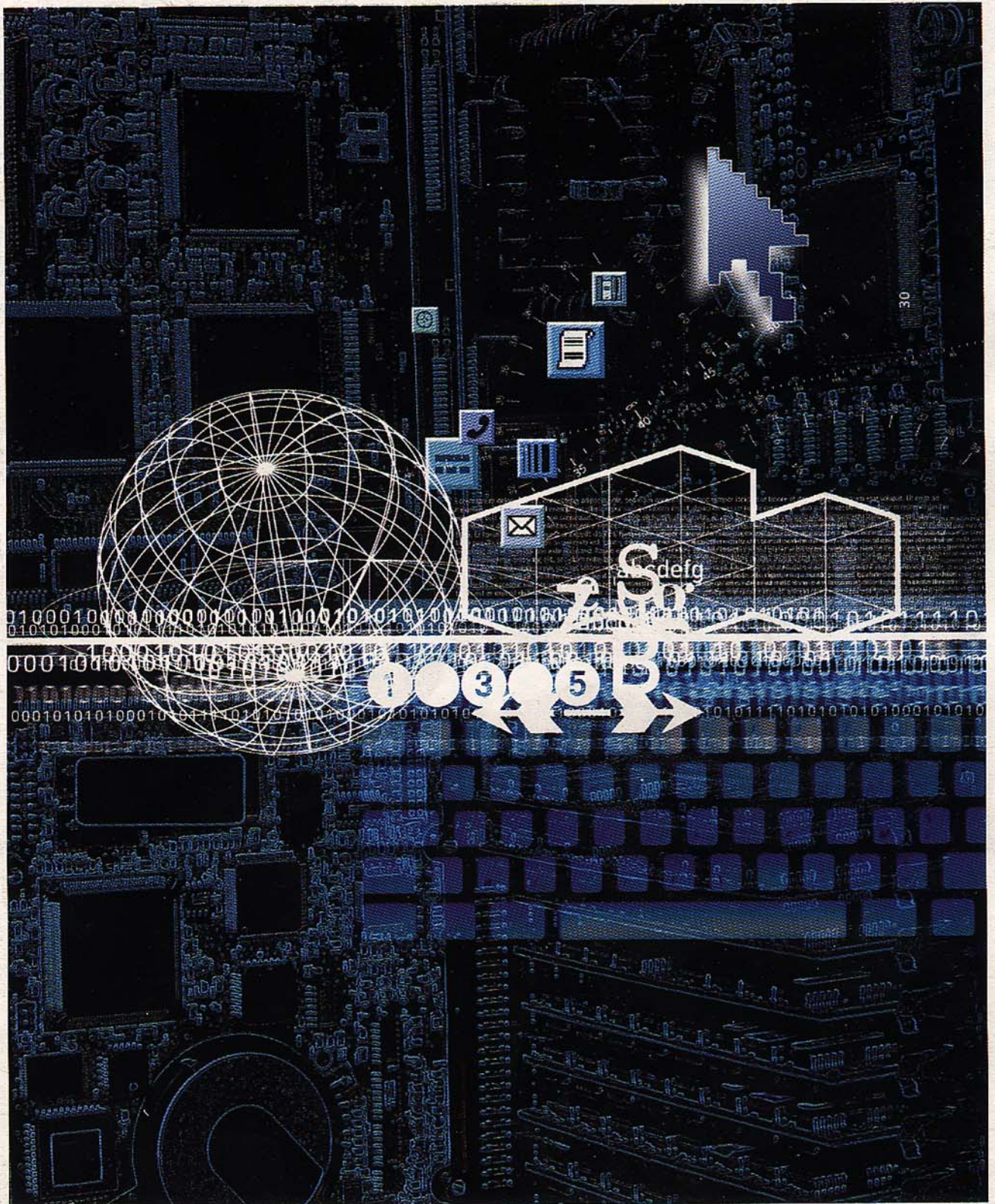


1^{er} Trimestre de 1999

REVISTA DE TELECOMUNICACIONES DE ALCATEL



SOFTWARE: EL MOTOR DE NUEVAS SOLUCIONES INNOVADORAS

La Revista de Telecomunicaciones de Alcatel es una publicación técnica de Alcatel que presenta de manera rigurosa sus investigaciones, desarrollos y productos en todo el mundo.

CONSEJO EDITORIAL

Peter Radley
Presidente

Philippe Goossens
Editor Jefe

Catherine Camus
Editora Jefe Adjunta
Directora de la edición francesa, París

Mike Deason
Director de la edición inglesa, París

DIRECTORES

Emmanuel Darmois
Asesor Editorial

Andreas Ortelt
Director de la edición alemana, Stuttgart

Gustavo Arroyo
Director de la edición española, Madrid

Isabelle Liu
Director de la edición china, Beijing

Ann Paulsrud
Asistente editorial

Las direcciones de las oficinas editoriales pueden encontrarse en la última página de la Revista.

En esta publicación no se hace ninguna mención a derechos relativos a marcas o nombres comerciales que puedan afectar a algunos de los términos o símbolos utilizados. La ausencia explícita de dicha mención no implica, sin embargo, la falta de protección sobre esos términos o siglas.

Revista técnica editada por Alcatel España, S.A.
Domicilio social: c/ Ramírez de Prado, 5
28045 Madrid, España
Depósito legal: M21998/1998
ISSN: en curso
Imprime: COBRHI, S.A.
Edición española: 8.000 ejemplares
© Alcatel España, S.A.

REVISTA DE TELECOMUNICACIONES DE ALCATEL

1^{er} Trimestre 1999

SOFTWARE: EL MOTOR DE NUEVAS SOLUCIONES INNOVADORAS

2 **Editorial**
A. BRAVO

4 **Introducción**
E. DARMOIS

Mejora de los Procesos de Software

6 **Gestión del cambio de procesos en conmutación**
Ch. Ebert

9 **Reducir plazos en el desarrollo del software**
Ph. Thirionet

11 **Análisis de puntos de función para estimar la fiabilidad del software orientado a objetos**
G. Desoblin

13 **Especificación del uso y pruebas estadísticas**
A. Beck

16 **Una verificación formal de interacciones entre facilidades**
J. Leneutre, R. Tingaud

Eficiencia en Producción de Software

19 **Nueva disponibilidad de sistemas y servidores de telecomunicaciones**
D. Rasseneur, Ph. Richard

28 **Estrategia de la plataforma de software para gestión de red avanzada**
M. Uszynski, P. Wardé

37 **Nuevos servicios de información multimedia**
J-D. Chillet, P. Vignard

44 **Mejora de los servicios suplementarios del Alcatel 1000 S12, usando Java y tecnologías Internet**
Y. Depré, H. Vanderstraeten, W. Van Leekwijck

53 **Entorno de creación de servicios: flexibilidad, apertura y evolución**
M. Genette

Arquitecturas Avanzadas de Software

60 **Arquitecturas software, líneas de producto y entornos de trabajo**
G. Donnan, J. Jourdan

67 **Tecnología de agentes y sus aplicaciones**
F. Carrez

77 **Glosario**

78 **Abreviaturas de este número**

Si desea recibir más información sobre cualquiera de los temas de este número, contacte por favor con emmanuel.darmois@alcatel.fr

**ALCATEL**



A. BRAVO

EDITORIAL

SOFTWARE: EL MOTOR DE SOLUCIONES INNOVADORAS

■ Nuevas Necesidades, Nuevas Soluciones

Las (r)evoluciones que se están produciendo en el mundo de las telecomunicaciones están dirigidas a satisfacer los deseos de los usuarios por alcanzar nuevas formas de vida, de organización del trabajo y de relación social. Son lo suficientemente visibles como para mencionar sólo las más importantes: la eclosión de Internet y la convergencia de voz y datos; la emergencia de los móviles y la convergencia fijo/móvil; la multiplicación de terminales y de modos de acceso (por ejemplo, los satélites). Todos estos cambios responden a la demanda de los usuarios de nuevos servicios y conducen a un crecimiento constante de la complejidad de los sistemas.

Por otro lado, estos cambios se producen a diferentes velocidades según el país o la región geográfica que se trate, las modalidades de liberalización o desregulación de los distintos mercados, y la naturaleza de los actores ya instalados y/o emergentes. Es útil, por ejemplo, observar como el notorio crecimiento de los segmentos de datos y de móviles cursa en paralelo con un incremento, un poco menor, del número de usuarios conectados a segmentos fijos, lo cual no es, por otra parte, significativo en términos de tráfico. También hemos de tomar conciencia de que todos los segmentos de mercado requieren nuevas soluciones, lo que au-

menta la complejidad y los costes potenciales de implantación.

Como protagonistas de primer nivel en la industria de las telecomunicaciones, Alcatel está permanentemente decidida a tener en cuenta la amplia diversidad de las demandas y necesidades de sus clientes, con el fin de proponerles soluciones globales y avanzadas. Una característica común a estas soluciones es que todas ellas se basan en la integración progresiva de la tecnologías más avanzadas en el campo del software.

■ Nuevas Oportunidades

Hoy día, el software ocupa un lugar dominante en los sistemas de telecomunicaciones, un protagonismo que va más allá de lo que podría suponer una simple incorporación de funcionalidades a los equipos. Ostenta un papel esencial, por un lado, en la diferenciación entre productos (incluso, aunque las prestaciones intrínsecas se suelen suministrar por equipos cada vez más normalizados) y, por otro, en la diferenciación de las diversas soluciones (al asegurar un mayor nivel de apertura entre los sistemas).

Su introducción masiva en equipos, redes y aplicaciones se traduce en la existencia de una base instalada de software cada vez mayor y, desde este punto de vista, Internet no hace más que potenciar este fenómeno.

El primer reto es desplegar el software en el momento oportuno y con la calidad requerida. En particular, el software tiene que ser probado y esta es una de las principales cualidades -que aumenta conforme crece la base instalada- a tener en cuenta por los equipos de desarrollo. Por supuesto, esto también es cierto para los grandes programas de software informáticos como, por ejemplo, los sistemas operativos. Sin embargo, aunque la fiabilidad sigue siendo, con razón, una de las principales preocupaciones de los operadores, la necesidad de introducir nuevos servicios lo más rápidamente posible impone, cada vez más, el concepto de calidad gradual.

