiva [AKB79] y basándonos en los trabajos sobre el aprendizaje en el campo de la programación hemos establecido un repertorio de esquemas que modelan

¹ URL: <http://lmi.ls.fi.upm.es/~mp>

dichos contenidos [BGGM00a]. Para la definición de nuestros esquemas, hemos utilizado el modelo descriptivo ofrecido por los *patrones de diseño* [BGG00a]. Surgido originalmente en el contexto de la arquitectura [ALE79], en los últimos años este modelo ha venido demostrando su efectividad en la representación de sistemas y procesos complejos como el desarrollo de software [GHJV95], la enseñanza [CL99], la organización de empresas [BEE00] o la gestión documental [RÜP98].

Sin embargo, al intentar estructurar este conocimiento para integrarlo en un contexto más amplio con otros contenidos y gestionarlo en una organización educativa o de formación, hemos detectado la necesidad de un modelo más completo que ofrezca mayores facilidades de representación. Al tomar una perspectiva más alta de nuestro problema, lo hemos reformulado como la búsqueda de un modelo que permita representar y organizar cualquier conocimiento en cualquier organización. Tres cuestiones han hecho que recobre auge este problema. Por un lado, la comprensión del valor añadido que el conocimiento puede aportar a la organización. Por otro, el cambio de perspectiva y las problemáticas que la aplicación de las nuevas tecnologías pueden provocar en la transmisión del conocimiento en el marco de los nuevos paradigmas educativos [DUA00]. En tercer lugar, el crecimiento de Internet y la explosión de los sistemas de comercio electrónico y de formación a distancia han acentuado problemas que ya existían previamente relacionados con la interoperabilidad entre sistemas y la definición de marcos semánticos comunes entre organizaciones [GRU93].

La comunicación y el intercambio de datos e información dentro de una organización y entre diferentes organizaciones es hoy en día dependiente de la definición, la modelización, el diseño y la implementación de los conceptos subyacentes a la organización y a los compartidos por las organizaciones respectivamente. Los asuntos involucrados no están solamente relacionados con la infraestructura y la tecnología. En realidad, existen problemas fundamentales alrededor de la comprensión de qué conceptos son los que se modelan, cómo están relacionados dentro y fuera de la organización, qué conceptos se comparten con organizaciones externas, y cuáles son los mecanismos que se deben adoptar para permitir a los sistemas funcionar adecuadamente, comunicarse y operar mutuamente a todos los niveles.

Ante esta situación, hemos estudiado las posibilidades que pueden ofrecer los grafos conceptuales [SOW84], la metamodelización y la notación de conceptos [ESC94, WER95, GER98]. La propuesta de [GER98] es un buen punto de arranque para la elaboración de sistemas de conocimiento interrelacionado. Sin embargo, cuando se trata de aplicarlo para modelar entidades concretas en problemas habituales de las organizaciones que puedan surgir en casos como el nuestro, no dota suficientemente de significado a los elementos del modelo, no distingue claramente lo relevante de lo irrelevante y no da ninguna facilidad para su comprensión. Además no ofrece

ninguna pauta para diseñar e implementar sistemas de gestión del conocimiento basados en él.

Este trabajo es una primera aproximación a la interpretación de este modelo en el marco epistemológico de los patrones de diseño, explotando las cualidades de los patrones como entidades autónomas y las cualidades emergentes que ofrecen cuando se interrelacionan en entidades de orden superior conocidas como lenguajes de patrones. Como resultado de este trabajo se establecen unas pautas que permiten aplicar fácilmente el modelo para, por un lado, definir y modelizar sistema de conocimiento y por otro, diseñarlo e implementarlo como un sistema de software orientado a objetos.

El resto del artículo está organizado como sigue: en la sección 2 se hace una breve descripción de los conceptos de patrones de diseño y lenguajes de patrones. La sección 3 está dedicada a presentar escuetamente el modelo de grafo conceptual que sirve de formalismo guía para la propuesta aquí presentada basada en patrones de diseño. En la sección 4, se presenta el lenguaje de patrones propuesto para modelar la representación del conocimiento mediante grafos conceptuales. Finalmente, la última sección está dedicada a conclusiones y trabajos futuros.

