Búsqueda de servicios para asistir en el desarrollo de una Línea de Productos de Software

Maximiliano Arias, Alan De Renzis, Agustina Buccella, Alejandra Cechich, and Andres Flores

GIISCO Research Group,
Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue, Neuquén, Argentina
{maximiliano.arias,alanderenzis,agustina.buccella,alejandra.cechich,
andres.flores}@fi.uncoma.edu.ar
http://giisco.uncoma.edu.ar

Abstract. El desarrollo de Líneas de Productos de Software (LPS) es un paradigma de desarrollo basado en reuso muy vigente en la actualidad. Este paradigma plantea la existencia de una base común de funcionalidades definidas para un dominio particular de la cual se pueden instanciar productos con características similares. Estas funcionalidades pueden ser comunes a todos los productos o variar según las necesidades específicas de un producto particular. Existe un gran número de procesos de desarrollo definidos para el desarrollo de LPS que buscan reducir tiempo y esfuerzo. Para esta tarea los recursos semánticos, como las taxonomías, permiten organizar estos servicios para asistir en el desarrollo de nuevas funcionalidades y la instanciación de nuevos productos. A medida que una LPS crece estos recursos semánticos también crecen y se vuelven más difíciles de explorar, haciendo que su utilidad disminuya. Este trabajo propone un proceso de selección asistida de servicios que permita reducir o eliminar este problema. Este proceso ha sido instanciado en el contexto de una LPS cuyo desarrollo se basó en una taxonomía de servicios derivada de estándares geográficos y fue validado mediante evaluación experimental con expertos en el área.

Keywords: Líneas de productos de software, taxonomía de servicios, recuperación de información

1 Introducción

El desarrollo de Líneas de Productos de Software [5, 12] es un paradigma centrado en la identificación temprana de servicios comunes y variables que definen las funcionalidades de los productos a ser derivados. Los servicios comunes son aquellos que constituirán la base común o plataforma de la LPS, la cual será similar para todos los productos. Los servicios variables, en cambio, son los que

proveen la flexibilidad necesaria para permitir adaptarse a las necesidades o requerimientos particulares de las organizaciones o clientes. La identificación de estos dos tipos de servicios, comunes y variables, es una tarea fundamental dentro de todo desarrollo de las LPSs ya que repercutirá en el reuso efectivo que se haga de la plataforma desarrollada y la cantidad de productos derivados de la misma. A su vez, para maximizar dicho reuso, en la literatura han surgido propuestas [1, 4] que combinan el desarrollo de LPSs con el desarrollo basado en componentes [16]. Así, construir una LPS mediante el desarrollo de componentes reusables permitirá aprovechar las ventajas de poseer componentes ya diseñados e implementados para formar parte de los productos derivados a partir de la misma. Sin embargo, tanto la tarea de identificar y seleccionar estos servicios, como la selección de componentes reusables no es una tarea sencilla.

Existen en la literatura varias metodologías y técnicas que asisten a las tareas de diseño y desarrollo de las LPSs. Muchas de ellas se basan en la aplicación de recursos semánticos [20, 27, 28, 31] como herramientas de soporte a la definición de las funcionalidades comunes y variables del dominio abarcado en la LPS. Por ejemplo, en los trabajos presentados en [20, 28] se definen y aplican ontologías para modelar y razonar dentro de los dominios. En [28] se utiliza una ontología para detectar inconsistencias durante la derivación de nuevos productos basados en componentes reusables, y en [27] se utiliza para definir los requerimientos de software comunes y variables. En el mismo sentido, algunos trabajos definen taxonomías como recursos semánticos para mejorar ciertas tareas del desarrollo de las LPSs. Por ejemplo, en [24, 30] las taxonomías son utilizadas para la gestión de la variabilidad y clasificación de funcionalidades. Sin embargo, a pesar de que estos trabajos están orientados a mejorar tanto la comunicación de los participantes (usuarios, ingenieros de software y desarrolladores), como la eficiencia en diferentes etapas del proceso de desarrollo; los mismos poseen limitaciones en cuanto a la forma en que los recursos son aplicados y aprovechados. Es sabido que el lenguaje utilizado por los usuarios finales y la terminología de un dominio específico en general no es familiar a todos los participantes [14]. Así, el hecho de realizar esfuerzos específicos que ayuden a dicha comunicación podrían mejorar y agilizar las tareas para las cuales los recursos semánticos han sido definidos.

En particular, en este trabajo nos basamos en el uso de taxonomías para guiar el desarrollo de funcionalidades comunes y variables de una LPS e identificar posibles componentes a reusar. Como esfuerzos iniciales hacia una mejor entendibilidad entre todos los participantes, en trabajos previos [7, 8, 25] hemos desarrollado una taxonomía basada en estándares del domino, en particular el dominio geográfico. Sin embargo, el lenguaje de la taxonomía estandarizada resulta demasiado técnica con respecto a la que utilizan los usuarios o viceversa. Esta diferencia dificulta la tarea de los ingenieros de software y desarrolladores que deben definir las funcionalidades de acuerdo a la identificación de servicios en la misma.