Un segundo reto, que quieren hacernos olvidar aquellos que proponen soluciones "sencillas", es el del "ciclo de vida" de un sistema desde su instalación hasta su retirada. Las convergencias entre voz y datos, fijo y móvil, y telecomunicaciones y multimedia, ofrecen ejemplos de gestión compleja de software avanzado para los cuales debemos de ser capaces de ofrecer soluciones diferenciales y apropiadas, según el tipo de red involucrada.

Además, la optimización del desarrollo de software evoluciona hacia la reutilización de funciones existentes para desarrollar nuevos servicios. La Red Inteligente está basada sobre una arquitectura global que puede ajustarse a la medida de servicios con diversas dimensiones (por ejemplo, abonados, rendimiento, etc.), lo cual facilita la creación de nuevos servicios

(mensajería, prepago, etc.), que, a su vez, pueden ser utilizados en contextos más amplios (principalmente en Internet).

La gestión de software significa también ser capaces de desplegar funciones en diferentes localizaciones de la red. Las arquitecturas software están jugando un protagonismo clave en la tercera generación de servicios de movilidad, particularmente en la generación del Entorno Virtual de Hogar (VHE, Virtual Home Environment).

■ Nuevos Impulsos

El software es uno de los principales motores de crecimiento para Alcatel, y su desarrollo es uno de los procesos funda-

mentales que se extienden desde la definición del producto hasta su despliegue en el mercado. Por esta razón, Alcatel lleva, desde hace algunos años, desarrollando un programa de gestión de calidad. Las constantes mejoras y los éxitos alcanzados se deben al hecho de que los ingenieros dedicados a la investigación y desarrollo trabajan en un nuevo marco cultural que combina excelencia y cambio, con complejidad e innovación.

La puesta al día de la gestión de software supone también anticiparnos a los cambios tecnológicos, al trabajo sobre nuevas soluciones (sistemas, algoritmos y metodologías de desarrollo) y disponer las transferencias necesarias entre equipos de desarrollo. Alcatel está realizan-

do un importante esfuerzo de investigación para llegar a analizar y dominar las tecnologías básicas de software. Los términos claves en este campo son: arquitecturas, servidores, informática distribuida, objetos y componentes, lenguajes de descripción y middleware. En este número de la *Revista Telecomunicaciones* se repasan estos conceptos y el constante esfuerzo técnico por la innovación en estas áreas.

Con este espíritu de convergencia como telón de fondo, el esfuerzo de Alcatel aboga por la estrecha integración de los mundos de la informática y las telecomunicaciones tanto en sus productos como en su organización. Esta es nuestra elección estratégica.



Alain Bravo
Director Técnico de Alcatel

INTRODUCCIÓN

E. DARMOIS

La experiencia en procesos y tecnologías software permite ofrecer a los usuarios sistemas y servicios diferenciados de gran calidad.

■ Introducción

Nunca como hasta ahora los servicios de telecomunicaciones habían cambiado y evolucionado con tanta rapidez; diariamente observamos y leemos los signos de esta transformación y, sobre todo, utilizamos sus resultados: Internet, movilidad y nuevos servicios. En todos los casos, el software está en el corazón de estos cambios, contribuyendo de manera significativa al valor añadido de los sistemas de (tele)comunicaciones.

El software es una tecnología básica utilizada para resolver problemas extremadamente variados y para ofrecer soluciones a, virtualmente, infinitas permutaciones de problemas. Aunque diferentes en forma, las soluciones software se parecen en que todas ellas requieren conocer cómo usar innovadoras tecnologías y procesos de desarrollo. El software también es un medio para desplegar las nuevas aplicaciones y servicios que el mercado de las telecomunicaciones demanda. La extraordinaria flexibilidad proporcionada por el software implica que las nuevas e innovadoras soluciones se pueden concebir y desplegar más rápidamente, y que nuevos criterios de diferenciación y de valor pueden ser creados para nuestros clientes. Este número de la *Revista de Telecomunicaciones de Alcatel* examina algunos aspectos de los dos objetivos clave del software: marcar el territorio y construir algo nuevo.

■ Excelencia

Los sistemas de comunicaciones se encuentran entre los más complejos de im-

plementar y actualizar para cualquier experto. Frente a esta complejidad, que tiende a crecer a lo largo del tiempo, el proceso de desarrollo del software necesita una constante mejora. Para Alcatel, este es uno de los procesos nucleares de la compañía y, de forma global, se ha establecido una clara y duradera estrategia de mejoras. En una serie de artículos cortos revisamos y debatimos algunas de las líneas de evolución que Alcatel se propone utilizar de manera general allá donde sean más eficaces.

La constante adaptación de los procesos de desarrollo de grandes sistemas, en un esfuerzo por incrementar tanto su calidad como su capacidad de respuesta, se trata en el artículo *Gestión de los Cambios de los Procesos en Conmutación* que describe los resultados obtenidos en este sector clave, donde el tamaño del software y de los equipos de desarrollo involucrados es de importancia vital. En él se hace énfasis en dos mejoras de la calidad. La primera, es un método global de verificación con el se pueden detectar, en la fase más temprana, los errores de software reduciendo significativamente su número al final de un ciclo. La segunda, es un preciso control de los requisitos sobre la base de nuevas funciones, por lo que se pueden reducir los tiempos de desarrollo y hacer que los nuevos servicios estén disponibles lo más rápidamente posible.

Se aporta también un breve repaso de tres aspectos complementarios al propósito de mejorar los procesos de desarrollo del software: reducir los tiempos de desarrollo, la evaluación previa del esfuerzo de desarrollo según el tipo de funciones requeridas, y el uso de técnicas estadísticas en las pruebas.

Finalmente, la descripción de los métodos formales aplicados en *Una Verificación Formal de Interacciones entre Facilidades* muestra los beneficios que se esperan de la utilización de formalismos matemáticos para demostrar las complejas propiedades de protocolos o sistemas en la fase de especificación, muy pronto en el ciclo de vida de desarrollo del software.

■ Eficacia

Para desplegar los nuevos servicios más rápidamente, se debe mejorar la eficacia en el desarrollo del software en dos direcciones complementarias. El software eficaz y escalable debe basarse en arquitecturas abiertas y flexibles para que las facilidades se puedan ofrecer en forma de servidor. Estas arquitecturas software, como las plataformas de ejecución, subyacen en la misma esencia de la experiencia de Alcatel. Igualmente crucial es la necesidad de contar con plataformas que ofrezcan la calidad requerida de servicio y que incorporen los avances más evolucionados de la industria de las tecnologías de la información. El enfoque de Alcatel en este campo se resalta en el artículo *Nueva Disponibilidad de Sistemas y Servidores de Telecomunicación*.

La gestión de red es otro campo donde se necesita un desarrollo racionalizado de software: se tienen que gestionar un gran número de elementos de red diferentes, mientras se ofrecen funciones genéricas (gestión de alarmas, gestión de configuración, etc.). El artículo *Estrategia de la Plataforma de Software para Gestión de Red Avanzada* describe un método racional para desarrollar aplicaciones basadas en una

plataforma común. En particular, esto abre el camino a la máxima utilización de las técnicas de generación automática de código, que mejoran significativamente la productividad y fiabilidad del software.

■ Flexibilidad

El uso intensivo de la flexibilidad que el software proporciona significa que se pueden ofrecer nuevos servicios a los usuarios con nuevos requisitos (información personalizada, movilidad, comercio electrónico, etc.). Esto presupone la capacidad de definir, muy pronto en el ciclo de vida, la naturaleza del servicio, su funcionalidad (su amigabilidad, en particular), la arquitectura software subyacente y todas las tecnologías requeridas (técnicas multimedia, lenguajes, middleware, etc.).

Dos ejemplos ilustran esta exploración de los nuevos servicios. El artículo *Nuevos Servicios de Información Multimedia* analiza como, mediante la aplicación de diferentes tecnologías (especialmente, la multimedia), los usuarios pueden obtener información personalizada en diferentes entornos y sobre distintos terminales. En *Mejora de los Servicios Suplementarios del Alcatel 1000 S12, usando Java y las Tecnologías Internet* se muestra como la definición de una arquitectura de tres ni-

veles ofrece al usuario una nueva y sencilla forma de definir los requisitos mediante una interfaz web normalizada.

La flexibilidad exigida a los servicios puede conllevar una potencial proliferación de desarrollos divergentes. Para evitar esta explosión, es esencial disponer de un entorno de creación genérico que sea fácil de usar y fiable. El entorno de creación de servicios de Alcatel fue uno de los primeros en satisfacer estos requisitos en el marco de las Redes Inteligentes. La forma como ha evolucionado gradualmente para tener en cuenta los nuevos requisitos, al tiempo que se aseguran las inversiones existentes, se describe en *Entorno de Creación de Servicios: Flexibilidad, Apertura y Evolución*.

■ Evolución

Como se ha mencionado anteriormente, las arquitecturas software han sido cruciales para la escalabilidad de sistemas software intensivos. En los últimos años se han sucedido importantes cambios con la emergencia de la Informática de Objetos Distribuidos, mediante la cual se está comenzando a realizar un método orientado a componentes software, similar al empleado en el hardware. Se espera obtener los mismos beneficios que los logrados en el hardware.

Los dos últimos artículos de la Revista exploran dos facetas de la gestión de arquitecturas. En *Arquitecturas Software, Líneas de Producto y Entornos de Trabajo* el objetivo es el concepto de "línea de productos" y en cómo se adapta dentro de arquitecturas más flexibles y en diferentes plataformas. Este trabajo se está llevando a cabo en un laboratorio conjunto recientemente creado por Alcatel y Thomson CSF. Finalmente, con los "agentes" (agentes móviles, agentes inteligentes) descritos en *Tecnología de Agentes y sus Aplicaciones*, vemos el esquema del nuevo método global de la arquitectura de aplicaciones, cómo se comunican y cómo cooperan.

