

# Proceso Extendido de Chung con Análisis del Dominio, Identificación de Aspectos y Estándares de Calidad

Francisca Losavio<sup>1</sup>, Alfredo Matteo<sup>1</sup>, Irma Pacilli<sup>1,2</sup>

<sup>1</sup>Laboratorio MoST, Centro ISYS, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela  
{francisca.losavio, alfredo.matteo}@ciens.ucv.ve, pacilli\_irma@hotmail.com

<sup>2</sup>Instituto Universitario de Tecnología Caripito-Monagas, Venezuela

## RESUMEN

La Ingeniería del Dominio (ID) considera la elicitación de requisitos para una familia de aplicaciones, identificando propiedades comunes y reutilizándolas en distintas aplicaciones. Chung y Supakkul (2004), proponen un proceso que integra Requisitos Funcionales (RF) y No Funcionales (RNF) en el modelo de casos de uso y llega a una arquitectura inicial utilizando el enfoque dirigido por objetivos de Yu y Mylopoulos (1998). Un proceso de análisis de dominio propuesto por Losavio, Matteo y Rahamut (2008) utiliza un modelo de calidad de acuerdo al estándar ISO/IEC 9126-1 para la especificación temprana de los RNF, pero no trata la identificación de las incumbencias transversales como posibles aspectos. El aporte principal de este trabajo consiste en definir un *Proceso Extendido de Chung* incorporando, por una parte un proceso de análisis del dominio (PAD-CAL) que considera estándares de calidad para la especificación de RNF, y por otra parte, la identificación temprana de incumbencias transversales, conocido como el enfoque “early-aspects. Para ello se actualiza el proceso de Losavio, Matteo y Pacilli (2009), con el nuevo estándar ISO/IEC 25010 y se utiliza el enfoque de Moreira, Brito y Araújo (2002), para la reestructuración de los RNF comunes y la identificación de los posibles candidatos a aspectos. El Proceso Extendido de Chung propuesto facilita la comunicación y la reutilización por el uso de estándares y puede ser fácilmente incorporado a un proceso de ID en un contexto de cadenas de producción de software, hacia una industrialización del procesos de desarrollo de software.

**Palabras clave:** ingeniería del dominio, análisis del dominio, estándar de calidad, ISO/IEC 25010, RNF, aspectos.

## ABSTRACT

Domain Engineering (DE) considers requirements elicitation for a family of applications, identifying common properties and reused in different applications. Chung and Supakkul (2004), proposes a process to integrate functional (FR) and non-functional (NFR) requirements in the use case model, to build an initial architecture using the Goal-Oriented approach of Yu and Mylopoulos (1998). A domain analysis process proposed by Losavio, Matteo and Rahamut (2008), using a quality model based on the ISO/IEC 9126-1 standard for the early specification of NFR; however the identification of crosscutting concerns is not treated. Our main contribution is the definition of an *Extended Chung Process*, including on one hand a domain analysis process (PAD-CAL) considering quality standards to specify NFR, and on the other hand, the early identification of crosscutting concerns, according to an early-aspects approach. For this, two issues are considered: an update of the domain analysis process of Losavio, Matteo and Pacilli (2009) with the new ISO/IEC 25010 standard (2011), and the approach of Moreira, Brito and Araújo (2002), to restructure common NFR and identify possible candidate aspects. The proposed process facilitate communication and reuse, due to the use of standards, and it can be easily incorporated to a DE process in the context of software product line, towards the industrialization of the software development process.

**Keywords:** domain engineering, domain analysis, product quality standard, ISO/IEC 25010, NFR, aspects.

## 1. Introducción

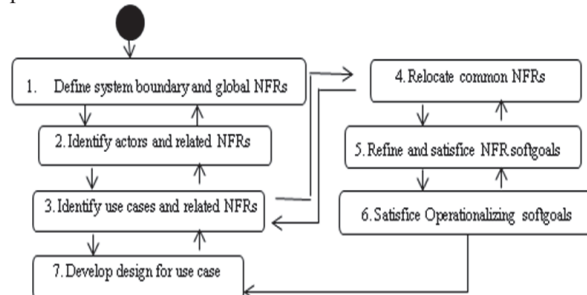
Para la Ingeniería de Software, la *calidad de un producto* debe ser considerada en etapas tempranas del proceso de desarrollo. En este contexto se define como el conjunto de características o propiedades deseadas que deben estar presentes en el producto, desde el punto de vista del usuario y del producto en si [ISO01]. Por lo tanto, todo sistema de software debe satisfacer estas características deseadas y una manera de validar parte de estos requisitos, a los que denominaremos *requisitos de calidad*, es considerar una *arquitectura documentada* del sistema como elemento central, en vista de que ésta debe responder a los requisitos exigidos por el sistema, tanto funcionales (RF), porque algunos componentes son introducidos para responder directamente a estos requisitos, como no funcionales (RNF) porque otros

componentes son introducidos para responder a objetivos de calidad precisos. Esto sólo se puede lograr siguiendo procesos que permitan verificar la satisfacción de los requisitos iniciales y reutilizar soluciones ya probadas. Una arquitectura documentada se refiere a que cada decisión arquitectónica está justificada sobre las bases de haber verificado que se cumplen los requisitos de calidad exigidos, como son las propiedades que caracterizan una solución arquitectónica, por ejemplo la interoperabilidad de los componentes. Los RNF se han asociado a los requisitos de calidad pero aún no se ha desarrollado un marco estándar, a pesar de los esfuerzos de [CNY\*00] con el framework NFR ó “Non Functional Requirements framework”, en el cual se puedan representar e integrar fácilmente al proceso de diseño de las aplicaciones tal y como ocurre con los RF, que se representan comúnmente a través del modelo de casos de uso de [BRJ99].

