Técnicas de análisis de dominio: organización del conocimiento para la construcción de sistemas software

Ricardo Eito Brun

Universidad Carlos III de Madrid, Getafe (Madrid). reito@bib.uc3m.es.

Resumen

La comunidad dedicada al diseño de sistemas informáticos ha propuesto una serie de técnicas conocidas como "análisis de dominio", que tienen como finalidad la captura y adquisición del conocimiento que deben almacenar y procesar las aplicaciones informáticas orientados a una misma función o uso. El análisis de dominio ofrece herramientas para capturar la información crítica sobre las entidades, datos y procesos característicos de un área de actividad, y facilitar la especificación, el diseño y la construcción de los sistemas que soporten dichas actividades. Como disciplina, el análisis de dominio pretende la reutilización intensiva del conocimiento y capitalizar la experiencia adquirida en el diseño y construcción de sistemas informáticos de un mismo tipo. En la comunicación se analiza la evolución de las metodologías de análisis de dominio, los principales hitos en la historia de su desarrollo – tanto desde un punto de vista conceptual como experimental– sus principios teóricos y la sinergia con las técnicas de procesamiento automático de documentos y la minería textual. Cobra especial relevancia la evolución de las técnicas de análisis de dominio hacia la formulación de ontologías y sistemas de representación del conocimiento complejos. Esta evolución constituye un excelente ejemplo de permeabilidad entre la ingeniería del software, la recuperación textual y la organización del conocimiento.

Palabras clave: Análisis de dominio, Ontologías, Representación del conocimiento, Diseño de sistemas informáticos, Minería textual.

Abstract

The community of users evolved in the design of Information System has proponed a set of techniques usually referred to as "domain analysis". Their purpose is to help designers in the acquisition of the knowledge that Information Systems are expected to store and process. Domain analysis tries to synthesize the commonalities of information systems that serve the same purpose. Domain analysis offers tools and techniques to gather critical information about the entities, data and process of a specific business activity. These tools and techniques

will also help designers complete specifications and designs during the development of new Information Systems. Domain analysis is closely related to information and knowledge representation and reuse, as this is one of its key objectives: capitalize the experience and know-how acquired in the design of complex information systems. This contribution describes the most relevant methodologies applied for domain analysis, the milestones in the development of these techniques, its theoretical basis and the synergies with automatic text processing and text mining techniques. It is especially relevant the evolution of domain analysis toward ontology analysis and design. This demonstrates the relationship between software engineering, text processing and knowledge organization.

Keywords: Domain Analysis, Ontology, Knowledge representation, Information systems design, Text mining.

1 Introducción

El diseño y construcción de sistemas informáticos se basa en la aplicación de una serie de técnicas y metodologías que aseguran la evolución de unas especificaciones formuladas en lenguaje natural hasta su representación en código escrito en un lenguaje de programación determinado. Para facilitar esta evolución, se han propuesto distintas metodologías y técnicas de diagramación que representan la estructura y el comportamiento de los sistemas de información. Se pretende así evitar cualquier malinterpretación y ambigüedad en las especificaciones y en su evolución posterior. Estos formalismos no sólo facilitan la representación de los sistemas de información y los hace más comprensibles para las distintas personas que participan en su desarrollo, también hacen viable su posterior mantenimiento y reutilización en futuros proyectos.

El valor de las metodologías depende de en qué medida se disponga de una clara definición del problema que se pretende solventar, y del contexto en el que operará el sistema de información. Esto requiere una adquisición precisa del conocimiento y de la información que gestionará el sistema de información, y de las reglas que regirán su funcionamiento.

La comunidad dedicada al diseño de sistemas de información ha propuesto una serie de técnicas conocidas como “análisis de dominio”, que tienen como finalidad la captura y adquisición de la información y del conocimiento que deben gestionar los sistemas informáticos de un mismo tipo o familia. Con esto nos referimos a los sistemas orientados a una misma comunidad de usuarios, función, actividad o uso. El análisis de dominio pretende capturar la información crítica sobre las entidades, los datos y los procesos que caracterizan un área de negocio en particular, con el objetivo de facilitar la especificación, el diseño y la construcción de aplicaciones que soporten dichos procesos. A partir de la información procedente de estos documentos - y de los sistemas informáticos existentes – se podrá elaborar una conceptualización del área de actividad y de los procesos para los que se idearon los sistemas informáticos tomados como base del análisis.

