

# LENGUAJES DE PATRONES DE DISEÑO DE *SOFTWARE* BAJO UNA PERSPECTIVA COGNOSCITIVISTA

Alan Calderón Castro

## RESUMEN

A la fecha se han publicado muchos catálogos de patrones asociados con el desarrollo de software. Los lenguajes de patrones son el medio hasta ahora considerado ideal para organizar el creciente conocimiento sobre patrones. Sin embargo, en la comunidad de autores y usuarios de patrones de diseño de software persisten inquietudes sobre cómo mejorar la expresividad de los lenguajes de patrones, las representaciones existentes, el esquema organizativo del conocimiento que promueven y cómo valorar la calidad de los lenguajes de patrones entre otros temas. En este trabajo se elabora una concepción teórica nueva de los lenguajes de patrones, basada en un marco de referencia cognoscitivista, que abarca a la vigente pero la amplía, de manera que al final se logran obtener nuevas respuestas para algunos de los problemas pendientes. Las consecuencias prácticas de este replanteamiento son tratadas en [Calderón, 2003a] y [Calderón, 2003c].

**Palabras clave:** Lenguajes de patrones, patrones de diseño, modelos semánticos para lenguajes naturales

## ABSTRACT

Presently many pattern catalogs for software design have been published. Pattern languages have been considered the best way to organize this increasing knowledge. Nevertheless, unsolved issues persist about how to enrich pattern language expressiveness, how to represent its organization, how to measure the quality of pattern languages and how to improve it. This paper addresses some of these topics from a new perspective of pattern languages based on some cognitive theories. This new perspective does not reject concepts so far accepted about pattern languages traits, but it includes them in an enriched cognitive framework aimed to solve these problems. Practical topics derived from this proposal are addressed on [Calderón, 2003a] and [Calderón, 2003c].

**Key words:** Pattern Languages, Patterns, Semantic Models for Natural Languages

---

\* Universidad de Costa Rica - Escuela de Ciencias de Computación e Informática.  
calderon@eccilucr.ac.cr

## Introducción

Alexander escribió sobre lenguajes de patrones, teniendo como marco de referencia concreto a la arquitectura. Para aplicar las nociones que desarrolló en [Alexander, 79] al campo del diseño de *software* es indispensable construir una interpretación apropiada. Curiosamente esto no se ha hecho y quizás es por esto que en la comunidad de autores y usuarios de patrones de *software*, persisten ciertas dudas de carácter teórico, práctico y metodológico. Desde principios de la década de los noventa, pero principalmente a partir de la publicación de [Gamma, 95], se ha estado usando intensivamente el concepto de patrón. Sin embargo, no es para nada evidente qué significado tienen todos los conceptos que se presentan en [Alexander, 79] para los diseñadores de *software*. De hecho, en la comunidad de usuarios y autores de patrones de *software*, solo se ha transferido hasta la fecha el concepto de patrón y en el caso de “lenguaje de patrones” se confunde frecuentemente con la noción de “estructura de lenguaje de patrones” y hasta con un catálogo de patrones que muestra asociaciones relevantes entre los patrones<sup>1</sup>. Es posiblemente debido a este vacío teórico que dentro de la comunidad de diseñadores de *software* se identifican algunos temas no resueltos tales como:

1. La misma diferenciación clara entre lenguaje de patrones, estructura de lenguaje de patrones, catálogos de patrones y sistemas de patrones.
2. ¿Cómo distinguir entre un buen lenguaje de patrones y otros no tan buenos? ¿Cuáles podrían ser los criterios? O lo que es lo mismo, ¿cómo se puede mejorar un lenguaje de patrones?
3. ¿Cómo mejorar la expresividad de un lenguaje de patrones?
4. ¿Cómo representar mejor un lenguaje de patrones?
5. ¿Cómo organizar el conocimiento de patrones que crece rápidamente<sup>2</sup>?

La estrategia en este trabajo consiste en transferir las nociones sobre los lenguajes de patrones, con base en un marco de referencia cognoscitivista, con el fin de intentar dar una respuesta a estos problemas que tenga sustento teórico. Además se trata de justificar teóricamente el patrón para sistemas de patrones (denominados aquí lexicones de patrones) de [Calderón, 2003a]. Para esto, primero se exponen y analizan brevemente las conceptualizaciones de Alexander, tal como aparecen en [Alexander, 79]. Luego se plantea un nuevo enfoque teórico de carácter cognoscitivista. Finalmente se plantea una interpretación adecuada para los conceptos de completitud funcional y completitud morfológica presentados en [Alexander, 79]. De esta manera, se puede concluir dando una respuesta nueva a los problemas teóricos y prácticos planteados en esta introducción.

## Lenguajes de patrones en Alexander

En esta sección se exponen las nociones de lenguaje de patrones, completitud funcional completitud morfológica y estructura de un lenguaje de patrones, de Alexander, tal como han sido tratadas en [Alexander, 79]. La intención es construir un contexto que sirva de punto de partida para la propuesta que se hace en las secciones subsecuentes.

Las siguientes citas provienen todas de [Alexander, 79] y son traducciones mías:

1. “La gente puede darle forma a edificios, y lo ha hecho por siglos, usando lo que yo llamo lenguajes de patrones. Un lenguaje de patrones provee a cada persona que lo usa el poder de crear una infinita variedad de edificios únicos y nuevos, de la misma forma en que su lenguaje ordinario le da el poder de crear una variedad infinita de oraciones.” (página 167).
2. “*Es en este sentido que un sistema de patrones forma un lenguaje.* Cuando el constructor de establos aplica los patrones de un establo, uno a otro en un orden apropiado, es capaz de crear un establo. El establo siempre mantendrá las relaciones requeridas por los patrones; sin embargo, los tamaños, los ángulos, y las relaciones dependen de las necesidades de la situación y de la inspiración del constructor. La familia de establos producida por este sistema comparte todas las características morfológicas especificadas en las reglas (éstas son las leyes morfológicas que hemos indicado), pero más allá de esto, existe una variedad literalmente infinita.” (página 183).
3. “Un lenguaje de patrones es un sistema más complejo de esta categoría<sup>3</sup>. Los elementos son patrones. Los patrones tienen estructura, que muestra cómo un patrón es en sí mismo un patrón de patrones más pequeños<sup>4</sup>. Y también hay reglas, incluidas en los patrones, las cuales describen la forma en que pueden crearse, y la forma en que deben combinarse con respecto a otros patrones. Sin embargo, en este caso, los patrones son a la vez

elementos y reglas, de tal manera que las reglas y los elementos son indistinguibles. Los patrones son elementos. Y cada patrón es también una regla que describe las posibles combinaciones de los elementos—que a su vez son patrones.” (página 185).

4. “*Un lenguaje ordinario como el Inglés es un sistema que nos permite crear una variedad infinita de combinaciones unidimensionales de palabras, llamadas oraciones.*” (página 185).
5. “*Un lenguaje de patrones es un sistema que nos permite crear una variedad infinita de combinaciones tridimensionales de patrones que llamamos edificios, jardines, pueblos.*” (página 186).
6. “En resumen: ambos, lenguajes ordinarios y lenguajes de patrones, son sistemas combinatorios finitos que nos permiten crear a voluntad una variedad infinita de combinaciones únicas, apropiadas a circunstancias diferentes.

---

| Lenguajes naturales  | Lenguajes de patrones                                |
|--|--|
| Palabras   | Patrones   |
| Reglas gramaticales y significados que originan asociaciones | Patrones que especifican asociaciones entre patrones |
| Oraciones  | Edificios y lugares                                  |

---

7. “*A este punto, hemos definido el concepto de lenguaje de patrones claramente. Sabemos que es un sistema finito de reglas que una persona puede usar para generar una variedad infinita de edificios diferentes— todos miembros de una familia—y que el uso del lenguaje permitirá a la gente de un pueblo o ciudad generar el balan-*

*ce exacto entre uniformidad y variedad que da vida a un lugar.* “ (página 191).

