

Descubriendo la Arquitectura de Sistemas de Software Híbridos: Un Enfoque Basado en Modelos i^*

Juan Pablo Carvallo
Universidad del Azuay
Av. 24 de Mayo 7-77 y Hernán Malo, Apartado 01.01.981 Cuenca, Ecuador
jpcarvallo@uazuay.edu.ec

Xavier Franch
Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3 (Campus Nord, C6) E-08034 Barcelona (Catalunya, Spain)
franch@lsi.upc.edu

Resumen

La mayoría de sistemas de software modernos se construyen integrando componentes de diversa naturaleza (comerciales, código libre, componentes legados, etc.), formando arquitecturas híbridas. La construcción de este tipo de sistemas se caracteriza por la adquisición de diversos componentes a proveedores externos a la organización que se integran con algún software hecho a medida. La correcta aplicación de este enfoque requiere la temprana identificación de los servicios que el sistema deberá brindar y su agrupación en dominios atómicos (los actores del sistema), que serán posteriormente reemplazados de una manera “oportunistista”, por los componentes más apropiados, independientemente de su naturaleza y origen. En este artículo abordamos este aspecto y proponemos un método basado en la utilización de modelos i^ , que permite identificar los actores del sistema y su estructuración en una arquitectura de partida. El método es ilustrado con un caso práctico en una empresa de telecomunicaciones.*

1. Introducción

La mayoría de sistemas de software modernos se construyen integrando componentes de software de diferente naturaleza y orígenes, en lo que se ha dado por llamar *Sistemas de Arquitectura Híbrida* [1]. Los componentes de software utilizados en este enfoque arquitectónico incluyen componentes desarrollados por terceros, generalmente conocidos como componentes “Off-The-Shelf” (OTS) [2] (p.e., componentes comerciales (COTS) [3], componentes gratuitos y de código abierto (FOSS) [4] y servicios Web [5]), así como software desarrollado en casa o a la medida y sistemas legados.

En este tipo de enfoque, los sistemas son construidos de manera “oportunistista” [6], considerando simultáneamente el entorno y la estrategia organizacional, los componentes disponibles en el mercado, su capacidad para ser integrados en un sistema e interactuar de manera transparente a los usuarios, y los recursos necesarios para su adopción e integración.

La especificación de requisitos, la selección de los componentes requeridos, y su adaptación e integración en una arquitectura común, son algunos de los problemas que han sido ampliamente estudiados y documentados en la literatura [7, 8]. Sin embargo existen problemas más básicos que permanecen abiertos y requieren de mayor estudio por parte de la comunidad científica. Entre ellos se citan: la identificación de las necesidades estratégicas para los cuales el sistema es requerido en primer lugar; el descubrimiento de los servicios específicos (asociados a dichas necesidades) que el sistema deberá brindar; y la agrupación de los servicios relacionados en dominios atómicos, que estructuran la arquitectura genérica del sistema y describen la funcionalidad mínima que debe ser cubierta por cada uno de los componentes que lo integran.

Este artículo aborda estos temas y propone el método DHARMA basado en el uso de diagramas i^* [9], que permite identificar la arquitectura de un sistema basado en componentes. Los componentes genéricos que estructuran esta arquitectura, pueden ser luego reemplazados de manera oportunistista por componentes de diversa naturaleza y orígenes formando un sistema híbrido. Este método ha sido probado en diversos casos prácticos, incluyendo la definición de la arquitectura de los sistemas informáticos de la empresa de telecomunicaciones, que es utilizado como caso de estudio en este artículo.

Este artículo está estructurado en 5 secciones además de la introducción: la sección 2 presenta una breve

introducción al modelado en *i**; la sección 3 introduce el caso de estudio; la sección 4 incluye un descripción general del método DHARMA; las sección 5 detalla cada una de las actividades principales del método; y la sección 6 presenta las conclusiones y líneas de trabajo futuro.

2. Introducción al modelado en *i**

El objetivo del modelado en *i** es razonar acerca del entorno de una organización y sus sistemas de información. El marco *i** propone el uso de dos tipos de modelos, que corresponden a un distinto nivel de abstracción: los modelos de *Dependencias Estratégicas* (SD por sus siglas en inglés) representan el nivel intencional y los modelos de *Racional Estratégico* (SR por sus siglas en inglés) representan el nivel racional.

Un modelo SD consiste de un grupo de nodos que representan *actores* y un grupo de dependencias que representan las relaciones entre ellos, de tal forma que un actor (*dependier*) depende de algún otro (*dependee*) para satisfacer una *dependencia*. La dependencia es un elemento intencional que puede ser un *recurso* (requerido por los actores para alcanzar sus objetivos), una *tarea* (que pueden ser considerada una actividad que produce cambios en el mundo), un *objetivo* (equivalente a un requisito funcional) o un *objetivo blando* (que puede interpretarse como un requisito no funcional). También es posible definir la importancia (*fuerza*) de la dependencia para cada uno de los actores involucrados utilizando tres categorías: *abierta*, *comprometida* o *crítica*.

Un modelo SR por su parte, permite visualizar los elementos intencionales en la frontera de un actor a fin de refinar el modelo SD con capacidades de razonamiento.

Las dependencias del modelo SD están vinculadas a elementos intencionales dentro de la frontera del actor. Los elementos incluidos en el modelo SR se descomponen conforme a dos tipos de vínculos:

- Vínculos *medios-fin* que establecen que uno o más de los elementos intencionales son los medios que contribuyen a la consecución de un fin. Diversas variantes del lenguaje definen de distinta manera la naturaleza de los elementos conectados; así, aunque con frecuencia el medio es una tarea, otras variantes como Tropos [10] permiten descomposiciones objetivo-objetivo, entre otras.
- Vínculos *tarea-descomposición* que establecen la descomposición de una tarea en diferentes elementos intencionales. Existe una relación AND cuando una tarea se descompone en más de un elemento intencional.

La notación gráfica se presenta en la Fig. 1 en la que se ilustra la descomposición de un sistema de tutoría a estudiantes. A la izquierda presentamos el modelo SR de un tutor y las relaciones jerárquicas entre los elementos intencionales internos. A la derecha se presentan las dependencias SD entre un estudiante y el tutor.

Los actores se pueden especializar en *agentes*, *roles* y *posiciones*. Una posición cubre roles. Los agentes representan instancias particulares de personas, máquinas o software dentro de la organización y ocupan posiciones (y como consecuencia, juegan los roles cubiertos por esas posiciones). Los actores y sus especializaciones pueden ser descompuestos en otros actores utilizando la relación *es-parte-de*.