2. Patrones y Lenguajes de Patrones

En los últimos años, se ha generado una fuerte corriente de trabajo sobre patrones y lenguajes de patrones en el mundo del desarrollo de software. Este movimiento en informática se basa en los trabajos del arquitecto y profesor Christopher Alexander, promotor de estas ideas en el mundo de la arquitectura [ALE79].

Un patrón de diseño es una solución a un problema en un contexto. Para tener el grado de generalidad que haga al patrón aplicable a muchos problemas distintos, se enuncia mediante una abstracción de una forma concreta de solución que se repite en contextos específicos. Pero un patrón es algo más que una solución probada a un problema recurrente. El problema ocurre en un determinado contexto y en presencia de diversas restricciones o ligaduras que muchas veces se contradicen. La solución propuesta conlleva una estructura que compromete a las restricciones para establecer un equilibrio entre estas "fuerzas". Usando la forma del patrón, la descripción de la solución intenta capturar el conocimiento detallado de resolución de problemas en un dominio poseído por un experto a otros que se podrían beneficiar de ese conocimiento para aplicarlo en otro dominio. Hasta cierto punto, un patrón es un intento de establecer de una manera clara y expresiva, el modo de actuar frente a un problema o una clase de problemas dictado por el "buen oficio". Al patrón se le refiere por un nombre que sirve como un "asa conceptual" por donde agarrarlo

y facilitar el diálogo sobre la perla de información que guarda, estableciendo un vocabulario sencillo y unificado.

Los patrones se pueden aglutinar formando colecciones que constituyen un vocabulario para comprender y comunicar ideas. Cuando tales colecciones se tejen con habilidad para formar un todo cohesionado que revele las estructuras y las relaciones de sus componentes para cumplir un objetivo compartido, entonces estamos hablando de lo que Alexander bautiza como *lenguaje de patrones*. Un lenguaje de patrones define una colección de patrones y las reglas y pautas para combinarlos con un estilo que podríamos denominar “arquitectónico”, pues define una forma de construir estructuras a todos los niveles de escala y en todos los grados de diversidad. En realidad, un lenguaje de patrones se compone de un léxico de patrones y una gramática que establece cómo unirlos para formar estructuras sintácticas. Idealmente, los buenos lenguajes de patrones son *generativos*, es decir, capaces de generar todas las posibles frases a partir de un vocabulario de patrones rico y expresivo. En palabras de Alexander: “*No solamente nos dicen qué reglas hay que seguir en las soluciones sino que nos muestran cómo construir soluciones –tantas como queramos- que satisfagan las reglas*” [ALE79].

Un lenguaje de patrones forma una entidad *orgánica*, en el sentido de que, además de que los patrones colaboran a resolver un problema de orden superior que no es explícitamente resuelto por ninguno de ellos individualmente, existe un equilibrio “perfecto” entre las necesidades de las partes y las necesidades del todo, formando un todo morfológica y funcionalmente completo. Los lenguajes de patrones definen un sistema *hologramático* [MOR90] en el sentido de la idea formulada por Pascal: “*No puedo concebir al todo sin concebir a las partes y no puedo concebir a las partes sin concebir al todo*”. El conocimiento aportado por cada patrón reentra sobre el lenguaje y lo que aprehendemos sobre las cualidades emergentes del lenguaje reentra sobre los patrones. El conocimiento de los patrones se enriquece por el del lenguaje y el del lenguaje por los patrones, en un mismo movimiento productor de conocimientos. Un ejemplo de esto es el efecto de los *comportamientos emergentes* [BEE00]: patrones que surgen de una espontánea interacción local muy densa entre entidades, lo cual da como resultado sistemas autoorganizativos que son adaptativos, abiertos y capaces de efectos multiescala. En otras palabras, los lenguajes de patrones ofrecen un proceso dinámico para la resolución de problemas dentro de su dominio que indirectamente conduce a la resolución de un problema mucho más amplio.

