Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 1, No. 2, pp. 12-22, Diciembre 2014
Selección de los Mejores Artículos de CoNCISa 2014

Modelos de Variabilidad con Requisitos no Funcionales en un Contexto de Producción Industrial de Software

Victor Esteller[1], Francisca Losavio[2], Alfredo Matteo[2], Oscar Ordaz[2]
vesteller@uc.edu.ve, francislosavio@gmail.com, alfredo.matteo@ciens.ucv.ve, oscarordaz55@gmail.com

[1] Departamento de Computación, Facultad de Ingeniería, Universidad de Carabobo, Valencia, Venezuela
[2] Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela

Resumen: La producción industrial de software es enfocada por la Ingeniería de Software como Líneas de Productos de Software (LPS) y se centra en representar características comunes y variables de productos, para lo cual se realiza un "modelo de variabilidad". Para la realización de dicho modelo se han propuesto diferentes enfoques, modelos, métodos y técnicas, las cuales tratan esencialmente la variabilidad respecto a los requisitos funcionales (RF); mientras que el problema de la variabilidad de los requisitos no funcionales (RNF) no ha sido aún resuelto. Por lo tanto, el objetivo de este trabajo es identificar aspectos claves que deben estar presentes en un proceso general de diseño del modelo de variabilidad que considere también los RNF, por lo que se hará una revisión sistemática de la literatura referente a los enfoques, modelos, métodos y técnicas vigentes que consideran modelos de variabilidad que explícitamente integran RF y RNF. Finalmente, a partir del estudio realizado, se hará una primera propuesta para un proceso de diseño de un modelo de variabilidad con RF y RNF, donde se considerará el estándar ISO/IEC 25010 para la especificación de los RNF para unificar la terminología del dominio respecto a las propiedades de calidad.

Palabras Clave: Línea de Productos de Software; Modelo de Variabilidad; Requisitos no Funcionales; ISO/IEC 25010; Revisión Sistemática.

Abstract: Industrial software production is focused by Software Engineering as Software Product Lines (SPL) and focuses on representing common and variable characteristics of software products, for which a "variability model" must be constructed. For the realization of this model different approaches, models, methods and techniques have been proposed, treating essentially the variability with respect to functional requirements (FR); while the problem of variability of non-functional requirements (NFR) has not treated as much and it is still an open a problem. Therefore, the aim of this work is to identify key aspects that must be present in a general design process of the variability model considering also NFR. In consequence, a systematic review of the literature on approaches, models, methods and techniques that consider existing variability models explicitly integrating NFR and FR is performed. Finally, from the results of the present study, an initial proposal for a design process of a variability model with FR and NFR is presented, considering the standard ISO/IEC 25010 to specify NFR and unify the terminology of the domain with respect to to quality properties.

Keywords: Software Product Line; Variability Model; Non-Functional Requirements; ISO/IEC 25010; Systematic Review.

I. Introducción

El desarrollo de sistemas de software a gran escala ha sido un desafío para investigadores y desarrolladores, y ha sentado las bases de la disciplina de la Ingeniería de Software, principalmente por lo complejo de la elaboración de un sistema como un todo o de cada uno de sus componentes individuales, y en particular, en lo referente a sus exigencias de calidad [1].

En este contexto, las Líneas de Productos de Software (LPS) [2], han sido propuestas para construir sistemas complejos, donde cada uno de sus componentes puedan ser reutilizados para el desarrollo de otros sistemas similares con requisitos distintos, facilitando a los desarrolladores la opción de administrar la configuración de una arquitectura genérica o arquitectura de referencia, es decir de seleccionar apropiadamente las propiedades comunes y variables de cada producto de software a reutilizar, para así generar nuevos productos.

Una LPS es definida como una colección o familia de sistemas de software que comparten un conjunto común y gestionado de características para un dominio, las cuales satisfacen necesidades específicas de un segmento particular del mercado y que son desarrolladas de una forma preestablecida, a partir de un conjunto común de activos de software [2]. Por otra parte, en [3][4] se utiliza la definición de un dominio, como el conjunto mínimo de propiedades que describen con precisión

una familia de problemas, para la cual se requiere una aplicación o solución computacional.

El término activo de software denota a un sistema, componente o producto de software diseñado expresamente para ser utilizado múltiples veces en el desarrollo de diferentes sistemas o aplicaciones; no solamente se consideran como activos a sistemas ejecutables sino también, por ejemplo, una especificación de requisitos, un modelo de negocio, una especificación de diseño o un patrón arquitectónico [5].

El proceso de desarrollo clásico de las LPS, denominado también proactivo o descendente, considera dos procesos que se llevan a cabo en paralelo: la Ingeniería del Dominio (ID) y la Ingeniería de Aplicación (IA) [6][7][8]. La ID captura información y representa el conocimiento sobre un dominio determinado, con la finalidad de crear activos de software reutilizables en el desarrollo de cualquier nuevo producto de una LPS [5].

Un aspecto central de la ID es construir la Arquitectura de Referencia (AR), o arquitectura del dominio, que consiste en una arquitectura genérica, cuyo núcleo básico de componentes está constituido por el modelo de características (en inglés "feature model"), que representa requisitos del software, funcionales o no, y el modelo de variabilidad (que representa las distintas maneras de generar nuevos productos reutilizando las características comunes del núcleo básico).

Un modelo de variabilidad [8] puede estar incluido o no en el modelo de características. Sus elementos conceptuales principales son los puntos de variación (en inglés "variation points") o componentes genéricos, que indican donde hay variabilidad en la AR y sus instancias o variantes (en inglés "variants") que son las componentes con soluciones concretas para configurar los nuevos productos. A partir de la AR debe ser posible generar todos los productos de la familia.

Para el desarrollo de los activos principales en una LPS, se requiere de un proceso de análisis del dominio, en el cual se identifica la parte común y variable de una familia de productos, formulándose así el modelo de variabilidad. Para esto, el estudio de la variabilidad en el análisis del dominio se ha convertido en un paradigma importante donde se han propuesto diferentes enfoques, modelos, métodos, aproximaciones y técnicas de análisis. En [5] se hace una revisión de métodos para el modelado de la variabilidad en general, y se considera que todas estas prácticas de análisis de dominio ayudan a construir, definir o utilizar modelos de variabilidad. Así mismo, en [9] se realiza una revisión de métodos de manejo de la variabilidad, sin tratarlos explícitamente como modelos.

Sin embargo, en ninguno de los trabajos [5][9] se tratan explícitamente los requisitos no funcionales (RNF), que en la literatura son también denominados indistintamente como atributos o requisitos de calidad.

Los enfoques comúnmente empleados para modelar la variabilidad en LPS enfatizan a la funcionalidad capturada a través de los requisitos funcionales (RF) y modelada generalmente por casos de uso. Por lo tanto, dentro de la óptica de las tendencias actuales, es importante que el proceso de especificación de la variabilidad tome en cuenta también los RNF y los correspondientes objetivos de calidad, tanto de RF como de RNF [12], porque permiten asegurar que el producto de software alcanzado cumpla con las características de calidad deseadas, para estar acorde con un proceso de producción industrial.

Con respecto al tópico: "tratamiento específico de la variabilidad de los RNF", que es el objetivo de este estudio, se han realizado numerosas investigaciones en la literatura. Para esto se hizo una revisión sistemática de acuerdo a la metodología de B. Kitchenham [10], y consistió en una búsqueda dirigida por estrategias, la selección y el filtrado de la literatura relativa al tópico de investigación planteado; con esta técnica se seleccionaron diez (10) trabajos que están directamente relacionados con el tópico y que se analizarán en el transcurso del presente trabajo.