Es importante hacer notar que Alexander:

- Siempre mantiene una analogía entre lenguaje natural y lenguaje de patrones.
- Casi siempre omite una definición directa de lenguaje de patrones y más bien construye el concepto en términos de para qué sirve a las personas.
- Cuando intenta una definición directa, dice que se trata de un sistema finito de reglas<sup>5</sup>, muy probablemente porque ésa era la noción prevaleciente de lenguaje natural en el momento en que escribió su obra; sin embargo, de nuevo insiste en para qué sirve un lenguaje de patrones.

Sobre los conceptos de estructura de lenguaje de patrones, completitud funcional y morfológica en [Alexander, 79] se afirma:

1. *“La estructura del lenguaje es creada por medio de la red de conexiones entre los patrones individuales: y el lenguaje tiene o no vida, como totalidad, en la medida en que los patrones forman una unidad completa.”* (página 305).
2. *“Cada patrón entonces, depende tanto de los patrones más pequeños que contiene, como de los patrones más grandes en que es contenido.”* (página 312).
3. *“En esta red, las conexiones entre los patrones son igualmente parte del lenguaje como los patrones mismos.”* (página 314).

4. *“Pero aun cuando tenga los patrones conectados unos con otros en una red, de tal manera que formen un lenguaje, ¿cómo puedo saber si un lenguaje es bueno?”* (página 315).
5. *“Un lenguaje—de patrones—es bueno, lo que significa que posibilita crear algo completo, cuando es morfológica y funcionalmente completo.”* (página 316).
6. *“Es morfológicamente completo, cuando los patrones juntos forman una estructura completa, con todos sus detalles, sin ningún vacío.”* (página 316).
7. *“El lenguaje es morfológicamente completo cuando puedo visualizar, de manera muy concreta, la clase de edificios que genera.”* (página 316).
8. *“Y es funcionalmente completo cuando el sistema de patrones tiene esa consistencia propia peculiar en que los patrones, como sistema, generan solamente aquéllas fuerzas que pueden resolver —de tal manera que el sistema como unidad, cobra vida, sin que esté sujeto a la acción auto-destructiva de los conflictos internos”* (página 316).
9. *“Y el lenguaje es funcionalmente completo, cuando el sistema de patrones que configura permite que las fuerzas internas se resuelvan mutuamente.”* (página 317).
10. *“En ambos casos—completitud funcional y morfológica, el lenguaje está completo solo cuando cada patrón del lenguaje individualmente está completo.”* (página 318).
11. *“Porque obviamente el lenguaje no puede estar completo como unidad,*

mientras cualquier patrón individual esté incompleto. Cada patrón debe tener suficientes patrones ‘debajo’ para que sea morfológicamente completo. Y también cada patrón debe tener suficientes patrones debajo, de manera que se puedan resolver los problemas que se generan al aplicarlo.” (página 318).

Es importante hacer notar que:

1. Se establece claramente la diferencia entre lenguaje de patrones y estructura de un lenguaje de patrones.
2. La noción de lenguaje de patrones y estructura de lenguaje de patrones plantea el dualismo típico entre sintáxis y semántica.
3. Sin embargo, las asociaciones o conexiones entre patrones son consideradas parte del lenguaje también, inclusive parte de los patrones.
4. El tipo de asociación “X completa a Y” es el más importante en un lenguaje de patrones, y de hecho en ella se basan los conceptos de completitud funcional y morfológica.
5. Al igual que en un lenguaje de patrones, las nociones de Alexander sobre patrón, lenguaje de patrones, estructura de lenguaje de patrones, completitud funcional y completitud morfológica (de un lenguaje de patrones y de los patrones mismos), conforman un marco de referencia coherente, de hecho se trata de un sistema de conceptos interdependientes del que podría interpretarse que emerge un *metalenguaje de patrones*.

6. Alexander siempre escribió sobre arquitectura. Este metalenguaje está intrínsecamente ligado a un dominio de experiencia específico, el urbanismo y la arquitectura. Es obvio que la aplicación de este metalenguaje a otro dominio requiere un esfuerzo de transferencia, y este es uno de los objetivos de este trabajo.

En la siguiente sección se propone un marco de referencia que abarca al de Alexander y lo extiende con el objetivo de lograr una transferencia completa al dominio del diseño de *software*. Además una diferencia fundamental entre esta propuesta y la de Alexander radica en la manera como aquí se caracterizan los lenguajes naturales y el supuesto que aparentemente él maneja, sobre su naturaleza y estructura. Desde la perspectiva del marco teórico que se maneja en este trabajo, no es coherente mantener una división tajante entre sistema semántico y sintáctico.

### **Un marco de referencia cognoscitivista para los lenguajes de patrones**

En esta sección se plantean los elementos básicos de un marco de referencia cognoscitivista para comprender los lenguajes de patrones y su dinámica. El principal sustento teórico de esta propuesta proviene de [Lakoff, 87] y [Alexander, 79], sin embargo también se hacen referencias a [Suchman, 87] y a [Ellis, 93]. La intención principal es transferir al dominio del diseño de *software* las nociones de Alexander, sobre la base de un marco teórico más amplio. El objetivo final es:

1. por un lado, construir una respuesta adecuada a los temas no resueltos citados al final de la introducción,
2. y por otro lado, justificar teóricamente el patrón para lexicones de patrones propuesto en [Calderón, 2003a].

### **Caracterización de un lenguaje de patrones**

Un lenguaje de patrones puede caracterizarse como una especialización de un lenguaje natural (es decir un subsistema simbólico empotrado en un sistema más amplio que es un lenguaje natural) cuyas categorías naturales han sido construidas para describir y organizar planes para resolver problemas de diseño de *software*, en forma heurística y en un dominio de diseño específico. Los planes son los patrones y constan a su vez de cinco aspectos básicos, "El Contexto" (C), "El Problema" (P), "La Solución" (S) y "Las Asociaciones con otros patrones" (A)—de acuerdo con [Alexander, 79]. Finalmente "Las consecuencias de aplicar la solución" (CS) que han agregado los ingenieros de *software*. Toda tupla (C, P, S, A, CS) tiene un nombre significativo que constituye una palabra de la especialización y por ende refleja las escogencias gramaticales preferidas por la comunidad de usuarios y autores de patrones.

Esta nueva caracterización para lenguajes de patrones merece una explicación detallada, particularmente del concepto de plan y de la relación entre lenguajes de patrones y lenguajes naturales. En primer lugar, el término "plan" se usa aquí en el sentido de Suchman:

"Como construcciones del sentido común, los planes son constituyentes de la acción práctica, pero la constituyen

como artefactos de nuestro *razonamiento acerca* de la acción, no como un *mecanismo* que genera la acción. Nuestras proyecciones imaginarias y nuestras reconstrucciones retrospectivas son los medios por los cuales aprehendemos la acción situada y razonamos sobre ella, mientras la acción en sí misma, en contraste, es esencialmente transparente para nosotros como actores." (traducción mía de la página 39 de [Suchman, 87]). En contraste con la noción clásica de "plan", "... La visión alternativa es que los planes son recursos para la acción situada, pero no determinan de manera decisiva su curso. Mientras el plan presupone las prácticas asimiladas y las circunstancias cambiantes de la acción situada, la eficiencia de los planes como representaciones proviene precisamente del hecho de que no representan estas prácticas y circunstancias en todos sus detalles concretos" (página 52 de [Suchman, 87]). "El término –acción situada– enfatiza el punto de vista de que cada secuencia de acciones depende de manera esencial de las circunstancias materiales y sociales en que se da." (traducción mía de la página 50 de [Suchman, 87]).

