

Lecciones aprendidas en la construcción de un *lexicón* de patrones de análisis de dominio

Lessons learned in building a *pattern lexicon* for domain analysis

Alan Calderón Castro, M.Sc.

Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica. San José, Costa Rica
alan.calderon@ecci.ucr.ac.cr

Recibido para revisión: 29 de Septiembre de 2008, Aceptado: 28 de Noviembre de 2008, Versión final: 16 de Diciembre de 2008

Resumen—Como resultado preliminar del análisis de patrones del dominio de sistemas de información empresarial, se describen ocho lecciones aprendidas para la creación de un *lenguaje de patrones de análisis de dominio* y su representación mediante un *lexicón de patrones de análisis de dominio*. Estas pautas metodológicas se han obtenido al organizar conocimiento de un dominio para usos didácticos.

Palabras Clave—Análisis de dominio, Familias de productos de software, Patrones de análisis, Lenguajes de patrones.

Abstract—As a preliminary result of the analysis of domain patterns for business processes eight learned lessons are described aimed at the creation of a *domain analysis pattern language* and its representation through a *domain analysis pattern lexicon*. These guidelines have been obtained through the organization of domain knowledge for didactical use.

Keywords—Domain analysis, Software product lines, Analysis patterns, Pattern languages.

I. INTRODUCCIÓN

El análisis de dominio para la construcción de *familias de productos de software* conlleva ciertos riesgos¹ que pueden ser mitigados al sistematizar el conocimiento de expertos en el dominio. Los lenguajes de patrones propuestos en [1] permiten tal sistematización. La Figura 1 presenta el marco conceptual de referencia para el presente trabajo que se describe brevemente a continuación:

- a. Patrón: En [1] se afirma que un patrón describe un problema que se reitera en un entorno de diseño y a continuación describe el núcleo de la solución a dicho problema, de tal forma que este núcleo de solución puede ser usado millones de veces sin repetir nunca una sola solución específica.
- b. Patrón de análisis: En el prefacio de [13] se afirma que los patrones de análisis son estructuras conceptuales que reflejan procesos de negocios en lugar de estructuras de software. Los patrones de análisis plantean núcleos de solución a problemas recurrentes de modelado de procesos de negocios. Se han publicado varios catálogos de patrones de análisis como [19], [13], [25], [7], [8] y [9]; además de numerosos artículos que describen colecciones pequeñas de patrones de análisis, como por ejemplo [10], [11], [12], [15], [18] y [26].

1. En [6] se enumeran por ejemplo: discernimiento incorrecto entre los requerimientos globales de la familia de productos y los requerimientos de productos específicos, insuficiencia de generalización, exceso de generalización, identificación incorrecta de puntos de variación.

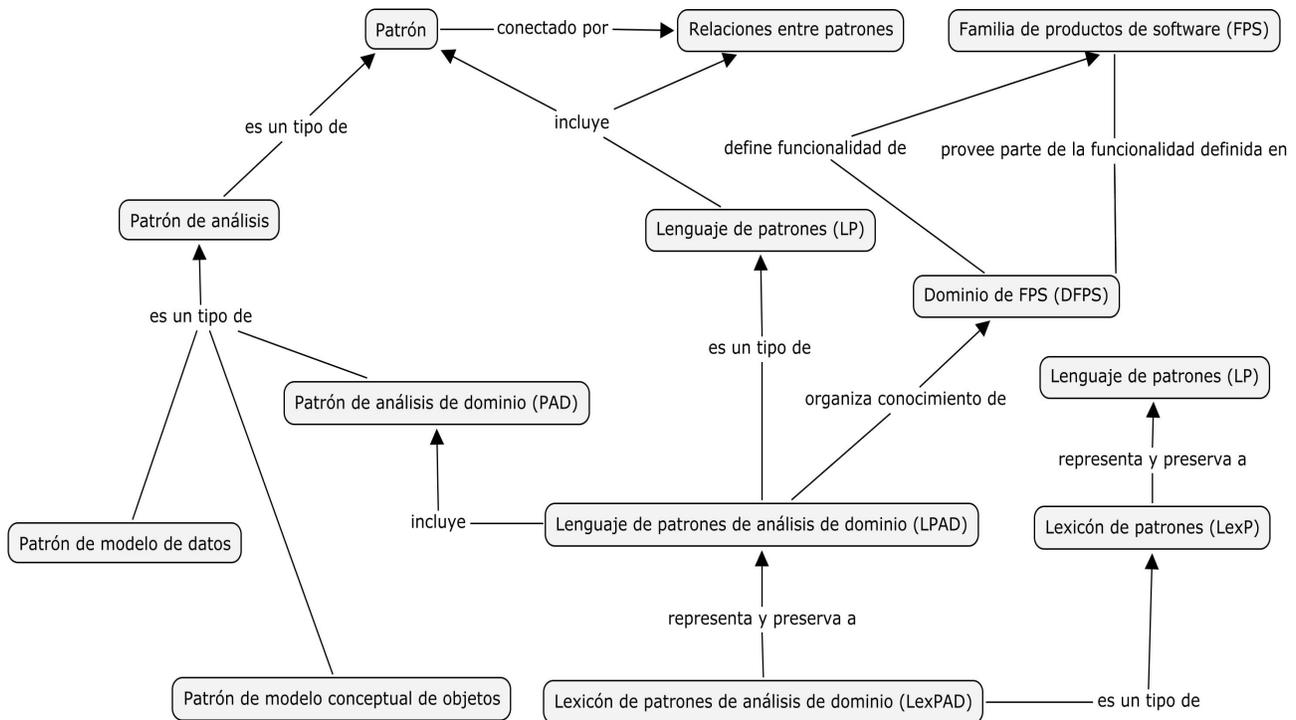


Figura 1. Marco conceptual de referencia.