A pesar de la existencia de múltiples aproximaciones sobre el modelado de las características esenciales comunes de los productos de un dominio y de su variabilidad, se observan muchas deficiencias, sobre todo en el abordaje de los RNF que no son directamente percibidos por el usuario, son difíciles de cuantificar, son especificados muy vagamente y generalmente se consideran en etapas posteriores al análisis. Estos aspectos incrementan enormemente la complejidad y disminuyen la eficiencia en el desarrollo, dificultando el mantenimiento, contradiciendo los principios del paradigma de la orientación a objetos.

Esto implica la necesidad de unificar criterios y el abordaje de los RNF en etapas tempranas, ya que de ellos se derivan los atributos de calidad asociados a los RNF, que deben ser medidos [11] y tomados en cuenta, ya que condicionan la calidad global del sistema. Además, se debe recalcar que las funcionalidades principales del dominio en general, derivadas de los RF, también exigen objetivos precisos de calidad. Es de resaltar que en un contexto de producción industrial, la calidad del producto representa un objetivo primordial a conseguir, cualquiera que sea la disciplina o enfoque empleado, es de máxima importancia; debe recordarse que la calidad del proceso determina la calidad del producto, [1]. Se define la calidad del producto como el conjunto de características o propiedades deseadas que deben estar presentes en el producto, considerando el punto de vista del usuario y del producto mismo [11].

Por otra parte, el hecho de considerar estándares de calidad, representa un aspecto central en la propuesta de este trabajo, ayudando a la construcción de un vocabulario común respecto a la calidad del producto de software. mediante el modelo de calidad, constituyendo así un activo de software importante para la construcción de la AR; debe recordarse que esta arquitectura genérica contiene el modelo de variabilidad para la generación de los diferentes productos. Establecer el modelo de calidad del dominio o vista de calidad del dominio [13], ayuda a definir los RNF globales del sistema, así como los requisitos de calidad que se derivan de las funcionalidades del usuario o funcionalidades implícitas, que son comunes a una familia de productos del dominio y representan la vista de calidad como un activo, siendo parte del conocimiento del dominio [4].

El tratamiento explícito de la variabilidad de RNF implica considerar requisitos de calidad no cuantificables directamente, como por ejemplo alta seguridad, disponibilidad e interoperabilidad; tomar en cuenta estos requisitos de calidad

implica, seleccionar variantes adicionales en la configuración de los productos específicos. Estos RNF de alto nivel no son medibles directamente y deben ser refinados hasta llegar a los elementos medibles. La idea es llegar a una granularidad que permita determinar la variabilidad no funcional para obtener productos concretos.

Por lo antes expuesto, determinar las características comunes y variables de una familia de productos es ampliamente estudiado y modelado durante el análisis del dominio, desde el punto de vista funcional; sin embargo no es así en lo que respecta a los RNF [8][12], hacia donde se dirige actualmente la investigación en el contexto de este trabajo. En este sentido Sepúlveda y Cachero [35] en respuesta a esta problemática, hacen una revisión de métodos de gestión de RNF en las LPS, sin embargo es un trabajo corto y no concretizan su propuesta de cómo facilitar la comunicación entre los grupos de trabajo.

Como consecuencia de la problemática planteada, nuestro trabajo tiene como objetivo principal revisar y comparar los diferentes modelos vigentes para la especificación de la variabilidad en la etapa de análisis del dominio, en particular aquellos modelos que contemplan también el tratamiento de la variabilidad respecto a los RNF.

Los resultados obtenidos en la revisión serán utilizados como base para plantear los aspectos esenciales que debe tener un modelo de variabilidad que integra RF y RNF, y un proceso de diseño del modelo de variabilidad de RF y RNF para una AR, inspirado en [32]. Además, se plantea cómo asociar a las características no funcionales del modelo de variabilidad, una especificación estándar utilizando la norma ISO/IEC 25010 [11].

El presente trabajo, además de esta introducción, se estructura de la siguiente forma: la segunda sección presenta la revisión sistemática que se siguió para seleccionar los enfoques. La tercera sección describe las técnicas, enfoques y modelos utilizados para expresar la variabilidad de RNF especificados como requisitos de calidad. La cuarta sección presenta la comparación y el análisis de resultados. En la quinta sección se formula una primera propuesta del proceso general de diseño del modelo de variabilidad con RF y RNF. Finalmente en la sexta sección se dan las conclusiones y perspectivas.

II. Revisión Sistemática

Para el estudio de la literatura vigente en el tema, se siguió la metodología de Revisión Sistemática o "Systematic Review" de B. Kitchenham [10], por sus ventajas en el área de la Ingeniería de Software para buscar, seleccionar y filtrar el enorme volumen de material literario disponible en un tema de investigación, y seleccionar aquellos trabajos directamente relacionados y de mayor interés; para tal fin se siguieron las fases o pasos recomendados en [10]:

A. Estrategia de Búsqueda y Fuentes de Información

La cadena de búsqueda utilizada en esta investigación se construyó de acuerdo a las siguientes pautas:

- Derivar los términos principales basado en los tópicos a ser investigados
- Determinar e incluir sinónimos y términos relacionados
- Combinar el orden de los términos de búsqueda

De acuerdo a estas estrategias se construyó la cadena de búsqueda, quedando de la siguiente manera:

<<software product line AND variability AND (quality attributes OR quality requirements OR non-functional requirements) >>

Nótese que los términos "quality attributes", "quality requirements" y "non-functional requirements" con frecuencia son usados indistintamente en la literatura.

Entre las Fuentes consultadas se pueden citar: IEEExplore, ACM Digital Library, Citeseer Library, Google Scholar. Además de los artículos conocidos y referenciados en las revisiones de [14][15]. También se consultaron como fuentes las actas "in extenso" de los congresos internacionales RCIS (2010) (Proceedings of the International Conference on Research Challenges in Information Science) y VaMoS (2010) (International Workshop on Variability Modelling of Software-intensive Systems).

B. Selección de Estudios

Se encontraron 401 artículos entre todas las fuentes mencionadas. Luego se removieron los duplicados. En una primera etapa de selección, revisando los resúmenes en caso de no tener el texto completo, sólo se incluyeron los que trataban el manejo de la variabilidad en LPS, considerando atributos o requisitos de calidad o requisitos no funcionales, como se especificó en la pregunta de búsqueda, resultando 37 trabajos.

C. Extracción de Información y Síntesis

Estos 37 trabajos fueron analizados detalladamente y sólo se seleccionaron 10 trabajos, ya que planteaban explícitamente un método, enfoque o técnica donde se aborda la variabilidad no funcional en LPS [12][15][16][17][18][19][20][21][22][23], constituyendo estos 10 trabajos un marco común de análisis.

III. Variabilidad de Requisitos no Funcionales

En los 10 trabajos seleccionados se encontraron diversos enfoques para análisis del dominio que consideran la variabilidad, donde se mencionan o toman en cuenta explícitamente los requisitos de calidad. Estos enfoques que contemplan técnicas, modelos y métodos, fueron seleccionados por su completitud, en cuanto a definición y organización de sus procesos, y buena documentación de los artefactos que se generan, además de ser bien conocidos por la comunidad científica del área. Gran parte de ellos coinciden en utilizar el Grafo de Interdependencias de "Softgoals", del inglés SIG ("Softgoals Interdependency Graph") [24] como herramienta para integrar RF y RNF, en el modelo de características y de variabilidad.