Hay enfoques [Lam01][YM98] en el proceso de desarrollo que consideran relacionar los RF con los requisitos de calidad a los cuales éstos responden. Existen propuestas donde se han logrado integrar RF y RNF en un modelo de casos de uso extendido, como se muestra en [MBA02][SSB\*04] y [CS04]; pero sin llegar a ser un estándar específico y comúnmente aceptado. En la práctica, dentro del contexto de un desarrollo basado en aspectos tempranos ó “early aspects” [MBA02][SSB\*04], el enfoque de Chung resulta prometedor y parece ser de fácil aplicación; es claro que esta afirmación se podrá comprobar en al medida en que avancen las investigaciones en el área. Recordamos que una *incumbencia* (“concern”) es cualquier propiedad considerada de interés para el sistema y una *incumbencia transversal* (“crosscutting concern”) es una incumbencia que entrecruza a varias funcionalidades. Un *aspecto* es el mecanismo que permite tratar a una incumbencia transversal. El problema de las incumbencias transversales es que, si no son identificadas y tratadas a tiempo, producen un código poco flexible que dificulta la evolución del sistema, contradiciendo el paradigma de la orientación a objetos.

El objetivo de este trabajo es extender el proceso propuesto por Chung y Supakkul [CS04], el cual denominaremos para simplificar *Proceso de Chung* (ver figura 1), para integrar RF y RNF en el modelo de casos de uso y bajo un enfoque orientado a objetivo ó “goal-oriented approach”. Chung y Supakkul utilizan para esta integración el framework NFR [CNY\*00].

Nuestra proposición, inspirada en [LMR08], especifica los requisitos de calidad asociados a RF y RNF de acuerdo al nuevo estándar ISO/IEC 25010 [ISO11], que considera la calidad del producto de software descrita por un modelo de calidad. Se actualiza así el trabajo de [LMP09], en el cual se usó el estándar ISO/IEC 9126-1 [ISO01], versión previa al ISO/IEC 25010.



**Figura 1:** Proceso de Chung. Diagrama de Actividad, tomado textualmente de [CS04].

Se considera la importancia de seguir estándares que permitan asegurar un lenguaje común para el entendimiento de todos los involucrados en el proyecto de software. En nuestro enfoque, la extensión al Proceso de Chung, al cual denominaremos *Proceso de Chung Extendido* (ver figura 4), consiste en la modificación del primer y cuarto paso del Proceso de Chung que se muestra en la figura 1. En el primer paso incluimos como etapa inicial del proceso, el análisis del dominio para la reutilización del conocimiento sobre la arquitectura y la identificación de restricciones o propiedades globales sobre una familia de aplicaciones del dominio. En el cuarto paso planteamos la identificación de las

incumbencias transversales existentes (siguiendo el enfoque denominado “early aspects”, donde se contempla que las incumbencias transversales se toman en cuenta a nivel de las primeras etapas del desarrollo del software). Un *dominio* [Ber92], es el conjunto mínimo de propiedades que definen una familia de problemas para los cuales se requieren soluciones computacionales. En este contexto un dominio se considera una familia de productos. El resultado de este análisis es el punto de partida para la justificación de los requisitos globales de la aplicación o sistema a ser construido; en el Proceso de Chung no se justifica la obtención de estos requisitos globales; es entonces complementado por nuestro enfoque con la modificación de las etapas antes mencionadas y con el uso de estándares [ISO11][LCM\*04], para la especificación de los RNF. Además, la especificación de RNF a través del modelo de calidad, facilita su identificación temprana, por lo cual se complementa el Proceso de Chung con técnicas de identificación de aspectos, según Moreira, Brito y Araujo [MBA02]. Esta investigación es parte del proyecto ADIRE Nro. PG-03-7310-2008/2 bajo los auspicios del Consejo de Desarrollo Científico (CDCH) y Humanístico de la Universidad Central de Venezuela.

Además de esta introducción y de la conclusión, el presente documento contiene una segunda sección donde se reseñan los estándares de calidad y la elección del modelo de calidad propuesto en el estándar ISO/IEC 25010 [ISO11]. En la tercera sección se presenta el enfoque orientado a objetivos. En la cuarta sección se describe el enfoque de Moreira, Brito y Araujo [MBA02]. En la quinta sección se describe el proceso propuesto PAD-CAL. En la sexta sección se describe el Proceso Extendido de Chung, considerando el framework NFR de análisis y diseño orientado a objetivos propuesto en el Proceso de Chung [CS04], extendido con el tratamiento de las incumbencias transversales de [MBA02] y PAD-CAL. Finalmente, en la séptima sección se ilustra la aplicación de la propuesta con un caso de estudio en el dominio de aplicaciones basadas en Servicios Web [W3C04].

## 2. Estándares de calidad del producto de software

Normalmente las propiedades o *requisitos de calidad* inherentes al producto a construir, son directamente definidas como RNF [ISO06], pero es bueno señalar que también las funcionalidades del sistema llevan implícitas la aplicación o cumplimiento de un requisito de calidad (objetivo de calidad), que corresponde a un RNF (funcionalidad implícita), como por ejemplo la *seguridad* exigida por la funcionalidad *control de acceso*. Existen varios modelos de calidad en la literatura como son los propuestos en [Dro94][BS99] y [ISO11] entre otros, para especificar la calidad de productos de software; en este documento utilizaremos el estándar ISO/IEC 25010 [ISO11], representado en el cuadro 1, donde se muestra la representación jerárquica de cada una de las ocho características principales de calidad del producto de software. El modelo puede ser refinado para incluir nuevas sub-sub características; es un modelo genérico que debe ser instanciado para un producto específico. Preferimos este estándar por pertenecer al grupo de normas ISO que tienen una amplia difusión y aceptación en la comunidad internacional. Este nuevo estándar, de reciente adopción