Se puede argumentar que un proceso de diseño de *software* es un continuo de acciones situadas porque se dan efectivamente una serie de circunstancias sociales y materiales que la determinan. Las prácticas establecidas en ingeniería de *software*, los estándares aceptados a nivel de toda la industria y los que se aplican a nivel local en la organización desarrolladora de *software*, son ejemplos de circunstancias sociales. Por otro lado, las limitaciones presupuestarias, los componentes de *software* que se reutilizan, las mismas herramientas usadas son ejemplos de circunstancias materiales. Más específicamente, las metas de adaptabilidad y reutilizabilidad

establecidas como parte de un proceso de diseño, son circunstancias determinadas socialmente que definen si es oportuna o no la misma aplicación de patrones. Introducir patrones en un proceso de diseño incrementa los costos al requerir más tiempo que cuando el resultado final solo se optimiza con respecto a un producto específico y las metas de reutilizabilidad y adaptabilidad no son tan importantes. Por lo tanto, los patrones se aplican en el contexto de un proceso de diseño determinado por circunstancias materiales y sociales.

Por otro lado, un patrón como tal no determina de manera mecánica un diseño: un patrón no genera mecánicamente acciones de diseño. Si bien la especificación de un patrón incluye una estructura y unas consecuencias de aplicarlo, las acciones de diseño que resultan responden completamente al problema específico al que es aplicado. Como es bien sabido, la aplicación de cada patrón debe considerar las restricciones técnicas del proceso de diseño. Los patrones son un recurso en un proceso de diseño, no lo generan. Por lo tanto, se puede ver que un patrón es un plan en el sentido en que apunta Suchman.

Para los usuarios y autores de patrones, ésta puede ser una verdad de Perogrullo en el mejor de los casos<sup>6</sup>. Sin embargo, desde un punto de vista teórico, es importante establecer con claridad la caracterización del concepto de “patrón” desde las perspectivas de los procesos cognoscitivos asociados al diseño de *software*. En una sección posterior, se verá que este enfoque teórico sí tiene consecuencias prácticas, pero antes de proceder a analizar algunas, es importante completar el nuevo marco de referencia.

En segundo lugar, ningún patrón tiene sentido en forma aislada. Un patrón

está enraizado en un sistema más amplio, denominado por Alexander “lenguaje de patrones”. A continuación se argumenta además que un lenguaje de patrones está enraizado en un sistema todavía más amplio: en un lenguaje natural. Existen asociaciones sutiles entre un lenguaje de patrones y el lenguaje natural en que está inmerso y se desarrolla. Sin embargo, no se trata simplemente, como ha afirmado Alexander, de una analogía. No es que las palabras son *como* los patrones y las frases *como* las construcciones arquitectónicas. Un lenguaje de patrones *es de hecho* una especialización de un lenguaje natural.

Un lenguaje de patrones es de hecho una especialización de un lenguaje natural porque, como era de esperarse, un análisis superficial de algunos de los catálogos de patrones publicados, revela evidencia abrumadora de los procesos de categorización típicos de los lenguajes naturales, tales como el uso de metonimias y metáforas, el principio del dominio de experiencia y el de los tres niveles de categorización, básico<sup>7</sup> u ordinario, sub-ordinario y supra-ordinario.

### 1. El uso de metonimias y metáforas

En particular considérese por ejemplo la denominación de los patrones en GoF. Lakoff elabora la conceptualización de los procesos de *metonimia* y *metáfora*:

1. Una metonimia se caracteriza como “...una situación en la cual una subcategoría o miembro o submodelo se usa (frecuentemente para un propósito inmediato y delimitado) como representante de toda la categoría. En otras palabras, estos son casos donde una parte (una subcategoría, miembro o submodelo) reemplaza a la categoría completa –ya sea durante

un razonamiento, un reconocimiento, etc.” (pág. 79 de [Lakoff, 87]).

2. Una metáfora “...tiene un dominio fuente, un dominio meta, y una transformación del dominio fuente al dominio meta.” (pág. 276 de [Lakoff, 87]). También “Los modelos metafóricos son correspondencias que se establecen entre un modelo proposicional o un modelo basado en un esquema de imagen de un dominio dado hacia estructuras en otro dominio”<sup>8</sup> (pág. 113 de [Lakoff, 87]).

Entre los patrones de GoF es evidente que hay muchos cuyo nombre obedece a una metonimia y a la vez a una metáfora, por ejemplo:

1. “Bridge”.
2. “Chain of Responsibility”.
3. “Facade”.
4. “Observer”.

En todos estos casos se puede corroborar que un objeto miembro o un aspecto específico del modelo de la solución del patrón ha sido escogido para representar a todo el patrón pues de hecho lleva el mismo nombre (ésta es la metonimia). Pero más aún, ¿de dónde vienen los nombres para estos objetos que forman parte esencial de la estructura genérica que plantea el patrón como solución? Igualmente, en todos estos casos, la imagen de la estructura de solución es interpretada como una realización de un esquema de imagen que corresponde con el que subyace la palabra escogida. En el caso de “Bridge”, subyace el esquema de imagen de un puente. En el caso de “Chain

of Responsibility”, el de una cadena. En el caso de “Facade”, el de una fachada de una obra arquitectónica y en el caso de “Observer” el de un objeto que capta imágenes o de una persona en actitud observadora o atenta. La intencionalidad de todos estos patrones se sintetiza muy bien al usar estas metonimias y metáforas. No se puede separar el término usado para denominar a estos patrones de las categorías naturales con las cuales está sutilmente asociado.

Dicho sea de paso, esta es la razón por la cual es imprescindible traducir los lenguajes de patrones al lenguaje natural nativo de quiénes los quieran usar. Por ejemplo para un aprendiz de programación hispanoparlante, el término “Session Facade” no va a sugerir las mismas imágenes que “Fachada de sesión”, y en caso de que lo hiciera es porque logra traducir fácilmente las palabras. En otros casos como “Bridge” puede ser que del todo no logre captar una sola imagen en contraste con “Puente” que de hecho revela un aspecto esencial de la estructura de este patrón tal como se describe en GoF.

## 2. El principio del dominio de experiencia

“Si existe un dominio básico de experiencia asociado con A, entonces es natural que todas las entidades de este dominio estén en la misma categoría que A” (pág. 93 de [Lakoff, 87]).

Por ejemplo, un “dominio de aplicación” como el que los ingenieros de *software* denominan “sistemas de información empresariales”, es un dominio de experiencias. POSA 1 y POSA 2 identifican tres grandes dominios de experiencia: “Patrones Arquitectónicos”, “Patrones de diseño” y “Frasas idiomáticas” (“idioms” en inglés), que corresponden



con lo se conoce como diseño arquitectónico, diseño de objetos y diseño detallado o programación respectivamente.

### 3. Tres niveles de categorización

El análisis de Lakoff plantea que los procesos de categorización revelan tres niveles: nivel básico u ordinario, sub-ordinario y supra-ordinario:

“...las categorías no están organizadas simplemente en forma jerárquica de lo más general a lo más específico, sino que además están organizadas de tal manera que las categorías cognoscitivamente básicas están en el medio de una jerarquía. El proceso de generalización procede de abajo hacia arriba y el de especialización de arriba hacia abajo.” (pág. 13 de [Lakoff, 87]).

“...las categorías del nivel básico u ordinario preponderan funcional y epistemológicamente en lo que respecta a factores tales como: la percepción gestáltica, la formación de imágenes, la asociación con movimientos motores, la organización del conocimiento, la facilidad de procesamiento cognoscitivo (aprendizaje, reconocimiento, memoria, etc.), y la simplicidad de la expresión lingüística.” (pág. 13 de [Lakoff, 87]).

En el nivel supra-ordinario preponderan las funciones y el propósito como atributos, el procesamiento de imágenes no se da, es lo que Lakoff denomina categorías construidas por “un esfuerzo de imaginación”.