En lo que sigue, se hará una revisión detallada de los enfoques: Definiciones jerárquicas [16]; Modelo de redes bayesianas (BBN, "Bayesian Belief Network") [17][33][34]; Modelo basado en metas [18][25]; F-SIG ("Feature-Softgoal Interdependency Graph") [19]; COVAMOF ("ConIPF Variability Modeling Framework") [20]; Svamp ("Software Variability Modeling Practices") [21]; IQ-SPLE ("Integrated Quality Software Product Line Engineering") [22]; Método de análisis de RNF orientado a características para LPS [23]; Modelo de características extendidas EFM ("Extended Feature Model") [15], y finalmente el Método de Gurses [12]. Es de hacer notar que sólo se mostrarán los gráficos que están

estrechamente relacionados con la propuesta que se realiza en esta investigación.

A. Definiciones Jerárquicas

Esta propuesta [16] consiste en la definición de un modelo jerárquico para la ingeniería de requisitos, donde los requisitos para diferentes productos son representados en la misma jerarquía. A diferencia de otros métodos para especificar requisitos, este método analiza los requisitos de acuerdo a las categorías objetivos de diseño y decisiones de diseño, en lugar de enfatizar solamente los RF basados en las necesidades del cliente.

La jerarquía está estructurada como un árbol lógico "AND", en la cual los nodos superiores representan objetivos de diseño o RNF, tales como: controladores arquitecturales y atributos de calidad, que se supone que el sistema debe satisfacer. Los otros nodos representan decisiones de diseño o RF. Cada nodo en la definición jerárquica tiene una prioridad que refleja la importancia de él, con respecto a su padre.

Las prioridades representan productos específicos y están dados en forma de tuplas. Ellos son utilizados como mecanismo para describir productos en la definición jerárquica. Cuando hay un conflicto entre un objetivo y una decisión de diseño, se dice que el requisito es parcialmente satisfecho por la decisión de diseño.

Este método de decisiones jerárquicas ayuda a los diseñadores en el manejo de los requisitos y como interrelacionarlos, específicamente para resolver conflictos e inconsistencias entre decisiones de diseño y objetivos. También facilita la tarea de encontrar requisitos perdidos por medio de un recorrido en reverso en el árbol jerárquico. Además de esto, si un requisito es considerado ambiguo, es fácilmente re-definible agregando nuevos nodos de objetivos de diseño. Sin embargo, a medida que en este árbol se tenga mayor cantidad de productos, se incrementa la complejidad del mismo, pudiendo ser ineficiente.

B. Redes Bayesianas de Conocimiento (BBN)

Este enfoque [17][33][34] propone un método basado en aproximaciones para predecir la calidad en una línea de productos de software. El BBN ("Bayesian Belief Network") se basa en un modelo para determinar explícitamente el impacto de las variantes (decisiones de diseño especiales) sobre los atributos de calidad del sistema.

Este método contempla dos artefactos principales: modelo de características tipo FODA [26], el cual es utilizado para capturar los RF y el modelo BBN para capturar el impacto de las variantes funcionales sobre los atributos de calidad. Los atributos de calidad son representados como nodos en el modelo BBN, y son especificados con escalas. Por ejemplo: "alta" o "baja", y esas escalas son especificas para cada dominio, así de este modo en un dominio dado, la instancia "alta" referente por ejemplo al tiempo de respuesta de la característica eficiencia, es más o menos igual a 0,5 ms.

Se utilizan arcos dirigidos para relacionar una decisión de diseño o variante con un atributo de calidad. Probabilidades condicionales son utilizadas para cuantificar las relaciones conceptuales; un valor de probabilidad condicional sobre los arcos, refleja el conocimiento de los expertos del dominio referente a qué tanto una decisión de diseño influye en un atributo de calidad.

C. Modelo Basado en Metas (GORE)

En este enfoque [18], se propone la utilización de técnicas de la Ingeniería de Requisitos Orientada a Metas, del inglés "Goal-Oriented Requirements Engineering (GORE)" [27], aplicada al contexto LPS. Específicamente, se plantea un modelo basado en metas para representar la variabilidad, así como también se provee de una técnica visual de análisis de variabilidad. Este modelo es una versión restringida del modelo de metas del conocido "Framework NFR", del inglés "Non-Functional Requirements Framework" [24], el cual ofrece una aproximación sistemática para definir RNF, su trazabilidad y sus interdependencias. Además ayuda a los diseñadores a considerar las acciones necesarias para asegurar la calidad. El "Framework NFR" está a su vez basado en el SIG ("Softgoals Interdependency Graph").

Un SIG o grafo de interdependencias de "softgoals", es un diagrama propuesto para modelar RNF por medio de metas no funcionales o "softgoals", las cuales se cumplen en un cierto contexto de acción, por ejemplo Seguridad [Acceso a Cuentas] indica el nombre del "softgoal", "Seguridad", el cual debe ser satisfecho en el contexto de acción o componente funcional "Acceso a Cuentas" [18][27][28].

De acuerdo a lo observado, este método genera dos artefactos o sub-modelos: un modelo de metas funcionales ("hardgoals") y un modelo de metas no funcionales ("softgoals"). El primero representa objetivos funcionales y el segundo representa las condiciones o criterios que el sistema debe cumplir desde el punto de vista no funcional; las metas no funcionales se identifican específicamente con los RNF respectivos y requisitos de calidad, y se representan por el diagrama SIG. En [28], sólo se utiliza un SIG que contiene la información integrada de RF y RNF, lo cual facilita la trazabilidad de los requisitos de calidad.

La operacionalización de un "softgoal" es decodificada en el sub-modelo de metas funcionales como tareas o mecanismos que deben ser implementados para que el "softgoal" sea satisfecho. Ambos sub-modelos son árboles "AND" y "OR".

Las prioridades están dadas por cada "softgoal" en una escala percentil y las correlaciones son utilizadas para representar relaciones entre las funcionalidades "hardgoals" y los "softgoals". Las correlaciones tienen diferentes etiquetas de influencia (--, -, ?, +, ++).

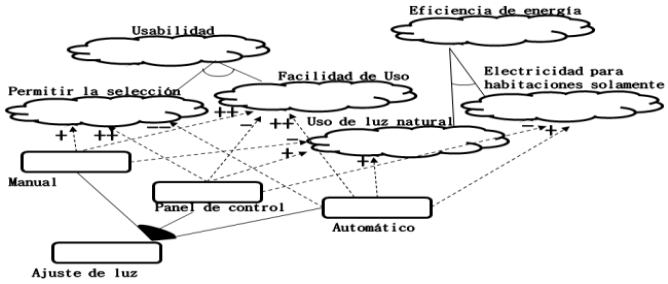

Figura 1: Combinación de Diagramas SIG con Metas Funcionales "Hardgoals" (Rectángulos Blancos) y no Funcionales "Softgoals" (Nubes Grises)

Estas etiquetas cualitativas son convertidas en valores cuantitativos, de la siguiente manera: El valor 1 para satisfactorio y otro valor para negación (ver Figura 1).

D. F-SIG (Grafo de Interdependencias entre "Features" y "Softgoals")

El objetivo principal de este método [19] es proveer un "framework" que permita representar decisiones de diseño en forma de interdependencias entre variantes funcionales y requisitos de calidad. Para ello se propone un nuevo modelo, que consiste en la unión de un modelo de características y un modelo de "softgoals", resultando en una extensión del FODA clásico [26], del inglés ("Feature-Oriented Domain Analysis"), al cual denominan F-SIG.