Así, en este trabajo, como contribución para mejorar la comunicación y definición de la funcionalidad y como herramienta para cualquier otra metodología que utilice esta clase de recursos semánticos, proponemos la definición e imple-

mentación de un *proceso de selección asistida* de servicios que permita reducir el esfuerzo de búsqueda de requerimientos de software dentro de una taxonomía de servicios. El proceso presentado se divide en dos actividades: preprocesamiento e indexación. Para facilitar estas tareas se utilizan técnicas y herramientas conocidas en el campo de Recuperación de Información (IR[1]) [3].

2 Motivación

Una LPS se enfoca en un dominio particular y se basa en la identificación y aplicación de un conjunto de funcionalidades construidas a partir de servicios, los cuales deben ser identificados como comunes o variables dependiendo el dominio para la cual se está desarrollando la LPS.

La Figura 1 muestra el proceso de desarrollo de una LPS [7]. Este proceso se divide en dos tipos de análisis: el *análisis de dominio* y el *análisis organizacional*. Durante el primero los ingenieros de software analizan las necesidades generales del dominio, que serán utilizadas para la construcción de la base de funcionalidades comunes. El segundo se enfoca en estudiar el contexto definido por la organización para la cual se desea instanciar un sistema de la LPS.

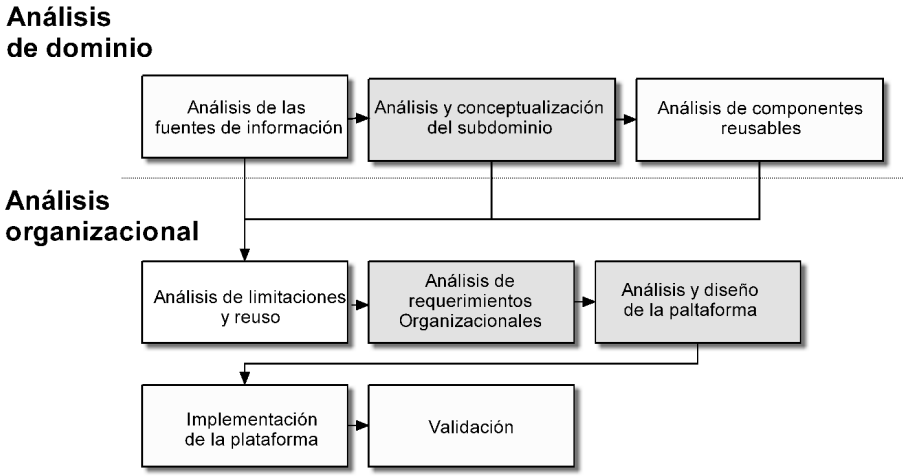

Fig. 1. Etapas del proceso de desarrollo de una LPS.

Durante las etapas de *Análisis y conceptualización del dominio, Análisis de requerimientos organizacionales, y Análisis y diseño de la plataforma* se realizan las tareas de análisis de requerimientos y diseño de funcionalidades a partir de las necesidades detectadas en el dominio y en la organización. Esta tarea implica la selección de servicios que permitan describir tales funcionalidades, por lo que

[1] Information Retrieval

resulta imperativo contar con alguna forma estructurada de administración y disposición de los servicios.

La Figura 2 muestra el proceso de diseño de una nueva funcionalidad para una LPS, siguiendo una metodología de desarrollo guiada por taxonomías [8], y se realiza durante las fases coloreadas en gris que se observan en la Figura 1. En primer lugar se puede observar que la funcionalidad requerida por los participantes es traducida a un conjunto de requerimientos atómicos (paso 1). Esta tarea se realiza como parte de la elicitación de requerimientos y el conjunto obtenido se utiliza para obtener los servicios de la taxonomía correspondiente. De esta manera, se busca para cada uno de los requerimientos individuales (en lenguaje natural) su contrapartida en la taxonomía. Debido a que la taxonomía podría no tener un alto nivel de especificidad (o granularidad fina de servicios), se podría identificar que un requerimiento no cuente con un servicio específico candidato. En este caso se debe marcar el requerimiento como posible incorporación y se debe evaluar esta decisión y actualizar la taxonomía – en caso de que se decida agregarlo a la misma (paso 2) –. De esta manera, se corresponden los requerimientos a servicios candidatos y este conjunto es utilizado para la elaboración de documentación de diseño como diagramas de variabilidad, secuencia, y otros que sean necesarios. En particular, a partir de este conjunto de candidatos se debe derivar el diagrama de componentes correspondiente a esa funcionalidad, para luego buscar en un repositorio los componentes ya existentes que se correspondan con los servicios seleccionados (paso 3). Finalmente, como parte de este proceso se construye la documentación necesaria para el diseño detallado de uno o más componentes que permitan implementar la funcionalidad requerida.