- c. Familia de productos de software (FPS): “...una FPS² es un conjunto de sistemas intensivos en software que comparten un conjunto administrado de características que satisfacen las necesidades específicas de un segmento particular de un mercado o de un tipo de misión crítica y que son construidos a partir de un conjunto básico de activos de una manera prescrita.” ([6], pág. 5). Un ejemplo típico de una FPS es el conjunto de aplicaciones para ofimática de Microsoft que incluye MS-Word, MS-Excel, MS-Power Point, etc.
- d. Dominio de familia de productos de software (DFPS): “Un dominio es un cuerpo de conocimientos especializados, un área de pericia, o una colección de funcionalidad relacionada. Por ejemplo, el dominio de telecomunicaciones es un conjunto de funcionalidades de telecomunicación, que a su vez consiste de otros dominios tales como la conmutación (el “switching”), los protocolos, la telefonía y las redes. Una línea de productos de software es un conjunto específico de sistemas de software que provee parte de esta funcionalidad” ([6], pág. 14).
- e. Patrón de análisis de dominio (PAD): Es un patrón de análisis que sistematiza conocimiento de un dominio asociado a una FPS. Por tanto la descripción del problema

y el núcleo de su solución debe ser útil en el contexto de la construcción de productos de una FPS.

- f. Patrón de modelo de datos: Es un patrón de análisis que representa el núcleo de la solución mediante un modelo de datos³.
- g. Patrón de modelo conceptual de objetos: Es un patrón de análisis que representa el núcleo de la solución mediante un modelo conceptual de objetos⁴.
- h. Lenguaje de patrones (LP): Alexander define un lenguaje de patrones como “...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 balance exacto entre uniformidad y variedad que da vida a un lugar.” ([1], pág. 191).
- i. Lenguaje de patrones de análisis de dominio (LPAD): En [4] se afirma que “...un lenguaje de patrones de análisis orientado a un dominio de aplicación específico (o LPAD) provee a cada ingeniero de software que lo usa el poder de crear una infinita variedad de productos de software únicos y nuevos, pertenecientes a una misma FPS, de la misma forma en que su lenguaje ordinario le da el poder de crear una variedad infinita de oraciones.”
- j. Lexicón de patrones (LexP): En [5] el concepto de *lexicón*

2. Se ha traducido “software product line” como “familia de productos de software”.

3. Se trata de un modelo de entidades y relaciones orientado al diseño de una base de datos relacional.

4. Se trata de un modelo de objetos que representa el contexto del sistema, según el “Proceso Unificado de desarrollo de software”, véanse las páginas 112 a 115 de Jacobson, Booch y Rumbaugh (2000).

de *patrones* se ha planteado en primera instancia como una síntesis de las propuestas de [3] y [24] incluyendo además características valiosas de los esquemas organizativos de otros catálogos como el de [14], el de [2], el de [13], el de [19] y el de [25]. En términos generales un *lexicón de patrones* es un sistema para organizar patrones mediante la representación de lenguajes de patrones. La estructura de un *lexicón de patrones* incluye las características de los sistemas descritos y además:

- Integra la representación de lenguajes de patrones (LP) complementarios.
 - Facilita la evolución de los LP.
 - Facilita el aprendizaje por parte de los ingenieros de software de los LP representados.
- k. *Lexicón de patrones de análisis de dominio (LexPAD)*: En Calderón (2007) se afirma que un *lexicón de patrones de análisis de dominio* es un LexP que organiza PADs mediante la representación de uno o más LPAD complementarios para el desarrollo de una FPS.

El problema que se aborda en este trabajo consiste en ¿cómo crear un LPAD y representarlo mediante un LexPAD? dados los aportes de un conjunto de expertos en un DFPS. El énfasis del problema se hace en el proceso para sintetizar un LPAD y representarlo mediante un LexPAD. Dado el carácter metodológico del problema, el objetivo del presente trabajo es simplemente explicar una serie de lecciones aprendidas al construir un LexPAD con fines didácticos para una FPS del dominio de sistemas de información empresarial. De esta manera, aunque no se pretenda todavía hacer un planteamiento metodológico definitivo, la idea es aproximarse a un futuro método.

En la siguiente sección se presenta un ejemplo de dominio que servirá de base para explicar de manera más concreta las lecciones aprendidas. En la tercera sección se presentan las lecciones aprendidas sobre cómo analizar los aportes de los expertos de dominio. En la cuarta sección se presentan las lecciones aprendidas sobre cómo hacer una síntesis de los aportes mediante la construcción del espacio de patrones y su categorización. En la quinta sección se hace un esbozo de un método que constituye una hipótesis de trabajo para futuras investigaciones. Finalmente en la sexta sección se hace un recuento de lo expuesto y se perfilan acciones futuras para concretar un método definitivo.

II. UN CASO ESPECÍFICO DE DOMINIO

En esta sección se describe brevemente el DFPS que se ha usado para la creación de un LPAD y su representación mediante un LexPAD. Este trabajo se ha hecho con fines puramente didácticos, la intención es proveer a los aprendices de análisis orientado a objetos con UML (“Unified Modeling Language”) un LexPAD que facilite su proceso de aprendizaje

mediante su exposición al conocimiento sistematizado de un dominio específico. Al crear este LPAD se han elaborado una serie de pautas metodológicas preliminares que se exponen en las siguientes secciones.

El DFPS se ha delimitado implícitamente con base en modelos provistos por [19], [25], [13], [9] y diversos autores de artículos como los citados previamente en relación con patrones de análisis. Los autores presentan los modelos provistos principalmente como *patrones de modelos de datos* y *patrones de modelos conceptuales de objetos* orientados en términos generales a resolver problemas de modelado de procesos de negocio que son típicamente soportados por sistemas de información empresarial de carácter operativo. Cada patrón incluye un modelo (de datos o conceptual de objetos) y una explicación sobre las entidades u objetos y relaciones que lo conforman. Algunos procesos de negocio considerados por los autores son:

- a. Administración de productos de la empresa:
 - Actualización continua del catálogo de productos.
 - Aprovisionamiento de productos.
- b. Administración de inventarios:
- c. Administración de personas relacionadas con la empresa.

III. ANÁLISIS DE LOS APORTES DE LOS EXPERTOS DE DOMINIO

En esta sección se describen las lecciones aprendidas en relación con cómo analizar los aportes de los expertos en el dominio. El objetivo de estas heurísticas es construir una categorización “multinivel” (varios niveles de categorización es decir categorías que contienen a otras categorías) del conocimiento aportado y hacer una puesta en común del mismo.