Es utilizado como método de análisis y modelado del dominio, que usa un modelo de características al estilo FODA para representar variabilidad. A su vez, en el F-SIG se consideran contribuciones explicitas e implícitas (donde una contribución es explicita cuando la selección de una variante influye directamente en un requisito de calidad, mientras es implícita cuando la selección de un requisito de calidad no está dado directamente por la selección de una variante funcional).

Al igual que el Modelo basado en metas [18], se tienen dos modelos FODA, el primero referente a funcionalidades de comportamiento del sistema y el segundo referente a decisiones de diseño, ambos relacionados con un modelo de "softgoals" (ver Figura 2).

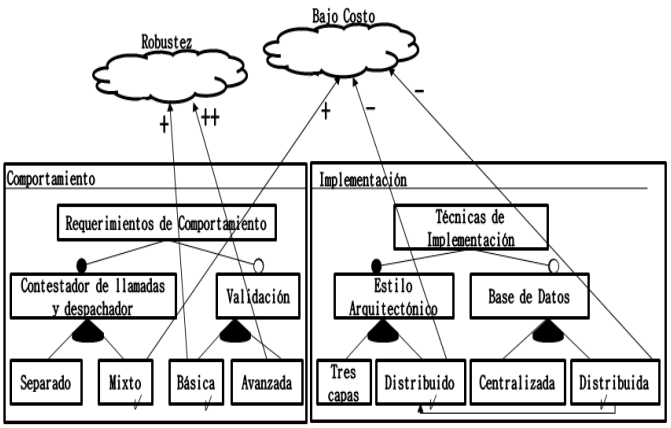

Figura 2: Ejemplo de un F-SIG que Muestra la Combinación de un SIG con Metas Funcional y no Funcional

E. COVAMOF

Es un "framework" para el modelado de la variabilidad en una familia de productos de software [20], contempla todas las capas de abstracción de la familia de productos, específicamente: - el modelo de características estilo FODA en el nivel más alto de abstracción que sería referente a las funcionalidades; - Diseño Arquitectónico en un nivel intermedio; e - Implementación de componentes en el nivel más bajo.

Este "framework" se basa en la utilización exhaustiva de puntos de variación y dependencias para capturar la variabilidad. Un punto de variación puede representar una funcionalidad, un artefacto o un valor de alguna variante de producto, mientras que una dependencia restringe la forma en que se seleccionan uno o más puntos de variación. Además proporciona un gráfico llamado CVV ("COVAMOF Variability View"), en el cual se refleja la variabilidad con las dos vistas de variación y de dependencia.

Los atributos de calidad pueden ser modelados con las dependencias, las cuales pueden especificar una propiedad que determina el valor de un atributo de calidad, tal como desempeño o utilización de la memoria. Aunque este "framework" no muestra claramente la manera de especificar y cuantificar el valor de dicho atributo.

F. Svamp ("Software Variability Modeling Practices")

Este enfoque [21] sintetiza y formaliza conceptos existentes en los enfoques de modelado de características tradicionales. A su vez, considera tanto el modelado funcional como la variabilidad de atributos de calidad en un nivel arquitectural. La especificación arquitectural se realiza por medio de diferentes puntos de vista. Específicamente, un modelo FODA y un modelo de punto de vista estructural especifican la estructura y las funcionalidades, así como también la variabilidad.

Adicionalmente, Svamp utiliza varios modelos para especificar una familia de productos de software: - un modelo de Kumbang [21], que consiste en modelos de puntos de vista estructurales y de características para representar las funcionalidades del sistema, expresados en UML 2.0, es decir su arquitectura; - un modelo de atributos de calidad que considera los requisitos de calidad asociados a los componentes en el punto de vista estructural del modelo de Kumbang; - un modelo de variabilidad de la calidad para expresar la variabilidad de esos atributos de calidad. Estos dos últimos modelos pertenecen a un perfil QA, del inglés "Quality Attribute profile". Cada uno de estos tres modelos es definido por una ontología, donde cada ontología provee un metamodelo para el modelado de los conceptos. Cada ontología del modelo de calidad define la dimensión técnica de un requisito de calidad. Por ejemplo, la ontología seguridad está definida por activos de seguridad, atributos, hilos, soluciones y métricas, mientras que la ontología fiabilidad define procesos, métodos, modelos y métricas. De este modo, cada ontología de calidad tiene asociados requisitos de calidad específicos.

Este enfoque provee una herramienta automatizada, desarrollada bajo una plataforma eclipse, que puede utilizarse para crear el modelo de Kumbang como instancia de una ontología. El usuario puede especificar las características funcionales de la familia de productos, los elementos arquitecturales y las restricciones entre ellos. La herramienta verifica que al menos una configuración válida pueda ser derivada del modelo. La herramienta tiene la desventaja de no ser interoperable, así como la carencia de poder transformar diferentes modelos de forma automática.

G. Método de Análisis de RNF Orientado a Características para LPS

Este método [23] se concentra en el análisis de variabilidad no funcional y en el modelo de decisiones orientado al "Framework NFR" [24]. Tiene como artefacto de entrada un modelo que combina un FODA para las características funcionales, con un SIG basado en el "Framework NFR" para los RNF. Como salida final se obtiene un modelo de decisión integrado de características funcionales estilo FODA y "softgoals".

El método consta de cuatro fases las cuales se describen a continuación:

1. Construcción del contexto

Se construye un modelo de características de contexto sobre el FODA inicial, el cual se lleva a cabo de acuerdo al dominio en estudio, del cual se derivan las incumbencias no-funcionales en forma de escenarios de ejecución, eventos, conceptos sociales, etc.

2. Identificación de la variabilidad no funcional

Se observan las variaciones no-funcionales del "framework NFR" con respecto al SIG inicial, soportándose en el modelo de contexto y en las plantillas o en inglés "templates NFR", esto es la codificación de un contexto genérico en el cual se deben satisfacer los RNF. Las variaciones no funcionales en el SIG pueden originarse de distintas maneras y para ello se realizan tres actividades de análisis de las cuales se puede obtener una perspectiva no funcional completa, incluyendo todos los "softgoals" necesarios en el dominio, su configuración de variabilidad, los niveles de "softgoals" candidatos y su operacionalización. Las actividades en cuestión son:

- Análisis de presencia de objetivos: se evalúa la presencia de "softgoals" de acuerdo a los "templates NFR" y a la información de contexto del dominio. Puede utilizarse para evaluar la existencia de "softgoals" en diferentes aplicaciones y también para proveer heurísticas para el análisis de variabilidad no funcional. Luego los "softgoals" no concernientes son removidos y los "softgoals" reservados son evaluados para convertirse en obligatorios u opcionales.
- Análisis de niveles NFR: se determina el nivel (bajo-medio-alto) para cada "softgoal". Un alto nivel siempre reemplaza uno bajo, pero si un "softgoal" de alto nivel está en conflicto con otro "softgoal", el de bajo nivel debe reservarse. El concepto de conflicto entre "softgoals", en inglés "trade-offs", en la siguiente etapa de operacionalización, es parte significante en el análisis de variabilidad.
- Análisis de operacionalización: se evalúan las posibles operacionalizaciones de acuerdo al nivel de los "softgoals" reservados considerando sus dependencias. El resultado puede ser: completamente satisfactorio, parcialmente satisfactorio o insatisfactorio. De acuerdo a esto, la configuración de la variabilidad de los niveles de los "softgoals" relacionados debe ser ajustada.

