

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 06-16

Desarrollo de aplicaciones con enfoque SOA
(Service Oriented Architecture)

Andrea Delgado Laura González Federico Piedrabuena

2006

Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)

Delgado, Andrea; González, Laura; Piedrabuena, Federico

ISSN 0797-6410

Reporte Técnico **RT 06-16**

PEDECIBA

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, 2006

Desarrollo de aplicaciones con enfoque SOA ^ϕ (Service Oriented Architecture)

Andrea Delgado, Laura González, Federico Piedrabuena

*Universidad de la República,
Instituto de Computación,
Grupo de Ingeniería de Software,
Montevideo, Uruguay
{adelgado,lauragon,fpiedrab}@fing.edu.uy*

Febrero 2006

Resumen

El Proyecto Link-All, financiado por la Unión Europea y del que forma parte el Instituto de Computación (INCO) de la Facultad de Ingeniería de la Universidad de la República, ha definido y puesto en funcionamiento un sistema basado en SOA. Dicho sistema brinda servicios concretos a nivel del negocio sobre los que se van construyendo otros nuevos y que son utilizados por las distintas aplicaciones que se incorporan al sistema.

El Grupo de Ingeniería de Software (Gris) del mismo instituto, INCO, tiene un programa de construcción y prueba de modelos de proceso asociado al curso “Proyecto de Ingeniería de Software” para el cual se definió una metodología de desarrollo de aplicaciones basadas en SOA. En este trabajo se presentan el sistema Link-all basado en SOA, la metodología de desarrollo SOA del Gris y el desarrollo de una aplicación de Help-Desk para este sistema utilizando la metodología SOA definida.

Palabras clave: Ingeniería de Software, Procesos y Metodologías de desarrollo, Arquitecturas de Software, Service Oriented Architecture (SOA)

^ϕ Artículo publicado en las “V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC’06)”, Puebla, México, Febrero de 2006. ISBN 970-94770-0-5

1 Introducción

El Proyecto Link-all [1], financiado por la Unión Europea e integrado por diecisiete socios de Sudamérica y Europa, está dirigido a asistir en la apropiación de prácticas de tecnologías de la información y comunicaciones (TIC), apoyar la colaboración, promover el intercambio de experiencias, la transferencia de conocimiento y mejorar habilidades en tres sectores objetivo en Latinoamérica: artesanía, eco-agro turismo y cultura. El Sistema Link-all incluye herramientas diseñadas e integradas para proveer de una serie de facilidades de inclusión tecnológica, que fortalezcan la integración de actividades de desarrollo local que posibiliten y faciliten la inserción exitosa de comunidades remotas en el mercado global, a través del desarrollo de una red transnacional europeo-latinoamericana de cooperación.

El programa de construcción y prueba de modelos de proceso [2], llevado adelante por el grupo de Ingeniería de Software (Gris) [3], se inició en el año 2000. Este utiliza como banco de pruebas al curso “Proyecto de Ingeniería de Software”, donde los estudiantes contrastan la teoría con su aplicación práctica en proyectos sometidos a restricciones análogas a las de la industria. El objetivo central del programa es el desarrollo de modelos de proceso que puedan resultar adecuados para su transferencia a la industria. Hasta la fecha se han estudiado y adaptado varios modelos de proceso, como ser Rational Unified Process (RUP) [4] y eXtreme Programming (XP) [5]. La metodología de desarrollo orientado a SOA se inscribe en este programa [6].

2 Service Oriented Architecture (SOA)

SOA es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Se basa en cuatro abstracciones básicas: servicios, application frontend, repositorio de servicios y bus de servicios. Un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio, las restricciones para el consumidor, y una interfaz que expone físicamente la funcionalidad. Las application frontend consumen los servicios formando procesos de negocios. Un repositorio de servicios almacena los contratos de servicios y el bus de servicios interconecta las application frontend y los servicios.

Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio. Por su parte, los procesos del negocio se realizan en servicios orientados a procesos que se componen de secuencias definidas de invocaciones a servicios, mediante una orquestación de los mismos en lo que se conoce como coreografías de servicios. Un Business Process Management System (BPMS) es el aliado ideal para definir esta orquestación desde donde invocar los servicios necesarios para realizar el proceso establecido. El término Business Process Management (BPM) [7] refiere al conjunto de actividades que se realizan para optimizar o adaptar sus procesos de negocio a las nuevas necesidades organizacionales. Como en general están soportadas por herramientas de software, el término BPM es utilizado indistintamente para referirse a las herramientas y las actividades.

2.1 Clasificación de servicios

Una primera clasificación de SOA es en dos niveles: application frontend y backend. Las applications frontend si bien no son servicios, son elementos activos que inician los procesos de negocio y reciben los resultados, los servicios del backend se clasifican [8] en: básicos, intermediarios, centrados en procesos y empresariales públicos.