Se puede argumentar que el nivel supra-ordinario está presente en catálogos como GoF, POSA 1, POSA 2 y [Alur, 2001]:

| Catálogo                     | Categorías de nivel supra-ordinario  | Subcategorías de nivel supra-ordinario  |
|------------------------------|--|---|
| GoF<br>“Behavioral Patterns” | “Creational Patterns”<br>“Structural Patterns”   | “Creational Class Patterns”<br>“Creational Object Patterns”<br>“Structural Class Patterns”<br>“Structural Object Patterns”<br>“Behavioral Class Patterns”<br>“Behavioral Object Patterns” |
| POSA 1                       | “From Mud to Structure”<br>“Distributed Systems”<br>“Interactive Systems”<br>“Adaptable Systems”<br>“Structural Decomposition”<br>“Organization of Work”<br>“Access Control”<br>“Management”<br>“Communication”<br>“Resource Handling” |   |
| POSA 2                       | “Communication”<br>“Initialization”<br>“Service Access and Configuration”<br>“Event Handling”<br>“Synchronization”<br>“Concurrency”  |   |
| Alur                         | “Presentation Tier Patterns”<br>“Business Tier Patterns”<br>“Integration Tier Patterns”  |   |

El nivel ordinario está constituido por los patrones en sí y es el nivel que a la fecha se elaborado mejor en todos los catálogos. Aunque no se describen variantes ni combinaciones de patrones en ninguno de los catálogos referidos, en algunos casos sí se mencionan. Por ejemplo [Alur, 2001] menciona brevemente variantes de “Proxy” y discute sus diferencias. En [Alur, 2001] también se introducen secciones para estrategias de programación específicas en la mayoría de los documentos de patrones, algunas de éstas podrían considerarse en sí mismas patrones de programación (“idioms”) que son variantes de otros. Este tipo de evidencia permite por lo menos argumentar la existencia de un nivel sub-ordinario incipiente en algunos de los catálogos.

#### 4. Patrones y palabras

Finalmente, si un patrón es un plan, entonces representa una cantidad potencialmente infinita de realizaciones, cada una en el contexto de secuencias continuas de acciones situadas singulares. Aplicar un patrón es como usar una palabra, siempre debe ajustarse e interpretarse en el contexto específico. De hecho los patrones funcionan como las categorías de los lenguajes naturales.

Por lo tanto, dado que:

1. en el proceso de denominación de los patrones emergen esquemas de imagen provenientes de otras categorías naturales que quedan sutilmente asociadas a los patrones a través de sus nombres,
2. la organización de los catálogos de patrones publicados muestran características típicas de los procesos de categorización en los lenguajes

naturales tales como el principio del dominio, el uso de niveles de categorización como el supra-ordinario y el nivel básico,

3. cada patrón es un representante de infinita cantidad de realizaciones,

se puede argumentar razonablemente que los lenguajes de patrones no son análogos a los lenguajes naturales, sino que *son de hecho* “subsistemas” de los lenguajes naturales. El nivel básico de categorización viene dado precisamente por los patrones mismos. Los patrones son categorías. No solo representan a todas sus posibles realizaciones pero además cada patrón tiene naturalmente asociado otros patrones que son variantes y combinaciones. Un patrón que a su vez es *variante* de otro, típicamente introduce cambios en la estructura y por ende en las consecuencias de su aplicación. Una patrón que es una *combinación*, consiste en la fusión de dos o más patrones. En ambos casos la intencionalidad es más específica que en el patrón original; esto genera una diferencia principalmente en la estructura de solución y las consecuencias de aplicar el(los) patrón(es) original(es). Por ejemplo POSA 1 menciona variantes y combinaciones de “Proxy”. Las variantes y combinaciones constituyen el nivel sub-ordinario.

Una vez explicados los dos puntos novedosos en la caracterización de lenguajes de patrones, a saber el concepto de plan y la relación que subyace entre lenguajes de patrones y lenguajes naturales, es necesario replantear el modelaje de los lenguajes de patrones pues esta será la base para la organización del conocimiento de patrones, para mejorar la expresividad de los lenguajes de patrones y para derivar finalmente un patrón

para sistemas de patrones (este patrón se expone en [Calderón, 2003a]).

Si un patrón es una categoría de un lenguaje de patrones que a su vez es una especialización de un lenguaje natural, entonces es posible usar la teoría de Modelos Cognoscitivos Idealizados de [Lakoff, 87] para modelar los lenguajes de patrones, para comprender mejor su estructura y funcionamiento, así como la red de asociaciones entre sus categorías. De nuevo, aunque en primera instancia se haya llegado a un resultado de interés meramente teórico, es importante adelantar que sí se pueden derivar consecuencias prácticas. Es posible derivar nuevos principios organizativos para los lenguajes de patrones a partir de la estructura de los procesos de categorización naturales, tal como estos son comprendidos bajo el marco de la teoría de Lakoff. Más adelante se verá cómo se puede intentar una respuesta nueva a algunas de las cuestiones todavía pendientes en la comunidad de autores y usuarios de patrones.

Finalmente, es conveniente apuntar que este enfoque no contradice de ninguna manera el pensamiento de Alexander, pero que sin embargo sí provee un marco teórico cognoscitivista más elaborado. La caracterización funcional del concepto de lenguaje de patrones que usa Alexander se mantiene. Por otro lado, la gramática de un lenguaje natural puede verse de hecho como un sistema finito de reglas que obviamente se aplica también a un lenguaje de patrones como una de sus especializaciones. Sin embargo, en cuanto a la relevancia que Alexander da al sistema de reglas sí hay una diferencia. Esta propuesta asume el planteamiento de Ellis que indica que “Es fácil mostrar que la gramática es solo un nivel más general de organización

que el que soportan las palabras individuales, pero que teóricamente no tiene que verse como un proceso distinto” (traducción mía de la pág. 49 de [Ellis,93]). Desde la perspectiva de este autor lo central en un lenguaje natural es el sistema de categorización que está orientado a simplificar dominios de experiencias, y cuyo objetivo es resolver problemas, aprender y comunicarse: “Esta simplificación es el hecho central del proceso de categorización, y por lo tanto el hecho central del lenguaje y del conocimiento que este nos provee acerca de nuestro mundo” (traducción mía de la pág. 29 de [Ellis, 93]).

En la siguiente sección se plantea una nueva estructura<sup>9</sup> para los lenguajes de patrones que integra la propuesta de Alexander pero además la extiende bajo el marco teórico expuesto.

## Estructura de un lenguaje de patrones

Con base en la teoría de [Lakoff, 87], el trabajo empírico que complementa a esta propuesta (ver [Calderón, 2003c]) y el análisis de catálogos como GoF, POSA 1, POSA 2 y [Alur, 2001], se puede caracterizar la estructura de un lenguaje de patrones de la siguiente manera:

1. Organiza el conocimiento de patrones con base en cinco tipos de categorías naturales:
  - a. **Dominios de Diseño:** un dominio de diseño corresponde con un dominio de experiencias organizado por la estructura de un lenguaje de patrones. Por ejemplo, modelaje de requerimientos, diseño arquitectónico, diseño de objetos, diseño detallado