Las arquitecturas, componentes y el middleware son los ingredientes básicos de los sistemas de comunicaciones del mañana. El objetivo es ofrecer a los usuarios nuevas formas de comunicar, intercambiar o expresar requisitos y nuevos lenguajes para programar sistemas y producir software sin que haga falta, incluso, pensar en ello.

Emmanuel Darmois es jefe del Departamento de Software en el Corporate Research Center de Alcatel, en Marcoussis, Francia.

GESTIÓN DEL CAMBIO DE PROCESOS EN CONMUTACIÓN

C. EBERT

■ Introducción

Debido a que el Modelo de Capacidad de Madurez (CMM) suministra las normas para identificar los puntos fuertes y débiles del proceso de desarrollo del software y un plan de desarrollo para mejoras, la División de Sistemas de Conmutación (SSD) de Alcatel lo ha adoptado para sus actividades de Mejora del Proceso Software (SPI) a aplicar a las centrales digitales Alcatel 1000 S12. Aunque el CMM sirve como entorno para una evaluación repetible y para una planificación enfocada a acciones, la ventaja real se encuentra en el enfoque para alcanzar los objetivos externos e internos del negocio. En todos los

principales centros de desarrollo se realizan evaluaciones periódicas. En línea con el CMM, los resultados encontrados se analizan de acuerdo a sus impactos futuros en los objetivos de negocio de la SSD y, después, se los da prioridad en función de su mejora potencial. Basándose en esta clasificación, el plan concreto de mejoras se refina continuamente, dando lugar a un plan de acción con descripciones detalladas de los métodos de mejora con responsabilidades definidas, con estimaciones de esfuerzos, etc. También se hacen repetidamente evaluaciones del proceso de desarrollo y sus interfaces con consultores externos para seguir la implantación del plan de mejora.

■ Implantación del SPI

El programa SPI se implantó en paralelo con cambios en la organización para optimizar todos los procesos de la ingeniería. Este cambio rompió las antiguas estructuras y los equipos existentes. Las áreas de competencia, el trabajo en equipo y una potente organización por proyecto sustituyeron el enfoque basado en un inadecuado desarrollo sincronizado. De repente, el desarrollo se empezó a hacer de acuerdo a procesos definidos y a herramientas normalizadas. Los procedimientos informales se sustituyeron por reglas clave y por descripciones de procesos genéricos, se siguieron las des-

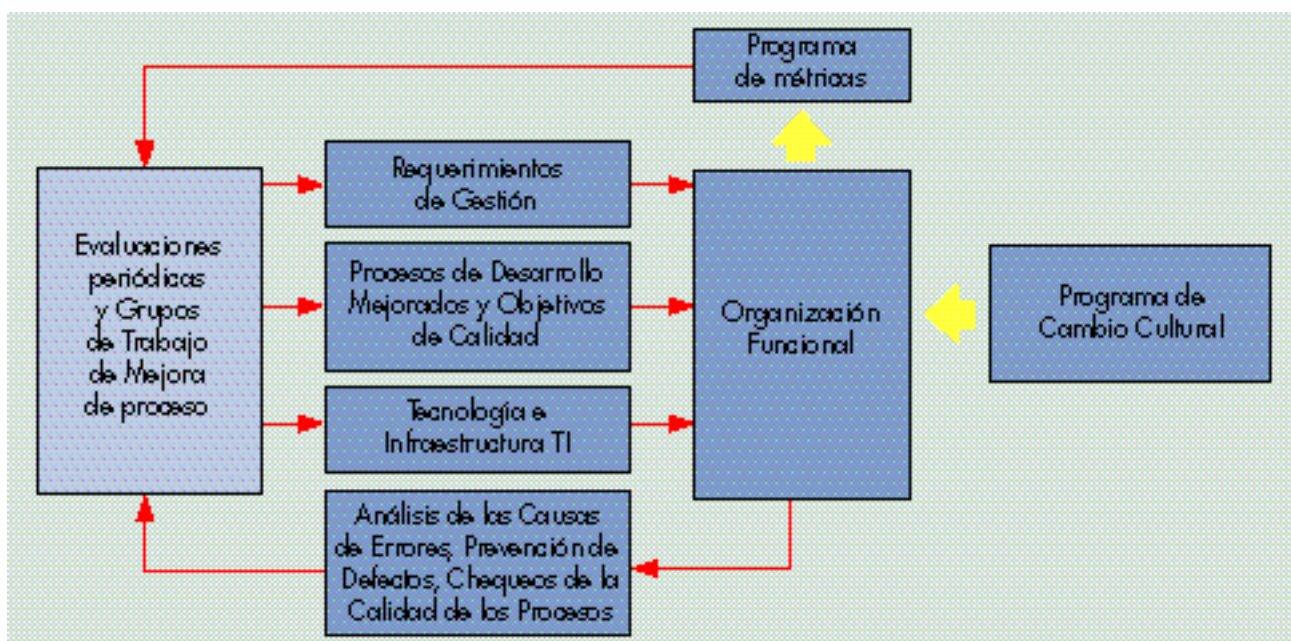


Figura 1: Las actividades de Mejora del Proceso Software en el Alcatel 1000 S12 ponen énfasis en la continua realimentación de los proyectos.

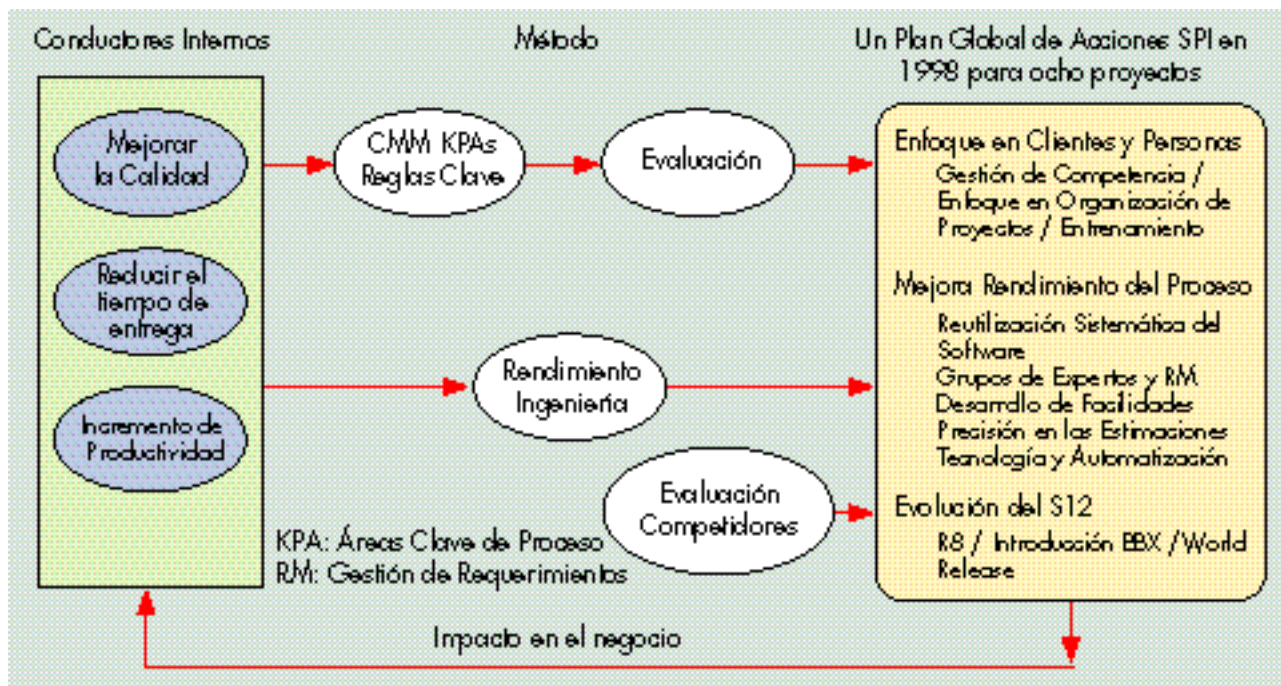


Figura 2: El Plan de Acción SPI de 1998 para el Alcatel 1000 S12 une los conductores internos del negocio con acciones concretas de mejora.

viaciones de los mismos y los directores se midieron contra ellos mismos. El plan de todo el programa SPI con sus relaciones en el desarrollo se muestra en la **Figura 1**.

Desde su inicio, el programa SPI vio el CMM como un medio de mejora, no como un fin. Las acciones de mejora se basan en los resultados de la evaluación para ayudar a alcanzar todos los objetivos de negocio. El plan de acción anual del SPI muestra como se obtiene esta unión con los objetivos de negocio y la evaluación frente a los competidores (**Figura 2**).

■ Situación Actual

El objetivo inicial, en 1995, fue mejorar la calidad y poner bajo control los proyectos. Esto ya casi se ha conseguido, y todavía se está mejorando la detección temprana de defectos. El Nivel 2 del CMM, que de hecho define las áreas clave del proceso para alcanzar el control del proyecto, se consiguió a comienzos de 1998. Hoy el enfoque no se hace sólo en la calidad y en la reducción del tiempo de desarrollo, sino que también se pone en la mejora sustancial del rendi-

miento de la ingeniería. Los análisis de nuestros clientes y de su situación competitiva y la evaluación frente a los competidores y las compañías líderes de software indican que la eficiencia y los ciclos cortos de respuesta son factores claves! El plan de acción de 1998 (**Figura 2**) tiene en cuenta este cambio de objetivos y dirige nuestros esfuerzos hacia los conceptos: "desarrollo del proceso por facilidades" y "world release". Los principales paradigmas son el desarrollo incremental, el trabajo en equipo y los costes a medida reducidos. Actualmente, el Alcatel 1000 S12 está dedicado totalmente a implantar estos principios.