Los pasos (1) y (2) de la Figura 2 se enfocan en la estructuración del requerimiento del usuario hacia la taxonomía de servicios del dominio particular sobre el cuál se enfoca la LPS y al refinamiento de la misma. Durante estos pasos es de importancia identificar los servicios que se corresponden con los requerimientos definidos y escritos en lenguaje natural. Esta tarea no es un proceso trivial y requiere de un análisis de texto en lenguaje natural, tanto para el requerimiento como para los servicios presentados en la taxonomía. Conocer la estructura de la taxonomía y los servicios que contiene facilita de cierto modo el proceso, pero incluso así puede requerir un gran esfuerzo (y tiempo), determinar el conjunto de servicios candidatos al explorar la taxonomía de manera manual. Aún así, la identificación de un servicio o conjunto de servicios para el diseño y desarrollo de una funcionalidad, servirá de soporte para la posterior recuperación de componentes reusables – que se efectúa en el paso (3).

Como se expuso anteriormente, la tarea de corresponder requerimientos de software a una taxonomía de servicios requiere un esfuerzo considerable. En este trabajo se propone agilizar el proceso de desarrollo de nuevas funcionalidades a partir de requerimientos mediante un *proceso de selección asistida* de servicios. De esta manera se podrá facilitar la etapa de identificación, análisis y diseño de funcionalidades comunes y variables, reduciendo posibles errores que surjan de una inadecuada selección de servicios.

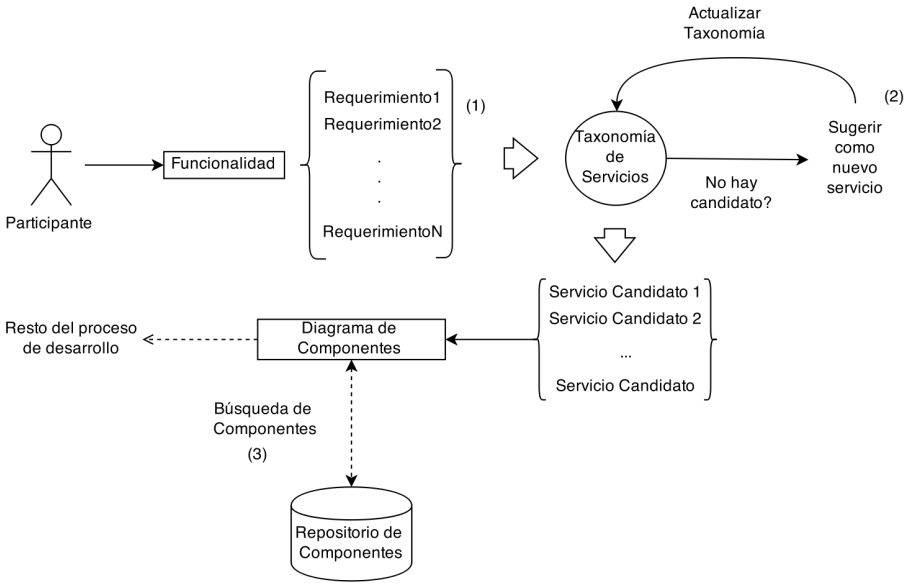

Fig. 2. Proceso de diseño de una funcionalidad para la LPS.

3 Propuesta

El *proceso de selección asistida* de servicios aplica herramientas y técnicas ampliamente utilizadas en el campo de IR , cuyo objetivo es facilitar la localización de determinados datos u objetos, y las relaciones existentes entre ellos. Los sistemas automatizados de IR se utilizan para reducir la "sobrecarga de información" – causando que una persona tenga dificultad en la comprensión de un problema– donde los errores en la toma de decisiones pueden deberse al exceso de información [21].

De esta manera, los mecanismos de búsqueda desarrollados para tratar elementos escritos en lenguaje natural, pueden ser aplicados para identificar servicios de la taxonomía que se asemejen al requerimiento que se está analizando, o bien agregar aquellos que no se encuentren dentro de la misma. Además, si se realiza esta tarea de manera iterativa, para cada uno de los requerimientos de los distintos sistemas que se desarrollen a partir de la LPS se puede a su vez alcanzar la construcción de una taxonomía de servicios cada vez más completa.

Para realizar el *proceso de selección asistida*, se desarrolló una estrategia que permite indexar una taxonomía de servicios expresada como una estructura de árbol y un método de preprocesamiento para reducir la diferencia existente entre las descripciones en lenguaje natural de los participantes y el lenguaje técnico de una taxonomía. Para poder llevar a cabo el preprocesamiento se hizo uso de la base de datos léxica del idioma inglés, WordNet y está inspirada en las teorías psicolingüísticas de la memoria léxica humana actual [22]. WordNet

organiza las palabras (sustantivos, verbos, adjetivos y adverbios) en conjuntos de sinónimos correspondientes a conceptos léxicos. Las relaciones entre conceptos son presentadas como puntos de unión semánticos entre conceptos relacionados. Es muy utilizado en numerosas aplicaciones relacionadas con el procesamiento del lenguaje natural.