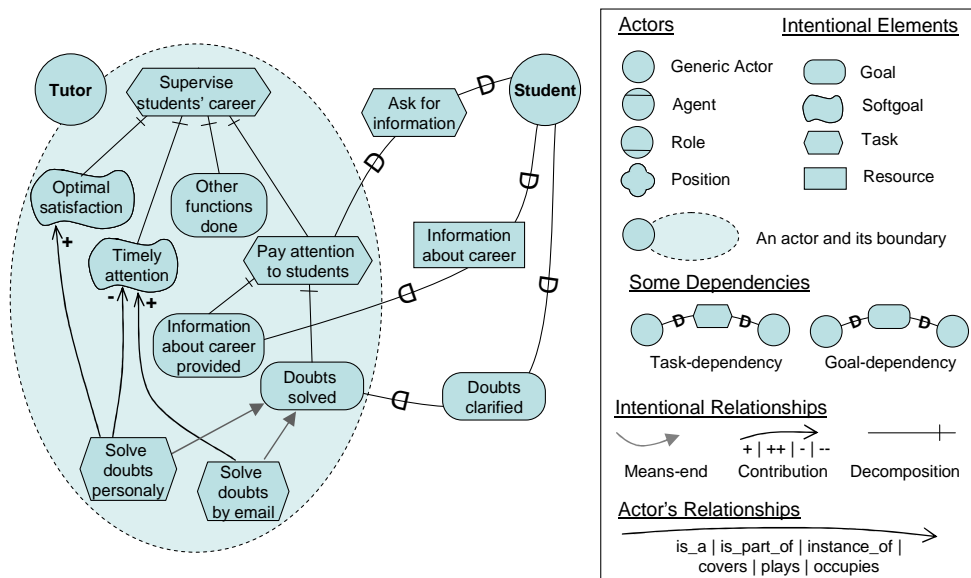


Figura 1. Ejemplo de un modelo *i** para un sistema de tutoría académica.

3. El caso de estudio

ETAPATELECOM S.A. es una empresa de telecomunicaciones fundada en el 2002 con base en la ciudad de Cuenca, Ecuador. Aunque de naturaleza municipal, la empresa cuenta con licencia para la provisión nacional de servicios de telefonía fija (pública y domiciliar), portadores de datos (enlaces locales y nacionales) y valor agregado (acceso a Internet). Si bien la empresa es de carácter privado, los fondos para su capitalización son de origen público provistos por el municipio que la constituyó.

Para cumplir con su estrategia de despliegue, ETAPATELECOM ha tenido que enfrentar la selección y adopción de varias tecnologías, incluyendo diversos componentes OTS requeridos para el sistema de información que soporta su operación. En este proceso, la empresa ha utilizado bajo diversas formas modelos de calidad [11] y técnicas de modelado usando i^* desde el año 2004 para soportar varias actividades relacionadas con la adopción y desarrollo de tecnologías de la información con resultados satisfactorios. En una de las variantes de uso, ambas líneas de trabajo fueron combinadas mediante el método COSTUME para la construcción de modelos de calidad para sistemas compuestos. COSTUME [12] consta de 4 etapas claramente diferenciadas (v. Fig. 2): análisis y modelado del entorno de la organización (v. (1) en Fig. 2); descomposición del sistema en dominios de componentes OTS (2); construcción de modelos de calidad individuales para los dominios (3); y construcción final del modelo de calidad compuesto para el sistema (4).

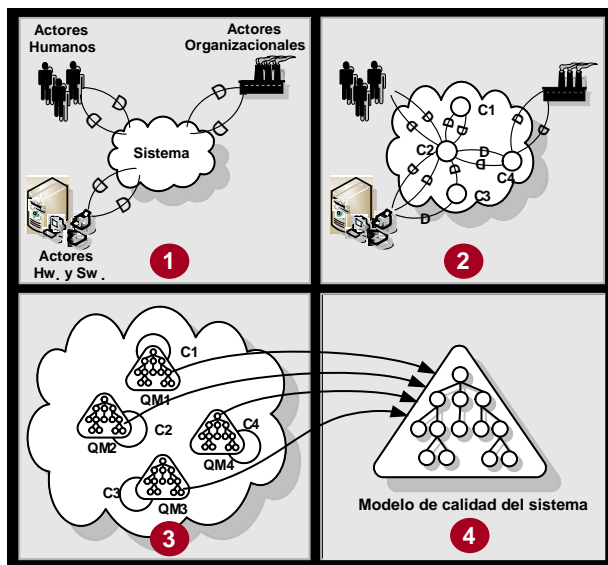


Figura 2. Actividades principales del método COSTUME.

4. Visión general del método DHARMA

El método DHARMA (abreviatura del inglés Discovering Hybrid ARchitectures by Modelling Actors) está basado en los principios del método COSTUME teniendo en cuenta la diferencia de objetivos: así como COSTUME pretende construir modelos de calidad para analizar el sistema, DHARMA se dirige a la definición de arquitecturas software. Como resultado, el proceso asociado al método DHARMA se inicia con el modelado del entorno de la organización y concluye con la identificación de la arquitectura genérica del sistema. Por arquitectura genérica entendemos la identificación de los actores que estructuran el sistema, los servicios que deben ser cubiertos por cada uno de ellos y las relaciones existentes entre los mismos.

El concepto de actor es pues central en el método DHARMA y por ello el marco i^* se revela altamente indicado. Los actores del sistema representan dominios atómicos para los cuales se pueden identificar componentes OTS. Por dominio atómico entendemos una agrupación de funciones que proporcionan un valor al usuario, de tal manera que ningún subconjunto propio de tal agrupación representa otro dominio significativo.

No hay que confundir la descomposición del sistema en actores, y la descomposición del sistema en componentes. El objetivo de este método no es el de identificar la arquitectura final del sistema, en la que los actores representan subsistemas directamente mapeables a componentes OTS individuales (aunque éste puede ser un caso particular), sino que pueden existir varios otros casos: un componente OTS que cubre los servicios de más de un actor; un actor cubierto por un solo componente; un actor cubierto por más de un componente, por razones de supervivencia y disponibilidad; o incluso algunos cuyos servicios no pueden ser cubiertos por componentes existentes, requiriendo de algún desarrollo a la medida.

El método ha sido concebido como cuatro actividades básicas que pueden ser iteradas o intercaladas (v. Fig. 3):

- **Actividad 1: Modelado del entorno de la organización.** La organización y su modelo de negocio son estudiados en detalle, a fin de identificar el rol que juega en relación a su entorno. Este análisis hace evidentes los diversos tipos de actores en su entorno y las necesidades estratégicas existentes entre ellos y la organización. Los modelos SD de i^* son utilizados para soportar el razonamiento y representar los resultados de esta actividad.
- **Actividad 2: Modelado del entorno del sistema.** En esta actividad se propone la introducción de un sistema en la organización (el sistema puede ser un sistema de información puro o un híbrido que incluya componentes de hardware, software o hardware con

software empotrado) y se analiza el impacto que éste tendría en relación a los elementos en el entorno de la misma. Las dependencias estratégicas identificadas en la actividad anterior son analizadas en detalle con el objeto de terminar cuáles pueden ser satisfechas directamente por el sistema, y cuáles son necesarias para que este mantenga su operación. Estas dependencias son redireccionadas y modeladas junto a los actores asociados en un diagrama SD del entorno del sistema. Este modelo incluye también a la organización como un actor en el entorno del sistema, en el que sus necesidades son modeladas como dependencias estratégicas sobre el mismo.