Las diferencias de enfoque sobre un mismo dominio entre los expertos es el problema principal con que se debe lidiar en esta primera etapa. No sólo es posible que los expertos hayan enfocado distintos procesos organizacionales (o *procesos de negocio*), sino también distintas subcategorías naturales del dominio en cuestión. Además, con toda seguridad los conceptos, sus atributos y sus interrelaciones identificados por los expertos diferirán también. Por tanto, se ha encontrado que es útil realizar una categorización inicial de los modelos, un análisis de las relaciones entre los modelos, una puesta en común de los modelos y una revisión final de la categorización inicial a la luz de todo este análisis. A continuación se detalla cada una de estas tareas.

5. Tales como: 1) una subdivisión natural de la funcionalidad incluida en el DFPS, 2) las partes o módulos que se identifican en el diseño arquitectónico inicial de la FPS en construcción o 3) el agrupamiento de requerimientos funcionales por actor.

A. Categorización inicial de los modelos

Pensando en procesos organizacionales típicos del DFPS objetivo y en otros criterios naturales⁵ se deberá categorizar cada uno de los modelos aportados por los expertos. Esta categorización puede ser de varios niveles y no necesariamente es disyuntiva, en el sentido de que un modelo perfectamente puede acabar incluido en varias categorías del mismo nivel. La intención principal al elaborar esta categorización será la de estructurar el DFPS de tal manera que abarque todos los aportes de los expertos y que a la vez sea útil en su delimitación, lo cual es indispensable para la construcción de una FPS (véase páginas 60 y 61 de [16]). A la vez, esta categorización inicial inspirará la estructura del LexPAD para el LPAD correspondiente.

En las Figuras 2 y 3 se muestra una parte de la categorización que se hizo de los modelos provistos por los expertos. La Figura 3 muestra subcategorías de la categoría “Administración de Productos” de la Figura 2.

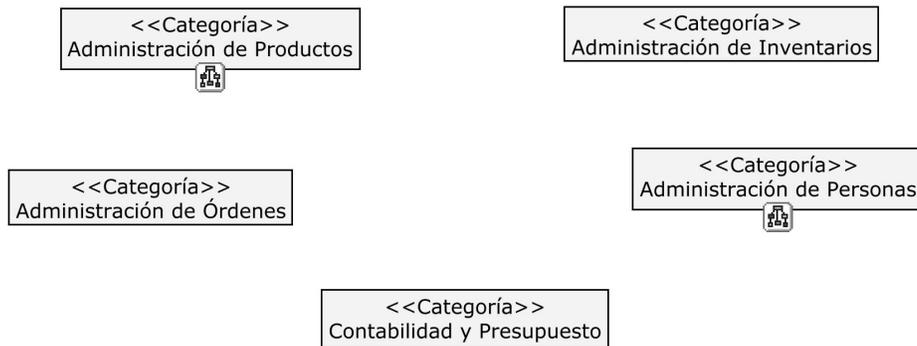


Figura 2. Primer nivel de la categorización de los modelos aportados por los expertos.

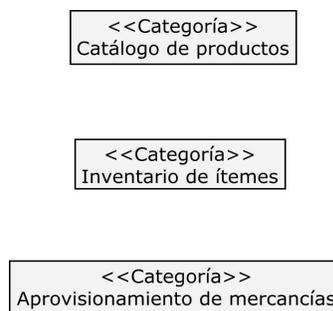


Figura 3. Ejemplo de sub-categorización de modelos aportados por los expertos.

C. Puesta en común de los modelos

Aún cuando dos modelos pertenezcan a la misma categoría e incluyan elementos similares, con toda seguridad se encontrarán diferencias en cuanto se los analiza con mayor detalle. Por ejemplo, entre los modelos de entidades y relaciones de [19] y [25] (de los expertos Hay y Silverston respectivamente) pertenecientes a las categorías de “Administración de productos” y “Administración de

B. Análisis de las relaciones entre los modelos.

Para identificar las coincidencias y diferencias entre los modelos provistos, se ha encontrado que es útil elaborar un diagrama que muestre cierto tipo de relaciones entre los modelos de tal manera que éstas se hagan explícitas. Particularmente ha sido de utilidad establecer relaciones de similitud, de complemento, de variación y de inclusión común de conceptos relevantes del dominio. La Figura 4 muestra un diagrama que interrelaciona modelos de datos provistos por [19] y [25]. En algunos casos ha sido útil mostrar los conceptos que se relacionan en un modelo. La figura presenta modelos de la categoría “Administración de Inventarios”. Se han incluido hipervínculos a anotaciones detalladas sobre las relaciones entre los modelos. Cada uno de los nodos como el denominado “Fig.4.1_48” representa un modelo específico provisto por algún experto. En este caso se ha incluido el número de la figura y la página del documento en que aparece. Idealmente, se debería asociar de alguna forma el nombre del experto que provee el modelo.

inventarios” se encontraron diferencias significativas en términos de los nombres con que se designan las entidades, los atributos asignados a las entidades (y sus nombres por supuesto), las relaciones entre las entidades y el nivel de generalidad de las entidades. Además Hay ha incluido entidades que Silverston no incluye y viceversa. Para avanzar hacia la meta de crear un LPAD que abarque todos los aportes de los distintos expertos, ha resultado útil construir modelos que

característica o atributo posible que permitirá clasificar una instancia de “PRODUCT TYPE – PT”.

Los atributos con el prefijo <<M>> son comunes en ambos expertos y por tanto se han considerado obligatorios, mientras

que los que tienen el prefijo <<O>> no son comunes y por tanto se han considerado opcionales. Los atributos con el prefijo <<K>> representan claves de acceso que se señalaron en modelos de ambos expertos, aunque no necesariamente hayan coincidido en su nombre.