Estos elementos se combinaron con el motor de búsqueda provisto por Apache, Lucene[2]. De esta manera, se construyó un buscador de requerimientos capaz de encontrar un conjunto de servicios relevantes para un requerimiento dado, generando un ranking de aquellos que más se asemejan al requerimiento buscado.

La Figura 3 muestra el proceso definido por el esquema propuesto. Podemos observar que, en primera instancia, la taxonomía es preprocesada e indexada. Para esta tarea, se explora la misma aplicando las tareas de separación en términos y enriquecimiento para cada uno de los servicios contenidos en ella. Así, se genera un archivo para cada uno de estos servicios y se indexan.

Por otro lado, por cada requerimiento que se desea realizar una búsqueda se efectúan tareas de preprocesamiento y el requerimiento enriquecido es buscado en la taxonomía indexada. Cabe destacar que tanto el requerimiento a buscar como la taxonomía son sometidos al mismo proceso de preprocesamiento, separando en términos las frases que los componen y obteniendo, mediante el uso de WordNet [22], una versión enriquecida de los mismos.

A continuación se describe cada una de estas dos etapas del proceso.

3.1 Preprocesamiento

Tanto en las LPSs como en cualquier desarrollo de software convencional, los requerimientos de software para el desarrollo de nuevas funcionalidades se obtienen a partir de las necesidades expresadas por los usuarios. Dichos requerimientos se expresan en lenguaje natural, pero incluyendo una cantidad de expresiones coloquiales – en la jerga propia del dominio de los usuarios. Una taxonomía de servicios o cualquier recurso semántico estandarizado está generalmente descrito mediante lenguaje natural de carácter técnico y específico, con el objetivo de reducir ambigüedades. Por estas razones existe una diferencia entre los lenguajes utilizados, que genera una dificultad extra cuando se necesita de realizar una búsqueda en una taxonomía mediante un conjunto de requerimientos. Para resolver este problema, la etapa de preprocesamiento se divide en dos actividades: separación en términos y enriquecimiento.

Separación en términos Toma como entrada una frase escrita en lenguaje natural, y realiza una separación semántica de términos [13]. Se realiza una detección de términos separados por espacios y de términos que se encuentran unidos en una palabra pero que siguen alguna convención. Para esto agrega un nivel semántico a este proceso, contemplando casos que no respetan las convenciones de nombrado. Utilizando WordNet se analiza cada posible término

[2] https://lucene.apache.org/core/

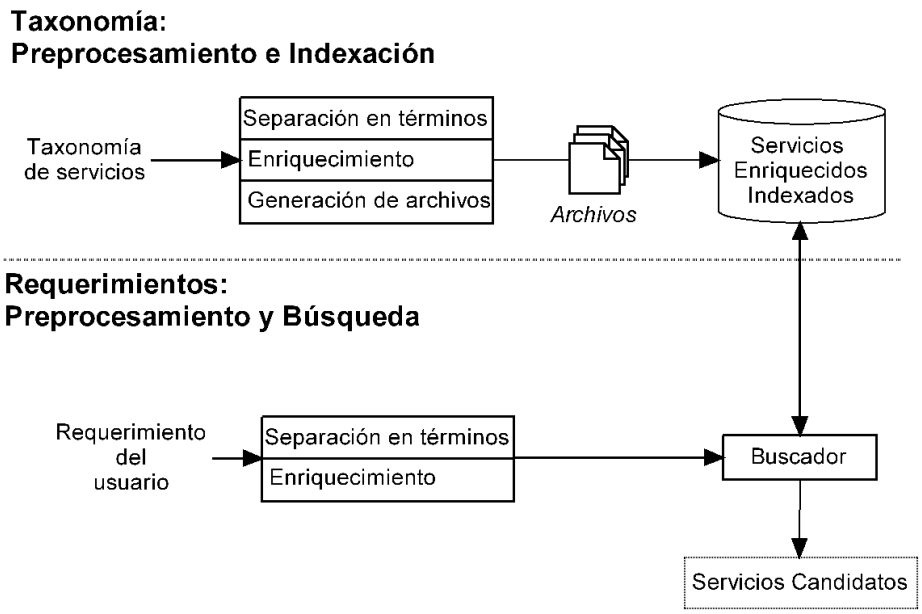

Fig. 3. Proceso de búsqueda de un requerimiento dentro de la taxonomía

y se determina cual es la separación correcta. Una vez obtenida una lista de términos, se eliminan aquellos considerados como StopWords. En el campo de la Recuperación de Información y el Procesamiento del Lenguaje Natural, las palabras/términos que no ayudan a determinar la semántica de una frase, que no poseen un peso semántico suficiente, que son irrelevantes para determinar si un conjunto de términos es similar a otro, y ciertas palabras de cada dominio específico, componen las listas de stopwords [9].