Adecuación funcional	Eficiencia	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Portabilidad
Complejidad	Comportamiento en Tiempo	Coexistencia	Ser reconocido como apropiado	Madurez	Confidencialidad	Modularidad	Adaptabilidad
Correctitud	Uso de Recursos	Interoperabilidad	Capacidad de ser aprendido	Disponibilidad	Integridad	Reutilización	Capacidad de ser instalado
Capacidad de ser apropiado			Capacidad de ser operado	Tolerancia a fallas	Capacidad de no ser rechazado	Capacidad de ser analizado	Capacidad de ser reemplazado
			Protección de errores del usuario	Capacidad de recuperarse	Capacidad de ser auditado	Capacidad de ser modificado	
			Estética de la Interfaz Usuario		Autenticidad	Capacidad de ser probado	
			Capacidad de ser de fácil acceso				

**Cuadro 1:** Características y Sub-características del modelo de Calidad ISO/IEC 25010 (traducción en castellano no oficial)

oficial, sustituye al ISO/IEC 9126-1. Establecer el modelo de calidad para el dominio de la aplicación ayuda a definir los *requisitos no funcionales globales*, válidos para todo el dominio, así como algunos requisitos de calidad para las funcionalidades del usuario comunes a la familia de aplicaciones del dominio y representan una vista de calidad del conocimiento del dominio.

**3. Enfoque Orientado a Objetivos**

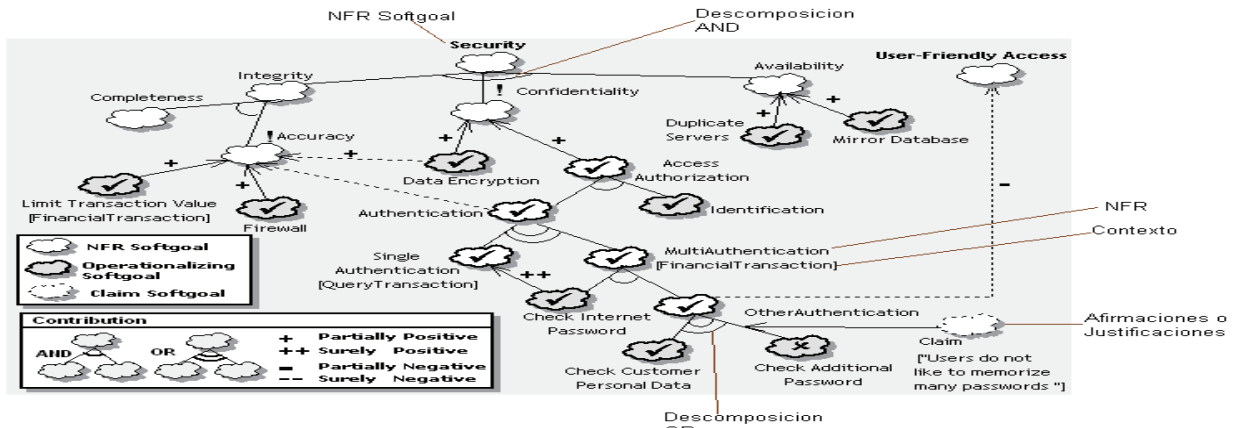
El enfoque orientado a objetivos de software, denominados en la literatura *objetivos blandos*, del inglés “softgoals” [DL93][Lam01][YM98], se traduce en la satisfacción de objetivos no funcionales, a través de las contribuciones positivas de las *operacionalizaciones*, que son actividades de bajo nivel, mecanismos o componentes de software, introducidos para la satisfacción de un objetivo requerido por los mismos. Es decir, los softgoals [CNY\*00], son objetivos difíciles de expresar ya que no son funcionalidades requeridas directamente por la interacción del usuario con el sistema; un ejemplo es la seguridad, ilustrada en la figura 2. Estos objetivos se descomponen y refinan en una estructura de árbol llamado el *Grafo de Interdependencia de Softgoals*, denominado en la literatura “Softgoal Interdependency Graph” (SIG) [CNY\*00], el cual es un grafo que recoge el criterio del

desarrollador sobre estos objetivos y que muestra la interdependencia entre ellos; se dividen en *subobjetivos* hasta alcanzar la operacionalización, que indica un componente de software que se introduce para satisfacer el objetivo, en un enfoque descendente o “topdown”. Luego, se procede desde el componente que operacionaliza el objetivo, hacia arriba ó “bottom up”, evaluando las *contribuciones* de cada rama del árbol; aquellas que tengan mayor peso positivo serán las que se deben ir logrando para satisfacer el objetivo principal de más alto nivel.

En la figura 2, se observa un ejemplo de un SIG donde se descompone el RNF *seguridad* ó “security”, indicando que se deben satisfacer los RNFs *integridad* ó “integrity”, *confidencialidad* ó “confidentiality” y *disponibilidad* ó “availability” para poder satisfacerlo.

**4. Integración de incumbencias transversales al modelo de casos de uso**

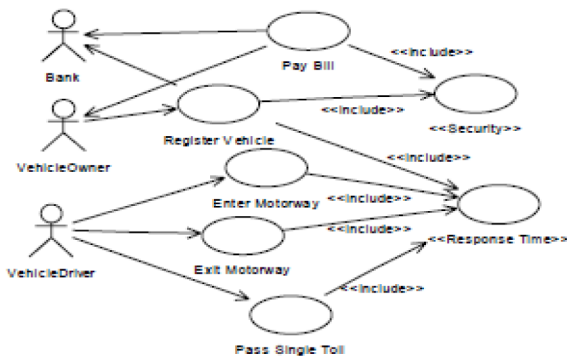
Moreira, Brito y Araujo [MBA02] plantean un modelo dentro de la ingeniería de requisitos, el cual considera los atributos de calidad transversales; para esto proponen un proceso compuesto de tres actividades principales: identificación, especificación e integración de requisitos. La primera actividad consiste en identificar todos los re-



**Figura 2:** Grafo de Interdependencia (SIG) del RNF seguridad ó “security”, tomado textualmente de [SSB\*04], con comentarios en castellano para mayor legibilidad.