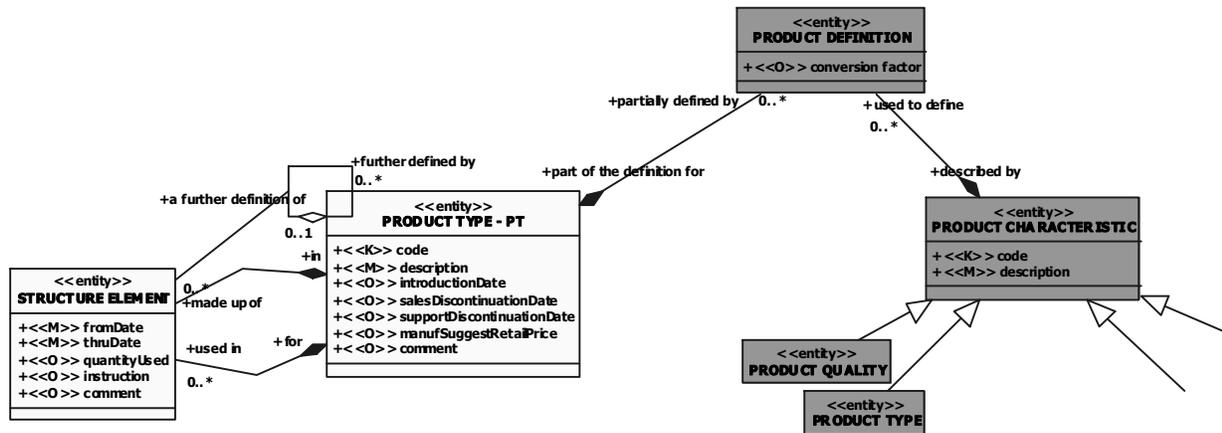


Figura 5. Trozo de un diagrama de puesta en común entre modelos de Hay y Silverston.

Finalmente un riesgo que conviene tener presente en esta tarea es la “sobre-generalización”, es decir, introducir algún concepto tan general que no aporte nada a la comprensión del dominio sólo con el fin de unificar dos o más conceptos similares incluidos por diversos expertos. Por ejemplo, Hay y Silverston difieren en cuanto a lo que consideran un producto como una cosa que una empresa u organización ofrece a sus clientes o usuarios. En Hay estos productos siempre son tangibles, es decir, no considera los servicios; mientras que Silverston considera ambas posibilidades y las diferencia. Un error garrafal en este contexto habría sido “sobre-generalizar” incluyendo en el modelo consolidado solamente una entidad para referirse tanto a un producto tangible como a uno intangible. Evidentemente en este caso lo mejor es incluir en el modelo consolidado entidades tanto para representar productos tangibles como intangibles y a la vez una entidad abstracta que represente un producto en general, que es de hecho la solución a que llega Silverston. Este último hecho facilitó la consolidación de ambas propuestas, sin embargo es imaginable una situación en que los modelos, por sí solos, no permitan una consolidación idónea. En tal caso sería indispensable una discusión de los expertos a fin de intentar llegar a un acuerdo.

D. Revisión de la categorización inicial

La categorización inicial sólo debe considerarse tentativa. Durante el análisis detallado para la puesta en común inevitablemente es muy posible que tal categorización sea cuestionada. Por ejemplo, en el caso de los modelos de Hay y Silverston se logró determinar, como ya se ha mencionado, que las categorías “Administración de Productos” y “Administración de Inventarios” están mucho más fuertemente

relacionadas para Hay que para Silverston. Esto se concreta en términos de la perspectiva que mantiene Hay de que los productos son ejemplos de activos. Al ser estas entidades centrales en cada una de las categorías correspondientes, se planteó la duda sobre si deberían mantenerse como categorías independientes o más bien fusionarse. La respuesta no se obtuvo con el análisis resumido hasta aquí. Más bien, la clave llegó después al empezar a elaborar sobre los distintos usuarios (o actores como se les conoce en el UML) de los datos de los modelos analizados y sobre sus necesidades específicas. En la siguiente sección se describen tareas relacionadas con los actores y sus necesidades específicas.

IV. SÍNTESIS DE LOS APORTES DE LOS EXPERTOS DE DOMINIO

En esta sección se describen las lecciones aprendidas en relación con cómo estructurar el espacio de patrones al interior de cada una de las categorías de la estructura “multinivel” previamente elaborada. El objetivo de estas heurísticas es perfilar cada uno de los PAD, los tipos de relaciones relevantes entre estos y sus conexiones concretas. Como se verá, esto termina también de resolver cuestiones pendientes sobre la categorización inicial.

La escogencia de una entre las distintas opciones para la estructuración del espacio de patrones es el problema principal con que se debe lidiar en esta segunda etapa. Este problema incluye el de la categorización de los patrones para el cual se tiene una solución? más sólo tentativa? basada en la categorización de los modelos. Puesto en términos formales, de lo que se trata es de elaborar un “hipergrafo” que en su estructura difiere de un grafo en el hecho de que un nodo puede

representar a su vez a otro grafo que sería su contenido. Por tanto, se tiene en un “hipergrafo” *nodos simples* y *nodos complejos*. De acuerdo con la caracterización de un LPAD en [4], los PAD son los *nodos simples* (sus variantes y combinaciones también), mientras que las categorías de patrones son los *nodos complejos*. El “hipergrafo” a elaborar debe encapsular el conocimiento del dominio aportado por los expertos en términos de PADs, categorías de PADs y sus interrelaciones. No es difícil imaginar que, para encapsular un mismo cuerpo de conocimientos basado en modelos aportados por expertos en un dominio, pueden elaborarse muchos “hipergrafos” posibles y de aquí deriva la dificultad en esta etapa. Para resolver este problema se ha encontrado útil identificar, para cada categoría plausible de PADs, los actores de los productos de la FPS objetivo, identificar los requerimientos funcionales que estos actores demandan y por ende las características funcionales de los productos de la FPS, identificar un PAD central, otros periféricos y sus interrelaciones, y finalmente elaborar cada uno de los PADs. Puesto en otros términos, la estrategia ha sido estructurar el espacio de patrones a partir de una estructuración previa del espacio de características funcionales⁶ que es considerado crucial en el desarrollo de FPS según [16].

A. Identificación de actores

La mayoría de los patrones de análisis publicados son *patrones de modelos de datos* o *patrones de modelos conceptuales de objetos*. Pocos incluyen diagramas de actividades, estados y transiciones o de eventos. Ninguno de los analizados para este trabajo incluye modelos de casos de uso⁷. Para poder construir productos de una FPS no basta con indicar cuáles son las entidades relevantes, sus atributos e interrelaciones. Es imprescindible establecer con precisión cómo van a ser usados todos estos, para qué son necesarios y por tanto quiénes los necesitan. Los mismos atributos de un conjunto de entidades y sus interrelaciones pueden ser usados por unas personas en procesos puramente operativos y por otras en procesos de toma de decisiones.