Enriquecimiento Luego de obtener la lista de términos, se toma cada término y se utiliza WordNet para buscar los sinónimos (palabras con exactamente el mismo significado), hipónimos (palabra con un significado más específico) e hiperónimos (palabras con un significado más general). Estos términos son agregados a la lista con el objetivo de expandir el lenguaje de la misma. De esta manera son contemplados casos donde los usuarios no utilizan exactamente los mismos términos que aquellos definidos en la taxonomía. Para los hiperónimos e hipónimos sólo se considera un nivel hipo-hiperonímia – es decir que sólo se tienen en cuenta aquellos con una relación directa.

3.2 Indexación y búsqueda

En esta esta etapa, se reescribe el recurso semántico o taxonomía en función del conjunto de servicios que incluye, donde los mismos son derivados a archivos

de texto. Cada uno de estos archivos contienen el resultado de preprocesar las palabras de los servicios, y se los denomina utilizando el nombre completo del servicio dentro del árbol de la taxonomía. Luego los archivos de texto se indexan por medio del motor de búsqueda Lucene, para que puedan ser recuperados ante una búsqueda con respecto a requerimientos de los usuarios. Como se muestra en la Figura 4 el servicio "Show default base map" es preprocesado obteniendo el archivo "Show default base map.txt" que contiene las palabras que conforman la descripción del mismo (show,default, base y map) y todos aquellos términos agregados como resultado del proceso de enriquecimiento (entertainment, display, piece, etc). Luego de generar el archivo correspondiente al servicio, se indexa para que el motor de búsqueda pueda encontrarlo dentro de su espacio de búsqueda.

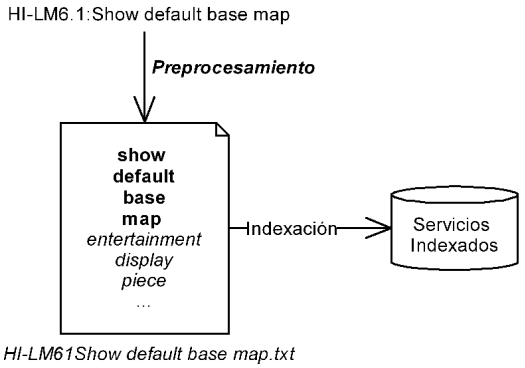

Fig. 4. Indexación de un servicio de la taxonomía

Tras indexar la taxonomía siguiendo el proceso explicado anteriormente se puede utilizar el motor de búsqueda para realizar consultas.

4 Instanciación del esquema y evaluación experimental

El *proceso de selección asistida* propuesto se validó mediante la construcción de un caso de estudio a partir de un conjunto de necesidades o requerimientos de usuario y una taxonomía desarrollada en trabajos previos [7]. Tanto el conjunto de necesidades como la taxonomía pertenecen al dominio geográfico, particularmente al subdominio de ecología marina. La taxonomía de servicios se construyó siguiendo el estándar definido en la ISO 19119[3] y cuenta con un total de 120 servicios organizados en forma jerárquica y almacenados en una base de datos. De esta manera, según el proceso presentado en la Sección 3.2, se generó e indexó un archivo para cada uno de los servicios.

[3] http://www.iso.org/iso/catalogue_detail.htm?csnumber=39890

4.1 Evaluación Experimental

Este experimento tiene como objetivo evaluar el desempeño del *proceso de selección asistida* y medir el esfuerzo que demanda a un ingeniero de software o desarrollador obtener la correspondencia de los requerimientos del sistema dentro de la taxonomía de servicios. Para esto se contó con la colaboración de un grupo de ingenieros de software que buscaron manualmente servicios de la taxonomía para un conjunto de requerimientos. Finalmente, se realizó un análisis del impacto en el esfuerzo para asociar requerimientos del sistema a servicios de la taxonomía, por un lado realizando una búsqueda de manera manual y por el otro utilizando el buscador de servicios implementado.

Configuración del experimento Como primera medida se definió un escenario experimental en el que fueron considerados los 120 servicios de la taxonomía previamente preprocesados e indexados. Con el objetivo de poder medir los resultados de la recuperación de servicios, se consideró el conjunto de necesidades de usuarios presentado anteriormente. Estas necesidades fueron definidas en conjunto con un grupo de biólogos marinos del Instituto de Biología Marina y Pesquera Almirante Storni (IBMPAS[4]) y el Centro Patagónico (CENPAT[5]) y se encuentran expresadas en lenguaje natural. Cada necesidad fue traducida, a uno o más requerimientos del sistema, también expresados en lenguaje natural. Luego de realizar esta traducción se obtuvo como resultado un conjunto de 58 requerimientos del sistema. Finalmente, para cada uno de estos requerimientos se estableció una correspondencia a un servicio de la taxonomía de forma manual. Esta tarea la realizó un experto en GIS, que además es uno de los creadores de la taxonomía de servicios utilizada.

Para este experimento participaron seis ingenieros de software pertenecientes al Grupo de Investigación de Ingeniería de Software del Comahue (GIISCO[6]), de los cuales varios no participaron en la creación de la taxonomía ni estaban familiarizados con el dominio. Se le entregó a cada uno de ellos una copia de la taxonomía en formato digital.