■ Recursos para el Desarrollo

Aproximadamente el 5% del esfuerzo total en el Alcatel 1000 S12 se utiliza en actividades tales como control de procesos, pilotaje, entrenamiento, mejora de herramientas o actividades mejoradas de seguimiento. El retorno de la inversión se alcanza sólo examinando la detección temprana de defectos. Tal como se mencionó más arriba, los chequeos de la evaluación y el control técnico de

todos los proyectos en desarrollo -para seguir la conformidad del proceso y la disponibilidad de las salidas y su calidad- es el núcleo de todo el programa de mejora. En lugar del establecimiento de Grupos de Proceso Software de Ingeniería (SEPG) sobre todo para evaluaciones, definimos un programa que se basa en una continua mejora del proceso a largo plazo.

Debido a que el control de cambios de proceso está llegando a ser una función de gestión y a que el proyecto SPI recibe la mayor prioridad entre todos los proyectos de ingeniería, el programa SPI ostenta una autoridad real. Los SEPGs locales son bastante pequeños, no obstante las contribuciones a tiempo parcial de expertos de diferentes áreas ayudan a difundir los mensajes del SPI y asisten en su implantación. Para evitar la actitud predominante que tiene como lema la necesidad de cambiar continuamente, la dirección senior del Alcatel 1000 S12 decidió, desde el comienzo, que cualquier cambio de proceso dentro de la ingeniería (por ejemplo, mayor eficiencia, reducción del tiempo de entrega, etc.) debería estar bajo el paraguas SPI. Todas estas decisiones garantizan la visibilidad de la iniciativa SPI.

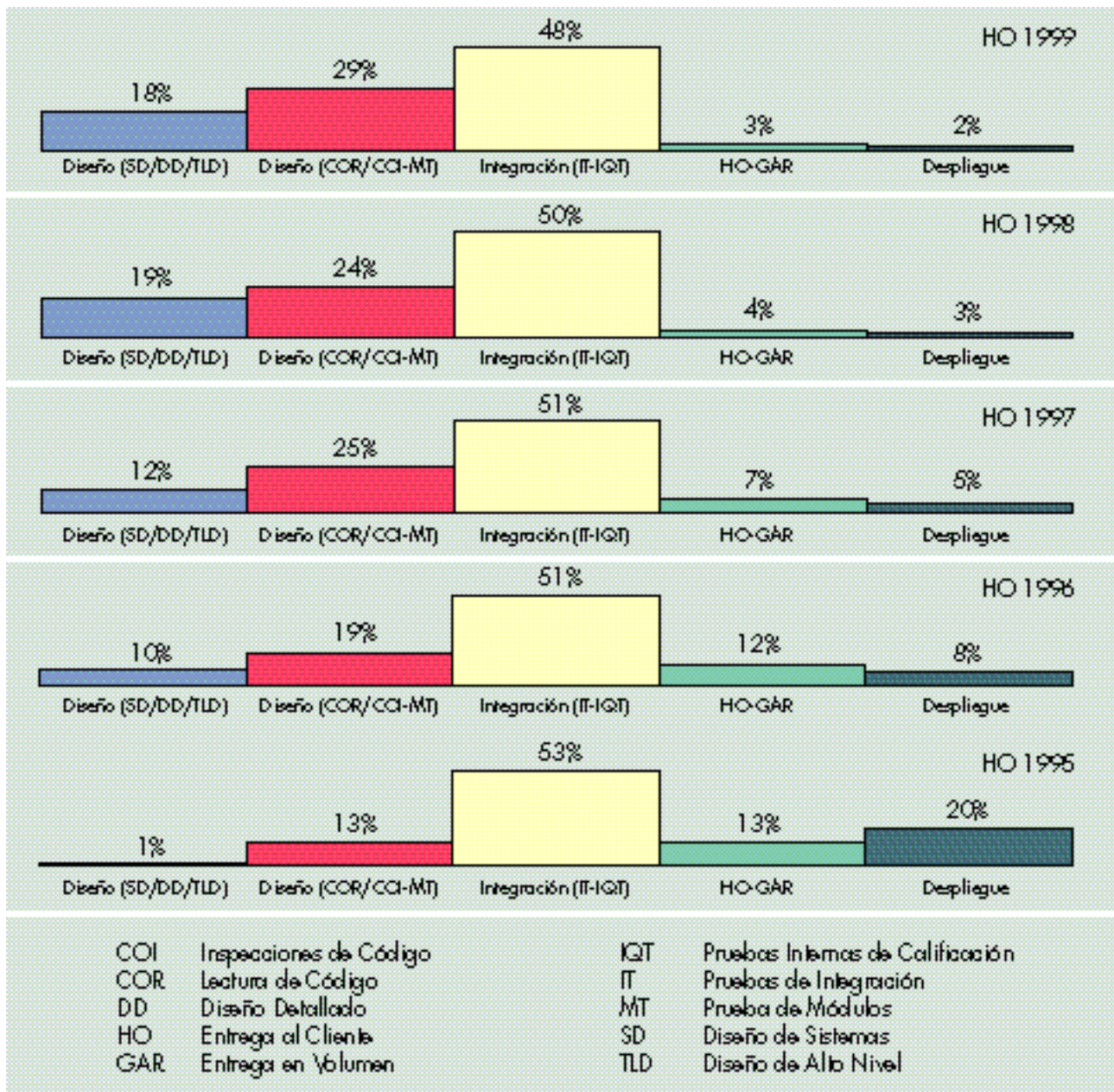


Figura 3: La distribución de defectos en todas las actividades de detección indica que, en la primera fase del SPI en Alcatel 1000 S12, el enfoque estaba en la detección temprana de los defectos. Al mismo tiempo, el número total de defectos está disminuyendo.

Con resultados medibles como los del segundo año (Figura 3) y la vital reorientación de los clientes, empezó la evolución y gradualmente se está desarrollando una nueva cultura.

Christof Ebert es Software Process Manager de la División de Sistemas de Conmutación de Alcatel en Amberes, Bélgica.

REDUCIR PLAZOS EN EL DESARROLLO DEL SOFTWARE

PH. THIRIONET

■ Introducción

El rapidísimo crecimiento de los sistemas de información, la impaciencia de los clientes y, a menudo, ciertos requisitos imprecisos, están convirtiendo el desarrollo de software en un negocio cada día más complicado. Esto aumenta considerablemente los riesgos de fallo, y uno de tales riesgos es el de no ser capaces de entregar a tiempo el producto solicitado.

Para reducir estos riesgos y el tiempo necesario para desarrollar software, pueden concebirse varias líneas de mejora. Modificaciones en la arquitectura del software, en el proceso de desarrollo y en las herramientas implicadas, pueden aumentar considerablemente la productividad. Además, la reutilización sistemática de componentes software cuidadosamente identificados y diseñados, puede reducir de manera espectacular el esfuerzo, el coste y el tiempo involucrados en el desarrollo de un producto.

Este artículo se centra concretamente en las mejoras potenciales del proceso de desarrollo del software.

■ Proceso de Desarrollo

La mayoría de los desarrollos de software utilizan aún ciclos de vida de Waterfall o del modelo en V [1] convencionales. Estos se adaptan mal a los frecuentes cambios exigidos por los clientes. Además, el final del ciclo de vida [2] debe alcanzarse antes de que pueda entregarse un producto al cliente, hacién-

dose mucho mayores los riesgos de que no sea capaz de realizar todas las funciones en el plazo permitido. Un ciclo de vida iterativo e incremental se adapta mejor a los requisitos cambiantes y proporciona un mejor control de los riesgos involucrados. En este caso, se generan y entregan al cliente varias versiones del software. Cada versión representa un incremento y ofrece un conjunto de facilidades cada vez más completo.

Esto significa que mucho antes, a lo largo del proceso, pueden identificarse los problemas de integración, tenerse en consideración las realimentaciones procedentes de versiones anteriores y suministrarse al cliente una parte de la funcionalidad antes de la fecha comprometida, teniendo en cuenta las prioridades impuestas por el contrato. No obstante, un proceso iterativo e incremental implica un nuevo planteamiento de la arquitectura y de su diseño.

■ Factores Claves

Las *arquitecturas* en capas y divididas no solamente aumentan la portabilidad, flexibilidad y capacidad de mantenimiento del software, también permiten añadir más sencillamente nuevas facilidades en cada versión del producto.

Una estrategia iterativa e incremental debería estar aliada con una determinación técnica de *reutilización* del software en versiones y proyectos posteriores con requisitos similares. Esto significa mirar más de cerca la arquitectura.

Son esenciales las *herramientas* adecuadas para administrar las prioridades de los requisitos, para asegurar la trazabilidad entre los requisitos y los componentes software que los implementan, y las correspondientes pruebas de integración y validación. Por tanto, se necesita una facilidad apropiada de automatización de pruebas debido al gran número de pruebas regresivas que deben hacerse sobre los componentes de versiones ya entregadas al cliente. Además, la gestión eficiente de librerías de componentes reutilizables solamente puede estimular la reutilización sistemática de estos componentes en otros proyectos.

■ Ciclo de Vida Iterativo

La División de Comunicaciones Móviles (MCD) de Alcatel desarrolla infraestructuras del Subsistema de Estaciones Base (BSS, Base Station Subsystem) para la red GSM. En este campo, los cambios en los requisitos del cliente son especialmente significativos y la competencia es tal que los plazos de desarrollo son cada vez más cortos.

Por consiguiente, MCD decidió introducir un ciclo de vida iterativo e incremental para los subsistemas involucrados. Partiendo de un ciclo de vida del modelo en V, el objetivo era definir las actividades asociadas con el software, cómo están ordenadas y sus puntos de sincronización, con el fin de definir un ciclo iterativo e incremental que tuviera en cuenta las condiciones impuestas por los componentes existentes y la viabili-

dad de un cambio de proceso de este tipo. Al mismo tiempo, MCD está estudiando evolucionar desde desarrollos software basados en subsistemas a desarrollos orientados a facilidades. El trabajo fue realizado conjuntamente por especialistas en programación y expertos en GSM y, a partir del mismo, se han identificado un ciclo de vida y un conjunto de acciones encaminadas a mejorar la administración de requisitos y la configuración, automatizando pruebas y organizando los recursos humanos.