Se ha encontrado que los lineamientos generales para identificar actores en los métodos de análisis orientado a objetos son igualmente útiles en este contexto. Por tanto, el análisis de las necesidades de distintos tipos de personas en el contexto de los procesos organizacionales o procesos del negocio ha sido suficiente para resolver esta tarea.

B. Elaboración del espacio de características funcionales

El espacio de características funcionales según [16] es un grafo cuyos nodos representan características funcionales provistas por una FPS y cuyos arcos representan relaciones

entre éstas. Las características pueden ser comunes a todos los productos de una FPS u opcionales. Dado un grupo de características opcionales puede definirse una como opción estándar y las demás como alternativas. A diferencia del método expuesto en [16] los LPAD propuestos en [4] organizan el espacio de características por medio del “hipergrafo” de categorías de patrones y patrones (incluyendo variantes y combinaciones de patrones). Cada nodo del “hipergrafo” encapsula un subconjunto del grafo de características. La intención con esta organización particionada o encapsulada es por un lado facilitar su comprensión y por otro facilitar su evolución futura, en forma análoga a como un sistema de software adecuadamente modularizado es más fácil de comprender y modificar.

Asignar características funcionales (y por ende actores) a todas y cada una de las categorías hipotéticas originalmente usadas para organizar los aportes de los expertos, constituye una prueba de existencia para las categorías de patrones y para los patrones que en definitiva conformarán el LPAD en proceso de creación. Si no fuera posible asignar un subconjunto del grafo de características funcionales (y por ende actores) a una categoría de modelos sería porque los modelos propuestos estarían incluidos, de alguna manera, en otras categorías. Por ejemplo, si los modelos fueran de entidades y relaciones, quizás las de los modelos de la categoría a la que no se le pueden adscribir características funcionales propias, estarían incluidas ya en otras categorías y por tanto sería innecesaria su inclusión en el LPAD en proceso de creación. Por otro lado, una vez asignado un subconjunto del espacio de características a una categoría de patrones, será necesario asignarlas a los patrones de la categoría, con lo cual se inicia la elaboración de los PAD. Cabe destacar que aún cuando la intersección de los conjuntos de características de todas las categorías de patrones puede no ser vacía, sin embargo todas y cada una de las categorías de patrones debe incluir un subconjunto de características que le es propio, es decir, que no comparte con ninguna otra categoría de patrones. Esta aserción deberá verificarse para toda categoría que sólo incluya patrones y no otras categorías.

Ahora bien, ¿cómo se obtienen las características funcionales? En el capítulo 5 de [16] se propone un método para su derivación, sin embargo en el contexto de este trabajo, se trata de describir las lecciones aprendidas sobre cómo crear un LPAD y representarlo mediante un LexPAD, *dados* los aportes de un conjunto de expertos en un DFPS. En esta circunstancia ha sido útil por un lado estudiar con detalle los comentarios que acompañan a los modelos provistos, y por otro imaginar diferentes usos posibles o diferentes necesidades que los modelos consolidados pueden satisfacer. La idea es complementar los modelos consolidados con características funcionales plausibles. Estas características luego serán usadas para estructurar el espacio de patrones.

6. Se refiere al lector al capítulo cinco de [14].

7. Se refiere al lector a [21].

C. Identificación de un PAD central, PADs periféricos e interrelaciones por cada categoría

Para asignar las características funcionales de una categoría de PADs a sus miembros y así estructurar el espacio de PADs propio de cada categoría ha resultado útil perfilar primero un PAD central a partir de las características con el atributo de “común”⁸ de entre el subconjunto asignado a cada categoría y luego identificar PADs periféricos o complementarios. En el PAD central también se podrán incluir características estándar pertenecientes a algunos grupos de características alternativas, así como aquéllas opcionales que quedarían aisladas bajo otras organizaciones plausibles del espacio de PADs.

La Figura 6 muestra el espacio de características de la categoría de patrones “Catálogo de Productos” de la categoría “Administración de Productos”, mientras la Figura 7 muestra

el espacio de características del PAD central de esta categoría de patrones. Las demás características han sido asignadas a los demás PAD que aparecen en el mapa de patrones de la Figura 8.

Como se ve en la Figura 8, el PAD central de una categoría típicamente es extendido por los periféricos, lo cual significa que al unir los modelos de los PADs periféricos a los del PAD central se obtiene un sistema de software cuyas características funcionales abarca la unión de los conjuntos de características de todos los PADs involucrados. La relación “sustituye clasificación lineal en” significa que la típica clasificación lineal de los tipos de productos en el PAD central se puede sustituir por una clasificación jerárquica o (o excluyente) una clasificación reticular⁹. Todos y cada uno de los PAD de cada CPAD deberá incluir las características funcionales que validan su existencia antes de proceder con su elaboración detallada.