requisitos de un sistema y seleccionar de ellos los atributos de calidad relevantes para el dominio de la aplicación y los stakeholders. La segunda actividad está dividida en dos partes: (1) especificar los requisitos funcionales utilizando un enfoque basado en casos de uso; (2) describir los atributos de calidad utilizando plantillas especiales e identificando los que atraviesan a los requisitos funcionales. La tercera actividad propone un conjunto de modelos para representar la integración transversal de los atributos de calidad y los RF. Nótese que en nuestro contexto, un atributo de calidad es un requisito de calidad, de acuerdo a la terminología del estándar SQuaRE [ISO06]. Las dos primeras actividades hacen uso de tablas para mostrar datos relevantes; pero en la tercera actividad la integración propuesta compone los atributos de calidad con los RF, para obtener todo el sistema. Ellos utilizan los diagramas UML para mostrar esta integración. En un nivel alto de abstracción, el diagrama de casos de uso puede ser extendido para incluir nuevos casos de uso estereotipados por cada atributo de calidad y hacer que los casos de uso iniciales incluyan los nuevos de carácter transversal (ver figura 3).



**Figura 3:** Diagrama integrado de RF y RNF, tomado textualmente de [MBA02].

### 5. Proceso de análisis del dominio con especificación de requisitos de calidad (PAD-CAL) para una familia

A continuación se presenta el proceso PAD-CAL con la identificación de RNF globales, válidos para toda la familia en consideración, el cual se incluirá como **1er. paso** del Proceso Extendido de Chung, que se muestra en gris en la figura 4.

**Entrada:** la descripción textual del problema, para el cual se requiere como solución un sistema o aplicación de software; identificar la familia y dar los *estilos o soluciones arquitecturales de alto nivel* para la familia.

- a) Definir una *taxonomía de las funcionalidades principales* para la familia: listar las funcionalidades
- b) Definir el modelo de calidad, utilizando el estándar ISO/IEC 25010.
  - b.1) se especifica la *calidad arquitectural del dominio*, utilizando los objetivos de calidad críticos de una solución arquitectural genérica o estilo para la familia del dominio
  - b.2) se especifica la *calidad funcional del dominio*, donde por cada funcionalidad del paso a, se asocian los requisitos de calidad, como objetivos

que deben cumplirse para lograr las funcionalidades.

**Salida:** modelo de calidad para la familia del dominio

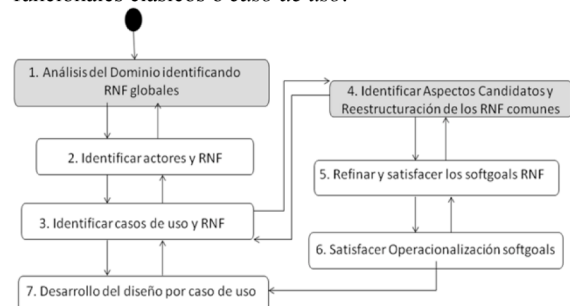
### 6. Proceso Extendido de Chung

En la sección 5 se describió PAD-CAL, el cual se incorpora como primer paso del Proceso de Chung [CS04], para la identificación de RNF globales.

Por otra parte, el Proceso de Chung definido en [CS04] propone utilizar el framework NFR, definido en [YM98] junto con el enfoque basado en escenarios o casos de uso, sobre las bases de dos premisas principales:

1. integrar los RNF a los diagramas de casos de uso, que expresan los requisitos funcionales principales respecto a los usuarios a través de los denominados *puntos de asociación* para clasificar los RNF y
2. establecer el alcance de reglas de propagación para relacionar las asociaciones propuestas con el modelo de casos de uso.

Los puntos de asociación se utilizan para realizar una taxonomía de los RNF y las relaciones de generalización/especialización son utilizadas para “propagar” el RNF, es decir incluirlo en el diagrama de casos de uso como una especialización del caso de uso base. Para los casos de uso por inclusión, denotados como <<include>> en el Lenguaje de Modelación Unificado, abreviado “UML” del inglés “Unified Modeling Language (OMG, del inglés Object Management Group [BRJ99]), el RNF es propagado también al caso de uso incluido. Esto es debido al hecho de que UML no dispone aún de elementos de modelación aceptados para los RNF. Sin embargo, en [MBA02] se propone considerar los RNF como casos de uso de inclusión ó <<include>>, como se explicó en la sección 4, lo cual nos parece atractivo, porque el RNF se expresa como un mecanismo que siempre deberá estar presente en la ejecución del caso de uso. Además, a partir de *tablas de composición* [MBA02] es fácil identificar las posibles incumbencias transversales en un contexto de diseño temprano de aspectos. Por lo tanto hemos incorporado este enfoque, descrito en [MBA02] en el paso 4, para la identificación de los aspectos candidatos. Tipos de Puntos de Asociación: se proponen cuatro asociaciones de los RNF relacionados con: el proceso de desarrollo del sistema denominados *globales*; entidades externas ó *actor*; de acceso, comunicación o intercambio de información denominados asociación *Actor-Caso de uso* y finalmente los requisitos funcionales clásicos ó *caso de uso*.



**Figura 4:** Proceso de Chung Extendido. Diagrama de Actividad integrando RF y RNF con análisis del dominio e identificación de aspectos candidatos.

Los tipos de RNF mencionados se aplican respectivamente en cuatro puntos del diagrama de casos de uso, específicamente en: la frontera del sistema, en el actor, en el caso de uso y en el enlace entre actor y el caso de uso. Estas asociaciones permiten una primera clasificación de requisitos no funcionales, a un alto nivel. Lo anteriormente descrito constituye el **4to. paso** del Proceso Extendido de Chung, que se muestra en gris en la figura 4. El diagrama de actividades ilustra un proceso iterativo e incremental a través del cual se integran los RF y RNFs, bajo las premisas planteadas.