Los servicios básicos son los fundamentos de SOA. Son proveedores puros y no mantienen un estado conversacional. Se dividen en servicios centrados en los datos y centrados en la lógica, donde los primeros tienen como propósito manejar los datos persistentes, almacenamiento, recuperación y gestión transaccional, y los segundos proveen encapsulamiento para cálculos complejos o reglas de negocio, tradicionalmente encapsulados en bibliotecas o frameworks.

Los servicios intermediarios son servicios sin estado que hacen de puente entre las inconsistencias técnicas o discrepancias conceptuales en el diseño. Son tanto consumidores como proveedores, mediando entre los distintos elementos que deben funcionar juntos. Se dividen a su vez en technology gateways, adapters, facades y functionality-adding.

Los servicios centrados en procesos encapsulan el conocimiento de los procesos de negocio de la Organización y controlan y mantienen el estado del proceso en ejecución. Son tanto consumidores como proveedores y desde el punto de vista técnico son la clase de servicios más sofisticada. Una de las principales ventajas que presentan estos servicios es que separan la lógica de los procesos, definiendo los procesos de negocio y su control, en base a orquestación de servicios existentes. Es en este tipo de servicios donde aparecen los BPMS y los lenguajes de modelado de procesos como BPML [7] (Business Process Modeling Language) y BPEL4WS [9] (Business Process Execution Language for Web Services) para orquestar servicios.

Los servicios empresariales públicos a diferencia de los anteriores son servicios que una empresa ofrece a socios y clientes externos. Estos servicios tienen requerimientos específicos de facturación de uso, interfaz, desacoplamiento y seguridad ya que las empresas deben acordar claramente como se realiza el uso de los mismos.

2.2 Niveles de clasificación de SOA

Al abordar un proyecto de construcción o transformación de un sistema basado en SOA, se plantea el desafío de cómo definir y estructurar los niveles de SOA. Estos niveles están relacionados directamente con los tipos de servicios que se incluirán. El nivel de “Fundamental SOA” se compone de dos capas de abstracción: application frontend y servicios básicos. Se identifican, definen e implementan servicios básicos del negocio que permitan que dos o más aplicaciones compartan la lógica de negocio. Aunque es un enfoque bastante simple provee las bases para ir avanzando hacia una SOA de mayor nivel. El nivel de “Networked SOA” agrega la capa de abstracción de servicios intermediarios que pueden incluir servicios del tipo facades, technology gateways, adapters y functionality-adding. Como se mencionó, los servicios intermediarios encapsulan parte de la complejidad de los servicios básicos.

El nivel de “Process-Enabled SOA” es el más completo, donde los procesos de la Organización se modelan en servicios orientados a procesos que mantienen además el estado. En este nivel se conjuntan el enfoque SOA con el de BPM, constituyendo el ideal del modelado de procesos en la Organización y su informatización independiente de la tecnología. Las applications frontend son más simples, y la capa de servicios intermediarios puede o no existir. Una ventaja de este nivel es que permite que la lógica de los procesos quede claramente separada del núcleo de lógica de negocio localizado en los servicios básicos.

Utilizando un BPMS para la definición y control de estos procesos, se evita la codificación de información y reglas de negocio directamente en la aplicación, lo que facilita la modificación, reconfiguración y optimización de los procesos mediante herramientas gráficas para definir los flujos de los procesos, por ejemplo en forma de Diagrama de Actividad de UML.

3 Enfoque SOA del Sistema Link-All

A nivel informático el proyecto Link-all está integrado por siete socios, pertenecientes a Colombia, España, Luxemburgo, Portugal y Uruguay. Esta situación agrega complejidades extras a las impuestas por un proyecto tecnológico de gran porte, como ser el idioma, las diferencias horarias y la fluidez en la comunicación. Esto impone que la solución deba ser altamente desacoplada para minimizar los problemas impuestos por esta situación.

El Sistema Link-all es un Sistema de Información Federado (SIF), orientado al web, basado en comunidades virtuales, tanto a nivel de vistas de datos como de funcionalidades, que representan las comunidades u organizaciones que hacen uso del sistema. Estas comunidades comparten un conjunto básico de funcionalidades, pero emplean procesos de negocios diferentes, por lo que la solución basada en SOA, en la que se pueden satisfacer estas necesidades simplemente orquestando los procesos de

diferente forma, permite concentrar el desarrollo en las generalidades y delegar las particularidades a la orquestación y configuración.

El Sistema posee además una serie de herramientas que lo complementan. Una de esas herramientas permite gestionar el ciclo de vida de los servicios y aplicaciones, como ser distribución e instalación. Otra brinda ayuda al usuario en base a su contexto, el que está formado por el proceso utilizado y características del perfil, como ser el idioma.