- **Actividad 3: Descomposición de los objetivos del sistema.** En esta actividad el sistema es analizado en detalle y descompuesto en una jerarquía de objetivos necesarios para satisfacer las dependencias estratégicas con su entorno. Los objetivos representan los servicios que el sistema debe proveer, para interactuar con los actores en su entorno. Un diagrama SR del sistema es construido utilizando descomposiciones medio – fin de tipo objetivo – objetivo (representado así una descomposición de objetivos en subobjetivos).
- **Actividad 4: Identificación de actores del sistema.** Los objetivos incluidos en el modelo SR de sistema son analizados y agrupados en actores que representan dominios atómicos. Los objetivos son asociados en grupos de servicios bien definidos, en base a un análisis de las dependencias estratégicas con el entorno y una exploración del mercado de componentes existentes. Las relaciones entre los diferentes actores que estructuran la arquitectura base del sistema, son descritas en base a la dirección de las asociaciones medio-fin existentes entre los objetivos incluidos en cada uno de ellos.

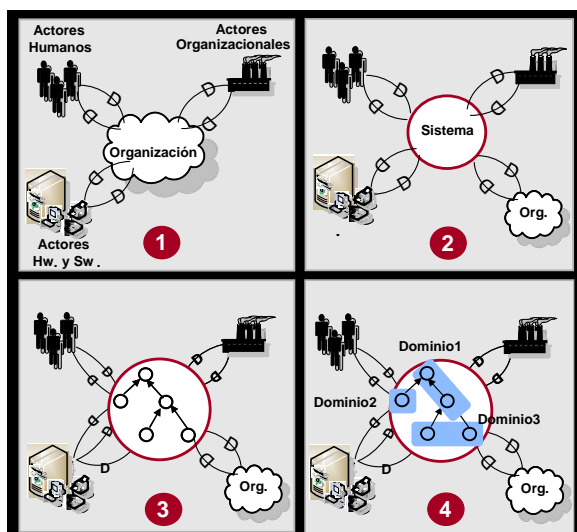


Figura 3. Actividades principales del método DHARMA.

Tabla 1. Ejemplos de patrones organizacionales.

Nombre	Definición
Plena Disponibilidad	Problema: El sistema debe ofrecer plena disponibilidad
	Consecuencias: + fault tolerance; + recoverability
Fácil Administración	Problema: El sistema debe ser fácil de administrar
	Consecuencias: + operability

En todo el proceso se sugiere el uso de artefactos que promueven la reusabilidad. En concreto, destacamos:

- Uso de patrones organizacionales [13]. Sugieren pedazos de modelos i^* adecuados para resolver un problema determinado, y muestran sus consecuencias respecto propiedades no-funcionales (v. Tabla 1 para ejemplo).
- Uso de taxonomías de dominios software [14]. Contienen dominios software ya identificados y jerarquizados (v. Fig. 4 para ejemplo).

La siguiente sección detalla cada una de las actividades del método y provee ejemplos concretos en relación al caso de estudio introducido en la Sección 3.

5. El método DHARMA en detalle

5.1 Modelado del entorno de la organización

El modelado del entorno de una organización requiere abordar al menos dos sub-actividades, la identificación de los actores de entorno y la identificación de sus dependencias con la organización.

Identificación de actores del entorno: Al estudiar el entorno de una organización varios actores de diversa naturaleza pueden ser identificados. Según nuestra experiencia los actores de entorno pueden ser al menos de cuatro tipos:

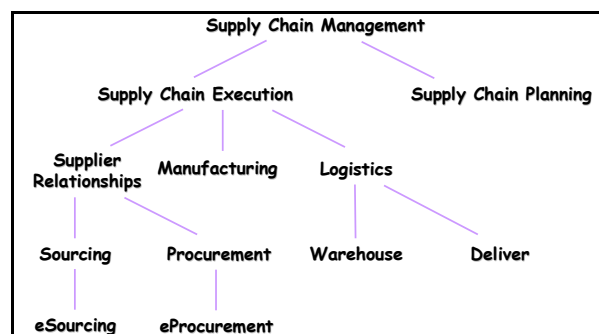


Figura 4. Extracto de una taxonomía para Business Applications.

- *Personas*, que representan diferentes tipos de usuarios de los servicios proporcionados por la organización (o por sus sistemas de información). P.e., administradores de sistema, usuarios finales, etc.
- *Organizaciones*, que representan organizaciones externas con sus propios objetivos sobre la organización o que le proveen servicios. P.e., organismos de control, socios similares, etc.
- *Hardware*, que representan dispositivos físicos que son gobernados por o proveen datos a la organización (o sus sistemas de información). P.e., escáneres, lectores de código de barra, firewalls, impresoras, etc.
- *Software*, que representan sistemas de software que proveen o reciben datos de la organización (o sus sistemas de información). P.e., servidores de correo, portales web, etc. Estos sistemas se consideran externos al sistema híbrido bajo análisis.

Existen varias técnicas descritas en la literatura que pueden ser utilizadas para facilitar la identificación de los actores en el entorno de una organización, entre ellas podemos citar: diagramas de caso de uso; técnicas de modelado de objetivos como las descritas en [15]; identificación de roles organizacionales soportada en la revisión de ontologías [16], estándares de cuerpos profesionales [17], o literatura sobre la teoría organizacional [18]; o la adopción de patrones sociales como los descritos en [19].

En el caso de ETAPATELECOM diversos actores fueron identificados de partida utilizando modelos específicos para la industria, p.e. eTOM [20], y posteriormente completados en talleres con un equipo multidisciplinario de funcionarios de la empresa, que incluyo representantes de las áreas financiera, administrativa, legal, comercial, técnica e informática. La naturaleza iterativa del método y las características propias de *i**, cuya naturaleza gráfica facilita la participación de usuarios no técnicos, fueron considerados factores de éxito en esta actividad. Un extracto de los actores identificados para en el entorno de la organización visualizado en la Fig. 5 se describen en la Tabla 2.