3. Integración de "softgoals"

Los "softgoals" identificados son incorporados al FODA inicial con variaciones desde una perspectiva no funcional. A su vez, se incorporan las operacionalizaciones de los "softgoals" relacionados. Las características funcionales se refinan de acuerdo a las incumbencias no funcionales que hayan sido adicionadas. De ser necesario, se añaden restricciones en los "softgoals" relacionados en el modelo de decisión.

4. Modelado de decisión orientado a "softgoals"

Finalmente, se obtiene un modelo de decisión que integra funcionalidades y "softgoals", donde se reflejan los conflictos entre "softgoals", dependencias, requisitos de entorno y restricciones.

Este método tiene como deficiencia que no considera la utilización de estándares de calidad, por lo cual puede presentarse deficiencias al momento de cuantificar y formalizar la especificación de RNF junto a los RF, imposibilitando a su vez, la refinación de RNF hasta llegar a elementos medibles.

H. Método IQSPLE

Este método [22][29], consta de ocho pasos, de los cuales los cuatro primeros que se refieren a procesos de ID, específicamente llevados a cabo en el espacio del problema y los cuatro últimos en la IA, en el espacio de la solución. Entre los modelos conocidos, este enfoque utiliza FODA [26], Estructuras jerárquicas, Modelo de casos de uso, Modelo de componentes y Modelo de calidad.

De los ocho pasos sólo se explicarán los cuatro primeros ya que corresponden a Procesos de ID, que es donde se ubica la presente investigación:

1. Análisis de requisitos

Se especifican las características comunes y variables. Al mismo tiempo se recopilan los requisitos de calidad. Se emplea el FODA tradicional

2. Modelado de características FODA y atributos de calidad

Además del FODA se genera de forma separada un árbol de calidad para modelar los atributos de calidad y sus valores típicos, para lo cual se emplea CVM, del inglés, "Compositional Variability Management". De esta manera, el árbol de calidad se puede utilizar para seleccionar los requisitos no funcionales. De igual forma, elementos particulares en el árbol se pueden relacionar con otros árboles (pudiendo inhabilitar la selección de alguna característica), y con modelos UML. En el caso, que se emplee una herramienta de soporte a UML; se utiliza un API para soportar la sincronización automática entre los modelos, haciendo que ciertas características de calidad desaparezcan si no tienen soporte en el modelo de solución

3. Modelado de artefactos de solución y especificación de la calidad

4. Por medio de un metamodelo basado en UML, los atributos de calidad y sus valores son adicionados a los modelos del espacio de la solución.

I. Modelo de Características Extendido EFM ("Extended Feature Model")

Este autor [15] al igual que F-SIG [19], COVAMOF [20], Svamp [21], BBN [17][33][34], IQSPLE [22] y Gurses [12], propone una extensión al modelo FODA clásico [26], relacionando las funcionalidades con respecto a los requisitos no funcionales que apliquen, aportando mejoras con respecto a la propuesta original, cuya notación es muy ambigua. En este enfoque los RNF son "equivalentes" a los atributos de calidad y son especificados como sub-características pudiendo ser medidos como un atributo previamente definido en el dominio.

De esta manera, los atributos de calidad pueden ser especificados por la relación entre uno o muchos con respecto a alguna funcionalidad. Los atributos de calidad pueden ser utilizados para reflejar información no funcional tal como: costo, velocidad, memoria RAM o tiempo de desarrollo que sean necesario para satisfacer las funcionalidades. Estos

atributos pueden contener valores de un rango específico pertenecientes a un dominio discreto o continuo (ej. Entero o real).

El EFM también puede ofrecer restricciones complejas referentes a RF y a RNF. Por otra parte, en [30] se propone una notación que se ha adoptado para ilustrar como una funcionalidad es decorada por atributos. De forma que las funcionalidades padre son decoradas con expresiones que son dependientes de los valores de los atributos de sus funcionalidades hijo (ver Figura 3).

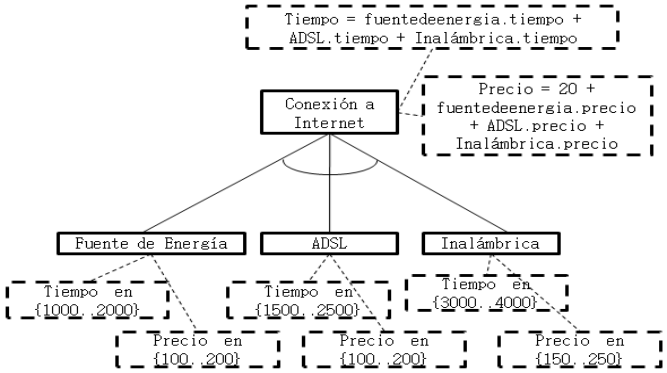

Figura 3: Ejemplo de un EFM

J. Método de Gurses

Este método [12] propone un proceso para construir un modelo de calidad complementario. Entre los modelos conocidos, también utiliza el modelo FODA clásico [26], el modelo EFM o FODA Extendido de [15], el "Framework NFR" de [24][23], el modelo F-SIG con la taxonomía de correlaciones de [19] para las interdependencias entre requisitos, y finalmente el estándar ISO/IEC 9126-1 (2001), incluyendo las métricas. Este método realiza el análisis del dominio y genera sus diferentes artefactos de la siguiente forma:

1. Se construye un diagrama de componentes del sistema de alto nivel (arquitectura del sistema).

2. Se genera un árbol FODA clásico por cada componente de la arquitectura, que contiene un grupo de características funcionales, que posiblemente son puntos de variación; las funcionalidades que no van a variar (comunes) para todos los productos no se colocan; se expresan las dependencias estructurales ("and", "or", "mandatory", etc.) entre las funcionalidades. Con lo que se obtiene básicamente un modelo de características funcional.

3. Cada funcionalidad de los árboles FODA es relacionada mediante una tabla utilizando las interdependencias implícitas "requires" o "excludes". Por ejemplo, la funcionalidad A "requires" B; si A es incluida, B también debe estar (A=>B); la funcionalidad A "excludes" B implica que A y B no pueden estar juntos, es decir A=> "NOT" B y B => "NOT" A. A<=>B se expresa como A=>B y B=>A.

4. Se hace un análisis cuantitativo con el modelo EFM [15], con etiquetas que contienen valores o rango de valores de restricciones impuestas a la funcionalidad.

5. Se realiza una unión disjunta de las etiquetas de las funcionalidades padres al estilo EFM. El modelo de calidad, que corresponde a un SIG, se obtiene relacionando las etiquetas compuestas de cada componente con las características de calidad de ISO/IEC 9126-1. Estas características de calidad son obligatorias para el producto (son etiquetadas como obligatorias en la terminología de FODA).

6. Se hace un análisis cualitativo con el "framework NFR", donde se construye un grafo SIG [24][23].

7. Se agregan al SIG los niveles "low", "medium", "high", como una escala de apreciación de la meta, al igual que [15] y luego se procede a hacer las operacionalizaciones.

8. Se construye el modelo de calidad complementario a partir del SIG con niveles cualitativos de apreciación ("low", "medium", "high").