El análisis de dominio está estrechamente vinculado a la reutilización de software, que busca la construcción de nuevos sistemas informáticos a partir de componentes, especificaciones o diseños creados en el pasado. La relación entre análisis de dominio y reutilización es evidente: disponer de una representación sistematizada del conocimiento característico de un área, hará

más sencilla la creación de sistemas, módulos y componentes que puedan aplicarse en múltiples escenarios y en la resolución de distintos problemas.

2 Definición y alcance

El término "*análisis de dominio*" tiene su origen en los trabajos de Neighbords a comienzos de la década de los 80, quien lo definió como "*la actividad que consiste en identificar los objetos y operaciones de un tipo de sistemas similares, dentro de un dominio de problema particular*". (Neighbords, 1981).

Otra definición clásica es la propuesta por *Prieto-Díaz* (1990), como "el proceso por el cual la información utilizada para el desarrollo de sistemas software se identifica, captura y organiza con el fin de hacerla reutilizable en la creación de nuevos sistemas".

Estas dos definiciones nos permiten identificar las principales características del análisis de dominio:

a) Se trata de una disciplina orientada a la captura y gestión de información y conocimiento.
b) Pretende abstraer los aspectos que describen y caracterizan un área de actividad o proceso, es decir, un "dominio" específico.
c) Parte de un conjunto de sistemas software existentes.
d) Tiene como objetivo identificar información que pueda reutilizarse en el diseño de futuros sistemas.

En relación al tercer punto, debemos recordar que al hablar de software se hace referencia no sólo a los programas informáticos propiamente dichos, sino también a toda la documentación generada durante su especificación y diseño, y a la que se usará para la operación, mantenimiento y uso del sistema. Señalaremos también que - en la práctica, y en las distintas metodologías que se han formulado para el análisis de dominio - el proceso de captura de información y conocimiento no se limita exclusivamente a los datos disponibles en documentos y aplicaciones informáticas; también se propone utilizar entrevistas, juicio experto, etc.

La principal característica del análisis de dominio es que su alcance no se restringe a la captura de la información necesaria para la resolución de un problema en particular, identificado para un grupo de usuarios determinados (como sucede con las técnicas y metodologías aplicadas en el diseño de aplicaciones informáticas). El análisis de dominio tiene una pretensión mayor: capturar y sistematizar el conocimiento en el que se basarían todas las aplicaciones informáticas diseñadas para satisfacer un escenario general, para cualquier comunidad de usuarios. Por ejemplo, si se diseña un sistema de información automatizado para gestionar la red de vehículos de una empresa, estaremos haciendo un análisis de una aplicación informática singular; frente a esto, el análisis de dominio buscaría respuesta a la pregunta: ¿qué información y conocimiento es aplicable a la gestión de redes de vehículos de cualquier empresa? ¿Cuál es la mejor forma de representar este conocimiento

para poder utilizarlo posteriormente en la creación de sistemas informáticos que satisfagan las necesidades de cualquier organización que opere en este ámbito de actividad?[1]

En los últimos años, el análisis de dominio ha sido foco de atención creciente debido a la importancia de la llamada PLSE (Product-Line Software Engineering), término que podríamos traducir como "ingeniería de software para líneas de producto", y que tiene como objetivo el desarrollo de aplicaciones informáticas a partir de una serie de características y activos clave para el área de actividad a la que va dirigida la aplicación.

La PLSE tiene grandes similitudes con la ingeniería y el análisis de dominio; de hecho, en los dos casos se trata de extraer las características comunes de un mismo tipo de sistemas. La principal diferencia entre una y otro está en el hecho de que la PLSE tiene como input principal el análisis de mercado, factor éste que no está presente en las técnicas de análisis de dominio tradicionales (Kwanwoo, 2002). No obstante, en la literatura sobre este tema generada en la década de los ochenta ya estaba descrita la importancia de utilizar análisis de mercado como primer paso para el análisis de dominio, en lo que se ha dado en llamar análisis de dominio orientado a producto, y que se aplicó en proyectos como STARS (Software Technology for Adaptable and Reliable Systems) y por IBM.