Ejecución del experimento Una vez que el escenario experimental fue definido e instanciado, se procedió a la ejecución del experimento. En principio se solicitó a cada desarrollador que establezca manualmente la correspondencia de un conjunto de 9 requerimientos (en promedio) a los servicios de la taxonomía. Para realizar la búsqueda de servicios relevantes se solicitó que registren el tiempo que les tomó explorar la taxonomía y seleccionar los servicios relevantes para los 9 requerimientos.

Por otro lado, cada requerimiento derivado a partir de una necesidad de usuario fue considerado como una consulta al motor de búsqueda. Esta consulta fue sometida al proceso de recuperación de servicios definido en la Sección 3. Para

[4] http://www.ibmpas.org/
[5] http://www.cenpat.edu.ar/
[6] http://giisco.uncoma.edu.ar

cada consulta fueron registrados los resultados proporcionados por el buscador. Estos datos incluyen por cada consulta (requerimiento) una lista ordenada de N servicios candidatos.

Para medir el desempeño del buscador y los resultados de la búsqueda manual realizada por los desarrolladores, se compararon los datos obtenidos usando como contrapartida los servicios relevantes seleccionados manualmente por los creadores de la taxonomía.

Resultados para Selección Manual Los resultados de la selección manual de servicios por parte de los seis desarrolladores expertos fueron medidos en base a la cantidad de aciertos y tiempo promedio que requirió analizar cada requerimiento y seleccionar el servicio correspondiente. Cada desarrollador demoró en promedio 22,5 minutos en analizar 9 requerimientos seleccionando el servicio en la taxonomía correspondiente a cada uno, por lo tanto el promedio de tiempo por requerimiento es de 2,5 minutos.

Por otro lado se registró el porcentaje de aciertos promedio en cuanto a la selección de servicios – es decir, observando cuáles de los servicios de la taxonomía seleccionados por los desarrolladores eran correctos. Para evaluar si la selección de los desarrolladores era correcta, fueron usados como referencia los servicios seleccionados por el experto de la taxonomía. Finalmente, el porcentaje de aciertos promedio en la selección manual fue de 56%.

Resultados del buscador de servicios En primer lugar medimos el desempeño del buscador de servicios utilizando dos métricas pertenecientes al campo de Recuperación de Información: *Precision-at-n* y *Recall* [15]. La métrica *Precision-at-n* calcula la precisión en diferentes puntos de la lista de candidatos retornados durante la selección. Formalmente, *Precision-at-n* para una consulta individual se define como:

$$Precision{-}at{-}n = \frac{RetRel_n}{n} \quad (1)$$

Donde $RetRel_n$ es el número de servicios de la taxonomía relevantes recuperados en las primeras n posiciones.

La métrica *Recall* mide el desempeño del proceso de selección de servicios de la taxonomía a partir de los servicios relevantes que recupera. La medida *Rec*all es 100% cuando cada servicio relevante es devuelto en la lista de candidatos.

La Tabla 4.1 resume los resultados para las métricas de *Precision-at-n* (con $n \in$ [1-6]) y *Recall* para el escenario experimental definido.

Discusión El resultado del *Recall* del buscador de 74% significa que, si un desarrollador lo utilizase para seleccionar los servicios a partir de un conjunto de requerimientos, su esfuerzo de búsqueda podría reducirse aproximadamente en un 74%. Esto se debe a que el tiempo de ejecución del buscador es insignificante y en el peor de los casos debería analizar 6 servicios (pues la precisión en 6 es

Métrica	*Valor*
Precision-at-1	0.65
Precision-at-2	0.75
Precision-at-3	0.82
Precision-at-4	0.86
Precision-at-5	0.96
Precision-at-6	1
Recall	0.74

Tabla 1. Precision y Recall en la recuperación de servicios de la taxonomía

de 1, es decir que todos los servicios correctamente recuperados se encontraron como máximo en la posición 6 de la lista de servicios candidatos).

Si analizamos que con un ejemplo de sólo 120 servicios, en donde una búsqueda manual es posible, los resultados son desalentadores, podemos imaginarnos que al aumentar la cantidad de servicios empeoraría estos resultados. Contar con un *proceso de selección asistida* sería una opción recomendable, ya que una parte de los servicios se correspondería de manera automática y el espacio de búsqueda se vería acotado a un número de servicios mucho menor.