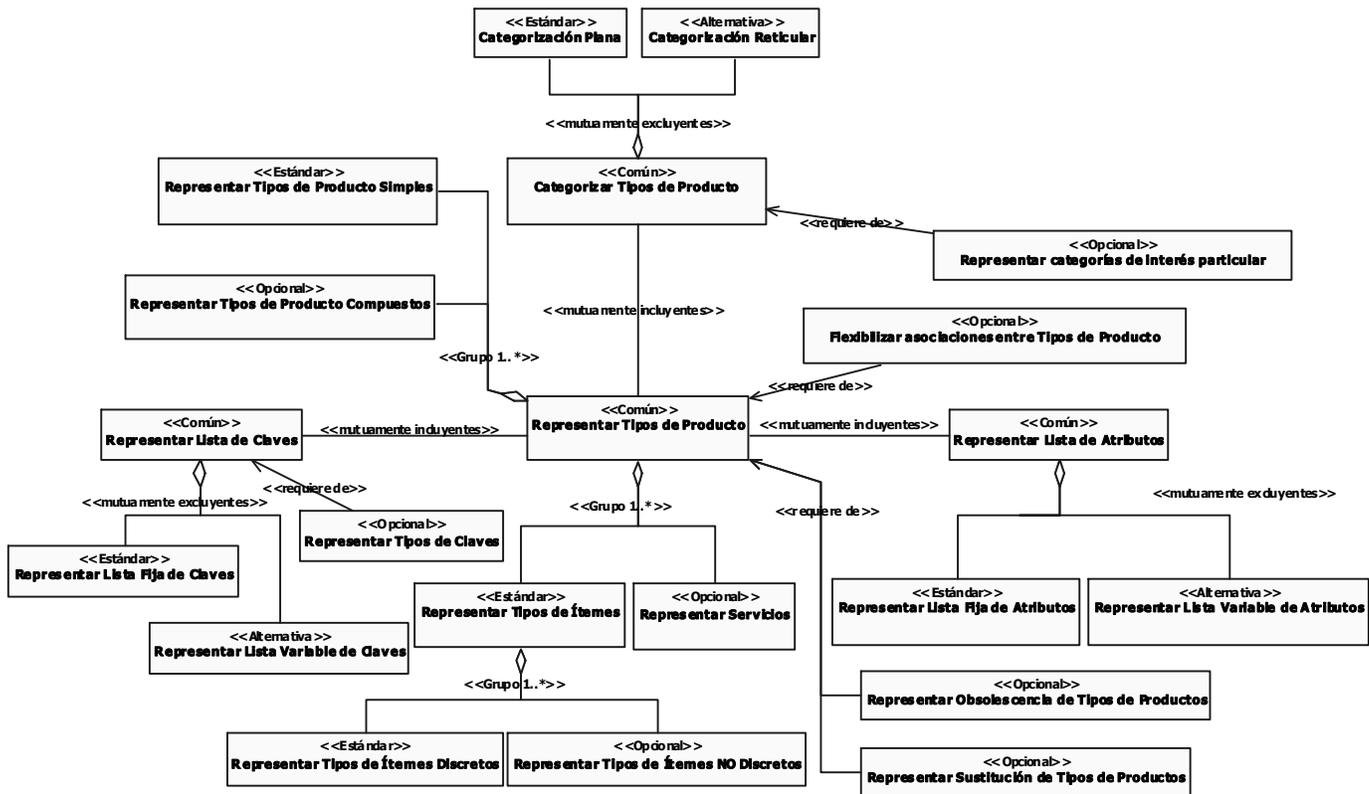


Figura 6. Características de la categoría de patrones “Catálogo de Productos”.

8. Es decir, provista por todos los productos de la FPS.
 9. Se entiende una clasificación reticular como aquella en que una clase de tipos de productos o un tipo de producto puede aparecer en varias categorías a la vez, es decir, no es una categorización estrictamente jerárquica.

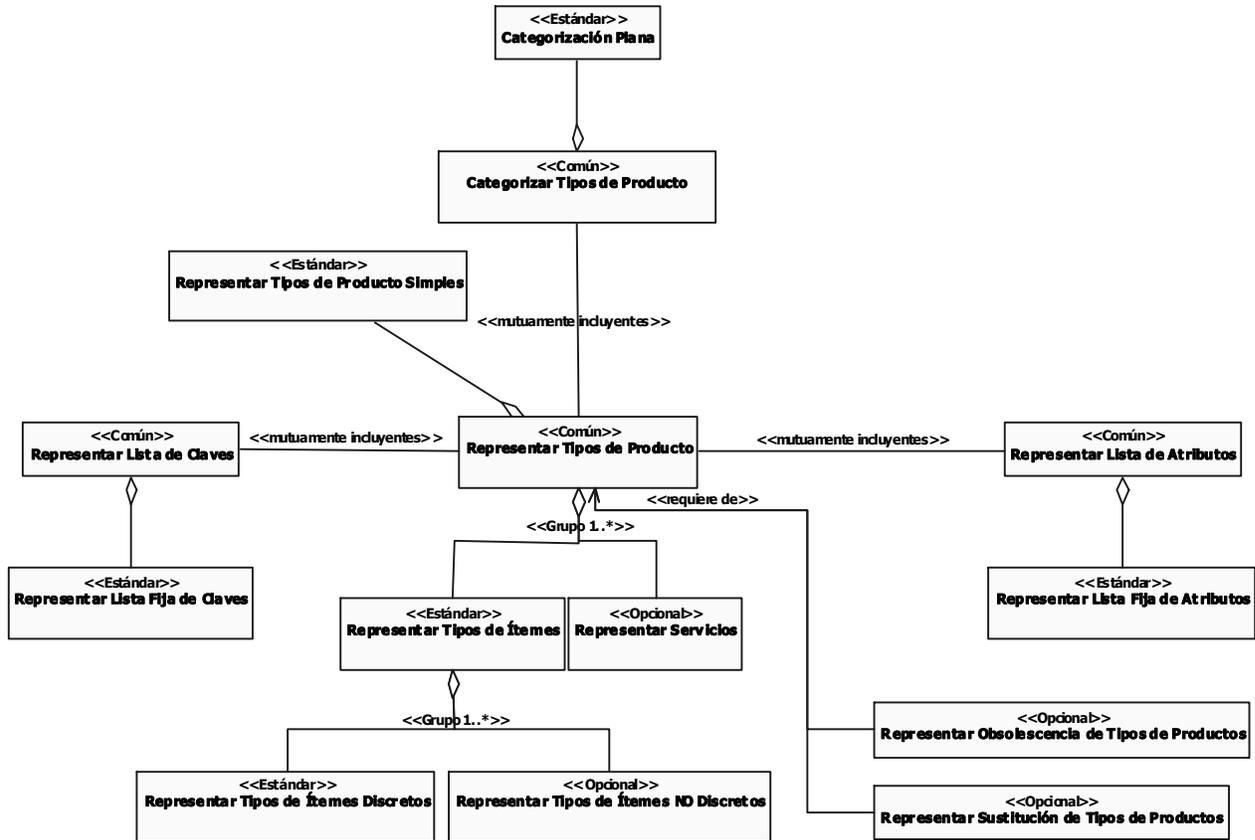


Figura 7. Características o “fuerzas” del PAD “Catálogo de productos simples”.