■ Conclusiones

La reducción de los plazos de desarrollo del software implica un gran número de factores que deben tenerse en cuenta conjuntamente si se quieren obtener mejoras signi-

ficativas. Además, la migración hacia el desarrollo iterativo e incremental solamente puede ser efectiva en coste si se estudia en las etapas iniciales, prestando atención a *qué ciclo de vida es mejor para el proyecto; si la arquitectura hace suficientemente fácil la creación de versiones; y si la gestión de la configuración es efectiva y funcional*. En un mercado de telecomunicaciones que cambia rápidamente, parece obvio que una respuesta a las demandas de los clientes cuando se habla de software es un ciclo de vida iterativo e incremental.

■ Referencias

- 1 S. McConnell, "Rapid Development: Taming Wild Software Schedules", Microsoft Press, ISBN: 1-556-159005, Julio 1996.

- 2 Un conjunto de actividades software entre el momento en que comienza la versión 1.0 de un sistema (estableciendo los requisitos del usuario) y el momento en que la misma versión se entrega al usuario.

Philippe Thirionet es ingeniero en el Corporate Research Center de Alcatel, en Marcoussis, Francia.

ANÁLISIS DE PUNTOS DE FUNCIÓN PARA ESTIMAR LA FIABILIDAD DEL SOFTWARE ORIENTADO A OBJETOS

G. DESOBLIN

■ Introducción

El Modelo de Capacidad de Madurez (CMM, Capability Maturity Model) es un marco que describe los elementos esenciales para obtener un proceso software efectivo. Está diseñado para ayudar a las organizaciones a mejorar su habilidad para conseguir sus objetivos de negocio, transformando los procesos caóticos en procesos software maduros y disciplinados. Las áreas esenciales de procesos en el Nivel 2 del CMM (el nivel de madurez de una organización software que conlleva un control básico de la gestión, además de la disciplina para hacer repetibles los casos de éxito) se centran en los controles esenciales de la gestión de proyectos, tales como la planificación de un proyecto software, su seguimiento y su supervisión. Una de las actividades primordiales en las fases iniciales en el desarrollo del software es la de producir estimaciones de esfuerzo precisas y fiables para obtener una eficiencia óptima en la gestión del proyecto. Sin embargo, la realización de tales estimaciones se basa fundamentalmente en la habilidad de predecir precisamente el tamaño del producto software y obtener, a partir de este tamaño, el esfuerzo necesario para desarrollarlo. Para mitigar el proceso de estimación e incrementar la fiabilidad de las estimaciones iniciales, hemos construido un modelo de estimación basado en el método de Análisis de Puntos de Función (FPA, Function Point Analysis).

■ Enfoque del FPA

Los Puntos de Función fueron introducidos en 1979 por A. Albretch (IBM) como una forma de medir el tamaño funcional de un producto software analizando los requisitos funcionales y cualitativos.

El método de FPA está compuesto de dos partes basadas en:

- Identificación de los componentes funcionales de interés para el usuario. Dichos componentes son o transacciones lógicas (por ejemplo, entrada, proceso, salida) o conjuntos lógicos de datos. El objetivo es el de cuantificar las funciones contenidas en el software en términos inteligibles para el usuario.
- Evaluación de la complejidad técnica de la aplicación software a construir; por ejemplo, son varios factores combinados que afectan al tamaño del software a desarrollar.

■ Experimentación

Con el fin de validarlo en un contexto de desarrollo software orientado a objetos para telecomunicaciones, el método se ha aplicado en varios proyectos software en su fase final. Concretamente, se aplicó a varios componentes software llamados objetos de Interfaces de Aplicación Programadas (API, Application Programming Interface) para un proyecto que maneja recursos de un terminal GSM. El proyecto se desarrolló en C++ usando la metodología orientada a objetos. El análisis se basó en los documentos de especi-

cación que contenían las descripciones funcionales de los objetos API.

Hemos tenido también que desarrollar un modelo de estimación adaptado al contexto de las telecomunicaciones ya que, en el modelo original, la definición y las reglas de especificación de los componentes funcionales se desarrollaron para aplicaciones en el campo de los negocios.

Por ejemplo, se ha incrementado la complejidad de algunos componentes de datos. La complejidad de un objeto API puede variar según la naturaleza de las entidades (objetos o aplicaciones) con las cuales se comunica.

■ De los Puntos de Función al Esfuerzo

La última parte de nuestro trabajo consistió en convertir los puntos de función en el esfuerzo requerido para desarrollar los objetos API. El esfuerzo estimado comprendía el necesario para su desarrollo, para probarlo y para producir la documentación relacionada.

Primeramente, usamos tablas estadísticas para determinar los rangos del factor de productividad. El análisis de varios objetos API nos permitió refinar posteriormente dichos valores de productividad dependiendo de nuestro contexto.

■ Discusión de los Resultados

Se produjeron estimaciones de esfuerzo para una docena de componentes softwa-

re API. Todas las estimaciones presentaban un límite inferior (estimación optimista), un valor medio y un límite superior (estimación pesimista). Para cada uno de los objetos considerados el esfuerzo gastado para desarrollarlo estuvo dentro de los límites estimados, salvo una excepción. Así, la precisión de la estimación fue de un 87%.

Según los factores de productividad que establecimos, fuimos capaces de refinar el rango de estimación entre los valores pesimistas y optimistas de tal modo que, en la mayoría de los casos, la estimación obtenida no difirió del esfuerzo real gastado en más de un 30%.

Una gran ventaja del método de FPA es que establece un framework riguroso con definiciones precisas y reglas que limitan las desviación entre las estimaciones.

■ Conclusiones

Nuestro modelo de estimaciones basado en los Puntos de Función parece idóneo para producir estimaciones iniciales de esfuerzo precisas para componentes software de telecomunicaciones orientados a objetos, usando las especificaciones de los mismos como una única entrada. Ob-

viamente, su precisión radica en la calidad de la documentación usada para producir la estimación.

Podemos ahora extender el modelo automatizando el proceso de estimación y diseñando una base de datos histórica para capitalizar la experiencia obtenida en los proyectos software de telecomunicaciones orientados a objetos.

Gilles Desoblin es ingeniero en el Corporate Research Center de Alcatel, en Marcoussis, Francia.

ESPECIFICACIÓN DEL USO Y PRUEBAS ESTADÍSTICAS

A. BECK

■ Introducción

Cualquier iniciativa con éxito sobre Mejora del Proceso del Software (SPI, Software Process Improvement) debería dar como resultado programas de mayor calidad. Además de cualidades tales como funcionalidad, eficacia y facilidad de uso, la fiabilidad es un tema importante. La fiabilidad, que está estrechamente relacionada con el número de defectos en el sistema software, se consigue principalmente a través de un diseño y una implementación apropiados. Sin embargo, la fiabilidad puede mejorarse mediante las pruebas adecuadas. Las pruebas son esenciales para la verificación de un sistema. Debido a la calidad mejorada del software, el enfoque de las pruebas ha pasado de la detección de defectos a la certificación de la calidad. No obstante, las pruebas exhaustivas han llegado a ser demasiado caras debido a la creciente complejidad del software que sufre el dilema de "cobertura frente a coste". Una aproximación prometedora es la de las pruebas estadísticas basadas en un modelo del uso, un método adoptado de la metodología Cleanroom [1].

El principal objetivo del experimento Especificación del Uso y Pruebas Estadísticas (USST, Usage Specification and Statistical Testing) fue investigar la aplicabilidad de las pruebas estadísticas basadas en un modelo del uso en el contexto de proyectos de telecomunicaciones a gran escala. USST se desarrolló para mejorar la fiabilidad del sistema entregado. Al concentrar los esfuerzos de las pruebas sobre su uso previsto, USST

ayuda a limitar los costes totales. Se esperaba que las pruebas estadísticas permitieran una predicción de la fiabilidad del sistema.

■ Ideas Básicas

Los desarrolladores de programas y los clientes normalmente perciben la fiabilidad del software de formas diferentes: las personas que desarrollan tienden a fijarse en las densidades de defectos (defectos por cada mil sentencias), mientras que los clientes están interesados principalmente en tasas de fallos (fallos por ejecución). Esto tiene que tenerse en cuenta cuando se diseñan casos de prueba. Con el fin de valorar la fiabilidad de un sistema, no es suficiente (algunas veces incluso no es necesario) alcanzar la cobertura completa del código. No obstante, es vital cubrir el uso del sistema.

En todos los sistemas reales de un tamaño razonable, la cobertura completa de las posibles ejecuciones no es ni posible ni económica, por lo que es inevitable el muestreo. Nunca se podrá estar completamente seguro de que el programa no contiene fallos o de que no fallará nunca. Por consiguiente, existen dos tipos de casos de prueba: casos de prueba explícitos, que se ejecutan durante las pruebas, y casos de prueba implícitos, que se ejecutan a través del uso real y diario del sistema. Por desgracia, la fiabilidad del sistema entregado se determina principalmente por estos últimos.

Puesto que solamente puede ejecutarse una pequeña parte de los posibles

casos de prueba, la fiabilidad del programa se determina por la probabilidad de que pueda fallar uno de los casos de prueba no ejecutados. Se necesita una deducción estadística para razonar sobre esto último; para estimar la fiabilidad se necesita una muestra estadística. Se pueden imaginar varias soluciones posibles para seleccionar un conjunto representativo de casos de prueba:

- La selección aleatoria de casos de prueba entre todas las posibles ejecuciones del programa no es ni posible ni eficiente, debido al muy elevado número de posibles ejecuciones. No tiene sentido crear miles de casos de prueba y pasar solamente unos pocos.
- La creación aleatoria de casos de prueba a partir de todas las entradas posibles conduce a "Monkey Tests" que son difícilmente realistas.
- La creación de un modelo del uso del programa y la generación aleatoria de casos de prueba a partir de este modelo, produce casos de prueba aleatorios y realistas a la vez. Para crear un modelo de este tipo pueden aplicarse Cadenas de Markov.