3.1 Arquitectura

El sistema Link-all está desarrollado orientado a una arquitectura Process-Enabled SOA, en la que se identifica un primer nivel de servicios de infraestructura, sobre el que se monta el segundo nivel con servicios orientados al dominio que luego son orquestados en un tercer nivel, en el que se construyen los procesos de negocio. Los servicios orientados al dominio son desarrollados por diferentes equipos, conformados por los diferentes socios técnicos del proyecto. Por esta razón era necesario disponer de una arquitectura que brinde bajo acoplamiento.

La capa de servicios de infraestructura ofrece funcionalidades independientes del dominio, que resuelven necesidades generales como ser autenticación, autorización, acceso a datos, etc. La capa de servicios orientados al dominio resuelve funcionalidades básicas específicas del negocio. La granularidad de estos servicios es baja, de forma que puedan ser reutilizados y así construir coreografías que permitan resolver funcionalidades más complejas. La capa de procesos de negocios ofrece principalmente coreografías de servicios pertenecientes al mismo nivel o a niveles inferiores. Esto permite resolver procesos de negocio complejos. En la figura 1 se muestran las capas de servicios definidas para el Sistema Link-all.

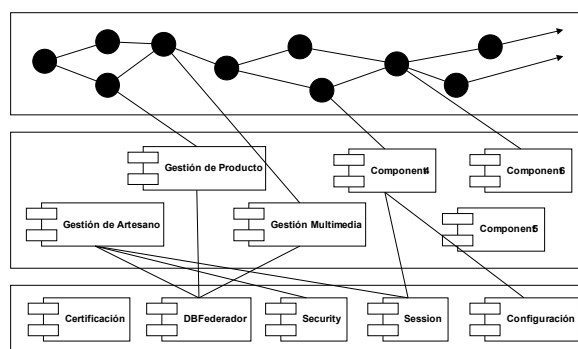


Figura 1 – Capas de servicios Sistema Link-all

3.1.1 Servicios de infraestructura

Los servicios de infraestructura conforman un framework que permiten gestionar el ciclo de vida de las aplicaciones orientadas a SOA, en un contexto particular en el que se manejan comunidades virtuales de usuarios en un ambiente federado de servidores Link-all. El servicio de Federación de base de datos tiene como principal objetivo exponer una visión homogénea de los datos de dominio que se encuentran físicamente distribuidos en varios servidores Link-all. Este componente provee un esquema de base de datos unificado y se encarga de la resolución de operaciones de acceso a datos en un ambiente federado, por lo que el acceso a los datos siempre será realizado a través de este servicio.

El servicio de sesiones se encarga del manejo de sesiones en el ambiente federado y actúa como repositorio unificado de la información de sesión de los usuarios. Esto permite contemplar la diversidad tecnológica de las aplicaciones que operan en el sistema. También expone funcionalidades para el registro de las acciones de un usuario a lo largo de su actividad en el sistema. El servicio de seguridad, actúa como proveedor de información sobre la autenticidad del usuario, y control de acceso. El servicio de configuración se encarga del mantenimiento de información de instalación y configuración y gestiona los parámetros que modifican el comportamiento del sistema.

3.1.2 Servicios primarios de dominio

Los servicios primarios de dominio ofrecen funcionalidades básicas del negocio, por lo tanto tienen una granularidad baja desde el punto de vista de complejidad del proceso de negocio que resuelven. Esto permite disponer de un conjunto canónico de funcionalidades, concentrando la tarea de desarrollo principalmente en la resolución de la complejidad del negocio y delegando las tareas de infraestructura a los servicios subyacentes. Como ejemplo de servicios que pueden encontrarse en este nivel se tiene mantenimiento de la entidad artesano o mantenimiento de contenido multimedia.

3.1.3 Servicios de procesos de negocio

Los servicios de procesos de negocio son orquestaciones de servicios primarios del dominio, por medio de las que se resuelven procesos de negocios complejos.

3.2 Tecnología

El sistema Link-all fue construido empleando un conjunto de tecnologías abiertas, probadas y difundidas, para asegurar un bajo costo de licenciamiento y maximizar la compatibilidad con otros productos. Para el desarrollo del sistema Link-all se utiliza el lenguaje de programación java, en particular Java 2 Enterprise Edition (J2EE) [10] que define un estándar para el desarrollo de aplicaciones empresariales multicapas basadas en componentes. Como servidor de aplicaciones J2EE se emplea JBoss Application Server (JBoss AS) [11]. Como RDBMS se emplea PostgreSQL [12] para gestionar la persistencia de la metadata necesaria para el funcionamiento del sistema, así como para almacenar datos específicos del negocio. Se utiliza el framework Apache Struts [13] que facilita la adopción de una variante del patrón de diseño Model-View-Controller (MVC) y Apache AXIS [14] como plataforma para web services. Se utiliza Eclipse [15] como entorno de desarrollo java.