5 Trabajos Relacionados

Existen diferentes propuestas enfocadas a la definición y aplicación de diferentes recursos semánticos que sirven de soporte para varias etapas dentro del desarrollo de software convencional como el definido para las LPSs. En particular, en trabajos como [2, 10] podemos observar como las taxonomías son utilizadas para mejorar y facilitar las tareas de identificación de componentes [2], completitud de requerimientos de software [10], gestión de variabilidad [30], clasificación de aspectos no funcionales [24] y evolución [29], entre otros. A su vez, existen varios trabajos proponiendo técnicas para la creación de taxonomías u ontologías que sean útiles en el desarrollo del sistemas de información [6, 11, 17, 23]. En general, todos los autores coinciden en que tanto la definición como el uso de estos recursos implica una continua evaluación y manipulación de estos recursos ya que los mismos serán continuamente accedidos y consultados. De esta manera, además de contar con estos recursos son necesarias también herramientas que asistan a todos los participantes para mejorar la entendibilidad de los mismos y sus usos futuros. ejemplo, en [18, 19] los autores presentan dos enfoques orientados a requerimientos de sistema, el primero para elicitación y el segundo de análisis. Ambos enfoques están basados en ontologías, donde una ontología para un dominio especifico se usa como conocimiento del mismo y juega un papel en el dominio semántico que da significado a los requerimientos. Este proceso se realiza a través de una función semántica. En ambas propuestas, un ingeniero de software debe asociar cada requerimiento a conceptos atómicos de una ontología, la cual no es una tarea sencilla. Para mejorar la eficiencia y tiempo de esta tarea,

el *proceso de selección asistida* presentado en este trabajo permitiría asistirlo en este trabajo.

Otro ejemplo en donde nuestra propuesta resultaría útil es en el trabajo presentado en [26]. Aquí los autores describen un explorador de ontologías que permite explorar todas las clases y atributos dentro de una ontología a partir de búsquedas realizadas por usuarios. Estas búsquedas se especifican completando valores en campos generados automáticamente a partir de la información proporcionada por la ontología. Cuando se explora la ontología, con el objetivo de encontrar correspondencias a partir de requerimientos del sistema, el usuario debe encargarse de inferir qué información completar en cada campo antes de realizar la búsqueda. Además, debe utilizar exactamente el mismo vocabulario que el definido por la ontología, ya que no dispone de una base de datos léxica para inferir similitudes entre palabras.

6 Conclusiones y trabajos futuros

El proceso de desarrollo de una LPS se centra en gran medida en el uso de servicios que se agrupan para formar funcionalidades. De esta manera, una gran parte del proceso de elicitación y análisis de requerimientos involucra la selección de servicios adecuados para la definición y el diseño de las funcionalidades que conformarán la línea. El esfuerzo de selección de servicios puede ser reducido mediante la introducción de una taxonomía que permita organizarlos en categorías útiles. Sin embargo, aún contando con esta taxonomía, la búsqueda y selección requiere de un gran conocimiento de la misma para minimizar el tiempo que implica explorarla y generar una correspondencia de servicios con requerimientos del sistema. De esta manera en este trabajo definimos e implementamos un *proceso de selección asistida* de servicios que asiste a tres fases dentro del proceso de desarrollo de las LPSs. El mismo se basa en la asistencia a los participantes de un proyecto (usuarios finales, ingenieros de software y desarrolladores) que permita una rápida y ágil identificación de los requerimientos de los usuarios finales en lenguaje natural y la correspondencia a servicios dentro de una taxonomía.

De acuerdo a los resultados obtenidos durante la evaluación experimental, concluimos que el uso de un *proceso de selección asistida* provee beneficios en cuanto a una reducción significativa del tiempo de selección de servicios y a una mejora en la calidad de esta selección. De esta manera se reduce el tiempo de desarrollo e instanciación de productos de una LPS y se mitigan posibles errores (derivados de una inadecuada selección) que podrían generar problemas en fases subsecuentes del proceso de desarrollo.

Como trabajo futuro, se propone extender este trabajo a otras etapas del desarrollo de las LPSs (Figura 2). En particular, nos interesa utilizar dicho *proceso de selección asistida* para agilizar la búsqueda de componentes previamente desarrollados que se correspondan con los servicios de la taxonomía y así asistir a la tarea de identificación de los mismos y maximizar el reuso.