Una Cadena de Markov puede describirse como un diagrama de estados con probabilidades asignadas a todas las transiciones. Este concepto es familiar a los ingenieros de programación y proporciona un modelo intuitivo. Cada estado del modelo corresponde a un estado en el uso del sistema. A partir del modelo pueden generarse casos de prueba aplicando un generador de números aleatorios; los

resultados de la prueba permitirán una estimación de la fiabilidad.

■ Gestión de la Complejidad de los Modelos del Uso

Normalmente, es difícil modelar un sistema con Cadenas de Markov porque tienden rápidamente a ser muy complejas. La única forma de hacer frente a la creciente complejidad es a través de una estrategia de "divide y vencerás". Son posibles varias aproximaciones:

- Abstracción, que significa concentrarse en las cosas esenciales y omitir los detalles que pueden añadirse durante la ejecución de las pruebas;
- Estratificación, que significa crear modelos diferentes para aspectos diferentes del sistema;
- Modelos jerárquicos.

La aproximación jerárquica es muy potente. El uso del sistema se describe primero de una forma sumamente resumida. En el segundo paso, las transiciones entre estados se amplían, donde sea apropiado, a modelos del uso. Este es un proceso recursivo. En cada nivel se añaden más detalles, lo que conduce a un árbol de modelos del uso.

Por desgracia, no existe un soporte de herramientas para manejar modelos jerárquicos. Para trabajar con ellos, pueden ampliarse los modelos o pueden generarse casos de prueba para submodelos y después ampliarse a casos de prueba completos.

Una forma especial de ampliar los casos de prueba es la combinación de Cadenas de Markov y expresiones gramaticales. Esta solución funciona bien para manejar órdenes del operador. Se comienza con un modelo del uso de alto nivel en el que cada estado describe un estado del objeto manejado. [Ejemplo: un número de abonado telefónico puede ser inexistente, estar asignado a un abonado analógico, estar asignado a un abonado a la red RDSI (Red Digital de Servicios Integrados) o a una PABX, etc.]. Los casos de prueba generados por un modelo de este tipo son completamente de alto nivel. Se amplían a casos

de prueba ejecutables por medio de una expresión gramatical que describe las posibles órdenes del operador.

Las cartas de estados son una notación adecuada para máquinas de estados jerárquicos, estados con registros con la historia y estados concurrentes. Simplificarían enormemente los modelos.

■ Pruebas Funcionales

El objetivo de las pruebas funcionales es verificar la corrección de todas las facilidades implementadas. Se detectarían muchos defectos y serían necesarias muchas actualizaciones del programa para corregirlos. Esto hace inútil la aplicación de la teoría de Markov para predicción de la fiabilidad. Sin embargo, las pruebas orientadas al uso son útiles cuando la cobertura es un tema importante durante las pruebas funcionales. La cobertura del código no tiene sentido en esta etapa del proceso de pruebas. Se define cobertura de la prueba como cobertura del modelo. La cobertura del modelo (estados y arcos) puede seguirse fácilmente y el progreso es más significativo que en una lista de pruebas. Para alcanzar la cobertura completa del modelo, normalmente es necesario un elevado número de casos de prueba generados aleatoriamente, la mayoría de los cuales están duplicados. La ejecución de todos estos casos de prueba no es posible. Alternativamente, podría generarse un conjunto mínimo de casos de prueba que cubriera el modelo. Esto último tiene la desventaja de no ser realista. Como compromiso, se generaron suficientes casos de prueba para alcanzar una cobertura completa del modelo y se descartaron todas las duplicaciones. Esto disminuye espectacularmente el número de casos de prueba sin deteriorar la calidad.

■ Pruebas de Certificación

El objetivo de las pruebas de certificación es valorar la calidad del programa, no depurarlo. No deberían comenzarse a no ser que el programa sea estable y casi libre de errores. Las pruebas de certificación son pruebas de calificación in-

terna que afectan a la decisión sobre la entrega del programa.

La teoría de Markov permite una predicción precisa de la fiabilidad. La fiabilidad calculada se representa en función del número de casos de prueba ejecutados. Esta representación se utiliza para observar el progreso de la prueba. Si no se observan fallos (o sólo fallos menores), la fiabilidad aumentará continuamente; pero si los fallos son más frecuentes (o más serios), la reducción de la fiabilidad con cada fallo observado hará que disminuya el valor total. Si la fiabilidad converge, puede pararse la prueba. Si la fiabilidad estimada es suficientemente buena, el programa puede entregarse después de corregir y eliminar los defectos detectados. Si no, debe pasarse un nuevo conjunto completo de pruebas después de rehacer el programa.

■ Resultados

El experimento mostró que las facilidades típicas de un conmutador RDSI pueden modelarse con máquinas de estados finitos "planas". La realización del modelo del uso de una facilidad condujo a la detección de algunos problemas de diseño con anterioridad a las pruebas. Las pruebas funcionales basadas en modelos fueron un 30% más eficaces que la integración convencional y las pruebas de facilidades (en términos de defectos por caso de prueba). Adicionalmente, los defectos fueron detectados antes. En aquellas facilidades que se probaron con USST, se encontraron solamente muy pocos defectos durante las pruebas de certificación. Los pocos problemas detectados después de la entrega fueron resultado de la integración con otras facilidades que habían sido probadas de forma convencional.

En conclusión, podemos afirmar que USST proporciona buenos resultados.

No obstante, la solución elegida también mostró limitaciones prácticas. Modelos multiusuario, interacciones entre facilidades y entradas complejas condujeron muy pronto a modelos "que explotaban". Se encontró una solución para entradas complejas (combinaciones con expresiones gramaticales), pero los modelos multiusuario y las interacciones

entre facilidades requieren técnicas de modelado más avanzadas.

■ Perspectivas Futuras

Este método mejora el proceso de pruebas. Antes de introducir el método de forma general en nuestro proceso de pruebas, deben producirse mejoras en la solución

del modelado, en especial en el manejo de los modelos jerárquicos. Estos temas están actualmente bajo investigación.

■ Referencias

- 1 H.D. Mills, M. Dyer, R.C. Linger: "Clean-room Software Engineering", *IEEE Software*, Septiembre 1987

Albrecht Beck es Jefe del Grupo de Ingeniería de Tráfico dentro del Departamento de Sistemas del Sistema 12 de Alcatel de la Switching Systems Division de Alcatel, en Stuttgart, Alemania.

UNA VERIFICACIÓN INFORMAL DE INTERACCIONES ENTRE FACILIDADES

J. LENEUTRE
R. TINGAUD

■ Introducción

Muchos ejemplos famosos, desde el error de programación en el Pentium hasta el fallo del Ariane 5, muestran que una pequeña equivocación en una aplicación software puede tener consecuencias económicas y personales espectaculares. Es por esto por lo que debe trabajarse mucho en el área de la calidad del software y, en particular, sobre el desarrollo de métodos formales.

Alcatel desarrolla servicios que se venden a operadores telefónicos. Este artículo examina cómo las aplicaciones de Alcatel utilizan los recientes desarrollos en métodos formales. Los avances en el área de especificación formal, deducción automatizada y comprobación de modelos se espera que tengan un impacto sobre la calidad del software en un ambiente industrial.

Para analizar este uso de métodos formales, se han elegido las interacciones entre facilidades. Además de a los servicios básicos, un usuario puede suscribirse a algunos servicios suplementarios, denominados facilidades, por ejemplo al servicio de Grupo Cerrado de Usuarios (GCU) que se describe brevemente a continuación.

De las interacciones entre estas facilidades pueden surgir muchas situaciones inesperadas. El problema es identificar las pruebas que deben realizarse antes de usar cada una de estas facilidades, con el fin de garantizar un conjunto de propiedades en cualquier contexto de utilización.

Lo que sigue, describe un caso de estudio de la interacción entre dos servicios suplementarios.

Se diseñó una prueba que fue probada formalmente utilizando una herramienta auxiliar de comprobación, que es necesaria, suficiente y mínima para asegurar que "dos usuarios conectados cualesquiera están autorizados a estar conectados" en presencia de las dos facilidades.

■ Descripción de los Servicios Suplementarios

Se consideran los siguientes servicios suplementarios:

- Grupos Cerrados de Usuarios (GCU). Un GCU permite a los usuarios formar grupos a los que, o desde los que, puede restringirse el acceso externo. Un usuario específico puede ser miembro de uno o más GCUs. Un miembro de un GCU puede recibir una llamada de cualquier usuario externo a este GCU si tiene el Acceso Saliente (AS) para responder al Acceso Entrante (AE). Miembros de un GCU específico pueden comunicarse unos con otros; no obstante junto con una suscripción GCU, un usuario puede tener restricciones adicionales que le impidan originar llamadas hacia otros miembros de este GCU, Barrera de Llamadas Salientes (BLIS), y/o recibir llamadas de otros miembros de este GCU, Barrera de Llamadas Entrantes (BLIE);
- Transferencia de Llamada Explícita (TLIE). La TLIE permite a un usuario transformar dos llamadas (una llamada activa y una llamada retenida, cada una de las cuales puede ser una llama-

mada entrante o saliente) en una nueva llamada entre el usuario conectado a la llamada activa y el usuario retenido.

■ Comprobar el Comportamiento Deseado frente al Comportamiento Real

En relación con las facilidades anteriores, se consideró la siguiente pregunta básica:

“¿Están dos usuarios cualesquiera que se comunican autorizados a comunicarse?”

“Autorizados” se refiere a las propiedades esperadas del programa, mientras que “usuarios que se comunican” se refiere a la implementación real, como se describe en la especificación.

Las situaciones en las que se permite una comunicación se muestran en la **Figura 1**.

Tanto las propiedades esperadas como la implementación se describen en documentos de la especificación.

Se especificaron formalmente la implementación (es decir las pruebas reales) y las propiedades usando lógica de primer orden. Se definieron dos predicados: $Mp(x,y)$ que significa “x tiene éxito al llamar a y” y $Mq(x,y)$ que significa “x está autorizado a llamar a y”, y se intentó probar que $Mp(x,y)$ implicaba de forma lógica a $Mq(x,y)$. De forma más general, se quiso encontrar las condiciones necesarias mínimas bajo las que esta implicación es válida.