4 Metodología de desarrollo SOA

Una metodología para el desarrollo orientado a servicios no es una nueva metodología, sino que se construye sobre el proceso o enfoque de desarrollo que se sigue en la Organización, agregando actividades y artefactos específicos para el desarrollo de los servicios, y por lo tanto cualquier proceso podría servir de base. Sin embargo, dada la naturaleza compleja y cambiante de los requerimientos del negocio en el contexto de las Organizaciones actuales, es recomendable que se siga un enfoque iterativo incremental como forma de lidiar con estos cambios, y con fuerte enfoque en la generación de productos intermedios como forma de ir obteniendo productos claves en el avance del mismo que provean visibilidad sobre el desarrollo. [8]

4.1 Proceso de desarrollo base

El proceso de desarrollo base es una adaptación del Rational Unified Process (RUP) [4] que ha ido evolucionando desde su primera versión definida por estudiantes, de la carrera de Ingeniero en Computación, en el año 2000 como tesis de grado [16]. Esta edición se compone de un esqueleto de actividades que es común a todos los tipos de desarrollo que se emprendan, y dos extensiones principales para realizar desarrollos específicos en OO y Genexus [17]. En este contexto, la metodología de desarrollo para aplicaciones basadas en SOA incorpora actividades y entregables específicos para el desarrollo orientado a servicios.

El proceso base tiene dos dimensiones, el tiempo y las disciplinas. En la dimensión del tiempo se definen cuatro fases: Inicial, Elaboración, Construcción y Transición, en las tres primeras fases se definen dos iteraciones de dos semanas cada una, y en la última fase una iteración de dos semanas, que completan junto con la semana de preparación previa y la semana de evaluación final, las dieciséis semanas del curso. En la dimensión de las Disciplinas se definen las disciplinas tradicionales del desarrollo de software: requerimientos, diseño, implementación y verificación, más las disciplinas de gestión: gestión del proyecto, de la calidad, de la configuración e implantación. En cada disciplina se definen actividades, roles encargados y participantes de las actividades, y entregables de entrada y salida. Se cuenta además con un cronograma que indica para cada semana que actividades se deben realizar y que entregables se deben obtener.

4.2 Roles, Disciplinas y Actividades para SOA

Entre los roles definidos los más importantes para la Extensión SOA son: el Arquitecto, quién tiene responsabilidad en el área de Diseño pero participa también en el relevamiento de requerimientos y como coordinador del desarrollo; el Analista, quién principalmente participa en el relevamiento de requerimientos; el Especialista Técnico, quién tiene a su cargo el estudio e investigación de las tecnologías disponibles a utilizar y el Implementador, quién tiene a su cargo la codificación de la aplicación en desarrollo. Una descripción completa del proceso base está fuera del alcance de este artículo y puede verse en su implementación actual en [18] opción Año 2005.

La Extensión SOA agrega una disciplina que está en el RUP pero no en el proceso base, que es el Modelado del Negocio, ya que como se mencionó, un aspecto importante en SOA es el modelado de los procesos de la Organización. Como parte central de la metodología definida se incorporan cinco actividades en la Disciplina de Diseño, que son las actividades claves del enfoque, en las que se identifican, clasifican, reutilizan, especifican, y definen los servicios de la aplicación, y la orquestación de los mismos para proveer los procesos del negocio modelado. Luego en la Disciplina de Implementación se agrega la actividad de implementación de servicios según fueron definidos en la etapa de diseño. Además, para cada actividad se definen los roles responsables, participantes y los entregables entrada y salida de las mismas.

4.2.1 Disciplina Modelado del Negocio

El propósito de la disciplina Modelado del Negocio es asegurar que clientes, usuarios finales, desarrolladores y otros involucrados tienen un conocimiento común de la Organización objetivo, derivar los requerimientos del sistema de software que son necesarios para apoyar a la Organización y comprender como el sistema de software se inscribe en la misma.

En esta disciplina participan el Analista y Arquitecto, quienes realizan principalmente el relevamiento de requerimientos a partir de reuniones con el cliente. Los entregables que se obtienen en esta Disciplina son la Evaluación de la Organización Objetivo que contiene aspectos claves de la misma, y el Modelo de Casos de Uso del Negocio donde se especifican los procesos de negocio identificados.

Evaluar la Organización Objetivo (MN1): Se debe describir el estado de la Organización en la que se implantará la solución, en términos de sus procesos actuales, herramientas, personas, clientes, competencia, desafíos tecnológicos, problemas y áreas de mejora, identificando claramente a los involucrados en el esfuerzo de modelado del negocio.

Identificar los Procesos de Negocio (MN2): En esta actividad se deben entender y describir los procesos que se realizan en el negocio en lo que tiene que ver principalmente con la aplicación que se va a desarrollar. Los procesos se describen como Casos de Uso del Negocio con los flujos correspondientes y se modelan con Diagramas de Actividad en UML, describiendo para cada uno los actores participantes y el flujo de ejecución de los mismos en la Organización.