3 Metodología

La metodología del análisis de dominio puede resumirse de la siguiente forma: partiendo de unas fuentes de información sobre el dominio (entendiendo por tal un área de actividad determinada), se completará un análisis usando unas técnicas que darán como resultado un *modelo del dominio.*

Las fuentes de información sobre el dominio - tal y como hemos señalado - consistirán en todo el conocimiento disponible de las aplicaciones existentes para el área de actividad: documentación de diseño, de operaciones, de usuario, estándares y normas aplicables, modelos, código fuente, y las aplicaciones propiamente dichas. También se incluirá en este grupo los usuarios y las personas con conocimiento experto en el área (los inputs del análisis de dominio no se limitan exclusivamente a fuentes documentales).

Partiendo del esquema anterior, en el que coinciden distintos autores, podemos encontrar algunas diferencias respecto a cuales son las actividades que constituyen el detalle del tratamiento a realizar, y la forma en que se representará el modelo de dominio. Así, Prieto-Díaz propuso un enfoque basado en el uso de sistemas de clasificación facetadas para la conceptualización del dominio.

McCain (1985) identificó tres actividades clave: a) identificación de las entidades reutilizables, b) abstracción o generalización para obtener características comunes y c) clasificación y catalogación para facilitar su recuperación y uso. La tercera actividad resulta significativa ya que, según este autor, los métodos propuestos para clasificar y facilitar la

[1] Estas dos preguntas nos ayudan a clarificar la diferencia existente entre *análisis de dominio* e *ingeniería de dominio*. Normalmente el primero sería una fase o sub-proceso dentro de la ingeniería de dominio, que tiene como objetivo sólo el análisis (entendiendo por tal la captura, abstracción y especificación del conocimiento característico de un área de actividad). La ingeniería también tiene como objetivo el diseño y desarrollo de un sistema que implemente y se base en dicho conocimiento.

recuperación de los ítems formarían parte del modelo de dominio, aspecto éste con el que no suelen coincidir el resto de autores.

Los resultados de la actividad de análisis, en los que se plasmará el modelo de dominio obtenido como resultado, incluirán un glosario o terminología con los conceptos significativos, las normas y estándares aplicables, una definición del dominio en la que se describe su alcance y ciertas reglas que determinan qué casos particulares estarían incluidos y cuales excluidos del dominio en cuestión, taxonomías con las relaciones entre los distintos conceptos, y los llamados "modelos de características" (features, en inglés).

Estos últimos resultan especialmente relevantes, ya que se usan para describir y enumerar las características o propiedades que tiene un sistema. En un análisis de dominio, éstas serán las características que serían aplicables y comunes a todas las aplicaciones informáticas y sistemas de información del mismo tipo, y también las que constituyan excepciones disponibles en un sistema particular, pero que no se consideran imprescindibles para la adscripción de un sistema al dominio en cuestión. Estas características se caracterizan porque son percibidas por un usuario final del sistema. Suelen representarse mediante diagramas jerárquicos en los que el sistema - como un conjunto - se subdivide en distintas características, éstas en otras con mayor detalle, etc. Los diagramas también representan las relaciones existentes entre las características y su carácter opcional u obligatorio. Un ejemplo de relación entre características es la relación "alternativa", que uniría dos características para indicar que un sistema debe tener exclusivamente una de las características unidas por esta relación; otra relación entre características ampliamente utilizada es la llamada Or-features, con la que se indica que un sistema puede tener una o más de las características unidas mediante esta relación. De esta forma se especifican - como parte del diagrama - las combinaciones de características que son válidas. A partir de esta información se podrán derivar las llamadas "instancias del sistema", es decir, las configuraciones válidas para que un producto forme parte del dominio, o dicho de otra forma, los distintos conjuntos de características que el sistema debe implementar y satisfacer. Estas relaciones entre características representan sus relaciones conceptuales y estructurales. Otras relaciones utilizadas en los diagramas de características son las relaciones "todo-parte", la generalización o especialización y la relación "implementada-por", con la que se pueden registrar aquellos casos en los que la implementación de una características requiere que previamente la implementación previa de otra característica diferente (podríamos hablar aquí de dependencias funcionales frente a las dependencias estructurales que se representan mediante las relaciones todo-parte y las de generalización/especialización).