9. Se hace una correspondencia entre las etiquetas compuestas del modelo cuantitativo y los objetivos del modelo cualitativo; se construye el árbol de ISO/IEC 9126-1 de forma descendente, del atributo de calidad a la funcionalidad; por ejemplo el atributo etiquetado "overall MTBF" corresponde a la sub-característica disponibilidad ("softgoal" en el SIG) y a la característica confiabilidad.

10. Se hace la correspondencia entre los rangos de valores especificados en las etiquetas del modelo cuantitativo y los niveles "low", "medium", "high" asociados a los "softgoals" del SIG.

11. Se relacionan las operacionalizaciones con las funcionalidades de donde provienen, en una tabla con "requires" y "exclude", construyendo un grafo similar al F-SIG [19].

12. Finalmente, se obtiene un modelo EFM con el modelo de calidad complementario.

IV. Comparación y Análisis entre los Enfoques, Técnicas y Métodos Estudiados

Para la comparación y evaluación de los diez métodos, técnicas y/o enfoques presentados en la sección anterior, se extenderá el "framework" propuesto en [14], el cual permite la comparación de métodos que modelan la variabilidad funcional y no funcional. Entre los aspectos generales tomados en cuenta para nuestra comparación, se tienen: los modelos para representar la variabilidad, el modelo de calidad para representar la variabilidad no funcional y el uso de estándares, la asociación entre características funcionales y no funcionales, y el empleo de herramientas automatizadas. A continuación se explica cada uno de los aspectos evaluados y su análisis de acuerdo a los métodos estudiados.

A. Modelos para Representar Variabilidad

De acuerdo a los resultados analizados, el modelo más utilizado para representar características funcionales es FODA [26], aunque en su definición original, las características pueden ser funcionales o no. Para representar requisitos de calidad los métodos más utilizados fueron el modelo SIG [27] y el modelo FODA Extendido (EFM) [15]. El enfoque IQ-SPLE [22], es muy expresivo al momento de representar la variabilidad, y a su vez está basado en "softgoals", por lo que es específico para caracterizar requisitos de calidad, ya que provee de estructuras especialmente pensadas para ese fin.

Sin embargo, es de resaltar la alta tendencia a realizar análisis de dependencias entre puntos de variación para especificar qué

requisitos no funcionales se ven implicados al seleccionar alguna variante de producto. Es decir que no está clara la trazabilidad entre los requisitos de calidad y los requisitos que los originan.

Una de nuestras proposiciones es que dentro del contexto de las LPS, un SIG pudiera servir de base para construir la arquitectura inicial del sistema, que después debe ser generalizada a una Arquitectura de Referencia; el modelo de calidad ISO/IEC 25010 puede ser utilizado para especificar el SIG.

Otro aspecto resaltante y en incremento es el énfasis que hacen los distintos métodos en la integración de modelos, principalmente con la finalidad de relacionar los RF con los RNF. Se ha observado que la trazabilidad entre los requisitos de calidad y su origen no está clara. A pesar de la importancia de considerar modelos de calidad en un proceso de producción industrial, solamente el enfoque de Gurses [12] y Svamp [21], utilizan y generan un modelo de calidad.

B. Modelo de Calidad para Representar la Variabilidad no Funcional y Uso de Estándares

La consideración de estándares ha tenido una baja aplicabilidad, en vista de que sólo Gurses [12] utiliza ISO/IEC 9126-1. A pesar que Svamp [21] genera entre sus artefactos un modelo de calidad, este no parece sustentado por ningún estándar.

A su vez, en el estudio de cada enfoque se determinó que la técnica de utilización de prioridades para requisitos de calidad ha tenido una aplicabilidad del 100%, lo cual puede ayudar (como se hace en [23]), a determinar los "softgoals" no concernientes que serán removidos y los "softgoals" reservados que serán evaluados para convertirse en obligatorio u opcional.

Por otro lado, se tiene que sólo IQ-SPLE [29] y Svamp [21] utilizan notación UML, lo cual resulta interesante de explotar en futuras propuestas, debido a su amplia aceptación a nivel de "stakeholders" y como lenguaje común en el área de la Ingeniería de Software.

Otro aspecto interesante que se puede extraer de este apartado de comparación es el análisis de niveles de variación de atributos de calidad, el cual tuvo una presencia de 50% entre los enfoques estudiados. Principalmente porque se encarga de determinar el nivel (bajo-medio-alto) para cada "softgoal". Donde un "softgoal" de alto nivel siempre reemplazará a uno de bajo nivel, pero si un "softgoal" de alto nivel está en conflicto con otro "softgoal" de bajo nivel, el de bajo nivel deberá conservarse.

C. Asociación entre Características Funcionales y no Funcionales

Se observa que la mayoría de los enfoques modelan características funcionales, mientras que una menor porción relaciona las características funcionales y no funcionales.

Por otra parte, el análisis cuantitativo tiene una consideración del 60% entre los enfoques, por la dificultad de asociar métricas a las características de calidad y funcionalidades a un alto nivel de abstracción [31]. Por lo cual sería conveniente considerar en propuestas futuras algún método o criterio que facilite dicha tarea a partir del conocimiento del dominio.

El análisis cuantitativo generalmente se representa con el modelo EFM [15]. Mientras que el análisis cualitativo (el cual tiene una consideración del 100% entre los enfoques) se hace apoyándose habitualmente en el modelo SIG.

Se observó la utilización de algunas técnicas, como la integración de modelos [22][23][29]. Adicionalmente, sólo IQ-SPLE [22][29], Gurses [12] y Svamp [21] proveen de una herramienta automatizada para generar modelos que de alguna forma representen la variabilidad.

Otro aspecto interesante, y que ha sido considerado en enfoques recientes es el análisis de la variabilidad composicional [22][23][29], en la cual además del FODA, se generan de forma separada un árbol de calidad para modelar los atributos de calidad y sus valores típicos, empleando CVM (Compositional Variability Management). De esta manera, el árbol de calidad se puede utilizar para seleccionar los requisitos no funcionales.

En nuestra propuesta, el árbol de calidad podría ser especificado según el estándar ISO/IEC 25010. La mayoría de los enfoques se concentran solamente en la ID, mientras que sólo COVAMOF [20], IQ-SPLE [29] y Gurses [12] consideran también la fase de IA y la relación entre ellas.

Pocos son los métodos que ofrecen un caso de estudio real completo en concordancia con sus etapas de realización. De igual manera, pocos son los métodos que poseen un modelo de proceso y conducen el proceso de forma sistemática y bien documentada; sólo un 30% de los métodos proveen de unos pasos claros para llevar a cabo el proceso de la variabilidad, como son [12][23][29].

Tabla I: Comparación entre Métodos de Modelado de Variabilidad que Consideran Requisitos de Calidad

Aspecto Evaluado / Técnica \ Método		GORE [18][27][28]	F-SIG [19]	COVAMOF [20]	Def. jerárquicas [16]	BBN [17]	EFM [15]	IQSPL [22]	Gurses [12]	Peng [23]	Svamp [21]
Modelos para representar la variabilidad	FODA [26]	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
	EFM [15]	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**
	SIG [27]	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
	Decisión [23]	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
	Ortogonal [8]	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**0**
	F-SIG [19]	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**1**
Modelo de Calidad para representar la variabilidad no funcional y uso de estándares	UML	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
	Estándares de calidad	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
	Modelo de calidad	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**
Asociación entre características funcionales y no funcionales	Modela características funcionales	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Relaciona características funcionales y no funcionales	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
Herramientas automatizadas		**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**

D. Herramientas Automatizadas

Se estudió la presencia de herramientas de soporte al método propuesto; se observó que sólo IQSPLE [29], Gurses [12] y

Svamp [21] poseen herramientas automatizadas de apoyo. Los aspectos antes discutidos pueden verse resumidos en la Tabla I.