4.2.2 Disciplina Diseño

El propósito de la disciplina de Diseño en SOA agrega los objetivos que tienen que ver con el modelado orientado a servicios para cumplir con los procesos de la Organización descritos en los Casos de Uso del Negocio. Especificar estos servicios, así como los componentes para implementarlos, reutilizar otros servicios dentro de la Organización, definir la secuencia de invocaciones a servicios necesaria para la ejecución de los procesos del Negocio en la orquestación de servicios que define la coreografía correspondiente, propendiendo a la utilización de un BPMS para su definición y control.

En esta disciplina participan el Analista, Arquitecto y los Especialistas técnicos, donde los primeros aportan su conocimiento de la Organización y del problema a resolver mientras los últimos aportan el conocimiento técnico de las herramientas a utilizar. El entregable que se obtiene en esta Disciplina es el Modelo de Servicios, que incluye la información de los servicios definidos.

Identificar y Categorizar Servicios (D6): a partir de la Descripción de la Arquitectura en sus vistas de Casos de Uso del Sistema y Lógica, se realiza un análisis de los subsistemas definidos, identificando

qué subsistemas realizan cada Caso de Uso del Sistema. A partir de esta correspondencia se identifican los servicios que deben ser provistos por cada subsistema para realizar los Casos de Uso del Sistema que componen los Casos de Uso del Negocio.

Especificar Servicios (D7): Para cada servicio identificado se realiza el contrato funcional, especificando para cada interfase definida los métodos que deben ser implementados para proveer el servicio acordado para la interfase. Para cada operación en la interfase se debe especificar: a) nombre del método, b) parámetros requeridos por el método, con nombre, tipo y descripción, c) valor retornado, indicando el nombre, tipo y descripción, d) lista de excepciones levantadas, e) breve descripción de la funcionalidad provista, f) precondiciones requeridas para la ejecución exitosa de la operación, g) post-condiciones válidas luego de la ejecución del método.

Investigar Servicios Existentes (D8): Se debe investigar con qué servicios cuenta la Organización, tanto básicos como de otros niveles, para reutilizar en la aplicación en desarrollo. Puede ser que se deban implementar servicios intermediarios para reutilizar otros servicios que no se adapten totalmente a los requerimientos de la aplicación.

Asignar Servicios a Componentes (D9): Se define quien proveerá la implementación de los servicios definidos según las interfaces especificadas, teniendo en cuenta los servicios existentes identificados, si existen componentes que los brinden o funcionalidades existentes para las cuales se pueda crear algún servicio intermediario que las provea. En la especificación de cada componente se indican las reglas de negocio y servicios que implementa, y componentes que utiliza.

Definir Orquestación de Servicios (D10): La orquestación de servicios para la implementación de procesos de negocio se muestra para cada proceso en un diagrama de secuencia que describa la interacción entre los distintos servicios involucrados. Se utilizará preferiblemente una herramienta BPMS para definir, ejecutar y gestionar la orquestación de dicho servicio.

4.2.3 Disciplina Implementación

El propósito de la disciplina de Implementación en una SOA agrega los siguientes objetivos a los definidos en el modelo base: implementar los componentes proveedores de servicios según lo establecido por la asignación de servicios a componentes, e implementar el ligamiento de servicios en las application frontend y en los servicios que llaman a otros servicios, utilizando la estrategia definida en la Organización. En esta Disciplina se incluye la actividad Implementar servicios (I13) que define como realizar la implementación de los servicios definidos, la que no se describe en este documento.

En la figura 2 se presenta como ejemplo el flujo de actividades para la Disciplina de Diseño modelado como Diagrama de Actividad de UML. Este flujo se ejecuta cada vez que se entra a una iteración en cada Fase del proceso, al momento de realizar las actividades de diseño que corresponda. Según la Fase e iteración en la que se esté se realizarán todas, algunas o ninguna de las actividades definidas, según las transiciones permitidas en el flujo. La agrupación de actividades como Definir Servicios (DA1) es la realización conjunta de las actividades D6, D7 y D8, ya que las tres se realizan a la vez por la relación que tienen, y además iterando con la actividad Describir la Arquitectura (D2) definida en el proceso base.