FODA (Feature-Oriented Design Analysis) es sin duda una de las metodologías más populares para el análisis de dominios (si bien sería más correcto hablar de “análisis de características”). En esta especificación se define una característica como una "propiedad prominente y distinguible por el usuario de un sistema”. Las características son el medio idóneo para diferenciar los aspectos comunes y las diferencias que presentan los distintos sistemas de información sobre los que se ha realizado el análisis de dominio. Los aspectos comunes se plasmarán en características obligatorias, y las diferencias o variaciones en características opcionales, o relacionadas mediante relaciones Or-features o mediante "alternativas".

4 Principales hitos en la formulación del concepto

En relación a los principales hitos y experiencias relacionadas con el análisis de dominio, el primer proyecto que debemos citar es CAMP (*Common Ada Missile Packages Project*) puesto en marcha en 1987 y considerado una implementación práctica de las teorías de Neighbors (a quién se atribuye el primer uso del término "*análisis de dominio*"). En el proyecto CAMP se estudiaron once sistemas de misiles para identificar aspectos comunes a todos ellos, a partir de los que se generaron plantillas comunes en lenguaje *Ada* para facilitar su implementación. Se suele señalar - en relación a este proyecto - cierta falta de detalle en lo relativo a cómo completar el análisis de dominio (Prieto-Díaz, 1999). Neighbors también desarrolló una teoría conocida como DRACO.

En la década de los 80 el concepto de análisis de dominio estaba estrechamente vinculado a la reutilización de software. Como ejemplo, encontramos autores que definen análisis de dominio como “un proceso continuo para crear y mantener la infraestructura necesaria para la reutilización [...] integrado en el proceso de desarrollo de software” (Arango, 1988). También caracterizó este periodo los intentos de automatizar la generación de código a partir de los resultados obtenidos por el análisis de dominio. Un ejemplo de esto es el proyecto FNIX desarrollado por Schlumberger-Doll Research en 1985, y en el que obtuvo como conclusión la necesidad de investigar sistemas de formalización del conocimiento de un dominio con mayor precisión para poder avanzar en el área de la programación automática. Otro proyecto significativo en los ochenta fue KATE y su concepto de extraer conocimiento a través de la crítica o "knowledge extraction by critique", con el que se buscaba ampliar el conocimiento registrado mediante críticas hipotéticas realizadas sobre una especificación inicial.

Uno de los principales hitos en el desarrollo del análisis de dominio ha sido el desarrollo del antes citado modelo FODA por parte del SEI (Software Engineering Institute). El modelo propuesto inicialmente ha sido la base de adaptaciones y desarrollos que trataron de mejorarlo. Así, ODM (Organization Domain Modeling) conjugó el modelo propuesto en FODA con el uso de clasificaciones facetadas popularizado por Prieto-Díaz. De forma similar, FeatuRSEB se presentó como una extensión del sistema RSEB (Reuse-Driven Software Engineering Business), que aplica las técnicas de diagramación del lenguaje UML (Unified Modeling Language) junto con el modelo FODA. La importancia de FODA se debe en parte al hecho de haber sido promovido desde el prestigioso SEI. Aunque pueda ser esta la causa por la cual esta técnica de análisis de dominio ha alcanzado una mayor visibilidad, hay otro aspecto decisivo: el hecho de que hablar de las características de un sistema informático es una práctica habitual, fácilmente comprensible, y usada normalmente para comparar sistemas informáticos. Se señala que se trata de abstracciones que tanto usuarios como equipos de ingeniería comparten y utilizan con frecuencia para caracterizar a los sistemas.

Pero sin ninguna duda el proyecto más relevante en la breve historia del análisis de dominio ha sido DARE (Domain Analysis and Reuse Environment), liderado por R. Prieto-Díaz y W. Frakes, y que establece una línea de continuidad entre el análisis de dominio, técnicas de minería textual y la representación del conocimiento basada en el uso de ontologías. Entendiendo por ontología una conceptualización compartida y explícita de un dominio, la relación entre las ontologías y el análisis de dominio resulta obvia.

En DARE, se propone un enfoque combinado para la captura de conocimiento, y que puede describirse de la siguiente forma: un conjunto de expertos establecen una ontología no

excesivamente detallada, que se tomará como punto de partida (este proceso recibe el nombre de top-down o de arriba a abajo, y se basa en el análisis conceptual del dominio). Simultáneamente, se completa un análisis de la documentación disponible sobre el dominio en cuestión, para extraer los términos y combinaciones de términos significativos de forma automática. Se utilizaron para ello técnicas basadas en la frecuencia estadística, co-ocurrencia y distribución de los términos en la colección. Tras esto, se procede a analizar los términos extraídos por el proceso automatizado de la colección de documentos con los propuestos en la ontología inicial, y se podrán realizar los ajustes oportunos en ésta hasta lograr un esquema que satisfaga y de cabida a los hallazgos de los dos sistemas de extracción/representación de conocimiento.