References

1. Colin Atkinson, Joachim Bayer, and Dirk Muthig. Component-based product line development: The kobra approach. In Patrick Donohoe, editor, *Software Product Lines*, volume 576 of *The Springer International Series in Engineering and Computer Science*, pages 289–309. Springer US, 2000.
2. Claudia Ayala, Pere Botella, and Xavier Franch. Construction of a taxonomy for requirements engineering commercial-off-the-shelf components. *Journal of Computer Science & Technology*, 5, 2005.
3. R Baeza-Yates, B Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
4. Joachim Bayer, Oliver Flege, Peter Knauber, Roland Laqua, Dirk Muthig, Klaus Schmid, Tanya Widen, and Jean-Marc DeBaud. Pulse: A methodology to develop software product lines. In *Proceedings of the 1999 Symposium on Software Reusability*, SSR '99, pages 122–131, New York, NY, USA, 1999. ACM.
5. Günter Böckle, Frank J van der Linden, and Klaus Pohl. *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media, 2005.
6. Denise Bruno and Heather Richmond. The truth about taxonomies. *Information Management Journal*, 37(2), 2003.
7. Agustina Buccella, Alejandra Cechich, Maximiliano Arias, Matias Pol'La, Maria del Socorro Doldan, and Enrique Morsan. Towards systematic software reuse of gis: Insights from a case study. *Computers & Geosciences*, 54:9–20, 2013.
8. Agustina Buccella, Alejandra Cechich, Matias Polla, Maximiliano Arias, Maria del Socorro Doldan, and Enrique Morsan. Marine ecology service reuse through taxonomy-oriented {SPL} development. *Computers & Geosciences*, 73(0):108 – 121, 2014.
9. Agustin Casamayor, Daniela Gody, and Marcelo Campo. Mining architectural responsibilities and components from textual specifications written in natural language. *Proceedings of the XI Argentine Symposium on Software Engineering*, 2010.
10. Brian D Chance and Bonnie E Melhart. A taxonomy for scenario use in requirements elicitation and analysis of software systems. In *Engineering of Computer-Based Systems, IEEE International Conference on the*, pages 232–232. IEEE Computer Society, 1999.
11. Carol EB Choksy. 8 steps to develop a taxonomy. *Information Management*, 40(6), 2006.
12. P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2002.
13. Alan De Renzis, Martin Garriga, Andres Flores, Alejandra Cechich, and Alejandro Zunino. Semantic-structural assessment scheme for integrability in service-oriented applications. In *Computing Conference (CLEI), 2014 XL Latin American*, pages 1–11. IEEE, 2014.
14. Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2002.
15. Daniela Godoy and Analía Amandi. Hybrid content and tag-based profiles for recommendation in collaborative tagging systems. In *Latin American Web Conference, 2008. LA-WEB'08.*, pages 58–65. IEEE, 2008.
16. George T. Heineman and William T. Councill, editors. *Component-based Software Engineering: Putting the Pieces Together*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

17. Ivo Hunink, Rene van Erk, Slinger Jansen, and Sjaak Brinkkemper. Industry taxonomy engineering: the case of the european software ecosystem. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pages 111–118. ACM, 2010.
18. Haruhiko Kaiya and Motoshi Saeki. Ontology based requirements analysis: lightweight semantic processing approach. In *Quality Software, 2005.(QSIC 2005). Fifth International Conference on*, pages 223–230. IEEE, 2005.
19. Haruhiko Kaiya and Motoshi Saeki. Using domain ontology as domain knowledge for requirements elicitation. In *Requirements Engineering, 14th IEEE International Conference*, pages 189–198. IEEE, 2006.
20. Marcello La Rosa, WilM.P. van der Aalst, Marlon Dumas, and ArthurH.M. ter Hofstede. Questionnaire-based variability modeling for system configuration. *Software & Systems Modeling*, 8(2):251–274, 2009.
21. Pattie Maes et al. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40, 1994.
22. George Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244, 1990.
23. Robert Nickerson, Jan Muntermann, Upkar Varshney, and Henri Isaac. Taxonomy development in information systems: Developing a taxonomy of mobile applications. In *European Conference in Information Systems*, 2009.
24. Mahdi Noorian, Ebrahim Bagheri, and Weichang Du. Non-functional properties in software product lines: A taxonomy for classification. In *SEKE*, volume 12, pages 663–667, 2012.
25. Matias PolÍa, Agustina Buccella, Maximiliano Arias, and Alejandra Cechich. Un modelo de metadatos para la gestión de la variabilidad en líneas de productos de software. In *Proceedings of ASSE 2014*, Buenos Aires, Argentina, 2014.
26. Anand Ranganathan, Robert E McGrath, Roy H Campbell, and M Dennis Mickunas. Use of ontologies in a pervasive computing environment. *The Knowledge Engineering Review*, 18(03):209–220, 2003.
27. Iris Reinhartz-Berger, Nili Itzik, and Yair Wand. Analyzing variability of software product lines using semantic and ontological considerations. In Matthias Jarke, John Mylopoulos, Christoph Quix, Colette Rolland, Yannis Manolopoulos, Haralambos Mouratidis, and Jennifer Horkoff, editors, *Advanced Information Systems Engineering*, volume 8484 of *Lecture Notes in Computer Science*, pages 150–164. Springer International Publishing, 2014.
28. Harvey Siy, Aaron Wolfson, and Mansour Zand. Ontology-based product line modeling and generation. In *Proceedings of the 2Nd International Workshop on Product Line Approaches in Software Engineering*, PLEASE '11, pages 50–54, New York, NY, USA, 2011. ACM.
29. Mikael Svahnberg and Jan Bosch. Evolution in software product lines. *Software Maintenance*, 11(6):391–422, 1999.
30. Mikael Svahnberg, Jilles Van Gurp, and Jan Bosch. A taxonomy of variability realization techniques. *Software: Practice and Experience*, 35(8):705–754, 2005.
31. Tewfik Ziadi and Jean-Marc Jezequel. Software product line engineering with the uml: Deriving products. In Timo Käköla and JuanCarlos Duenas, editors, *Software Product Lines*, pages 557–588. Springer Berlin Heidelberg, 2006.