3. Modelo de Conocimiento: Grafos Conceptuales

Los grafos conceptuales son un formalismo de representación del conocimiento en el que los objetos del universo del discurso son modelados mediante

conceptos y relaciones conceptuales que asocian unos conceptos con otros. Los grafos conceptuales han venido siendo usados y estudiados extensivamente por la comunidad científica. Un ejemplo de utilización del formalismo de grafo conceptual es el desarrollo de bases de conocimiento para empresas de consultorías, en las que se memorizan los métodos, el conocimiento práctico (“know-how”) y las habilidades de experto de sus consultores [GER98].

4. El Lenguaje de Patrones *Grafo Conceptual*

El lenguaje de patrones Grafo Conceptual está compuesto de 5 patrones generales: *Tipo de Concepto*, *Tipo de Relación*, *Grafo de Definición*, *Grafo de Restricción* y *Grafo de Regla*². Los 5 patrones forman un sistema generativo que sirve de base uniforme para definir los tipos de conceptos de un modelo.

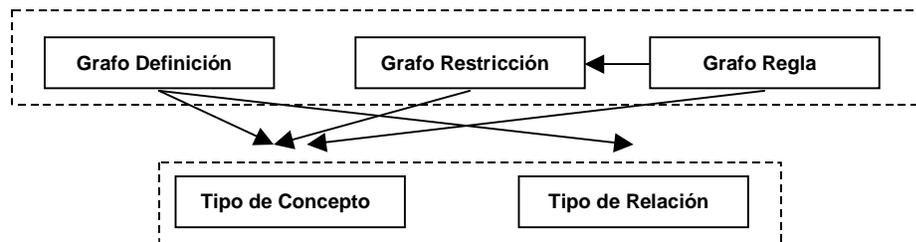


Figura 1. Lenguaje de Patrones: *Grafo Conceptual*

A continuación se describe cada uno de los patrones del lenguaje utilizando notación UML³.

4.1. Tipo de Concepto

Objetivo

Representar un dominio de conceptos.

Solución

Definir los tipos de conceptos como una jerarquía y definir el grafo que muestra las relaciones y asociaciones entre el tipo de concepto y otros tipos de conceptos.

Estructura y Participantes

² El patrón Grafo de Regla se define como una especialización del patrón Grafo de Restricción. Por este motivo, y por cuestiones de espacio del documento, hemos decidido no incluirlo en la presente versión del trabajo.

³ *Unified Modelling Language*.

En la figura 2 se describe la estructura del patrón *Tipo de Concepto*

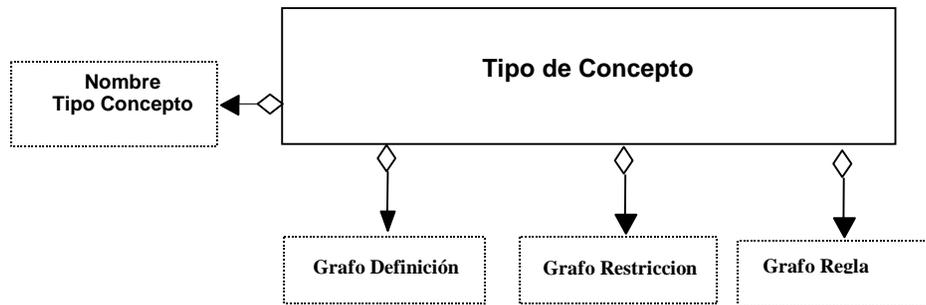


Figura 2. Patrón Tipo de Concepto

Nombre Tipo Concepto: Describe el nombre del tipo de concepto descrito por el grafo conceptual

Grafo de Definición: Define las relaciones conceptual que un concepto perteneciente a este tipo (o clase) tiene con otros tipos de conceptos.