La primera contribución del presente trabajo es integrar, en el primer paso del Proceso de Chung, denominado en [CS04] “*definir límites del sistema y requerimientos globales*” (ver figura 1), el proceso PAD-CAL descrito en la sección 5 (ver figura 4), utilizando el estándar de calidad ISO/IEC 25010 para la especificación de los requisitos de calidad asociados a los RNF, de tal manera de proporcionar una primera especificación y justificación razonada ó documentación, de los requisitos que afectan al dominio y a cualquier familia de aplicaciones que se encuentre dentro de éste. *Análisis del dominio* en este contexto implica identificar RF y RNF relativos a la familia de aplicaciones del dominio del problema a resolver. Para esta etapa, en [LMR08], se propone un proceso para una caracterización del dominio de aplicaciones basadas en servicios Web (WS), que será aplicado en la siguiente sección para el caso de estudio; este proceso es generalizado en PAD-CAL, para cualquier dominio y constituye en sí mismo un aporte importante de este trabajo. Cabe resaltar que el proceso original en [LMP09] utiliza el estándar ISO/IEC 9126-1 el cual hemos actualizados utilizando el nuevo estándar ISO/IEC 25010, recientemente adoptado por la comunidad internacional, como ya se destacó en la introducción.

## 7. Caso de Estudio: Sistema de Fijación de Precios (SFP) para el suministro de servicios en líneas aéreas

A continuación, se aplica el Proceso Extendido de Chung al problema del SFP.

### 7.1. Paso 1 - Análisis del Dominio (PAD-CAL)

Recalcamos que este paso, conformado ahora por el proceso PAD-CAL, es una propuesta original del presente trabajo, y no es considerado en el Proceso de Chung definido en [CS04].

**Entrada:** *Descripción del problema.* El problema aquí tratado como caso de estudio es propuesto en [CS04].

El SFP permitirá a las aerolíneas colaborar con sus proveedores a través de Internet para establecer los precios de los ítems de servicio que se ofrecen en los vuelos, tales como: leche, bebidas, comidas y artículos de limpieza. La aerolínea se encarga de mantener actualizado el sistema con los servicios que requiere. Cuando se crea o actualiza un servicio se le envía esa información a los proveedores, vía internet, para que ellos envíen su propuesta de precios. Si la propuesta es aceptada o rechazada, se le informa al proveedor y éste puede reenviar su propuesta hasta llegar a un acuerdo con respecto al precio del servicio.

**Identificación de la familia:** El SFP es un miembro de la familia de aplicaciones basadas en WS de tipo

*transaccional* (contratación de servicios vía Internet); una transacción se define como un conjunto de actividades agrupadas en una unidad; una transacción es cumplida como un todo, si una actividad falla, toda la transacción falla [W3C04].

**Estilo arquitectural para la familia de aplicaciones basadas en WS transaccionales:** según el Software Engineering Institute de Carnegie Mellon [Cms03], las Arquitecturas Orientadas a Servicios denotadas por “SOA” del inglés “Service Oriented Architecture” siguen estilo en capas, siguiendo un modelo de comunicación cliente/servidor.

#### a. Identificar funcionalidades principales de la familia:

Para el tipo de aplicaciones basadas en WS transaccionales, se identifican las siguientes funcionalidades básicas: *intercambio de datos*, *control de acceso* [LMR08].

#### b. Definir el Modelo de calidad – b.1 Especificar la calidad arquitectural para la familia de aplicaciones basadas en WS transaccionales:

la arquitectura SOA para estas aplicaciones debe responder a objetivos de calidad precisos. En [LMR08] se proponen: *compatibilidad* (*interoperabilidad*), *confiabilidad* (*disponibilidad*), *mantenibilidad* (*capacidad de ser modificado*), *portabilidad* (*adaptabilidad*); nótese que las sub-características se han colocado entre paréntesis a continuación de la característica principal de calidad, según el estándar ISO/IEC 25010. Debe observarse que SOA es parte del conocimiento del dominio; pueden existir otras soluciones arquitectónicas para ese dominio en particular. La elección de una solución particular debe obedecer a un proceso de selección basado en la experticia, es decir es una solución probada y aparece en un catálogo y en la satisfacción de sus requisitos de calidad, respecto a los requisitos de la aplicación particular. Este proceso de selección no es obvio y pertenece al área de investigación abierta de la evaluación de arquitecturas, donde se han realizado numerosos trabajos previos [LCM\*04].

#### b. Definir Modelo de calidad – b.2 Especificar la calidad funcional para la familia de aplicaciones basadas en WS transaccionales:

Para realizar *intercambio de datos* se requiere *seguridad*, *adecuación funcional* (*correctitud*) y *eficiencia* (*comportamiento en tiempo*) en la transmisión de datos, además de la *compatibilidad* (*interoperabilidad*) y *confiabilidad* (*disponibilidad*) por parte del servidor, que son aseguradas por la SOA. Para *control de acceso* se requiere *seguridad*.

**Salida:** el modelo de calidad para el dominio de aplicaciones basadas en WS transaccionales, cuyo representante es el SFP, está constituido por la unión disjunta de las características y sub-características de calidad definidas en b.1 y b.2, según el cuadro 1: *mantenibilidad* (*capacidad de ser modificado*), *eficiencia* (*comportamiento en tiempo*), *confiabilidad* (*disponibilidad*), *portabilidad* (*adaptabilidad*), además de las sub-características de funcionalidad *compatibilidad* (*interoperabilidad*), *adecuación funcional* (*correctitud*), y de la característica *seguridad*.



Los *RNF globales*, que deben ser cumplidos en general para todo el dominio y en consecuencia para la familia, se derivan directamente de este modelo de calidad del dominio, considerando los requisitos de calidad arquitecturales obtenidos en la actividad **b.1**, es decir: *compatibilidad (interoperabilidad)*, *confiabilidad (disponibilidad)*, *mantenibilidad (capacidad de ser modificado)*, *portabilidad (adaptabilidad)*. La calidad funcional, actividades **a** y **b.2**, será utilizada en el paso 3, para justificar los objetivos de calidad de los RF.