DARE - que es tanto una metodología de trabajo como una herramienta informática con el mismo nombre - incorpora una serie de utilidades para facilitar el mantenimiento de la ontología, agrupar términos similares en un mismo concepto o cluster, asignarlos a facetas, etc. La agrupación de términos similares se realiza a partir de su similitud, y ésta se calcula tomando como base la co-ocurrencia de los términos en una misma frase, párrafo, etc. Las agrupaciones obtenidas inicialmente tras el procesamiento automático de los documentos suelen traducirse en facetas, si bien en DARE la responsabilidad de seleccionar las facetas y decidir qué términos constituirán parte de las mismas se reserva a un editor humano. Una de las principales ventajas de esta aproximación es que se une en análisis experto con la extracción de términos de las colecciones de documentos, lo que garantiza la rápida identificación de nuevos conceptos (lo que sucede frecuente en determinadas áreas científicas y técnicas).

Además de DARE, otros proyectos desarrollados en los últimos años que podemos citar incluyen: ODE (Ontology Based Domain Engineering) o FAST (Family-Oriented Abstraction, Specification and Translation). También se han desarrollado herramientas y utilidades informáticas adicionales para facilitar la realización de análisis de dominio, como son 001 de Hamilton Technologies, PuLSE-BEAT del IESE (Institut Experimentelles Software Engineering), Odyssey (Federal University of Rio de Janeiro), FORM/ASADAL (Pohang University of Science and Technology, Korea), RequiLine (Research Group Software Construction RWTH) o GEARS (Big Lever). Las cinco últimas se publicaron con posterioridad al año 2000, lo que demuestra el interés existente en este área de la ingeniería del software y de la representación del conocimiento.

5 Conclusiones

En la década de los ochenta, las iniciativas dirigidas a la reutilización de software popularizaron el concepto de análisis de dominio, entendiendo por tal el proceso intelectual consistente en abstraer las características, vocabulario y reglas comunes a los sistemas informáticos de un mismo tipo. Se pretendía así extraer las características básicas de los sistemas de información destinados a un mismo sector, actividad o proceso.

Las técnicas de extracción y representación del conocimiento utilizadas por el análisis de dominio han hecho un uso intensivo del tratamiento automático de textos y de la minería textual, y también se han identificado similitudes con los procesos de diseño de ontologías. En la actualidad, el análisis de dominio es una práctica plenamente integrada dentro de la PLSE (Ingeniería de Software para Líneas de Productos). Los recientes avances en el área de las

ontologías constituyen una excelente oportunidad para revisar las técnicas y herramientas diseñadas hasta la fecha e identificar oportunidades de mejora en los modelos tradicionales.

Bibliografía citada

ARANGO, G. Domain Engineering for Software Reuse. Ph.D. Thesis, Department of Information and Computer Science. University of California, Irvine, 1988.

ESPECIFICACIÓN FODA. Disponible en: http://www.sei.cmu.edu/domain-engineering/FODA.html [Consultado: 10 nov. 2006]

FRAKES, W., PRIETO-DIAZ, R., AND FOX, C. DARE: Domain analysis and reuse environment. Annals of Software Engineering, 1998, n. 5, p. 125–141

KWANWOO LEE, KYO C. KANG, AND JAEJOON LEE. Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. En: ICSR: international conference on software reuse (7º. Austin TX. 2002), vol. 2319, p. 62-77.

NEIGHBORS, J. Software Construction Using Components. Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine, 1981.

NEIGHBORS, J. The Draco Approach to Constructing Software from Reusable Components. IEEE Transactions on Software Engineering, Sept. 1984, n. 10, p. 564-573.

PRIETO-DIAZ, R. Domain Analysis for Reusability. En: Proceedings of COMPSAC'87, Tokyo, Japan, (23-29), October, 1987.

PRIETO-DIAZ, R.. Domain Analysis: An Introduction. En: Software engineering Notes, ACM SIGSOFT, 1990, vol. 15, n.. 02, p. 47-54.