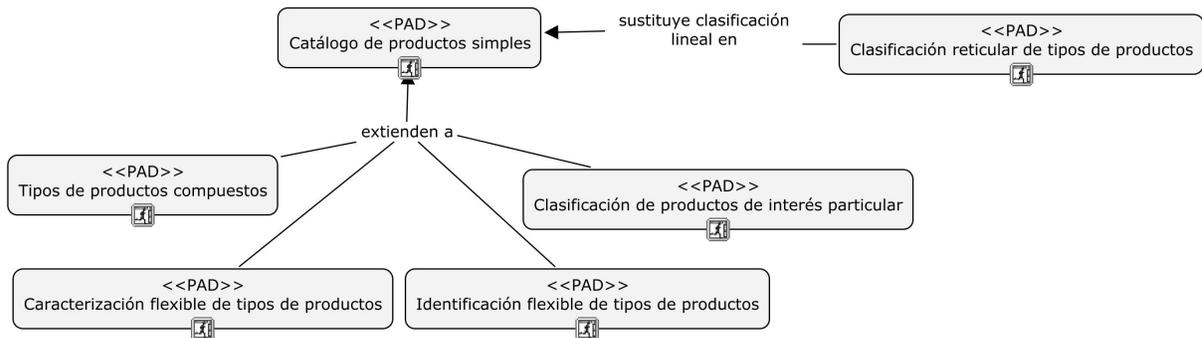


Figura 8. Mapa de PADs de la categoría “Catálogo de Productos”.

D. Elaboración de los PADs por cada categoría

Esta última tarea consiste en la especificación de los PAD siguiendo una plantilla adoptada para tales efectos. En [14] se afirma que un patrón debe incluir al menos cuatro aspectos esenciales: nombre, descripción del problema, descripción del núcleo de la solución y análisis de consecuencias. La plantilla adoptada durante la investigación que ha permitido derivar las lecciones que se presentan en este trabajo incluye estos cuatro aspectos esenciales.

En cuanto al nombre, el del PAD central debería referirse al escenario planteado por su grupo de características, mientras que los de los PAD periféricos deberían referirse a la característica funcional más relevante que incluyen. Para la descripción del problema se ha adoptado la costumbre entre los escritores de patrones de describirlo en términos de las “fuerzas” o condiciones que restringen la solución o demandan solución. En el caso específico de los PADs se ha considerado que las fuerzas son precisamente las características funcionales

consideradas en el PAD, las cuales serán siempre un subconjunto de las características funcionales de la categoría a que pertenece el PAD. Para describir el núcleo de la solución se ha incluido en la especificación de cada PAD un modelo de casos de uso con actores y objetos de frontera, un modelo conceptual de objeto o modelo de dominio, y un modelo de datos para el diseño de una base de datos relacional. Las consecuencias han sido identificadas y comentadas brevemente para cada PAD. Adicionalmente se ha considerado conveniente incluir una sección que describe reglas de aplicabilidad del patrón o condiciones bajo las cuales el PAD se considera aplicable y otra sección en que se orienta al analista sobre cómo aplicar el PAD a un contexto específico.

Una característica primordial de los lenguajes de patrones según la caracterización de Alexander es que los patrones se deben poder agregar uno a otro para conformar poco a poco una solución completa a un problema de diseño dado. Para lograr esto se han definido con precisión las relaciones entre los PADs así como los puntos en que sus modelos se pueden entrelazar. Se ha encontrado que dados dos elementos que conforman un punto de enlace posible entre dos PADs diferentes, conviene indicar una de dos tipos de relaciones a saber: “extiende a” o “sustituye a”. En el caso de que los elementos sean casos de uso, la relación “extiende a” se debe entender como la relación representada por el estereotipo estándar <<extends>>. En el caso de que sean clases, se debe entender como la relación de herencia o derivación. La relación “sustituye a” indica que al entrelazar los PADs involucrados, el elemento del PAD base se debe sustituir por el correspondiente elemento del PAD que extiende o amplía el modelo.

V. PRIMER BOSQUEJO DE UN MÉTODO

Para sintetizar las secciones tercera y cuarta, se puede establecer la siguiente aproximación inicial a un futuro método para la creación de LPAD y su representación por medio de LexPAD a partir de los aportes de un grupo de expertos en el contexto de un DFPS meta:

- a. Análisis de los artefactos aportados:
 - Categorización de los artefactos.
 - Análisis de las relaciones entre los artefactos.
 - Puesta en común.
 - Crítica de la categorización inicial.
- b. Síntesis a través de la estructuración del espacio de PADs por categoría:
 - Identificación de actores.
 - Elaboración del espacio de características funcionales (consolidación de la categorización del LPAD).
 - Identificación de un PAD central, PADs periféricos e interrelaciones.
 - Elaboración de los PADs.

Cabe enfatizar que la síntesis a través de la estructuración del espacio de PADs por categoría es una fase que se repetirá tantas veces como sea necesario. Dependiendo de la complejidad del dominio y de la cantidad de modelos provistos, es posible que sea necesaria la repetición de ambas fases para cada categoría de PADs.

VI. CONCLUSIONES

Se han descrito ocho lecciones aprendidas sobre cómo crear un LPAD y representarlo mediante un LexPAD a partir de una experiencia concreta sobre el DFPS asociado a los sistemas de información empresarial de apoyo a los procesos administrativos. El cuerpo de conocimientos base para esta experiencia han sido los modelos aportados por distintos expertos en el DFPS meta. La mayoría de estos modelos son de entidades y relaciones o modelos conceptuales de objetos. En la quinta sección se han resumido las lecciones aprendidas en una primera aproximación a un futuro método. Aunque no ha habido espacio suficiente para describir el detalle de la construcción de un LexPAD tal como es concebido en [4], sí se han explicado los aspectos fundamentales.

Aunque el objetivo de este trabajo no incluye bosquejar un futuro método con la participación intensiva de expertos en el dominio meta, sin embargo su interacción se ha señalado como importante en lo que es la “puesta en común” de los modelos provistos, la “identificación de actores” y la “elaboración del espacio de características funcionales”.