**7.2. Paso 2 - Identificar actores y RNF**

Los actores humanos que se pueden detectar de la descripción del problema son: *Proveedor*, *Gerente de Item de Servicio*, *Gerente de Aprobación de Propuesta*. El *Sistema de Facturación de Materiales* es también un actor, pero no humano. Un solo RNF, la *usabilidad*, es asociado a cada uno de los actores humanos que interactúan con el sistema mediante una interfaz-usuario, como se muestra en el cuadro 2.

Requisito Funcional (RF)	Calidad asociada (RNF)
a. Gestión de <i>item</i> de servicio	Usabilidad
b. Enviar requerimiento de propuesta (RP) (intercambio de datos)	Eficiencia (comportamiento en tiempo), seguridad, adecuación ( <i>correctitud</i> )
c. Enviar propuesta de precios (PP) (intercambio de datos)	Usabilidad, eficiencia (comportamiento en tiempo), seguridad, adecuación ( <i>correctitud</i> )
d. Aprobar propuesta de precios (PP) (intercambio de datos)	Usabilidad, eficiencia (comportamiento en tiempo), seguridad, adecuación ( <i>correctitud</i> )

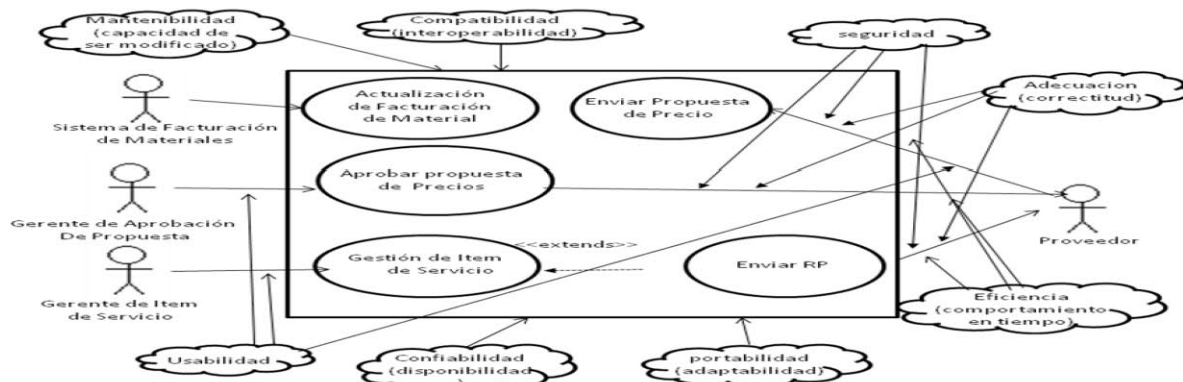
**Cuadro 2:** Lista de funcionalidades del SFP y sus requisitos de calidad.

**7.3. Paso 3 - Identificar casos de uso (RF) y RNF**

Con la ayuda del marco NFR [CNY\*00], de la descripción del problema y considerando las funcionalidades y sus propiedades de calidad, obtenidas en las actividades **a** y **b.2** del **paso 1**, se obtienen todas las funcionalidades del Sistema de Fijación de Precios, a las cuales corresponderán los casos de uso y los objetivos de calidad asociados, en la figura 5. La mayoría de los requisitos de calidad (RNF) son obtenidos observando la correspondencia con las funcionalidades principales detectadas en el **paso 1**, como se muestra en el cuadro 2.

La figura 5 muestra el modelo de casos de uso, todos los RNF asociados, además de los RNF globales del Sistema de Fijación de Precios.

**7.4. Paso 4 - Identificar Aspectos Candidatos y Reestructuración de los RNF comunes:**



**Figura 5:** Modelo de Caso de Uso para el SFP y los RNF Asociados.

Este paso, considerado en la propuesta inicial en [CS04], se ha modificado en este trabajo, siendo otra de las contribuciones aportadas, para indicar de manera explícita los RNF que pueden ser tratados como aspectos, es decir, “*candidatos a aspectos*”. Una de las premisas más importantes del enfoque orientado a aspectos es que si un RNF afecta a más de un caso de uso (funcionalidad), éste es considerado como una incumbencia transversal [MBA02] y es un posible candidato a ser tratado como un aspecto. Como se puede observar en la figura 5, existen RNF que afectan o intervienen en más de una funcionalidad (*seguridad, adecuación funcional, eficiencia y usabilidad*), por lo tanto podemos indicar que los mismos son candidatos a aspectos (ver cuadro 3). Bajo el enfoque orientado a aspectos, la solución para las incumbencias transversales es el encapsulamiento de las mismas en módulos separados, conocidos como aspectos, los cuales facilitan el mantenimiento y disminuyen el enmarañamiento del código.

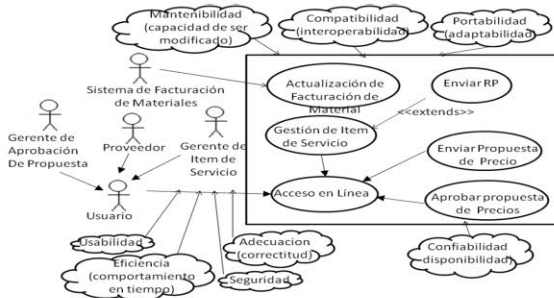
RNF común (Candidatos a Aspectos)	Funcionalidades afectadas
a. Usabilidad	Aprobar PP; Gestión de <i>item</i> de servicios; Enviar PP
b. Eficiencia (comportamiento en tiempo)	Enviar RP; Enviar PP; Aprobar PP
c. Seguridad	Enviar RP; Enviar PP; Aprobar PP
d. Adecuación ( <i>correctitud</i> )	Enviar RP; Enviar PP; Aprobar PP

**Cuadro 3:** Identificación de RNF candidatos a aspectos

El enfoque original en [CS04] no incluye el tratamiento de RNF como aspectos, pero plantea una solución enmarcada en la generalización/especialización reestructurando los RNF que se repiten y decide crear el caso de uso *Acceso en Línea* que generaliza *Gestión de Item de Servicio*, *Enviar PP* y *Aprobar PP*.