Identificación de dependencias: Una vez identificados los actores en el entorno de la organización, las dependencias entre ellos pueden ser graficadas utilizando modelos SD. La identificación de las dependencias puede ser una actividad laboriosa y el modelo resultante puede ser difícil de manejar debido a la notación gráfica utilizada en los modelos *i**. Por esta razón y para simplificar esta tarea proponemos utilizar las siguientes guías metodológicas:

Tabla 2: Extracto de actores en el entorno de ETAPATELECOM

Actor	Tipo	Descripción
Secretaría Nacional de Comunicaciones SENATEL	Org.	► Proveer licencias de operación ► Emisión de regulaciones y reglamentos
Organismo de control SUPERTEL	Org.	► Supervisar la calidad del servicio y el cumplimiento regulatorio
Proveedor de interconexión	Org.	► Proveer acceso a destinos fuera del área de cobertura de la organización (ej. Internacionales) ► Proveer transporte de datos fuera de la zona de cobertura de las redes del operador
Organización de recaudación	Org.	Cobrar planillas de servicios e impuestos
Proveedor de Directorio telefónico	Org.	Proveer listado o libro de números telefónicos
Distribuidor	Org.	Comercializar los servicios que la empresa provee
► Vendedor de servicios prepago	Org.	► Vender servicios prepago ► Proveer soporte a los usuarios
Usuarios	Persona	Utilizar servicios de telefonía y valor agregado
► Telefonía fija domiciliar	Persona	Utilizar servicios desde el hogar / oficina
► Telefonía pública	Persona	Utilizar servicios desde teléfonos públicos
► Franquicias telecentros	Org.	Revender servicios de telefonía y valor agregado

- Primero se deben identificar los objetivos para los cuales los actores de entorno dependen de la organización y viceversa, y representarlos en el modelo con dependencias de tipo objetivo. Para simplificar el proceso, deben omitirse las dependencias que no involucran a la organización, p.e. la existente entre el usuario y el proveedor de Directorio telefónico en la Fig. 5. El modelo debe mantenerse lo más simple posible enfocándose solamente en las dependencias entre los actores del entorno y la organización.
- Luego se deben identificar los recursos necesarios para satisfacer el cumplimiento de los objetivos propuestos y modelarlos como dependencias de tipo recurso. Considerar que los recursos pueden ser físicos o información.
- A continuación se debe analizar cada dependencia sobre la organización que sea de tipo objetivo en relación a catálogos de requisitos no funcionales, p.e. las características y subcaracterísticas del estándar ISO/IEC 9126 [21], con el propósito de identificar dependencias de tipo objetivo-blando que sean considerado fundamentales para su consecución.

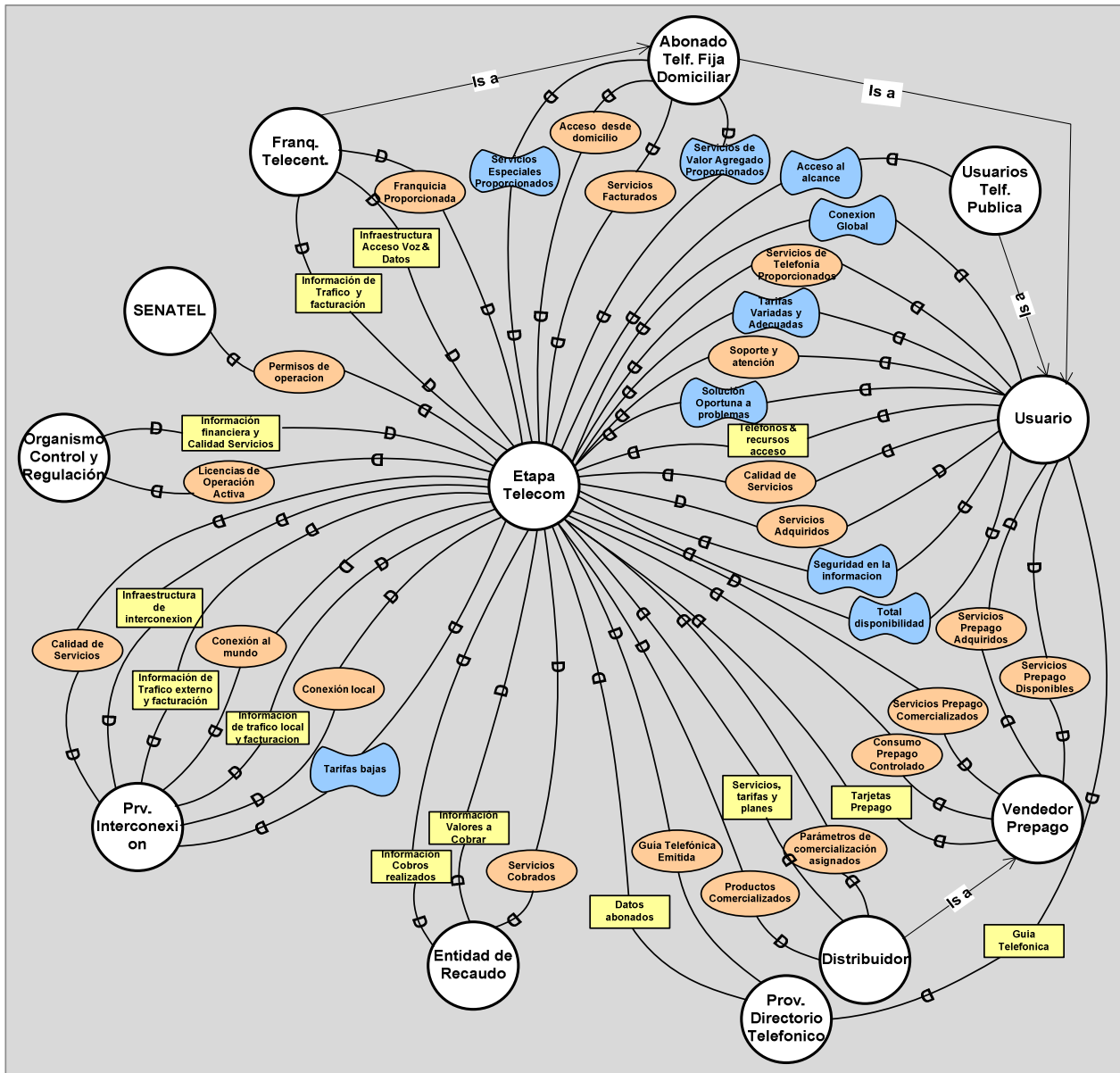


Figura 5: Modelo SD del entorno de la organización ETAPATELECOM S.A.

- Finalmente, aunque en algunos casos son inevitables, es preferible prescindir del uso de dependencias de tipo tarea debido a que pueden ser muy prescriptivas. Una dependencia de tipo tarea representa una forma particular de alcanzar un objetivo lo cual puede de alguna manera comprometer la objetividad del proceso.

En adición a estas guías, hemos explorado en otros contextos métodos más complejos que pueden ser adaptados para conducir esta actividad [22]. Un extracto de las dependencias identificadas en el caso de ETAPATELECOM se presenta en la Fig. 5.