- o programación. Todo lenguaje de patrones está asociado con un dominio de diseño específico. Aquí se aplica el principio del dominio de experiencia de Lakoff.
- b. Categorías de Patrones: por ejemplo los de la tabla de la página 9.
  - c. Patrones.
  - d. Variantes y Combinaciones de Patrones: por ejemplo las variantes y combinaciones de “Proxy” discutidas en POSA 1.
  - e. Categorías de asociaciones: por ejemplo cuando se afirma que un patrón X “complementa a”<sup>10</sup> otro Y, o que un patrón X es “similar a” Y en su propósito, o que un patrón X “contrasta con” otro Y en su propósito o consecuencias. Se puede identificar categorías de asociaciones propias de las categorías de patrones, como al afirmar que los “Patrones de Construcción” “complementan a” los “Patrones de Estructuración” de GoF. También categorías de asociaciones propias de los dominios de experiencia de diseño como al afirmar que los “Patrones de diseño de objetos” “complementan a” los “Patrones arquitectónicos”.
2. Dado un dominio de diseño específico, organiza el conocimiento de patrones con base en tres niveles de categorización: supra-ordinario, básico u ordinario y sub-ordinario. El nivel supra-ordinario consiste en un sistema de categorías de patrones. El nivel básico u ordinario consiste precisamente en un sistema de patrones. El nivel sub-ordinario consiste en un sistema de patrones que son variantes y combinaciones de otros patrones. Aquí se aplica el principio de los tres niveles de categorización de Lakoff.
    - a. Es un sistema abierto que desarrolla asociaciones con las estructuras de otros lenguajes de patrones. Esta evolución se da de manera natural conforme los dominios de experiencia de diseño se van entrelazando. En la medida en que esto se da, se puede hablar de dominios afines y por ende de lenguajes afines.
    4. Una categoría de un lenguaje de patrones, de cualquiera de los cinco tipos mencionados, puede ser modelada con un “Modelo Cognoscitivo Idealizado” (en adelante MCI) y por ende se estructurará con base en principios tales como: estructura proposicional, estructura basada en esquemas de imagen, metáfora y metonimia (ver pág. 68 de [Lakoff, 87]).
    5. El trabajo empírico que complementa esta propuesta muestra que son particularmente relevantes, para modelar la estructura de los lenguajes de patrones, los MCI de estructura radial o categorías radiales según Lakoff. Los rasgos principales de una MCI con estructura radial son (ver pág. 91 de [Lakoff, 87]):
      - a. Un modelo central denominado prototipo.
      - b. Modelos no centrales o periféricos que se asocian al central.
      - c. Los modelos no centrales asociados al central deben aprenderse pues son convencionales, no existen reglas generales para derivarlos.
      - d. Principios que *motivan*<sup>11</sup> la inclusión de los modelos no centrales.
    6. Como ya se ha visto, los nombres de los patrones ponen en evidencia procesos naturales de metáfora y

metonimia, lo cual establece vínculos sutiles entre un lenguaje de patrones y el lenguaje natural en que está empotrado.

7. La denominación de patrones que son variantes y combinaciones de otros obedecerá a reglas gramaticales propias del lenguaje natural. Estas reglas corresponden con niveles de centralidad y generalización en el sistema de categorización. Aquí se aplica el principio del continuo entre gramática y léxico de [Ellis, 93]. Por ejemplo los nombres de las variantes y combinaciones de “Proxy” en POSA 1 se puede explicar en términos de estas reglas<sup>12</sup>.

Es importante señalar que esta caracterización de la estructura de un lenguaje de patrones no contradice la de Alexander. La conceptualización de Alexander se fundamenta básicamente en la categoría de asociación entre patrones que podría enunciarse como “complementa a”. Existen muchos ejemplos de esta categoría de asociación en los catálogos publicados. Considérese tan solo uno a manera de ilustración: la asociación entre “Interpreter” y “Composite” en GoF. Se afirma que una aplicación de “Intérprete” puede programarse aplicando “Compuesto” para representar el árbol sintáctico abstracto. En otras palabras, al aplicar “Intérprete” en un contexto de diseño específico podrá usarse también “Compuesto” para representar el árbol sintáctico abstracto que forma parte de la solución de “Intérprete”. En este sentido “Compuesto” “complementa a” “Intérprete”. Alexander visualiza la estructura de un lenguaje de patrones básicamente como un digrafo<sup>13</sup> cuyos nodos son patrones y cuyos arcos repre-

sentan ejemplos de la categoría de asociación “complementa a”. Al incluir esta categoría de asociaciones en un lenguaje de patrones, se estará abarcando el aspecto esencial del concepto de estructura de lenguaje de patrones de Alexander.

Hay sin embargo otros dos aspectos importantes en el concepto de estructura de Alexander (ver pág. 314 de [Alexander, 79]):

1. Existen patrones que son puntos iniciales o puntos de entrada en un digrafo de patrones. Estos puntos de entrada representan metas en un proceso de diseño.
2. Cada punto de entrada representa para un usuario de patrones un conjunto finito de caminos a través del digrafo que representan secuencias alternativas de acciones de diseño (conocidas como secuencias de diseño). El diseñador va agregando un patrón a otro, siguiendo un camino a través del digrafo, y al final obtendrá el resultado deseado.

Asumir la estructura propuesta en este trabajo no impide que se puedan identificar puntos de entrada y secuencias de diseño, sin embargo la estructura como tal no puede formalizarse fácilmente como un digrafo plano como el que propone Alexander. Para empezar, tan solo como ejemplo, habría que considerar varios tipos de nodos, varios tipos de arcos, los caminos atravesarían niveles de categorización y dominios de experiencia, los arcos deberían mostrar propiedades como *gradualidad* y *centralidad*<sup>14</sup>. Aunque no hay espacio para tal desarrollo en este trabajo<sup>15</sup>, sí es posible construir criterios generales

para valorar un lenguaje de patrones<sup>16</sup> a partir de la estructura descrita, que en todo caso es el uso principal que da Alexander a su concepto de estructura de lenguajes de patrones. Los conceptos de completitud funcional y completitud morfológica están íntimamente ligados con el de estructura de lenguaje de patrones en el pensamiento de Alexander. En la siguiente sección se reinterpretan los conceptos de completitud de un lenguaje de patrones y se intenta así una nueva respuesta a algunas de las inquietudes todavía pendientes en la comunidad de autores y usuarios de patrones.

### Completitud morfológica y funcional en un lenguaje de patrones

Las nociones de completitud funcional y morfológica no han sido transferidas al campo del diseño de *software*<sup>17</sup>, a pesar de que otros aspectos de la propuesta de Alexander sobre patrones y lenguajes de patrones sí se han venido usando intensamente en los últimos ocho o diez años. Un resultado de esta situación es que no existen criterios para distinguir entre un catálogo de patrones en el que se incluye un diagrama con las asociaciones de la categoría “complementa a” y un lenguaje de patrones. No existen criterios para determinar si un conjunto de patrones es más completo que otro, ni mucho menos métricas que permitan establecer el grado de completitud de un conjunto de patrones. Ni siquiera está claro qué significará “completitud” en este contexto.

En el caso del diseño de *software* la situación es todavía más complicada que en la arquitectura porque buena parte de los esfuerzos de desarrollo de *software* están orientados a producir *software*

reutilizable. Este es *software* que en sí mismo no consiste en una aplicación final, sino que se construye para facilitar la elaboración posterior de aplicaciones finales. Es el caso de lo que en la jerga se conoce como “frameworks” o “marcos de trabajo”. También se ubican en esta categoría de *software* reutilizable las bibliotecas de funciones y las bibliotecas de clases. En todos estos casos se trata de *software* elaborado con el propósito de construir otros sistemas de *software* que probablemente<sup>18</sup> sí van a ser usados como aplicaciones finales. En la arquitectura no existe esta categoría de construcciones. Alexander considera jardines, casas, edificios, ciudades, pero nunca estructuras reutilizables, hechas solo para construir estas categorías de lugares.

Alexander introduce sus caracterizaciones de completitud morfológica y funcional de la siguiente manera:

“Pero aún cuando he logrado conectar los patrones uno con otro, en una red, de tal manera que formen un lenguaje, ¿cómo puedo saber si el lenguaje es bueno? ¿Es completo? ¿Debo agregar otros patrones? ¿Debo eliminar algunos? ¿Se sostienen entre sí? Y, principalmente, ¿me permitirán generar un jardín que sea vivo?...¿Cómo puedo estar seguro?” (ver pág. 316 de [Alexander, 79]).

Ahora bien, para poder decidir si es necesario agregar o eliminar patrones a un lenguaje, para poder valorar si los patrones “se sostienen entre sí”, se requiere una representación operacional del lenguaje de patrones. Además esta representación debe ser el resultado de una puesta en común, pues como el lenguaje natural, no se trata de una construcción individual sino más bien colectiva. De acuerdo con el marco teórico elaborado, no es posible ni tiene ningún valor práctico intentar aplicar las nociones normativas de completitud a un lenguaje de patrones en sí mismo.