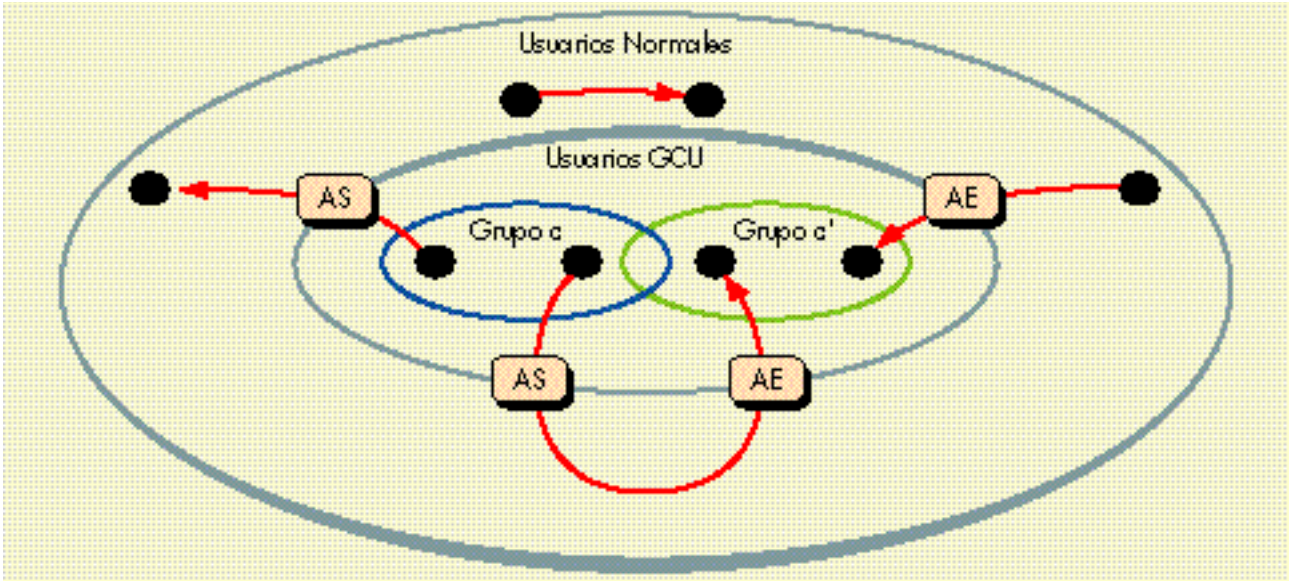


Figura 1a - Los cuatros casos en los que un usuario está autorizado a llamar a otro con una llamada normal (descripción general GCU).

■ Resultados

En un contexto de este tipo, primero se probó que $Mp(x,y)$ implicaba $Mq(x,y)$ cuando sólo estaba involucrada la facilidad GCU. Después, considerando la combinación de GCU y de TLIE, se encontró una condición mínima y necesaria, C, bajo la que se mantiene cierta la misma propiedad.

En la **Figura 2** se muestran algunas de las situaciones cubiertas. Estas situaciones podrían haberse perdido sin la ayuda de métodos formales.

Las pruebas se completaron utilizando Larch Prover (LP), que es una herramienta auxiliar de comprobación de lógicas de primero orden con igualdad. LP es un comprobador de teoremas semi-automático. Existe una gran cantidad de otras posibles herramientas, algunas de las cuales son totalmente automáticas. No obstante, herramientas auxiliares de comprobación, tales como LP, son a menudo más adecuadas para aplicaciones en gran escala.

■ Perspectivas y Conclusiones

La misma solución se ha aplicado a interacciones entre facilidades adicionales; se consideró también la combinación de tres facilidades y se llegó a resultados similares.

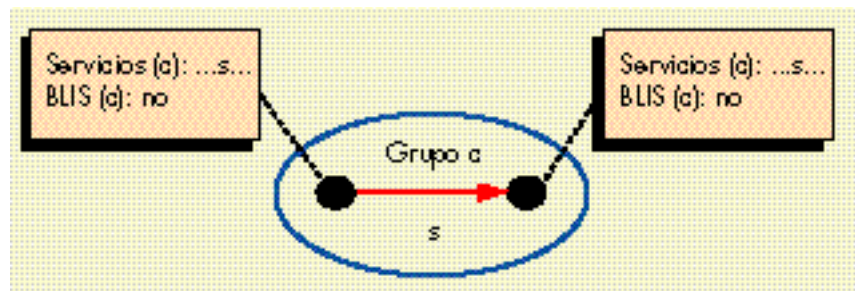


Figura 1b - El caso en que un usuario está autorizado a llamar con una llamada GCU a otro usuario dentro de un Grupo c (descripción general GCU).

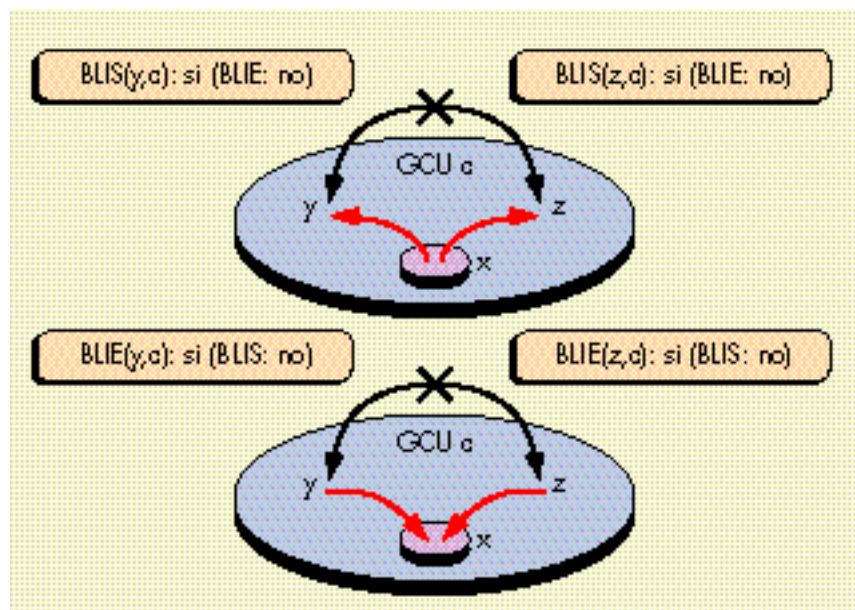


Figura 2 - Dos situaciones en las que debe prohibirse la TLIE: los usuarios "y" y "z" están ambos suscritos a BLIS (o a BLE, respectivamente), dentro del Grupo c, y están conectados al usuario x mediante dos llamadas dentro del Grupo c.

Se ha considerado aquí la propiedad: “¿Están dos usuarios, que están conectados, autorizados a estar conectados?” No obstante, el método se aplica a otras cuestiones, y se intenta trabajar en otras propiedades. Por ejemplo, se podría comprobar:

“¿Se carga al usuario con los costes esperados de las llamadas?”

De nuevo, la respuesta no es clara en el contexto de interacción entre facilidades.

En general, los beneficios de utilizar métodos formales no son fáciles de cuan-

tificar. En el área de las telecomunicaciones públicas, los beneficios, desde el punto de vista del usuario, son principalmente que incrementan la calidad de los servicios y reducen el número de comportamientos no esperados. En algunos casos, la identificación de lagunas, situaciones no definidas o incluso errores en la implementación, pueden también evitar algunas equivocaciones que podrían ser embarazosas para el suministrador del servicio.

Jean Leneutre es ingeniero en el Corporate Research Center de Alcatel, en Marcoussis, Francia.

Robert Tingaud es Jefe de Grupo de Investigación en el Corporate Research Center de Alcatel, en Marcoussis, Francia.

NUEVA DISPONIBILIDAD DE SISTEMAS Y SERVIDORES DE TELECOMUNICACIÓN

D. RASSENEUR
PH. RICHARD

Alcatel está basando sus nuevos servidores en productos y normas de la industria de las TI, para permitir a los operadores la introducción rápida de nuevos servicios.

■ Introducción

Satisfacer Nuevos Requerimientos

Los sistemas de conmutación de Alcatel ofrecen una calidad de servicio basada sobre sistemas de control, en conjunción con sistemas que operan en tiempo real, capaces de proporcionar continuidad de servicio en cualquier circunstancia y permitir el despliegue del software sin interrupción del servicio. Estos productos (particularmente la parte de control que opera en el núcleo de las centrales de Alcatel) están basados en un hardware y un software propietarios. En los años 80 y principios de los 90, no existían en el mercado de la Tecnología de la Información (TI) plataformas hardware y software que ofrecieran todos los criterios de calidad necesarios para los sistemas de conmutación. De esta forma, Alcatel era libre para desarrollar sus propias plataformas hardware y software sobre las que generar sus aplicaciones (centrales Alcatel 1000 y servidores asociados).

Hacia la mitad de los años 90, surgieron nuevos requisitos. La progresiva liberalización del mercado de la Telecomunicaciones en todo el mundo aumentó la competencia entre operadores, forzándoles a ofrecer nuevos productos diferenciados para nuevos abonados de servicios. Para cumplir con estos requisitos emergentes, Alcatel tuvo que desarrollar un creciente número de servicios en espacios de tiempo cada vez más cortos.

Durante el mismo periodo, el sistema operativo Unix se constituyó como una norma de mercado y también para los clientes de Alcatel. La aparición de este estándar contribuyó al desarrollo del concepto de taller de creación del servicio (la base de la Red Inteligente). Consecuentemente, Alcatel tenía que ser capaz de aprovechar los últimos desarrollos en procesadores de los grandes fabricantes, hacerlo virtualmente en tiempo real y a precios competitivos.