Nótese que el caso de uso *Acceso en Línea* implica tener *control de acceso*, funcionalidad detectada en la actividad **a**, y requiere seguridad según la actividad **b.1** (PAD-CAL, **paso 1**). Esta generalización afecta a los actores porque son diferentes para cada caso de uso, por lo tanto también se pueden generalizar los actores *Proveedor*, *Gerente de Item de Servicio* y *Gerente de Aprobación*, en un sólo actor llamado *Usuario* y asociar el RNF *usabilidad* a la asociación entre el actor *Usuario* y el caso de uso *Acceso en Línea*; cabe señalar, que por las reglas de propagación, esto implica que el RNF *usabilidad* se propaga a todos los casos de usos incluidos, tal como se observa en la figura 6. Por otra parte, también los RNF *comportamiento en tiempo*, *seguridad* y *correctitud* intervienen en todas las

transmisiones de datos: *Enviar RP*, *Enviar PP* y *Aprobar PP*; y por lo tanto se asocian ahora a la comunicación entre *Usuario* y *Acceso en Línea*, que es el punto de entrada de todos los actores al sistema.



**Figura 6:** Reestructuración de los RNF comunes en el Modelo de Casos de Uso del Sistema de Fijación de Precios (según el enfoque de Chung en [CS04]).

Se propone el uso de un gráfico combinando los enfoques propuestos en [CS04] y [MBA02], en el cual se reflejan los RNF globales y los RNF candidatos a aspectos como un caso de uso con una relación de <<include>>, tal como se puede observar en la figura 7.

**7.5. Paso 5 - Refinar y satisfacer los softgoals RNF**

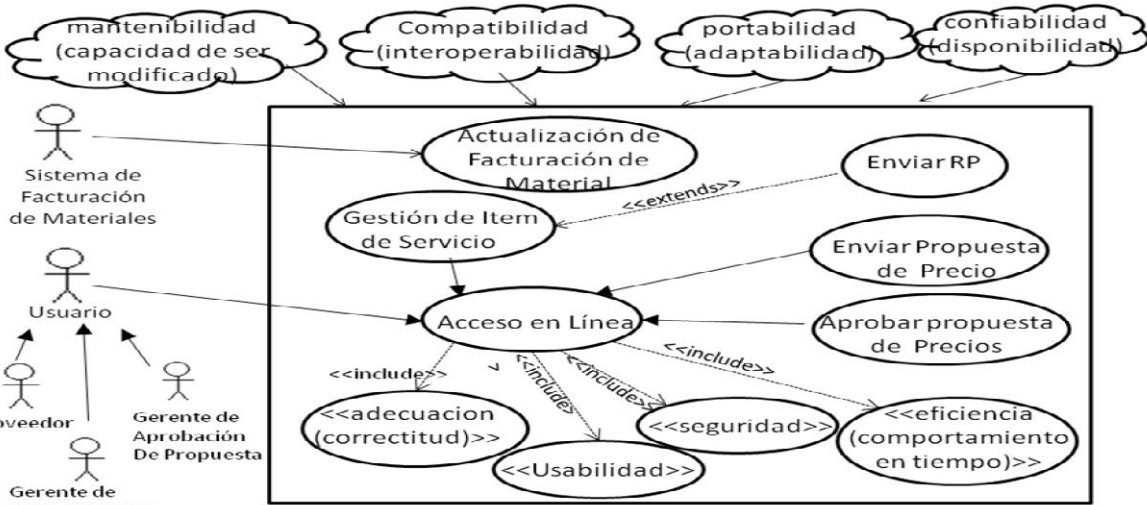
Se realiza un SIG por cada RNF asociado al Modelo de casos de uso. Los RNF globales se expresan en un SIG para la arquitectura; la figura 8, muestra el SIG para SOA determinada también en el análisis del dominio.

**7.6. Paso 6 - Satisfacer las operacionalizaciones de los softgoals**

A través de las operacionalizaciones en el Gráfico de Interdependencia para la arquitectura orientada a servicio (SOA), se pueden obtener los componentes de la arquitectura inicial o baseline para el Sistema de Fijación de precios, dentro de un enfoque de línea de producto.

**Contribución**

Se propone el Proceso Extendido de Chung como una



**Figura 7:** Reestructuración de los RNF comunes en el Modelo de Casos de Uso del SFP, según la combinación de los enfoques [CS04] y [MBA02], como resultado del paso 4 de Proceso Extendido de Chung.



**Figura 8:** Gráfico de Interdependencia de Objetivos (SIG) de SOA.

extensión al Proceso de Chung [CS04]. Por una parte, como una primera contribución, se incorpora en su primer paso, el proceso PAD-CAL para el análisis del dominio utilizando el estándar de calidad del producto ISO/IEC 25010 para especificar los RNF; se facilita la integración de los RNF y los RF en los puntos de asociación con los elementos del diagrama de casos de uso y la identificación de los RNF globales. El modelo de calidad como resultado del primer paso, facilita la trazabilidad de los RNF en los pasos subsiguientes. Por otra parte, como segunda contribución, se modifica el cuarto paso, combinando el enfoque del NFR framework [YM98] con el enfoque propuesto por Moreira, Brito y Araujo [MBA02], donde se identifican las incumbencias transversales y se determinan los RNF candidatos a aspectos. De esta manera se establece una visión más amplia, al combinar ambos enfoques y generar un gráfico que permite observar claramente los RNF globales, los cuales deben ser considerados en la arquitectura inicial.

**Conclusión**

La especificación de los requisitos de calidad por estándares internacionales [ISO11], comúnmente aceptados por la comunidad, facilita el entendimiento entre los grupos que desarrollan el proyecto de software, ofreciendo un lenguaje unificado sobre la calidad del software, del cual se carece.