Como especialización de un lenguaje natural, un lenguaje de patrones es un sistema vivo. Existe en la intersubjetividad y en la intrasubjetividad, se auto-re-elabora, auto-re-estructura y auto-re-genera en cada pensamiento y en cada conversación, en cada acción situada de diseño. Lo central en un lenguaje, su estructura de categorización existe a nivel cognoscitivo. Cada individuo usa su propio sistema de categorización único. Probablemente coincide en algún grado con el de los demás individuos porque ha sido construido precisamente a través de los procesos del lenguaje, pero no es idéntico y, sobre todo, sigue su propio proceso evolutivo<sup>19</sup>. Evidentemente no es posible ni tiene ningún valor práctico intentar normar los procesos de categorización natural de cada individuo. Sin embargo, sí es posible construir colaborativamente una estructura de categorización compartida. Una estructura objetivizada en la intersubjetividad: “La objetividad es un producto de prácticas sistematizadas, es decir, de los métodos que usan los miembros<sup>20</sup> para presentar sus experiencias singulares y circunstancias relativas de manera que sean mutuamente inteligibles” (ver pág. 57 de [Suchman, 87]). Esto es precisamente lo central en el lenguaje, simplificar las experiencias de un dominio para poder compartirlas, es decir hacerlas mutuamente inteligibles<sup>21</sup>. Para hacerlo, es imprescindible una representación operacional de las categorías que simplifican estas experiencias de diseño y las hacen mutuamente inteligibles. Esta representación es lo que se podría denominar un “lexicón de patrones”. En este sentido, los catálogos analizados pueden ser considerados de hecho lexicones de patrones<sup>22</sup>. Si se asume el marco teórico propuesto, no importa cuál sea la estructura de un lexicón de patrones, como

representación que es de una estructura de categorización objetivizada de un lenguaje de patrones, deberá valorarse con respecto a la estructura propuesta. En última instancia lo que se busca es mejorar un lexicón de patrones para mejorar el lenguaje correspondiente. ¿Mejorar en qué sentido? Un sistema de categorización objetivizado mejora en la medida en que facilita aún más a los individuos que usan el lenguaje:

1. La sistematización del conocimiento del dominio de diseño. Esto significa facilitar la generación, organización y simplificación del conocimiento obtenido de las experiencias de diseño, en un dominio específico.
2. El compartir conocimiento del dominio de diseño.
3. La resolución de problemas específicos del dominio de diseño, en forma individual y colaborativa.
4. Aprender a resolver problemas en el dominio de diseño.

A continuación se hace un primer intento por transferir los conceptos de completitud morfológica y funcional de Alexander al campo del diseño de *software*. Se parte del supuesto de que se trata de grados de completitud y no simplemente de “completitud” o “incompletitud”. No se pretende agotar el tema, ni mucho menos, sino más bien señalar posibles caminos para investigaciones posteriores. Por esta razón se ha adoptado un enfoque heurístico, planteando algunos criterios que podrían usarse para estimar cualitativamente la completitud morfológica y funcional de un lexicón de patrones. Todos estos criterios deben verse más

bien como hipótesis que habría que validar empíricamente de alguna forma.

Los criterios han sido redactados como estrategias para mejorar el grado de completitud morfológica y funcional de un lenguaje de patrones, con el propósito de que se evidencien las implicaciones prácticas del enfoque teórico descrito en este trabajo.

### **Criterios para completitud funcional**

El grado de completitud funcional de un lexicón de patrones y consecuentemente de un lenguaje de patrones, será mayor si:

1. Se logra aumentar el grado de completitud funcional a nivel supra-ordinario, es decir a nivel del sistema de categorías de patrones que constituye un dominio de diseño dado. Esto se puede lograr mediante:
  - a. Identificación de nuevas asociaciones de tipo “complementa a” en relación con otros dominios de diseño. Esto dependerá de la cantidad de asociaciones “trans-dominio” que se den en los niveles inferiores.
  - b. Identificación de nuevas asociaciones de tipo “complementa a” entre las categorías del dominio y también con categorías de otros dominios.
  - c. Aumento del grado de completitud funcional a nivel básico u ordinario.
2. Se logra aumentar el grado de completitud funcional a nivel básico u ordinario, es decir a nivel del sistema de patrones que constituye a cada una de las categorías de patrones. Esto se puede lograr mediante:
  - a. Identificación de nuevas asociaciones de tipo “complementa a” entre los patrones de cada una de las categorías del dominio.
  - b. Identificación de nuevas asociaciones del mismo tipo entre patrones de distintas categorías del mismo dominio.
  - c. Identificación de nuevas asociaciones “trans-dominio” del mismo tipo. Son particularmente importantes las relaciones de “implementación”, donde un patrón se puede usar como parte de la solución de otro.
  - d. Aumento del grado de completitud funcional a nivel sub-ordinario.
3. Se logra aumentar el grado de completitud morfológica a nivel sub-ordinario, esto es a nivel interno de cada patrón como categoría. Esto se puede lograr mediante:
  - a. Identificación de nuevas asociaciones del tipo “complementa a” entre los subpatrones de distintos patrones, es decir los patrones que son variantes y combinaciones de patrones.
  - b. Identificación de nuevas asociaciones “trans-dominio” del mismo tipo. Son particularmente importantes las relaciones de “implementación”, donde un subpatrón se puede usar como parte de la solución de otro.

### **Criterios para completitud morfológica**

El grado de completitud morfológica de un lexicón de patrones y consecuentemente un lenguaje de patrones, será mayor si:



1. Se logra aumentar el grado de completitud morfológica a nivel supraordinario, es decir a nivel del sistema de categorías de patrones que constituye un dominio de diseño dado. Esto se puede lograr mediante:
  - a. Enriquecimiento de la red de asociaciones que se haya logrado construir entre las categorías de patrones del dominio de diseño. Particularmente con más asociaciones de categoría “similar a” y “contrasta con”. Estas asociaciones aclaran el panorama de los problemas y contextos generales que abordan las distintas categorías de patrones.
  - b. Aumento de la fuerza de la *motivación* de las asociaciones entre las categorías de patrones del dominio, en el sentido que usa el término Lakoff. Al hacerlo serán aun más significativas las asociaciones. En este nivel, la fuerza de una motivación que subyace una asociación depende de cuántas asociaciones de la misma categoría existen en el nivel ordinario, de cuántas entre patrones de las dos categorías y de cuán centrales son estos patrones respectivamente.
  - c. Aclaración de las características de complejidad y envergadura de las estructuras de los diseños asociados con las categorías del dominio.
  - d. Aumento del grado de completitud morfológica de las categorías de patrones.

Nota: De los cuatro dominios identificados a la fecha, los mejor estructurados son el de diseño arquitectónico y el de diseño de objetos (mejor conocidos como patrones arquitectónicos y patrones de diseño). A la fecha no es posible dar una respuesta definitiva

sobre si tiene sentido considerar un dominio genérico de “modelaje de requerimientos” o si será mejor considerar distintos dominios asociados con lo que se conoce en ingeniería de *software* como “dominios de aplicación”. Lo mismo podría decirse sobre un hipotético dominio genérico de “idioms” o patrones de programación. Un análisis más detallado podría concluir que es mejor constituir dominios específicos para cada lenguaje de programación o al menos para cada tipo o paradigma de lenguajes (por ejemplo, programación procedimental, funcional, lógica, orientada a objetos, etc.).

2. Se logra aumentar el grado de completitud morfológica a nivel básico u ordinario, es decir a nivel del sistema de patrones que constituye a cada una de las categorías de patrones. Esto se puede lograr mediante:
  - a. Enriquecimiento de la red de asociaciones que se haya logrado construir entre los patrones de cada categoría, particularmente con asociaciones de categoría “similar a” y “contrasta con”, pero también “transformación de” y “extiende a”<sup>23</sup>. Estas categorías de asociaciones completan el panorama de los contextos y los problemas que abordan los patrones de una categoría, así como sus características generales de diseño<sup>24</sup> que promueven.
  - b. Aumento de la fuerza de la *motivación* de las asociaciones, en el sentido que usa el término Lakoff. Al hacerlo serán aun más significativas las asociaciones. En este nivel, la fuerza de una motivación que subyace una asociación depende de similitudes y contrastes que se han logrado hacer

explícitos entre el problema, el contexto, las consecuencias y las asociaciones de los patrones involucrados.