En conclusión, los resultados del estudio fueron que sólo dos (2) trabajos utilizan el modelo de calidad y sólo uno (1) utiliza un estándar para el modelo de calidad, el ISO/IEC 9126-1 (2001), lo cual consolida nuestra propuesta de usar su actualización, el ISO/IEC 25010 [11] para unificar la terminología sobre calidad del producto software en diferentes dominios, como una de las buenas prácticas de la Ingeniería de Software.

V. Primera Propuesta

En vista del análisis efectuado, formulamos una primera propuesta para un proceso de diseño de un modelo de variabilidad para una AR que considera RF y RNF, en el cual se toma como base la combinación de algunos de los modelos estudiados: FODA Extendido (EFM) [15], el enfoque GORE [18][27][28] con el SIG en la notación de Supakkul y Chung [28] y tal como es utilizado por Gurses [12], además se utiliza una técnica análoga al F-SIG [19] para determinar los conflictos o "trade-offs" entre los requisitos de calidad; la especificación de los "softgoals" se realizará mediante el modelo de calidad estándar ISO/IEC 25010; la construcción de la AR se inspira en proceso descendente (en inglés "bottom-up") o reactivo, basado en la refactorización de las arquitecturas de productos existentes [32]:

- A partir del EFM [15] se construye el modelo SIG [28] representando los RNF o "softgoals", como características del modelo de calidad estándar y refinándolos hasta las sub-características y atributos. Los atributos, en este contexto, corresponden a la "realización" de operacionalizaciones (en inglés "operationalization"), que son componentes arquitectónicos. Este SIG puede "traducirse" a un diagrama de componentes en UML 2.0 y representa una Arquitectura Candidata (AC). La AC contiene un modelo de variabilidad con las variantes funcionales (por el EFM) y no funcionales por las operacionalizaciones del SIG que representan las propiedades de calidad (o RNF) asociadas a las funcionalidades.
- Finalmente, la AR se construye a partir de la AC, generalizando las variantes de la AC, que constituyen los puntos de variación del modelo de variabilidad.

De acuerdo a lo anterior, se propone un primer esbozo de proceso de diseño del modelo de variabilidad con RF y RNF para construir la AR de una LPS; el proceso propuesto establece como premisa: tener información sobre una familia de productos similares en cuanto a funcionalidades básicas y estilos arquitectónicos en un dominio dado. Y consta de siete fases que se detallan a continuación:

Fase I – Construcción del Contexto

Como punto de partida se construye un modelo de características extendido referente al contexto del dominio de estudio, obteniendo así un EFM inicial [15].

Nota: esta fase se inspira en el método seguido en el modelo EFM [15], considerando también un análisis de las similitudes entre componentes y conectores de los productos tomados en cuenta para el estudio.

Fase II – Análisis de Presencia de Objetivos

Se evalúa la presencia de "softgoals" de acuerdo a la información de contexto del EFM [15]. Los "softgoals" identificados se asocian a una característica del ISO-25010. Esto permite evaluar la existencia de "softgoals" en diferentes configuraciones.

Nota: esta fase se inspira del método de Gurses [12], quien sólo considera las características no comunes en el EFM. En nuestro caso se toman las comunes y no comunes.

Fase III – Identificación de la Variabilidad No Funcional

Se especifica por medio del SIG obtenido en el paso anterior del EFM, cada una de las características de calidad asociada a cada "softgoal", especificado como en [28], componente (contexto) y la lista de requisitos de calidad asociados.

Fase IV – Análisis de Prioridad

En esta fase las características de calidad del ISO-25010 no concernientes en cuanto a presentar conflictos o contradicciones son removidas y las características de calidad reservadas son evaluadas para convertirse en obligatoria u opcional, estableciendo prioridades (baja-media-alta) para cada característica. Una prioridad alta siempre reemplaza una baja, pero si una característica de alta prioridad esta en conflicto con otra de menor prioridad, esta ultima deberá reservarse.

Nota: Esta fase utiliza el enfoque F-SIG [19].

Fase V – Operacionalización de Características

Se evalúan las posibles operacionalizaciones de acuerdo a la prioridad de los "softgoals" reservados considerando sus dependencias. El resultado puede ser satisfactorio, regular, insatisfactorio. De acuerdo a esto la configuración de variabilidad de la prioridad de los "softgoals" deberá ser ajustada. Las operacionalizaciones o atributos de calidad resultantes serán los componentes arquitectónicos para la siguiente fase.

Nota: esta fase se inspira parcialmente en [32].

Fase VI – Integración de Componentes

El SIG obtenido se traduce en esta fase a un diagrama de componentes UML, representando así una AC. Esta arquitectura candidata se compone de un modelo de variabilidad con las variantes funcional EFM, y un modelo de variabilidad con las variantes no funcional representado por la Operacionalización del SIG, que representan las soluciones a las propiedades de calidad asociadas a las funcionalidades.

Nota: esta fase se inspira también en [32].

Fase VII – Modelo Arquitectónico de Referencia

Finalmente, se construye la AR a partir de la AC [32], incorporando como componentes la generalización de las variantes de la AC como puntos de variabilidad.

Nótese que la trazabilidad entre los modelos y artefactos obtenidos en las diferentes fases del proceso aquí planteado será considerada en la propuesta final del proceso completo de diseño, actualmente en desarrollo. Por otra parte debe observarse que las *Fases I* y *II* del proceso son relativas a capturar conocimiento sobre el dominio, el cual es necesario

para establecer el modelo de calidad y los estilos arquitectónicos más utilizados; en el proceso "bottom-up" este conocimiento puede ser extraído de los productos existentes estudiados utilizando técnicas de reingeniería, pero un análisis del dominio facilita esta tarea y por esto se han incluido como fases iniciales del proceso.

VI. CONCLUSIONES Y PERSPECTIVAS

De acuerdo a lo observado, la variabilidad de los requisitos de calidad está muy relacionada con la variabilidad de los requisitos funcionales y con el comportamiento global esperado del sistema; sin embargo esta trazabilidad es difícil de expresar y no se percibe claramente en muchos de los métodos estudiados. Al momento de especificar la variabilidad, es importante considerar un proceso sistemático que considere las mejores prácticas citadas anteriormente en los diferentes enfoques, tales como priorización y contribución entre funcionalidades, tratar explícitamente la variabilidad de los atributos de calidad y su trazabilidad en relación a las funcionalidades, profundizar estudios sobre análisis de "trade-offs" y la utilización de estándares para representar al modelo de variabilidad. Adicionalmente, no sólo se debe hacer análisis cualitativo, sino que se debe incluir en la especificación análisis cuantitativos, para lograr tener métricas tangibles y así asegurar la calidad de la producción industrial. Se pueden resaltar los métodos [12][23][29] por sus procesos bien documentados.

Finalmente, se ha presentado una primera propuesta, a partir del análisis realizado, que combina algunos de los enfoques estudiados, para obtener un modelo de variabilidad funcional y no funcional que considera la especificación de la calidad mediante el estándar ISO/IEC 25010 [11]. La formalización de esta primera propuesta está en vía de desarrollo; paralelamente se está definiendo un caso de estudio en el dominio de los sistemas de salud integrados con manejo de historias clínicas electrónicas para su aplicación y validación.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por los proyectos grupales DISofT No. 2011001343 del PEII-Fonacit; DesCCaP No. PG-03-8014-2011 y DARGRAF No. 03-8730-2013-1 del CDCH; además, por el Postgrado en Ciencias de la Computación, Facultad de Ciencias, Universidad Central de Venezuela y el Banco Central de Venezuela.