Nuestro aporte principal, la extensión al proceso en [CS04], parece ser de fácil aplicabilidad en el contexto de métodos de diseño de líneas de productos de software y de



desarrollo de software centrados en la arquitectura, ofreciendo también un lenguaje unificado sobre la calidad del producto software. Por otra parte, definir un proceso general y reutilizable de análisis del dominio que pueda ser incorporado a métodos centrados en la arquitectura o de cadenas de producción, en un contexto de desarrollo de software orientado a aspectos, es un área abierta de investigación. Incorporar PAD-CAL para construir una arquitectura de referencia, por ejemplo en las fases de Inicio y Elaboración del Proceso Unificado, abreviado UP del inglés Unified Process [BRJ99], es una investigación en curso. Se han hechos trabajos que contemplan procesos con estándares de calidad y UP [LCMa\*04][CAU\*06], sin embargo no se considera allí un enfoque de análisis del dominio.

Otros trabajos en curso son, por una parte la comparación de enfoques similares, por ejemplo en [BM04] de integración de RNF y RF con escenarios con el enfoque orientado a objetivos y aplicar PAD-CAL a los enfoques que resulten más prometedores de la comparación; debe también experimentarse con un mayor número de casos de estudios realistas. Así mismo se está estudiando la especificación de la vista de calidad del conocimiento del dominio mediante ontologías [SJP\*07], para facilitar la integración de diferentes estándares, los cuales son utilizados en la práctica y la recuperación de las métricas correspondientes a los atributos de calidad.

#### Referencias

- [Ber92] Berard E. *Essays on Object-Oriented Software Engineering*, Prentice Hal, New Jersey, U.S.A.1992
- [BS99] Berghout E., Solingen R. *The Goal/Question/Metric Method (GQM) A practical method for quality improvement of software development*. London, McGraw Hill. 1999
- [BRJ99] Booch G., Rumbaugh, J. y Jacobson I. *The Unified Modeling Language User Guide*, Addison-Wesley, U.S.A. 1999
- [BM04] Brito I., Moreira A. *Integrating the NFR Framework in a RE Model*. Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design, UK. 2004,
- [Cms03] Carnegie Mellon. SEI (2003). *Client/Server Software Architectures-An Overview*.
- [CNY\*00] Chung L., Nixon B. A., Yu E., y Mylopoulos J. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers. 2000
- [CS04] Chung L., Supakkul S. *Integrating FRs and NFRs: A Use Case and Goal Driven Approach*. 2<sup>nd</sup> ICSE, pp 30-37. 2004
- [CAU\*06] Cooper K., Abraham S. P., Unnithan R. S., Chung L. y Courtney S. *Integrating Goal Models in the Rational Unified Process*. Journal of Visual Languages & Computing: 17(6), Dec. 2006, pp. 551-583. 2006
- [DL93] Dardenne A., Lamsweerde A. Von. *Goal-Directed Requirements Acquisition*. Science of Computer Programming, vol. 20, 179-190. 1993
- [Dro94] Dromey. *A Model for Software Product Quality*. 1994.
- [ISO01] ISO/IEC 9126-1. *International Organization for Standardization (ISO)*. Software engineering- Product quality - Part 1: Quality model. 2001
- [ISO06] ISO/IEC. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) 2006
- [ISO11] ISO/IEC 25010. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Systems and software quality models, ISO/IEC JTC1/SC7/WG6, 2011
- [Lam01] Lamsweerde A *Goal-Oriented Requirement Engineering: A Guided Tour*. 5<sup>th</sup> Int'l Symp. On RE, IEEE CS Press, pp. 249-261. 2001
- [LMP09] Losavio, F., Matteo, A. y Pacilli, I. Proceso dirigido por objetivos para análisis de dominio bajo estándares de calidad. Enl@ce Revista Venezolana de Información, Tecnología y Conocimiento, 6(3), 11-28. 2009
- [LMR08] Losavio F., Matteo A. y Rahamut R. *Web Services Domain Analysis Based on Quality Standards* 2<sup>nd</sup> ECSA, Cyprus, 2008, R. Morrison, D. Balasubramaniam, and K. Falkner (Eds.): ECSA 2008, LNCS Vol. 5292, 354-358 © Springer-Verlag Berlin
- [LCM\*04] Losavio F., Chirinos L., Matteo A. y Ramdane-Cherif A. *ISO quality standards for measuring architectures*. The Journal of Systems and Software (JSS). Vol. 72 (2), 209-223. 2004
- [LCMa\*04] Losavio F., Chirinos L., Matteo A., Lévy, N. y Ramdane-Cherif A. *Designing Quality Architecture: Incorporating ISO Standards into the Unified Process*. Information Systems Management (ISYM), Auerbach Publications, Vol. 21(1), 27-44. 2004
- [MBA02] Moreira A., Brito I. y Araújo J. *Crosscutting quality attributes for requirements engineering*. The 14<sup>th</sup> ICSE and SEKE'02, Ischia, Italy, ACM Press, pp 167-174. 2002
- [SJP\*07] Sancho P., Juiz C., Puigjaner R., Chung L. y Subramanian N., *An Approach to Ontology-aided Performance Engineering through NFR Framework*. Proc. WOSP'07, Buenos Aires, Argentina. ACM (Order No. 488073). Feb. 5-8, 2007. pp. 125-128
- [SSB\*04] Sousa G., Soares S., Borba P. y Castro J. *Separation of Crosscutting Concerns from Requirements to Design: Adapting a Use Case Driven Approach*. Early Aspects 2004: Workshop at AOSD, Lancaster UK pp 93-102.
- [W3C04] World Wide Web Consortium. *Web Services Architecture Requirements*. W3C Working Group Note 11 February. Copyright © 2004 W3C® Disponible en <http://www.w3.org/TR/wsa-reqs>