5.2 Modelado del entorno del sistema

Una vez concluido el modelado el entorno de la organización, se debe analizar el impacto que la introducción de un sistema tendría sobre los elementos incluidos en el mismo. Cabe anotar que el sistema no necesariamente tiene que ser un *sistema de información* construido exclusivamente por componentes de software, sino que también puede ser un sistema mixto que incluya componentes de hardware. En el caso particular de ETAPATELECOM, el sistema a modelar es un sistema de Telefonía IP (ver Fig. 6), lo cual de partida permite intuir que varios de sus componentes serán dispositivos de hardware o hardware con software embebido.

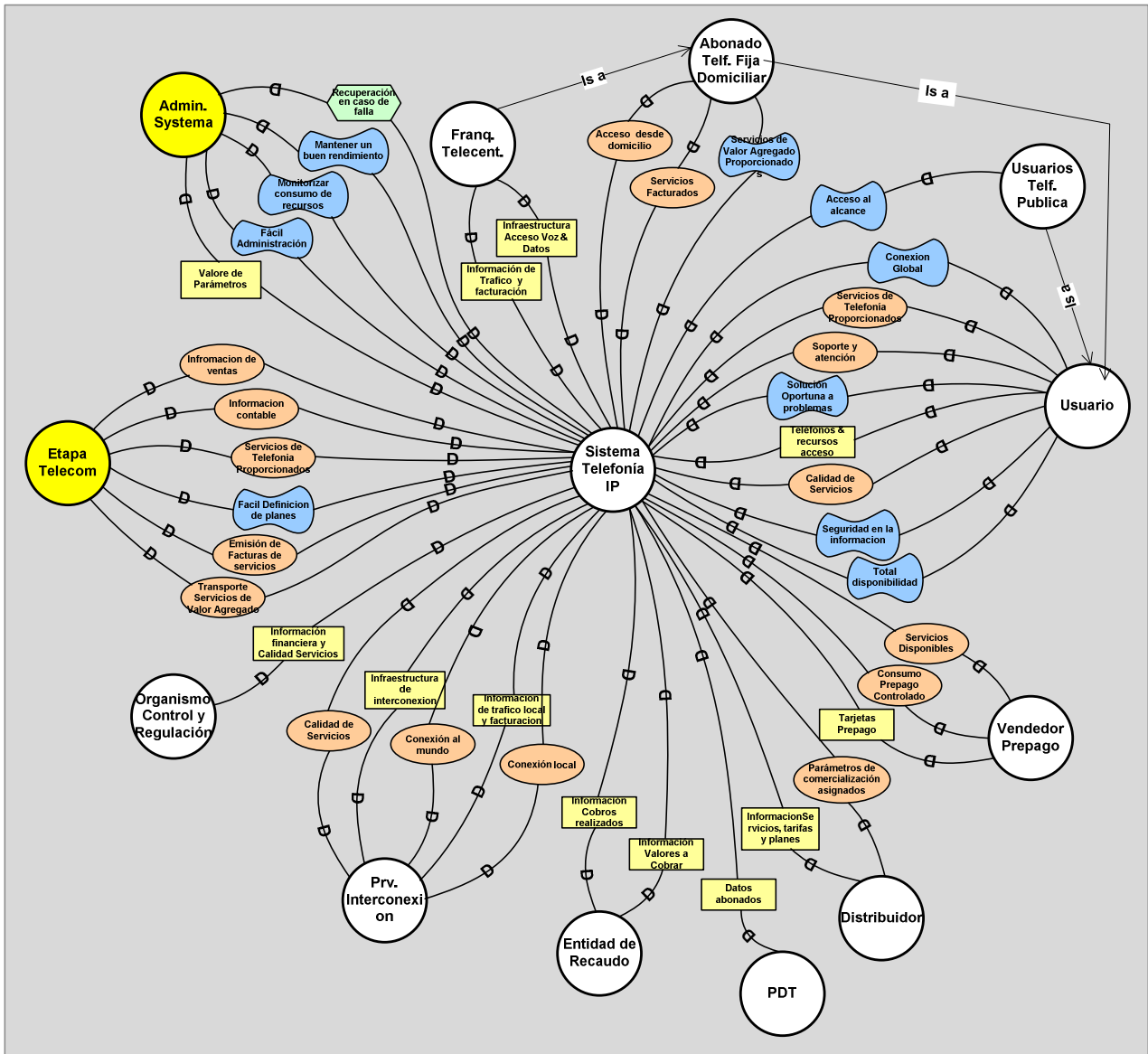


Figura 6: Modelo SD del Sistema de Telefonía IP para ETAPATELECOM S.A.

En términos sencillos, el objetivo de esta actividad es construir un modelo de entorno del sistema partiendo del modelo de entorno de la organización, en el que se identifiquen de manera temprana las dependencias relacionadas con el sistema (aquellas que podrían ser satisfechas por el sistema para el caso de dependencias entrantes en las que el sistema actúa como dependee o que serían necesarias para su operación en el caso de dependencias salientes en las que el sistema actúa como depender), y cuáles no estarían relacionadas con el sistema. Dos sub-actividades son requeridas para este propósito: la identificación de dependencias relevantes para el sistema y la identificación de nuevos actores en el entorno del sistema.

Identificación de dependencias relevantes para el sistema: En esta actividad las dependencias incluidas en el modelo de entorno de la organización, son analizadas de manera sistemática para determinar si deben mantenerse en el modelo de entorno del sistema de información. Cada dependencia es revisada para determinar si tiene sentido que el sistema de información actúe como depender o dependee en la misma. En caso afirmativo la dependencia es redireccionada al sistema de información y la organización es relevada de su rol sobre la misma, p.e., el recurso *Información Financiera y de Calidad de Servicio*, o el objetivo *Consumo Prepago Controlado*, de los actores *Vendedor Prepago* y *Organismo de Control* respectivamente, que recaían sobre

la organización (v. Fig. 5) ahora pasan a ser responsabilidad del sistema (v. Fig. 6). En caso contrario la dependencia es removida del modelo. Este es el caso del actor *SENATEL* y su objetivo asociado *Permiso de Operación* (v. Fig. 3); la organización es la que requiere un permiso de concesión para operar los servicios concesionados, pero esto es transparente para el sistema.

Identificación de actores adicionales en el entorno del sistema: Se puede deducir de los párrafos anteriores, que el modelo de entorno del sistema incluye un subconjunto de los actores y dependencias del modelo de entorno de la organización. Ahora bien, en el entorno del sistema pueden existir otros actores que no necesariamente participan en el entorno de la organización.

Al incorporarse el sistema de información, la organización pasa a ser un actor en su entorno, ya que lo requiere para satisfacer un conjunto de dependencias estratégicas, p.e., el objetivo de *Emisión de Facturas de Servicios* o la *Provisión de Información Contable*.