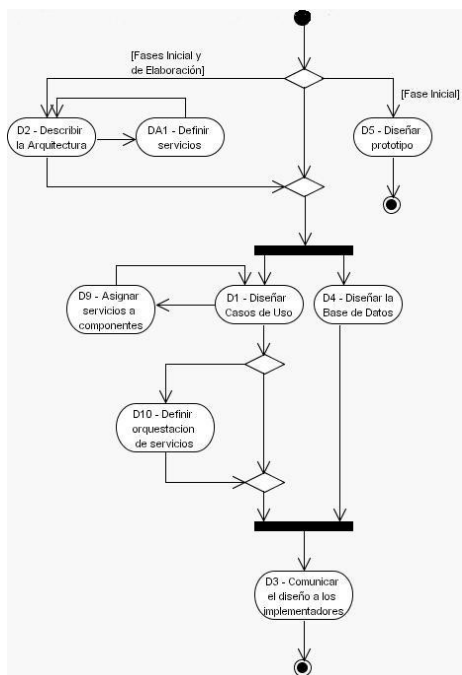


Figura 2: Flujo de actividades de Diseño

5 Caso de estudio: Aplicación Help-Desk

En un sistema como el Link-all que brinda servicios a múltiples usuarios distribuidos geográficamente y que poseen un bajo conocimiento informático, es imprescindible contar con múltiples canales de comunicación y apoyo al usuario. Es así que el sistema fue concebido para ocultar las complejidades técnicas y brindar soporte en todo momento. Un ejemplo de ello es el Help Contextual, una herramienta que brinda ayuda al usuario basada en el contexto, apoyándose en heurísticas para aquellas situaciones en las que no se dispone de información precisa. Otro caso es el Help-Desk, herramienta de gestión de incidentes y nuevos requerimientos que será objeto de estudio en la presente sección. Este caso de estudio se enmarca en el contexto de trabajo que se mencionó previamente y refiere al desarrollo de una aplicación de Help-Desk en el curso “Proyecto de Ingeniería de Software” que dicta el Grupo de Ingeniería de Software (Gris), donde se aplica la metodología definida por el Gris para el desarrollo de aplicaciones en plataforma SOA.

5.1 Descripción de la aplicación Help-Desk

Esta aplicación permite gestionar incidentes, situaciones de error o de comportamientos inesperados, y nuevos requerimientos, en un Sistema de Información Federado basado en metadata del sistema. Los desarrolladores del sistema, al igual que los usuarios, están distribuidos geográficamente. En lo que refiere al desarrollo, la plataforma de integración (los servicios de infraestructura y la base de datos del sistema) fue desarrollada en Uruguay, mientras que las aplicaciones (que resuelven la lógica de negocio) fueron desarrolladas por socios de Colombia, España, Francia, Luxemburgo y Portugal. Esto supone responsabilidades compartidas en cuanto al mantenimiento de las aplicaciones, lo que requiere de coordinación, procesos claramente definidos e información bien estructurada para el tratamiento de incidentes y requerimientos. En lo que refiere a los usuarios, además de la distribución geográfica existe un bajo nivel de alfabetización informática. Por lo que es importante respaldarlos en todo momento, lo que implica entre otras cosas, contar con una herramienta de soporte a usuarios.

En el contexto en el que se utilizará el Help-Desk, el principal objetivo es gestionar de manera sistemática y eficiente tanto los incidentes como los nuevos requerimientos del sistema. De esta manera se busca confeccionar una base de conocimientos explotable por el usuario final, que permita tipificar problemas y soluciones o alternativas (workaround). Otros objetivos que se persiguen son mantener registro de los incidentes, tiempos de resolución, tiempos de dedicación y estado actual de cada incidente de forma de mejorar la operativa. Este sistema debe integrarse completamente al Sistema

Link-all, consumiendo los servicios de infraestructura de forma de compartir el esquema de seguridad, por ejemplo. A su vez el Help-Desk debe ofrecer servicios que puedan ser consumidos por otras aplicaciones u herramientas, ejemplo de ello es que el Help Contextual pueda consumir servicios de la base de conocimientos para proveer ayuda sobre errores conocidos. La comunicación con los usuarios es fundamental en el Sistema Link-all. En este sentido el Help-Desk debe permitir mantener contactos tanto sincrónicos como asincrónicos. Debe mantener informado al usuario del estado de su incidente, vía mail de hitos importantes y de una interfaz grafica integrada al sistema de los eventos significativos. Por otro lado debe existir la posibilidad de mantener una conversación entre el usuario y el técnico encargado del incidente, por lo que se integrará una herramienta de Chat.

Debido a que el desarrollo del Sistema está a cargo de varios grupos, es fundamental poder catalogar los incidentes de forma de que pueda encaminarse al grupo del desarrollo de la aplicación que originó el error. Es importante contar con la posibilidad de reasignar un incidente al técnico más idóneo para el tratamiento del incidente o requerimiento. Si bien pueden identificarse tres tipos de usuarios (administrador, técnico y cliente), el sistema debe estar diseñado de forma que pueda reconfigurarse el conjunto de funcionalidades disponibles para cada usuario. De esta forma se podrán absorber los cambios generados por la evolución del Sistema Link-all.