REFERENCIAS

[1] I. Sommerville, *Ingeniería de Software*. 9na Edición. Prentice Hall. 2011.

[2] K. Lee, K. Kang, and J. Lee, *Concepts and Guidelines of Feature Modeling for Product Line Software Engineering*, in proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools, ISBN 3-540-43483-6, pp. 62–77, 2002.

[3] E. Berard, *Essays in Object-Oriented Software Engineering*. Prentice Hall, New York, NY, USA, 1992.

[4] F. Losavio, A. Matteo e I. Pacilli, *Proceso Dirigido por Objetivos para Análisis de Dominio bajo Estándares de Calidad*. Enl@ce Revista Venezolana de Información, Tecnología y Conocimiento, vol. 6, no. 3, pp. 13–30, 2009.

[5] G. Perrouin, *A Metamodel-Based Classification of Variability Modeling and Software Product Lines*, VARY Workshop, 2011.

[6] E. de Almeida, A. Alvaro, D. Lucredio, V. Garcia, and S. de Lamos Meira, *A Survey on Software Reuse Processes*, in proceedings of the International Conference on Information Reuse and Integration. IRI-IEEE, pp. 66–71, 2005.

[7] A. Helferich, G. Herzwurm, and S. Schockert, *Developing Portfolios of Enterprise Applications using Software Product Lines*, in proceedings of the Conference on Component Oriented Enterprise Applications (COEA), pp. 71–85, Erfurt, Germany, 2005.

[8] K. Pohl, G. Bockle, and F. van der Linden, *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, 2005.

[9] L. Chen, M. Ali Babar, and N. Ali, *Variability Management in Software Product Lines: A Systematic Review*, Software Product Line Conference, pp. 81–90, Pittsburgh, PA, USA, 2009.

[10] B. Kitchenham and S. Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Keele University and University of Durham, Technical Report EBSE-2007–01, 2007.

[11] ISO/IEC FCD 25010, *Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Quality Model*. International Organization for Standardization (ISO), 2011.

[12] O. Gurses, *Non-Functional Variability Management by Complementary Quality Modeling in a Software Product Line*, MSc Thesis, Graduate School of Natural and Applied Science of Middle East Technical University, 2010.

[13] F. Losavio and A. Matteo, *Reference Architecture Design Using Domain Quality View*, Journal of Software Engineering and Methodology, vol. 31, no. 1, pp. 47–61, 2013.

[14] L. Etxeberria and G. Sagardui, *Modeling Variation in Quality Attributes*, University of Mondragon, 2007.

[15] D. Benavides, S. Segura, and A. Ruiz, *Automated Analysis of Feature Models 20 Years Later: A Literature Review*. University of Seville, 2010.

[16] J. Kuusela and J. Savolainen, *Requirement Engineering for Product Families*, in proceedings of the 22nd International Conference on Software Engineering (ICSE), pp. 61–69, New York, NY, USA, 2000.

[17] H. Zhang, S. Jarzabek, and B. Yang, *Quality Prediction and Assessment for Product Lines*, in proceedings of the 15th International Conference on Advanced Information Systems Engineering. Springer-Verlag. vol. 2681, pp. 681–695, 2003.

[18] B. González, J. Leite, and J. Mylopoulus, *Visual Variability Analysis for Goal Models*, in proceedings of the 12th IEEE International Requirements Engineering Conference, pp. 198–207, 2004.

[19] S. Jarzabek, B. Yang, and S. Yoeun, *Addressing Quality Attributes in Domain Analysis for Product Lines*. IEEE Proceedings Software, vol. 153, no. 2, pp. 61–73, 2006.

[20] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch, *Modeling Dependencies in Product Families with COVAMOF*, in proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, ISBN 0-7695-2546-6, pp. 298–307, 2006.

[21] M. Raatikainen, E. Niemela, V. Myllarniemi, and T. Mannisto, *Svamp, An Integrated Approach to Modeling Functional and Quality Variability*, in proceedings of the 2nd International Workshop on Variability Modeling of Software-Intensive Systems (VaMoS), pp. 89–96, 2008.

[22] R. Oberhauser, M. Medak, and J. Bartholdt, *Integrating Quality Modeling with Feature Modeling in Software Product Lines*. in proceedings of the Fourth International Conference on Software Engineering Advance, pp. 365–370, 2009.

[23] X. Peng, S. Lee, and W. Zhao, *Feature-Oriented Nonfunctional Requirements Analysis for Software Product Line*. Journal of Computer Science and Technology, vol. 24, no. 2, pp. 319–338, 2009.

[24] L. Chung, B. Nixon, E. Yu, J. Leite, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Springer, Reading, Massachusetts, 2000.

[25] B. González, M. Laguna y J. Sampaio, *Análisis de Variabilidad con Modelos de Objetivos*. VII Workshop on Requirements Engineering (WER-2004), pp. 77–87, 2004.

[26] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Tech. Report, CMU/SEI-90-TR-21, SEI (Carnegie Mellon), Pittsburgh, PA 15213, 1990.

[27] P. Giorgini, J. Mylopoulus, and R. Sebastiani, *Goal Oriented Requirements Analysis and Reasoning in the Tropos Methodology*. Engineering Applications of Artificial Intelligence. vol. 18, no. 2, pp. 159–171, 2005.

[28] S. Supakkul and L. Chung, *Integrating FRs and NFRs: A Use Case and Goal Driven Approach*, 2nd ICSE, pp. 30–37, 2004.

[29] J. Bartholdt, R. Oberhauser, M. Medak, and A. Rytina, *Integrating Quality Modeling in Software Product Lines*. International Journal on Advances in Software, vol. 3, no. 1 & 2, pp. 161–174, 2010.

[30] D. Streitferdt, M. Riebisch, and I. Philippow, *Details of Formalized Relations in Feature Models Using ocl*, in proceedings of 10th IEEE International Conference on Engineering of Computer-Based Systems (ECBS 2003). USA IEEE Computer Society, pp. 45–54, 2003.

[31] F. Losavio, L. Chirinos, N. Lévy, and A. Ramdane-Cherif, *Quality Characteristics for Software Architecture*. Journal of Object Technology, vol. 2, no. 2, pp. 133–150, 2003.

[32] F. Losavio, O. Ordaz, N. Levy, and A. Baiotto, *Graph Modeling of a Refactoring Process for Product Line Architecture Design*, in proceedings of the XXXIX Latin American Computing Conference (CLEI 2013), vol. 1, Naiguatá, Venezuela, October 2013.

[33] N. Siegmund, *Scalable Prediction of Non-Functional Properties in Software Product Lines: Footprint and Memory Consumption*, Information and Software Technology, vol. 55, no. 3, pp. 491–507, 2012.

[34] A. von Rhein, *Strategies for Product-Line Verification*, in proceedings of the 35th International Conference on Software Engineering. San Francisco, CA, USA, 2013.

[35] S. Sepulveda y C. Cachero, *Requerimientos No Funcionales en las Líneas de Producto de Software y el Modelado de Características: Una Propuesta*. International Workshop on Software Engineering IWASE, vol. 3, Curicço, Chile, 2011.