Por otro lado, utilizando las mismas guías metodológicas descritas en la sección 5.1, otros actores en el entorno del sistema pueden ser identificados. Este es el caso del actor *Administrador del Sistema* (ver Fig. 6), cuya intencionalidad respecto al sistema se describe con los objetivos blandos *Fácil Administración* y *Monitorizar el Consumo de Recursos*. De la misma manera el sistema depende del administrador para satisfacer un conjunto de dependencias estratégicas entre las que se encuentra la provisión de los *Parámetros de Operación* (modelado como un recurso), el objetivo blando *Mantener un Buen Rendimiento* y la tarea *Recuperar en caso de Falla*.

5.3 Descomposición del sistema

El modelo de entorno del sistema permite comprender lo que los actores en su entorno esperan de él. Sin embargo, la identificación de los servicios específicos que debe proveer, requiere un examen más detallado. Los modelos *i* SR* permiten justamente esto: explorar de manera más detallada el interior del sistema por medio de estructuras jerárquicas medio-fin, descomposición de tareas y relaciones de contribuciones de objetivos blandos. No obstante, en DHARMA no queremos proveer un excesivo nivel de detalle que por su complejidad comprometa la utilidad del método. Hay que recordar que en este punto nuestro propósito primario es identificar los principales objetivos internos que deben ser alcanzados, a fin de que el sistema satisfaga las dependencias de los actores en su entorno.

Por tanto, en esta actividad nos enfocaremos en la descomposición del sistema mediante jerarquías medio-fin, de tipo objetivo-objetivo, ya que éstas describen la funcionalidad (servicios) que el sistema debe proveer a los actores en su entorno. Para ello proponemos la siguiente guía metodológica:

- Se debe identificar el principal objetivo del sistema y graficarlo como el objetivo raíz del diagrama.
- Luego se debe reducir este objetivo en sub-objetivos mediante vínculos objetivo-objetivo, en relación a las principales áreas funcionales que se espera que el sistema provea, y ligarlos a las dependencias externas siempre que sea apropiado. Esta primera descomposición se puede alcanzar explorando las dependencias que los actores de entorno tienen sobre el sistema.
- Por último, se debe repetir el paso anterior para cada uno de los sub-objetivos identificados, hasta que los sub-objetivos que se obtengan representen servicios lo suficientemente atómicos como para que no sea razonable continuar con su descomposición. Una regla que permite validar la descomposición es que todos los objetivos que sean hojas en la jerarquía se encuentren ligados a por lo menos una dependencia de entorno. Si uno de estos objetivos no se encuentra ligado a alguna dependencia externa, entonces puede ser removido, a no ser que sea considerado crítico para el cumplimiento de su predecesor.
- Algunos de los sub-objetivos de tipo *medio* pueden estar ligados a más de un objetivo *fin*. Esto sucede usualmente cuando un sub-objetivo de una rama de la descomposición requiere los servicios de otras ramas para ser satisfecho.
- El proceso se completa cuando todas las dependencias externas en el entorno del sistema han sido consideradas y ligadas a los sub-objetivos requeridos para ser satisfechas, en el caso de dependencias entrantes (en las que el sistema actúa como dependee), o a los que dependen de ellas, en el caso de dependencias salientes (en las que el sistema actúa como dependier).

Nuevamente en este caso, es preferible obviar las descomposiciones de tareas porque estas representan “cómo” hacer algo, en lugar del “porqué” tiene que ser hecho, que es lo que requerimos para identificar los servicios que el sistema tiene que proveer. Las tareas se requieren cuando se intenta desarrollar un sistema desde cero, pues ellas prescriben de manera detallada como se debe alcanzar un objetivo. Por el contrario, los objetivos son importantes en la construcción de sistemas basados en componentes pues describen los servicios que deben ser cubiertos por los componentes disponibles en el mercado.

En el caso particular de ETAPATELECOM, el objetivo raíz del sistema fue identificado como *Servicios de Telefonía Proporcionados*. Este objetivo fue descompuesto en cinco subobjetivos *Comunicación Establecida*, *Servicios Administrados*, *Administración Gestionada*, *Servicios Facturados* y *Recursos Proporcionados*, que fueron posteriormente descompuestos en otros más atómicos requeridos para satisfacer las dependencias con los actores del entorno (v. Fig. 7).