Para el desarrollo del Help-Desk se incorporaron nuevas tecnologías, teniendo en cuenta los objetivos de bajo costo de licenciamiento y compatibilidad con otros productos planteados por el sistema Link-all. Se emplea JBoss jBPM [19] como motor de workflow y BPM, que permite la creación de procesos de negocio que se coordinan entre personas, aplicaciones y servicios. Provee un framework para crear, coordinar y monitorear procesos de negocios. En el Help-Desk jBPM es utilizado principalmente para definir y gestionar el flujo que deben seguir los incidentes luego de haber ingresado al sistema. En las interfaces de usuario se emplea AJAX, [20], técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías existentes: HTML (o XHTML) y Hojas de Estilo en Cascada (CSS), Document Object Model (DOM) y JavaScript, XML y XSLT. Se emplea el protocolo Jabber [21] que permite que dos entidades en Internet intercambien mensajes y otro tipo de información casi en tiempo real. El sistema utiliza el servidor Jive [22] para implementar la mensajería instantánea entre los usuarios del Help-Desk. Jive es un servidor Jabber de código abierto.

5.2 Puesta en práctica de la metodología

El desarrollo del Help-Desk está siendo llevado adelante por dos grupos de 11 estudiantes cada uno, grupos 4 y 5 [18], que siguen el proceso base con la extensión OO y la Extensión SOA. Estos grupos son dirigidos por el docente responsable de la definición de la metodología de la Extensión SOA y tiene como clientes a dos integrantes del Proyecto Link-all. El director de proyecto tiene contacto con los grupos una vez por semana, donde se revisa el trabajo realizado en la semana previa y se discute la planificación de la semana siguiente, dentro de las Fases e iteraciones que indica el proceso base. Las reuniones con el cliente son frecuentes al comienzo para el relevamiento de requerimientos, y más espaciadas cuando el grupo comienza a tener liberaciones, que son presentadas al cliente y validadas con él. El director de proyecto está en estrecho contacto con el cliente para hacer el seguimiento del desarrollo y sus validaciones y contar con la perspectiva del cliente sobre el desarrollo del proyecto, además de la del grupo.

5.3 Resultados obtenidos

El Help-Desk también sigue el enfoque SOA Process-Enabled. Se desarrollaron varios servicios básicos, como ser la base de conocimiento, manejador de alerta, servicio de mail, y sistema de Chat, que hacen uso de los servicios de infraestructura del Sistema Link-all. Sobre los servicios básicos se construye una serie de servicios orquestados que resuelven la lógica de negocio que debe resolver el Help-Desk. La orquestación de estos servicios se realiza empleando un motor de orquestación y workflow, JBMP de JBOSS, extendido y configurado para el Help-Desk. En este motor se especifica también el flujo de trabajo que debe seguir un incidente.

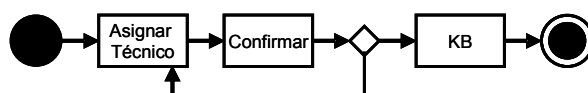


Figura 3: Ejemplo de flujo de incidentes

La figura 3 muestra que al ingresar un incidente al sistema, éste es verificado y encaminado al técnico, asociado a la categoría del incidente, que tenga menos carga de trabajo. El técnico atiende el incidente agregándole eventos donde registra los pasos seguidos, junto a la carga horaria insumida. Una vez obtenida una solución el incidente pasa al estado de confirmación, en donde se envía un mail al cliente para verificar que la solución sea satisfactoria. Una vez aceptada la solución, el incidente se cierra y pasa al estado donde es controlado para su posible incorporación a la base de conocimiento (KB).

A esta altura del proyecto los grupos han validado con el cliente el ejecutable de la Línea Base de la Arquitectura obtenido como liberación de la Fase de Elaboración, y que implementa los Casos de Uso relevantes a la Arquitectura. Ambos grupos han obtenido productos que cumplen con los requerimientos básicos pedidos por el cliente. En la Fase de Construcción, que está transitando, se irán incorporando el resto de las funcionalidades incluidas en el Alcance y que llevarán a la obtención del ejecutable final del proyecto. Han logrado incorporar el uso de los servicios de infraestructura provistos por la plataforma SOA del Link-all y definir los servicios que componen el Help-Desk, así como realizar la orquestación de servicios necesaria para cumplir con el proceso principal del Help-Desk, la Gestión de Incidentes, implementada en JBMP.

Si bien resulta temprano para evaluar la efectividad de la metodología definida, se detectó que debido a que los estudiantes no tenían experiencia ni conocimiento sobre este enfoque, tuvieron dificultades al momento de realizar la identificación y categorización de servicios. Este aspecto sin embargo, se menciona en la literatura como una de las partes más difíciles en la orientación a servicios, sobre todo en lo que tiene que ver con la identificación de la granularidad de los servicios, y la aplicación de principios de diseño como el de generalidad y anticipación al cambio para definirlos. Por otro lado la identificación de servicios a partir de los subsistemas de Software identificados en la Arquitectura plantea dificultades, así como la asignación de componentes a los servicios.