- c. La identificación de un patrón prototipo para la categoría de patrones, en caso de que no existiera, o cambio del existente por un patrón que resulte ser mejor como prototipo. Así será más fácil visualizar la categoría de problemas y contextos que abordan los patrones de la categoría; en particular será más fácil visualizar las características generales de diseño asociadas con cada categoría.
  - d. Una mayor afinidad semántica entre el nombre de cada categoría y el nombre del patrón prototipo correspondiente. Aquí afinidad semántica podría entenderse como el sistema de metáforas y metonimias que asocia el de una categoría con el nombre de su patrón prototipo. Por ejemplo, “Patrones de estructuración” es el nombre de una categoría de patrones. Un esquema de imagen prototípico del concepto de “estructura” es “Totalidad-Parte”. Se podría argumentar así que el patrón “Whole-Part” de POSA 1 es mejor ejemplo de la categoría “Patrones de Estructuración” que “Compuesto” de GoF, o sea, es el patrón prototípico.
  - e. Aumento del grado de completitud morfológica de los patrones de la categoría.
3. Se logra aumentar el grado de completitud morfológica a nivel sub-ordinario, esto es, a nivel interno de cada patrón como categoría. Un patrón representa no solamente un sistema potencialmente infinito de aplicaciones, sino también un sistema de sub-patrones o variantes y combinaciones de patrones. El primero

solo se puede representar intencionalmente a través de la especificación del patrón y para el segundo es posible construir un mapa de variantes y combinaciones que muestre sus asociaciones. Entonces el grado de completitud morfológica de un patrón se puede aumentar mediante:

- a. Enriquecimiento de la complejidad semántica del nombre del patrón introduciendo metáforas y metonimias<sup>25</sup>.
- b. La inclusión de la descripción detallada de una aplicación prototípica del patrón. Esta es una costumbre muy difundida entre los autores de patrones. Evidentemente se favorece con el ejemplo la caracterización del problema, el contexto y las características generales de diseño que promueve el patrón.
- c. El descubrimiento de patrones de variantes del patrón. Entre más se hayan documentado, mayor será la base de comparación y contraste del patrón original.
- d. El descubrimiento de patrones de combinaciones con otros patrones.

## Conclusiones

1. Se ha presentado un marco teórico nuevo que puede ayudar a comprender mejor los lenguajes de patrones, particularmente su organización y posible evolución.
2. Se ha mostrado que este nuevo marco teórico no contradice el pensamiento de Alexander, tal como ha sido presentado en [Alexander, 79], pero en cambio lo ubica en un contexto más amplio.

3. Se obtienen consecuencias prácticas al asumir este nuevo marco teórico:
  - Se establece una diferenciación clara entre lenguaje de patrones, estructura de lenguaje de patrones, sistemas de patrones (lexicones) y catálogos de patrones.
  - Se han construido los conceptos de completitud morfológica y funcional tratando de preservar la intención de los originales de Alexander, pero detallándolos y ampliándolos con base el marco teórico propuesto.
  - Los nuevos conceptos de completitud morfológica y funcional pueden ser la base de posteriores investigaciones tendientes a validarlos, con el fin de lograr que sean efectivos al tratar de mejorar los lenguajes de patrones que se van desarrollando en el campo del diseño de *software*.
  - El modelo propuesto para la estructura de un lenguaje de patrones sugiere un nuevo diseño para sistemas de patrones denominados aquí “lexicones de patrones”, lo que permitirá lograr mejores representaciones de lenguajes de patrones<sup>26</sup>.
  - Al introducirse las categorías de asociaciones como aspecto esencial del modelo de la estructura de un lenguaje de patrones, se recupera un tema que la comunidad de autores y usuarios de patrones ha considerado muy relevante para poder mejorar la expresividad de los lenguajes de patrones<sup>27</sup>.
  - Pero además, al proponer una estructura más compleja, es posible argumentar que para enriquecer la expresividad de un lenguaje de patrones no es suficiente con conseguir nuevas categorías de patrones, es necesario dar su lugar a los distintos tipos de categorías, así como a las asociaciones entre estas categorías.
- La estructura propuesta no es más sencilla que la que subyace los catálogos publicados hasta ahora, pero se argumenta que puede ser mejor para facilitar el aprendizaje de patrones y también la búsqueda de patrones.
4. A pesar de que la estructura propuesta es más compleja, la administración de un lexicón de patrones puede facilitarse si se provee la funcionalidad necesaria, tal como se propone en [Calderón, 2003a].
5. Es posible mejorar, desde el punto de vista de los criterios construidos, los catálogos de patrones publicados. Esto se ha hecho en [Calderón, 2003c] mediante la integración de varios catálogos y su organización según los lineamientos dados.
6. Este trabajo sugiere nuevas investigaciones o nuevos enfoques para:
  - a. Identificar nuevas categorías de asociaciones.
  - b. Validar criterios de completitud funcional y completitud morfológica.
  - c. Modelar formalmente la estructura de categorización con el fin de construir una base de conocimientos para mejorar el funcionamiento de un lexicón de patrones.
  - d. Elaborar y validar un diseño de cursos para la enseñanza de la ingeniería de *software* basada en patrones.

- e. Elaborar y validar un método para extraer conocimiento de patrones a expertos en diseño de *software*.
7. Esta propuesta se ha orientado hacia la obtención de implicaciones prácticas para el campo del diseño de *software*, pero al basarse en una teoría general del lenguaje muchos de los planteamientos posiblemente son transferibles a otras disciplinas de diseño.
8. Finalmente, es sugestivo terminar con un comentario especulativo de Rosch en [Rosch, 78]. Afirma esta investigadora que el descubrimiento del nivel básico de categorización y por ende los otros dos niveles, podría indicar que los lenguajes naturales evolucionan “desde el centro hacia fuera”, es decir desde el nivel básico hacia el supra-ordinario por generalización, y desde el nivel básico hacia el sub-ordinario por especialización. Si esto fuera así, se podría argumentar que la evolución de los lenguajes de patrones coincide perfectamente, pues de hecho lo que mejor se ha elaborado hasta el momento es el nivel básico en las cuatro dominios explorados.
3. Aquí el autor se refiere a una categoría de sistemas como la de los lenguajes naturales.
4. Menos literal, podría ser la traducción “más detallados”.
5. Lo que puede interpretarse como una reducción del lenguaje, como sistema, al sistema sintáctico. En años posteriores, muchos autores, entre ellos Lakoff y Ellis, han argumentado sólidamente en contra de este supuesto, (ver [Lakoff, 87] y [Ellis, 93]).
6. De hecho, probablemente más bien parezca inútil.
7. Aquí el término se usa en la primera acepción que indica el Diccionario de la Real Academia Española: “Común, regular y que sucede habitualmente”, lo que en este contexto significa el nivel de categorización común.
8. En estos casos por *modelo* debe entenderse “Modelo Cognoscitivo Idealizado” y por *dominio* “Dominio de Experiencia” (ver [Lakoff, 87]).

## Notas

1. Cabe aclarar que en [Alexander, 79] **no** se presentan estas confusiones.
2. También se ha detectado que cambia aunque no tan rápido como crece. Cambia por ejemplo conforme se van publicando diferentes versiones de un mismo patrón, ver por ejemplo la discusión sobre “Proxy” en [Buschmann, 96].
9. Estructura como estructura de categorización.
10. Alexander da particular importancia a esta categoría de asociación pues insiste en que en un lenguaje de patrones debe ser posible recorrer cadenas de patrones que completan o complementan el diseño de un edificio. En ingeniería de *software* estas han sido denominadas cadenas de patrones de diseño.