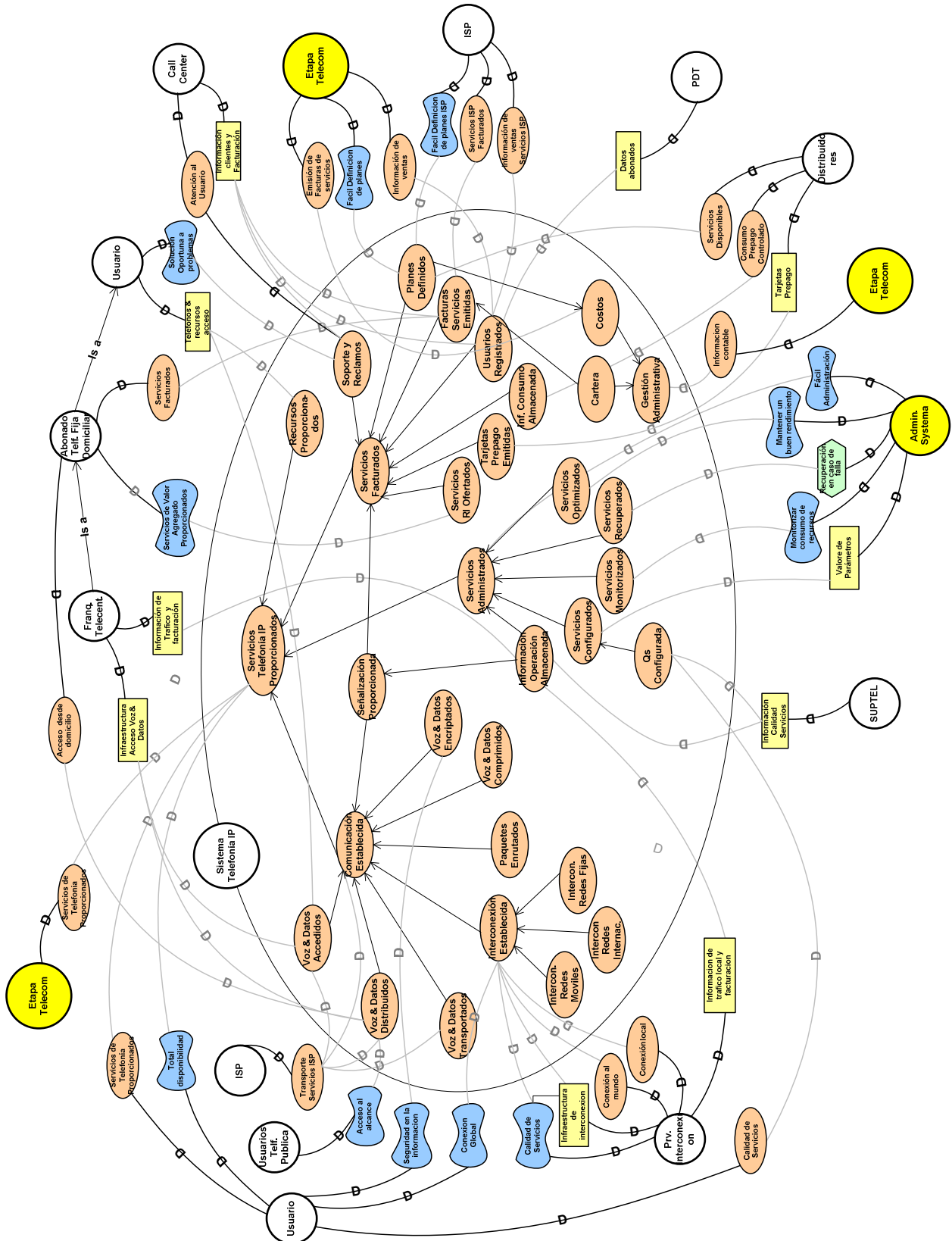


Figura 7. Extracto del modelo SR del sistema de telefonía IP para ETAPATELECOM.

5.4 Identificación de actores del sistema

La siguiente actividad del método propone la descomposición del sistema en subsistemas. Es importante empezar destacando que no adoptaremos una aproximación orientada a componentes sino orientada a actores. Los actores del sistema representan dominios atómicos de componentes, para los cuales no existe un mercado de componentes OTS que cubran tan solo una parte de la funcionalidad considerada para el dominio.

Los actores del sistema son identificados agrupando objetivos relacionados que se encuentre incluidos en la descomposición del modelo SR del sistema. En general, los actores del sistema podrán ser solamente software. Sin embargo, para el caso del sistema de Telefonía IP para ETAPATELECOM se decidió extender el alcance del método para incluir también actores de tipo hardware o hardware con software embebido.

Típicamente los actores en un sistema pueden actuar de dos maneras, como actores centrales o como actores de soporte:

- *Actores centrales*: Este tipo de actores proveen la funcionalidad central del sistema. Debido a ello, en muchos casos el sistema completo adopta su nombre. La mayoría de las dependencias comprometidas y críticas de los actores del entorno del sistema se encuentran ligadas a ellos. En el caso de estudio que nos compete el *Soft Switch* (central telefónica IP usualmente de última generación) y el sistema *ERP* (parcialmente modelado por efectos de espacio) comparten esta clasificación (v. Fig. 8). En el caso del *Soft Switch* este es un actor de tipo hardware con un fuerte componente de software embebido y el *ERP* un componente netamente de software. Debido a ello es fácil intuir que la integración de los actores del sistema requirió del desarrollo a la medida, de un conjunto importante de interfaces de integración hardware y software. Para su construcción se optó por un enfoque oportunista en el que se utilizaron servicios web, interpretación de interfaces HTML e invocación a métodos GET y POST, generación de vistas lógicas a bases de datos y herramientas CASE basadas en lenguajes de quinta generación.
- *Actores de soporte*. Los actores de soporte no proveen la funcionalidad central del sistema. Por el contrario proveen servicios adicionales requeridos por los actores centrales del sistema, para satisfacer algunas de sus dependencias con los actores del entorno. A modo de ejemplo, el sistema *ERP* necesita un *Sistema de*

Facturación para tazar los registros de comunicación (CDRs), y emitir las facturas de consumo de los servicios ofertados por la organización. Todos los actores de soporte tienen vínculos de dependencia con actores centrales, pero no necesariamente entre ellos. También pueden actuar en vínculos de dependencias con actores de entorno, pero usualmente no en relación a la funcionalidad central del sistema.

Si bien los sistemas pueden incluir más de un actor central, podrían no incluir actores de soporte pues éstos no son obligatorios (aunque éste no es el caso usual). Sin embargo es importante anotar que los actores de soporte en muchos de los casos hacen referencia a servicios específicos del negocio, al contrario de los actores centrales que usualmente son de naturaleza más genérica. Por ejemplo el sistema *ERP* incluye módulos de contabilidad, activos fijos e inventarios entre otros genéricos para varios tipos de organización. El Sistema de Facturación por su parte, suele ser específico para cada tipo de organización o instancias particulares de ellas, llegándose a convertir en un elemento generador de valor agregado que permita a una empresa diferenciarse de la competencia, por ejemplo por la flexibilidad en la definición de planes comerciales.

La identificación de los actores del sistema es guiada por los objetivos incluidos en el modelo SR del sistema. Estos objetivos revelan los servicios que se espera que sean cubiertos por los actores del sistema. Su asignación a actores del sistema puede ser apoyada por la revisión de varias fuentes de información, entre ellas se pueden citar:

- Revisión de componentes disponibles en el Mercado y las funciones que ellos proveen, Incluyendo documentos técnicos, revistas especializadas, demos y por supuesto portales especializados como INCOSE [23] que proveen información detallada de los principales objetivos y funcionalidades de varios dominios de componentes OTS.
- Taxonomías de componentes OTS como las disponibles en empresas de consultoría [24]. Aunque en muchos de los casos estas taxonomías no se encuentran bien estructuradas permiten una navegación jerárquica básica, a través de categorías y dominios de componentes relacionados, y la identificación de sus objetivos principales.

Un extracto de los actores identificados en el caso de estudio, y los objetivos que cubren, se puede observar en la Fig. 8.