Siendo el fin último de la investigación que motiva el presente resultado preliminar especificar un método para la creación de LPAD y su representación mediante LexPAD, aparte de una necesaria validación conviene:

- a. Contrastar estas pautas metodológicas preliminares con métodos afines que hayan sido bien documentados, como métodos de análisis de dominio (por ejemplo los discutidos en [16], [23], [20] o [22]) o métodos de adquisición de conocimiento (véase por ejemplo [17]).
- b. Explorar ventajas y desventajas entre la participación directa de los expertos del dominio y la puesta en común de sus aportes a través de modelos, así como su combinación idónea.
- c. Explorar con el uso de distintas notaciones para que los expertos plasmen sus modelos base en el proceso descrito.
- d. Explorar técnicas orientadas a comparar ontologías, considerando que los modelos de entidades y relaciones y los modelos conceptuales de objetos son de hecho similares a ontologías.
- e. Identificar y valorar herramientas para facilitar el proceso en cuestión.

El análisis de un DFPS a partir de los aportes de expertos y orientado a la creación de un LPAD, así como su representación

mediante un LexPAD es una propuesta alternativa. En contraste con las existentes, se enfatiza la construcción y sistematización de conocimiento del DFPS bajo el supuesto de que es el principal capital de una empresa cuyo posicionamiento se basa en la oferta de productos de una FPS. Este trabajo es un pequeño paso más dirigido a consolidar un método de ingeniería de FPS que aspira a enfatizar la construcción y sistematización de conocimiento con base en el concepto de lenguaje de patrones propuesto en [1].

REFERENCIAS

- [1] Alexander, C., 1979. *The Timeless Way of Building*. Nueva York: Oxford University Press.
- [2] Alur, D., Crupi, J. y Malks, D., 2001. *Core J2EE Patterns (Best Practices and Design Strategies)*. New Jersey: Sun Microsystems Press.
- [3] Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P. y Stal, M., 1996. *Pattern-Oriented Software Architecture (A System of Patterns)*. West Sussex: John Wiley & Sons. Conocido como POSA 1.
- [4] Calderón, A., 2007. Representación de lenguajes de patrones de análisis de dominio. En: *Ingeniería – Revista de la Universidad de Costa Rica*. Vol. 16, pp. 85-101.
- [5] Calderón, A., 2003. Un patrón para lexicones de patrones. En: *The Third Latin American Conference on Pattern Languages of Programming SugarLoaf PLoP, 2003*. Pernambuco, Brasil.
- [6] Clements, P. y Northrop, L., 2002. *Software Product Lines: Practices and Patterns*. Boston: Addison-Wesley.
- [7] Coad, P. 1992. Object-Oriented Patterns, *Communications of the ACM*. En Vol. 35(9), pp. 152-159.
- [8] Coad, P., 1997. *Object Models (Strategies, Patterns and applications)*. New Jersey: Yourdon Press.
- [9] Eriksson, H.-E. y Penker, M., 2000. *Business Modeling with UML: Business Patterns at Work*. New York: John Wiley & Sons.
- [10] Fernández, E., 2000. Stock Manager: An Analysis Pattern for Inventories. En: *Proceedings PLoP 2000*. Monticello, Indiana.
- [11] Fernández, E. & Yuan, X., 1999. An Analysis Pattern for Reservation and Use of Reusable Entities. En: *Proceedings PLoP 1999*. Monticello, Illinois.
- [12] Fernandez, E., Yuan, X. y Brey, S., 2000. Analysis Patterns for the Order and Shipment of a Product. *Proceedings*. En: *Proceedings PLoP 2000*. Monticello, Illinois.
- [13] Fowler, M., 1997. *Analysis Patterns: Reusable Object Models*. Boston: Addison-Wesley.
- [14] Gamma, E., Helm R., Johnson, R. y Vlissides J., 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley Publishing Company. Conocido como GoF.
- [15] Gaertner, N. & Thirion, B., 1999. Grafcet: an Analysis Pattern for Event Driven Real-time Systems. En: *Proceedings PLoP 1999*. Monticello, Illinois.
- [16] Gomaa, H., 2004. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Boston: Addison-Wesley.
- [17] Gonzalez, A. & Dankel, D. (1993). *The Engineering of Knowledge-Based Systems*. New Jersey: Prentice-Hall.
- [18] Grand, M., 1999. Transaction Patterns: A Collection of Four Transaction Related Patterns. En: *Proceedings PLoP 1999*. Monticello, Illinois.
- [19] Hay, D., 1996. *Data Model Patterns (Conventions of Thought)*. New York: Dorset House Pub.
- [20] Jacobson, I., Griss, M. & Jonsson, P., 1997. *Software Reuse: Architecture, Process and Organization for Business Success*. Boston: Addison-Wesley.
- [21] Jacobson, I., Booch, G. & Rumbaugh, J., 2000. *El Proceso Unificado de desarrollo de software*. España: Addison-Wesley.
- [22] Prieto-Diaz, R., 1987. Domain analysis for Reusability. En: *Compsac'87: Eleventh International Computer Software and Applications Conference Proceedings*, Los Alamitos, California. pp. 23-29.
- [23] Roslyn, N., Kogut, P. & Jackelen, G., 1994. Component Provider's and Tool Developer's Handbook CARDS. Reporte técnico desarrollado para la Corporación Unisys.
- [24] Schmidt, D., Stal, M., Rohnert, H., Buschmann, F., 2002. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. West Sussex: John Wiley & Sons. Conocido como POSA 2.
- [25] Silverston, L., Inmon, W.H. y Graziano, K., 1996. *The Data Model Resource Book (A Library of Logical*
- [26] Vaccare Braga, R., Germano, F. y Masiero, P., 1999. A Pattern Language for Business Resource Management. En: *Proceedings PLoP*. Monticello, Illinois.

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas



Escuela de Ingeniería de Sistemas

Misión

La misión de la Escuela de Ingeniería de Sistemas es fomentar y apoyar la generación o la apropiación de conocimiento, la innovación y el desarrollo tecnológico en el área de ingeniería de sistemas e informática sobre una base científica, tecnológica, ética y humanística.



Visión

La formación integral de profesionales desde el punto de vista científico, tecnológico y social que les permita adoptar, aplicar e innovar conocimiento en el campo de los sistemas e informática en sus diferentes aspectos, aportando con su organización, estructuración, gestión, planeación, modelamiento, desarrollo, procesamiento, validación, transferencia y comunicación; para lograr un desempeño profesional, investigativo y académico que contribuya al desarrollo social, económico, científico y tecnológico del país.



Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>