11. Lakoff utiliza el término “motivación” para indicar, en términos generales, que existe un parecido familiar entre el modelo prototípico de la categoría y los modelos no centrales (ver págs. 65, 91, 96 y otras de [Lakoff, 87]).
12. Otro ejemplo de POSA 2 es un patrón denominado “Wrapper Facade”. Los autores no usaron “Facade Wrapper”. Ambos, “Wrapper” y “Facade” son patrones de GoF. La escogencia hecha en POSA 2 no es para nada arbitraria y refleja entre otras cosas que el nuevo patrón se parece más a “Facade” que a “Wrapper” en su intencionalidad, pero que combina propiedades de ambos patrones.
13. Entiéndase por *digrafo* un grafo cuyos arcos tienen dirección, es decir representan una relación de orden entre los nodos.
14. Estos son considerados por Lakoff como efectos observables en procesos de categorización modelables a través de categorías radiales.
15. En [Calderón Castro, 2003a] se ha elaborado un modelo que especifica un sistema de *software* orientado a soportar el desarrollo de lenguajes de patrones y que se basa en la estructura caracterizada.
16. En un sentido estricto, lo que se puede valorar es una representación de la estructura de categorización compartida por quienes usan un lenguaje de patrones, o parafraseando a Suchman, la estructura de categorización objetivizada en la intersubjetividad.
17. Esto se ha concluido después de revisar los catálogos de patrones más depurados publicados hasta la fecha y que son considerados ejemplos a imitar por la comunidad de autores y usuarios de patrones (GoF, POSA 1, POSA 2 y [Alur, 2001]) y después de haber planteado la inquietud en el foro de discusión en Internet “patterns-discussion@cs.uiuc.edu” donde, participan muchos de los autores más connotados sobre el tema, por ejemplo Ralph Johnson co-autor de GoF, Frank Buschmann co-autor de POSA 1 y Douglas C. Schmidt co-autor de POSA 2.
18. Obviamente no se puede descartar que una pieza de *software* reutilizable se utilice para construir otra con intención de ser reutilizable también.
19. Alexander coincide totalmente al afirmar que “*Un lenguaje tiene vida solo cuando cada persona en un grupo social, o en un pueblo, tiene su propia versión del lenguaje.*” (véanse págs. 337-341 de [Alexander, 79]).
20. Los miembros de un grupo humano, por ejemplo la comunidad de los ingenieros de *software* que son autores o usuarios de patrones.
21. En esto Ellis y Suchman coinciden plenamente.
22. Por esta razón es que en [Calderón, 2003a] se propone un patrón para lexicones de patrones. Desde una perspectiva práctica, un lexicón de patrones es un sistema orientado a facilitar el uso de un lenguaje de patrones y reflejar su evolución natural, por lo que debe estructurarse en concordancia

con la estructura de un lenguaje de patrones. Debido a que un lenguaje de patrones es un sistema abierto y que se entrelaza naturalmente con otros, tal sistema de *software* debería representar en realidad a un sistema de lenguajes de patrones afines.

23. Según Lakoff, las asociaciones de estas categorías están *motivadas* directamente por relaciones entre los esquemas de imagen vinculados con las categorías. Si los esquemas de imagen se modelan como vectores de atributos, entonces las relaciones entre esquemas de imagen derivan de relaciones entre los vectores de atributos. En el caso de los patrones, los atributos podrían ser las características generales de diseño que promueven (ver nota 25).
24. Características generales relevantes pueden ser: robustez, eficiencia, flexibilidad ante cambios en los requerimientos, flexibilidad ante cambios en la base tecnológica (incluyendo hardware y *software*), reutilizabilidad, comprensibilidad.
25. Los patrones son categorías muy abstractas pues no es posible asociarles fácilmente una imagen de un objeto concreto. En el caso de los patrones de la arquitectura, por metonimia, se les puede asociar naturalmente el resultado final, a través de un lugar prototípico donde se ha usado el patrón. Este lugar tiene una imagen muy concreta. En el caso de los patrones de diseño de *software*, se acostumbra incluir un ejemplo prototípico en la especificación de cada patrón, pero al carecer de imagen sigue siendo bastante abstracto, como para ser

una buena síntesis del problema y el contexto del patrón. Sucede aquí algo similar a lo que ocurre en las secuencias de movimientos del Tai Chi: de hecho un patrón induce acciones situadas de diseño. Para resolver esto, los maestros de Tai Chi han usado nombres alegóricos, como “La Cigüeña” y “Cola de Pájaro”, que evidentemente son esquemas de imagen basadas en metonimias y metáforas del movimiento, de hecho, casi siempre, son imágenes de una postura inmersa en toda la secuencia de movimientos.

26. En [Calderón, 2003a] se describe un patrón para lexicones de patrones basado en la estructura descrita aquí.
27. En la convocatoria de la Conferencia ChiliPLoP 2003 se plantea el tema como uno de los tres objetivos centrales: “Nuestro propósito en este taller (el de expresividad de lenguajes de patrones) es entonces identificar cómo se pueden balancear estas fuerzas, y así obtener una representación de los lenguajes de patrones que sea más rica y más expresiva, de manera que se promueva la creatividad y colaboración sin sacrificar la claridad. Al mismo tiempo sería importante apoyar el trabajo del diseñador al tener que seleccionar patrones de una cantidad creciente de patrones y lenguajes de patrones, y la vez asegurar que sea fácil identificar las consecuencias de aplicar un patrón a un diseño específico.” (ver [http://hillside.net/chiliplop/2003/2003\\_expressiveness.htm](http://hillside.net/chiliplop/2003/2003_expressiveness.htm)). Además en [Calderón, 2003b] se describen algunas categorías de asociaciones nuevas, con lo que se pretende invitar a una sistematización

del conocimiento en lo que se refiere a categorías de asociaciones.

## Bibliografía

- [Alexander, 79] Alexander, Christopher. *The Timeless Way of Building*. Oxford University Press, Nueva York, 1979.
- [Alur, 2001] Alur, D., Crupi, J., Malks, D. *Core J2EE Patterns (Best Practices and Design Strategies)*. Sun Microsystems Press, E.E.U.U., 2001.
- [Buschmann, 96] Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., Stal, M. *Pattern-Oriented Software Architecture (A System of Patterns)*. John Wiley & Sons. West Sussex, Inglaterra, Octubre 1996. Conocido como POSA 1.
- [Calderón, 2003a] Calderón, Alan. Un patrón para lexicones de patrones. Aceptado en la tercera conferencia lationamericana PLoP: SugarLoaf PLoP 2003. Publicado electrónicamente en las memorias de la conferencia. Dirección del sitio en Internet: <http://www.cin.ufpe.br/~sugarloafplop/proceedings.html>. Accedido por última vez el 15 de junio del 2004.
- [Calderón, 2003c] Calderón, Alan. *Unifying pattern catalogs (towards a pattern for pattern lexicons)*. No ha sido publicado.
- [Ellis, 93] Ellis, John M. *Language, Thought and Logic*. Northwestern University Press. Evanston, Illinois, 1993.
- [Gamma, 95] Gamma, E., Helm R., Johnson, R. y Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, 1995. Conocido como GoF.
- [Lakoff, 87] Lakoff, George. *Women, Fire, and Dangerous Things (What categories reveal about the mind)*. The University of Chicago Press, Chicago, 1987.
- [Rosch, 78] Rosch, E. & Lloyd, B. *Cognition and Categorization*. Lawrence Erlbaum Associates, Publishers. New Jersey, 1978.
- [Schmidt, 2000] Schmidt, D., Stal, M., Rohnert, H., Buschmann, F. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. John Wiley & Sons. West Sussex, Inglaterra, January, 2000. Conocido como POSA 2.
- [Suchman, 87] Suchman, Lucy. *Plans and situated actions: The problem of human-machine communication*. Cambridge University Press, Cambridge, 1987.