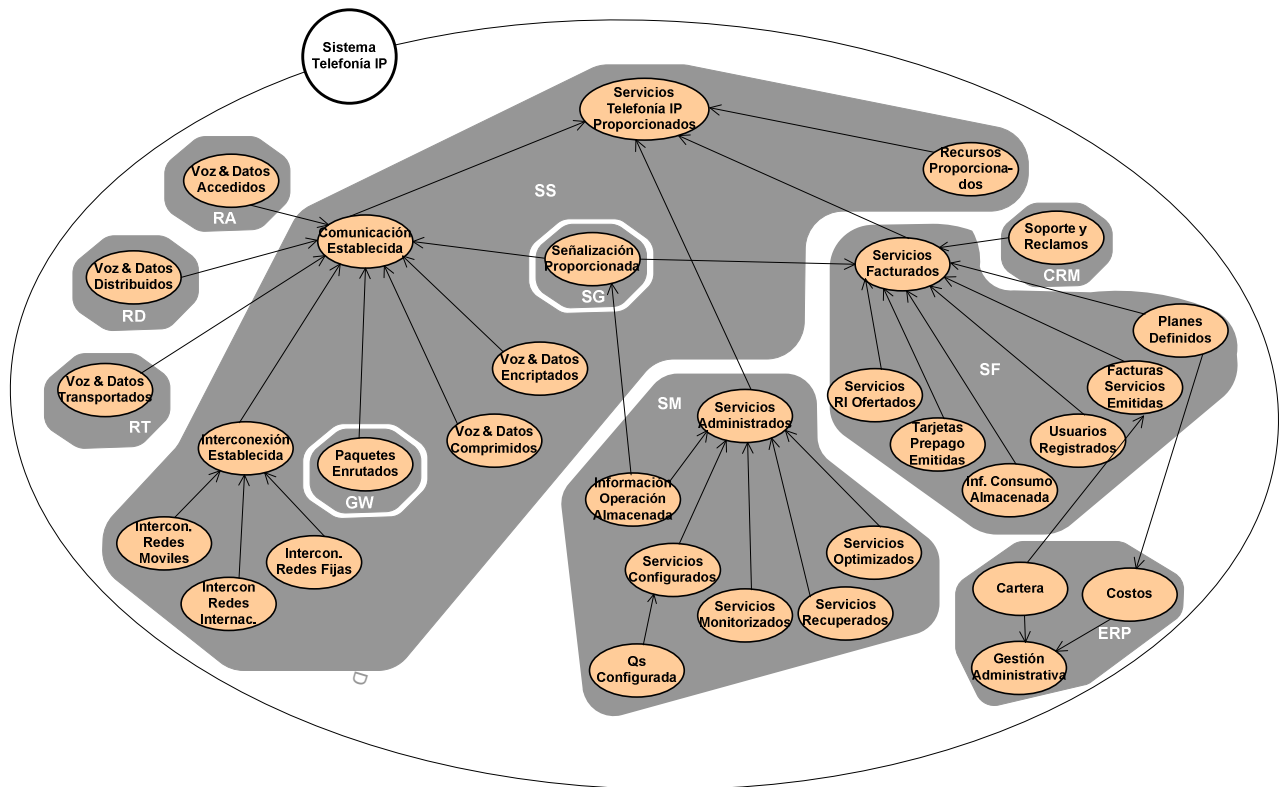


Figura 8. Arquitectura final del Sistema de Telefonía IP para ETAPATELECOM.

6. Conclusiones

En este artículo hemos presentado el método DHARMA para la determinación de arquitecturas software para sistemas híbridos. Este método se basa en el uso de modelos orientados a objetivos y actores expresados en el lenguaje i^* . La utilización de actores concuerda bien con las necesidades de identificar dominios atómicos de software que son la base para expresar la arquitectura. El método es el resultado de una línea de trabajo que se inició en 2006 [25] y ha sido probado en la práctica en el entorno empresarial para sistemas de gran complejidad.

Como ya se ha comentado, DHARMA se entronca con otros métodos elaborados por nuestro grupo en el que el marco i^* se utiliza para descomponer un sistema en actores que representan dominios de software. En concreto estos métodos son COSTUME [12] para la elaboración de modelos de calidad para sistemas compuestos, y RECSS [13] para la elicitación de requisitos en sistemas que abarcan diversos dominios. Una línea de trabajo futuro consiste en la integración de estos tres métodos en un método genérico que pudiera ser particularizado a estas necesidades u otras similares que pudieran surgir en el futuro, probablemente aplicando principios de Method Engineering [26].

Agradecimientos

Este artículo ha sido parcialmente apoyado por el proyecto español ref. TIN2007-64753.

Referencias

- [1] *Proceedings of the 7th International Intl. Conference on Composition-Based Software Systems (ICCBSS)*, IEEE, 2008.
- [2] J. Li *et al.* "A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components". *IEEE TSE*, 34(2), 2008.
- [3] A. Mohamed, G. Ruhe, A. Eberlein. "COTS Selection: Past, Present, and Future". CBSE 2007.
- [4] J. Feller, B. Fitzgerald. *Understanding Open Source Software Development*. Addison-Wesley, 2002.
- [5] M.P. Papazoglou. *Web Services: Principles and Technology*. Prentice-Hall, 2008.
- [6] G. Kotonya, S. Lock, J. Mariani. "Opportunistic Reuse: Lessons from Scrapheap Software Development". CBSE 2007.
- [7] C. Alves, F.M.R. Alencar, J. Castro. "Requirements Engineering for COTS Selection". WER 2000.
- [8] X. Burgués, C. Estay, X. Franch, J.A. Pastor, C. Quer. "Combined Selection of COTS Components". ICCBSS 2002, Springer.

- [9] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD Dissertation, University of Toronto, 1995.
- [10] F. Sannicolo, A. Perini, F. Giunchiglia. "The Tropos Modeling Language. A User Guide". Technical Report # DIT-02-0061, 2002.
- [11] X. Franch, J.P. Carvallo, "Using Quality Models in Software Package Selection". *IEEE Software*, 20(1), 2003.
- [12] J.P. Carvallo, X. Franch, G. Grau, C. Quer. "COSTUME: A Method for Building Quality Models for Composite COTS-Based Software Systems". QSI 2004, IEEE.
- [13] J.P. Carvallo, X. Franch, C. Quer. "Requirements Engineering for COTS-based Software Systems". SAC 2008, ACM/IEEE.
- [14] J.P. Carvallo, X. Franch, C. Quer, M. Torchiano. "Characterization of a Taxonomy for Business Applications and the Relationships among them". ICCBSS 2004, Springer.
- [15] A. Antón, C. Potts. "The Use of Goals to Surface Requirements for Evolving Systems". ICSE 1998, IEEE.
- [16] OpenCyc web page, <http://www.cyc.com/cyc>, last accessed April 2009.
- [17] SWEBOK web page. <http://www.swebok.org/>, last accessed April 2009.
- [18] R.L. Daft. *Organization Theory and Design*. Thomson, 1992.
- [19] A. Fuxman, P. Giorgini, M. Kolp, J. Mylopoulos. "Information Systems as Social Structures". FOIS 2001.
- [20] eTOM web page. <http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/1647/Home.html>, last accessed April 2009.
- [21] *ISO/IEC Standard 9126-1. Software Engineering – Product Quality – Part 1: Quality Model*, 2001.
- [22] V. Sai, X. Franch, N. Maiden. "Driving Component Selection through Actor-Oriented Models and Use Cases". ICCBSS 2004, Springer.
- [23] Incose web page, <http://www.incose.org/>, last accessed April 2009.
- [24] Gartner web page, <http://www.gartner.com/>, last accessed April 2009.
- [25] J.P. Carvallo. "Supporting Organizational Induction and Goals Alignment for COTS Components Selection by means of *i**". ICCBSS 2006, IEEE.
- [26] S. Brinkkemper. "Method Engineering: Engineering of Information Systems Development Methods and Tools". *IST 38(4)*, Elsevier, 1996.