Grafo de Restricción: Especifica condiciones complementarias y necesarias. Muestra topologías que no deben existir. Su evaluación se expresa como la conjunción de las condiciones.

Grafo de Regla: Especifica otras condiciones complementarias y necesarias que se expresan más fácilmente mediante reglas (relaciones disyuntivas).

4.2. Tipo de Relación

Objetivo

Representar relaciones entre tipos de conceptos.

Solución

Definir los tipos de relación como una jerarquía y definir el grafo que muestra las relaciones y asociaciones entre los tipos de conceptos establecidos por el tipo de relación.

Estructura y Participantes

En la figura 3 se describe la estructura de *Tipo de Relación*.



Figura 3. Patrón Tipo de Relación

Nombre Tipo Relación: Describe el nombre del tipo de relación.

Grafo de Definición: Define las relaciones conceptuales que un concepto perteneciente a este tipo (o clase) tiene con otros tipos de conceptos.

4.3. Grafo de Definición

Objetivo

Especificar los tipos de conceptos o relaciones.

Solución

Utilizar un patrón Definición [BUR00] que asocie objetos que representan instancias de la clase Tipo (conceptuales o relaciones) a un objeto de la clase *Grafo Genérico*. El uso de una función como modelo de representación asegura que un *Grafo de Definición* no puede estar asociado a dos Tipos (conceptuales o de relación) diferentes.

Estructura y Participantes

La figura 4 describe la estructura del patrón *Grafo de Definición*.

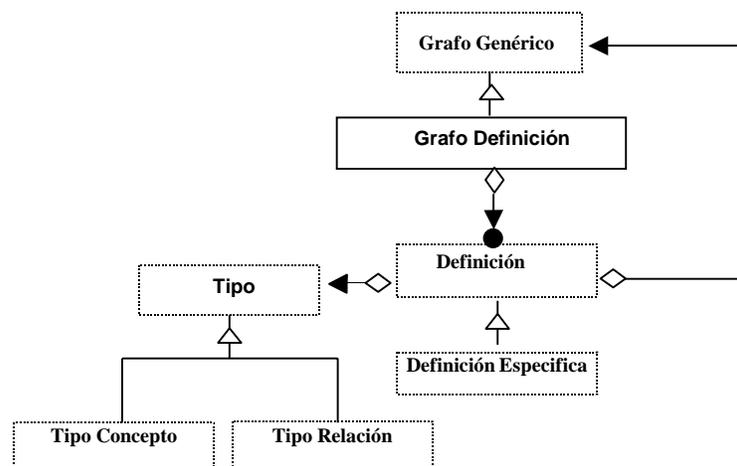


Figura 4. Patrón Grafo de Definición

Tipo: Clase de elementos que permiten definir y organizar el conocimiento de una manera genérica. Representa un conjunto de conceptos o relaciones que tienen las mismas propiedades. Los tipos están definidos a partir de las propiedades comunes de las instancias de tipos y permiten reagrupar las instancias bajo una misma semántica. Hay dos clases:

Tipo de Conceptos: que organizan las entidades

Tipo de Relación: que organizan las relaciones entre entidades.

Grafo Genérico: Describe un grafo conceptual genérico.

Definición: Define la relación unívoca entre elementos de *Tipo* y *Grafo Genérico*. Un grafo de definición puede incluir varios objetos de tipo *Definición* (símbolo • en la figura 4) que modelan diferentes aspectos del tipo de concepto.

Definición Específica: Define una relación específica entre un *Tipo* y un *Grafo* específicos.

4.4. Grafo de Restricción

Objetivo

Especificar las restricciones que definen un tipo de concepto.

Solución

Utilizar un *Objeto Restricción* (o *Predicado* [BUR00a]) para modelar cada restricción.

Estructura y Participantes

La figura 5 describe la estructura del patrón *Grafo de Restricción*.