6 Conclusiones y trabajo futuro

El sistema Link-all fue implementado por varios grupos de desarrollo pertenecientes a varios países de América del Sur y Europa, por lo que era importante mantener un bajo acoplamiento de manera de minimizar los problemas. Por esta razón el sistema fue desarrollado siguiendo un enfoque Process-Enabled SOA, en el que los servicios básicos fueron desarrollados para proveer funcionalidades de infraestructura que permitan integrar las aplicaciones que resuelvan los procesos de negocios. En este contexto se desarrolló una metodología acorde a este tipo de enfoque, la que fue puesta en práctica en el desarrollo de la aplicación Help-Desk para el Sistema Link-all.

Como resultado de la experiencia esta previsto orientar el desarrollo de la nueva versión del sistema Link-all hacia esta metodología. En la misma línea se prevé desacoplar totalmente los servicios de la capa de infraestructura y emplear un motor de orquestación de la misma manera que se hizo en el Help-Desk. También se pretende incorporar directorios de servicios de manera de poder independizar el proveedor del consumidor y así poder contar con una red de nodos Link-all.

En cuanto a la metodología definida para el desarrollo de aplicaciones en plataforma SOA, se la piensa ajustar con los aportes realizados por el desarrollo de los proyectos del Help-Desk. También se prevé incorporarle actividades específicas en las áreas de Verificación, Gestión del Proyecto, Calidad y Configuración, que refuercen la orientación a servicios en el desarrollo, la verificación de los mismos, la planificación y gestión de configuración en base a estos, y el aseguramiento de la calidad de los servicios definidos y de su diseño e implementación.

7 Referencias

- [1] Proyecto Link-all, Programa @lis, Unión Europea. <<http://www.link-all.org/>> [Consulta: diciembre de 2005]
- [2] Triñanes J., "TLREQ: Proceso para desarrollo a distancia", en III Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'03), Valdivia, Chile, p. 49-52, 2003
- [3] Grupo de Ingeniería de Software (Gris), Instituto de computación, Facultad de Ingeniería, Universidad de la República, <<http://www.fing.edu.uy/inco/cursos/ingsoft/gris>> [Consulta: diciembre de 2005]
- [4] IBM Rational Unified Process. <<http://www-130.ibm.com/developerworks/rational/products/rup/>> [Consulta: diciembre de 2005]
- [5] Beck, K. Extreme Programming Explained: Embrace Change, Addison Wesley Professional, 1999, ISBN 201-61641-6
- [6] Delgado, A. Metodología de desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), Agosto 2005, <<http://www.fing.edu.uy/~adelgado/ExtensionSOA/index.htm>> [Consulta: diciembre de 2005]
- [7] Business Process Management Initiative, <<http://www.bpmi.org/>> [Consulta: diciembre de 2005]
- [8] Krafzig, D. Banke, K. Slama, D. "Enterprise SOA, Service Oriented Architecture Best Practices" Prentice Hall, 2005, ISBN 0-13-146575-9
- [9] IBM et al, "Business Process Execution Language for Web Services(BPEL4WS), Version 1.1", Mayo 2003.
- [10] J2EE, <<http://java.sun.com/j2ee/>> [Consulta: diciembre de 2005]
- [11] JBoss AS, <<http://www.jboss.com/products/jbossas>> [Consulta: diciembre de 2005]
- [12] PostgreSQL, <<http://www.postgresql.org>> [Consulta: diciembre de 2005]
- [13] Apache Struts, <<http://struts.apache.org>> [Consulta: diciembre de 2005]
- [14] Apache AXIS, <<http://ws.apache.org/axis/>> [Consulta: diciembre de 2005]
- [15] Eclipse, <<http://www.eclipse.org>> [Consulta: diciembre de 2005]
- [16] Delgado, A. Pérez, B. Modelado del proceso de software - Informe final Proyecto de Taller V, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, año 2000
- [17] Visión General, ARTech, Febrero 2003, <http://www.genexus.com/DOCUM/GeneXus_VG.pdf> [Consulta: diciembre de 2005]
- [18] Proyecto de Ingeniería de Software, Instituto de computación, Facultad de Ingeniería, Universidad de la República, <<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/>> [Consulta: diciembre de 2005]
- [19] JBPM, <<http://www.jboss.com/products/jbpm>> [Consulta: diciembre de 2005]
- [20] AJAX, <<http://es.wikipedia.org/wiki/AJAX>> [Consulta: diciembre de 2005]
- [21] Jabber, <<http://www.jabber.org>> [Consulta: diciembre de 2005]
- [22] Jive, <<http://www.jivesoftware.org/messenger/>> [Consulta: diciembre de 2005]