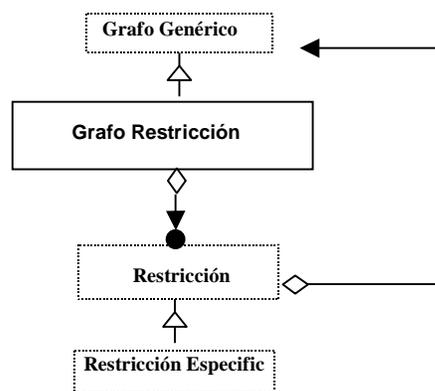


Figura 5. Patrón Grafo de Restricción

Grafo Genérico: Describe un grafo conceptual genérico.

Restricción: Define una condición genérica que debe cumplir el tipo de concepto definido. Un grafo de restricción puede incluir varios objetos de tipo *Restricción* (símbolo • en figura 5). El resultado de la composición de condiciones es la conjunción de todas.

Restricción Específica: Define una condición específica para un Grafo Específico.

5. Conclusiones y Trabajos Futuros

La presente propuesta trata de aplicar la tecnología de los patrones de diseño al modelado de grafos conceptuales. Como consecuencia se ha alcanzado una formulación de una parte sustancial del modelo ontológico descrito en [GER98] mediante un lenguaje de patrones de diseño. Por lo tanto, una propuesta futura deberá completar este lenguaje.

Asimismo, como trabajos futuros, nos planteamos la integración de mecanismos basados en patrones de diseño desarrollados en trabajos anteriores [BUR00], su aplicación a la elaboración de materiales educativos como cursos a distancia o cursos multimedia. Esto último nos permitirá evaluar la validez práctica de la propuesta.

Referencias

- [AKB79] J. R. Anderson, P. J. Kline y C. M. Beasley. A General Learning Theory and Its Application to Schema Abstraction. En G. H. Bower (ed.). *The Psychology of Learning and Motivation*, vol. 13, New York Academic Pres (1979), pp. 277-318.
- [ALE79] C. Alexander: *The Timeless Way of Building*, Oxford University Press, 1979.
- [BEE00] M. Beedle. SCRUM: A Pattern Language for Hyperproductive Software Development. Pattern Languages of Program Design 4. Addison-Wesley 2000.
- [BGG00a] J.M. Burgos, J.Galve y J. García. From Problems to Programs: A Pattern Language to Go from Problem Requirements to Solution Schemas in Elementary Programming. Proceedings of the 5th EuroPLoP'2000. <http://www.coldewey.com/europlop2000/papers>
- [BGG00b] J.M. Burgos, J.Galve, J. García y M.Sutil. *Enseñanza de la Programación basada en Esquemas* . VI Jornadas sobre la Enseñanza Universitaria de la Informática. Jenui'00
- [CL99] M.J Clancy y M.C. Linn. Patterns and Pedagogy. Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education, 1999, 37-42.
- [DUA00] J.M. Duart. Aprender sin Distancias. Nueva Revista, 70. Jul-Ago 2000.

- [ESC94] J. Esch. Contexts, Canons and Coreferent Types. In W. Tepfenhart, J. Dick, and J. Sowa, editors, Proceedings of the Second International Conference on Conceptual Structures, ICCS'94, pages 185{195. Springer-Verlag, 1994.
- [GER98] O. Gerbe, R. Keller, and G. Mineau. Conceptual Graphs for Representing Business Processes in Corporate Memories. In Proceedings of the sixth international Conference on Conceptual Structures (ICCS'98). Springer-Verlag, 1998.
- [GHJV95] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.
- [GRU93] Gruber T., *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, CA, 1993.
- [MOR90] E.Morín. Introducción al Pensamiento Complejo. Gedisa Eds. 1990.
- [RÜP98] A. Rüping. Writing and Reviewing Technical Documents. Proceedings of the 3rd EuroPLoP'98.
- [SOW84] J. F. Sowa. Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.
- [WER95] M. Wermelinger. Conceptual Graphs and First-Order Logic. In G. Ellis, R. Levinson, W. Rich, and J. Sowa, editors, Proceedings of ICCS'95, pages 323{337. Springer-Verlag, 1995.