Para hacer frente a este aumento de la demanda de nuevos servicios, Alcatel volvió sus ojos hacia el diseño y utilización de plataformas TI que combinaran las ventajas tanto de la industria informática como de las telecomunicaciones: ejecución en tiempo real, servicio ininterrumpido, posibilidad para aceptar extensiones de hardware y software, soporte para una gran variedad de productos flexibles y abiertos. Tales plataformas sirven como infraestructura básica para desarrollar y soportar gran número de aplicaciones de telecomunicación. Para sus aplicaciones avanzadas, servidores y sistemas de alta disponibilidad, Alcatel ofrece la plataforma ideal para satisfacer las necesidades del mercado: *Nectar*. Esta plataforma integral forma parte de la cada vez más extendida aplicación del concepto de red inteligente basada en aplicación de servidor al que evolucionan las centrales de Alcatel.

Funciones tales como conversión, cobro o, incluso, control de llamadas, que eran tradicionalmente realizadas por el núcleo de la central, son ahora

ejecutadas en una plataforma Unix unida a la Central de conmutación.

En el ámbito del software, Nectar aparece como un middleware intermedio para hacer a Unix más robusto, añadiendo las cualidades de servicio esperadas en una plataforma de control, mientras conserva las interfaces de aplicación que existen en Unix. Nectar puede utilizarse para desarrollar aplicaciones de telecomunicación que tienen todas las ventajas del mundo Unix (flexibilidad, apertura, catálogo de herramientas software y productos). Utilizando Nectar, estas aplicaciones también pueden ser mantenidas en la casa del abonado y proporcionar la calidad esperada de un servidor de telecomunicación.

En resumen, Nectar combina la experiencia de Alcatel -robustez, comportamiento y coste de la plataforma- (por ejemplo, la plataforma Alcatel 8300, que es bien conocida por nuestros principales clientes), con la sencillez del mundo Unix tan necesaria para aumentar la productividad propia, como la de los clientes.

Plataforma Nectar

Nectar es una plataforma hardware y software diseñada como un conjunto de servidores normalizados Unix relacionados con una Red de Área Local (LAN) segura, tal y como se muestra en la **Figura 1**. El software básico del fabricante del ordenador es ampliado por funciones adicionales de servicio tales como:

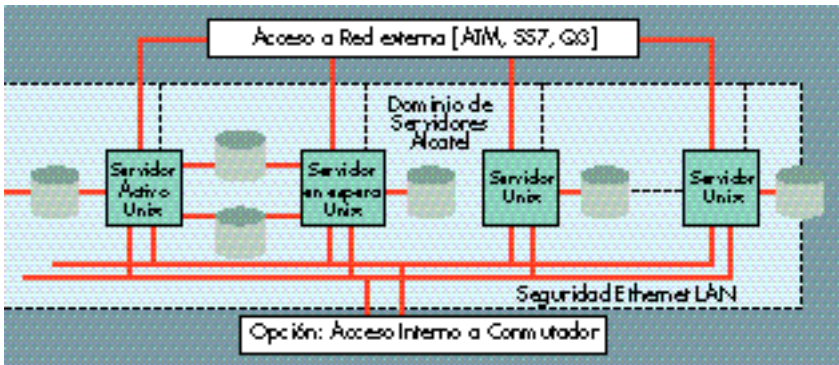


Figura 1 - Configuración típica Nectar.

- Sistema de operación más robusto: comprende las funciones que proporcionan servicio ininterrumpido, incluyendo la gestión de hardware y software y supervisión de la gestión en línea de la aplicación y actualizaciones de la plataforma.
- Servicios de telecomunicación de valor añadido genéricos: proporcionan elementos de funciones típicas en la red, tales como administración Q3 y pilas de protocolo del Sistema de Señalización n° 7 (SS7) en las bandas estrecha y ancha.

Deberá tenerse en cuenta que la plataforma Nectar utiliza solamente el hardware normalizado de TI y no un hardware propietario de Alcatel.

La arquitectura del hardware Nectar ha sido diseñada para satisfacer los siguientes requisitos principales:

- *Efecto de amplitud*: Los servidores son añadidos simplemente a la LAN cuando necesita ser aumentada la capacidad global de proceso.
- *Servicio ininterrumpido*: Las aplicaciones de Telecomunicación deben de ofrecer alta disponibilidad; el nivel comúnmente aceptable de fallo es una indisponibilidad menor a 5×10^{-6} , lo cual equivale aproximadamente a tres minutos de tiempo perdido por año.

Como se muestra en la **Figura 2**, el objetivo inicial de la plataforma Nectar son los servidores agregados a las centrales de conmutación Alcatel 1000 S12 y Alcatel 1000 E10. Estos servidores Unix manejan funciones como gestión de registro

de llamada de datos, administración y soporte, incluidas con el servicio de conmutación de la red inteligente. Este objetivo también incluye los nodos de la red inteligente, tales como los Puntos de Control de Servicio (SCP).

El segundo objetivo se relaciona con el control de aplicaciones tales como conmutación ATM 1000 BBX, o el Sistema Global para Comunicaciones Móviles (GSM) y los elementos de la red móvil de radio para el Servicio Global de Paquetes Radio (GPRS). En este caso, igual que en los servidores agregados, el propósito es integrar los servidores Unix estándar del mercado en el núcleo de los productos Alcatel, los cuales hasta ahora utilizaba hardware y software propietarios.

Para resumir, los objetivos de la plataforma Nectar son los distintos ele-

■ Visión de Conjunto

La plataforma Nectar consiste en un hardware estándar TI (el cual debe adaptarse en algunos casos, particularmente para satisfacer los requisitos normalizados medio ambientales en sistemas de telecomunicación) y un middleware Unix que ejecuta las funciones esenciales para los productos Alcatel. Este middleware consiste en un dispositivo de módulos software que puede ser utilizados conforme a las necesidades específicas de la aplicación.

Como se muestra en la **Figura 3**, los módulos software Nectar están organizados sobre el sistema operativo Unix. El primer módulo es el Servidor de Disponibilidad (AS) que realiza las funciones de seguridad. Es el primer nivel de protec-

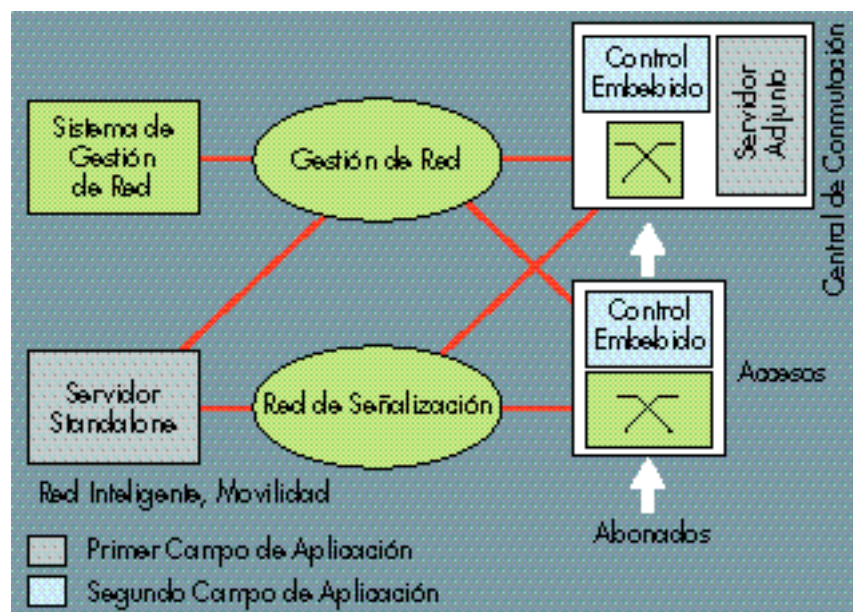


Figura 2 - Aplicaciones objetivo de la Plataforma Nectar.

ción previsto por Nectar. Este nivel consiste en submódulos de gestión de datos en tiempo real y en un sistema seguro de conmutación inter-proceso basado en el Protocolo de Control de Transmisión /Protocolo Internet (TCP/IP).

El sistema de gestión de datos ofrece un servicio de transacción distribuido sobre todos los servidores de la plataforma, de forma que pueden desarrollarse aplicaciones de alta disponibilidad. Los datos sensibles de las aplicaciones puede ser copiados en la memoria y/o grabarse transparentemente en disco. Los tiempos de acceso de lectura y escritura (una pocas décimas de microsegundo; ver apartado "Rendimiento") deben de satisfacer los requisitos de aplicaciones de control de llamadas en tiempo real.

El sistema de comunicación de la plataforma proporciona diferentes nodos de comunicación entre procesos

(tratamiento de datos, pregunta/respuesta, modo en conexión) con gestión dinámica de encaminamiento para mantener establecidas las conexiones si falla uno de los procesos de envío o recepción. Este servicio de comunicación es implementado como un nivel de sesión basado en el transporte normalizado Unix (TCP/IP).

El módulo servidor de disponibilidad fue desarrollado conjuntamente con Compaq y está incluido en su catálogo Unix.

El segundo módulo, el Gestor de Disponibilidad (AM), es exclusivo de Alcatel. Proporciona la configuración de la aplicación y las funciones de seguridad, un sistema de gestión de ficheros seguro y una protección en la plataforma Ethernet LAN. Con estas herramientas, la gestión de disponibilidad puede utilizarse para configurar los procesos de aplicación en todos los ser-

vidores de la plataforma; también proporciona un servicio de supervisión para comprobar que estos procesos operan correctamente. Si uno de ellos falla, la gestión de disponibilidad "principal" reconfigura la aplicación (preservando el contexto) con una indisponibilidad total que no excede de unos pocos milisegundos.

El sistema de gestión de protección de ficheros utiliza la técnica de espejo para proteger los datos del disco y asegurar que el fichero de acceso a las transacciones permanecerá indivisible (propiedad atómica) cuando ocurre un fallo. Esto significa que el sistema de ficheros no tiene por que ser comprobado (con el consiguiente tiempo de indisponibilidad de varios minutos) o ser reiniciado.

El servicio de protección de red de la plataforma soporta una conexión Ethernet LAN redundante. Cualquier fallo en una de las redes es, por lo tanto, transparente a la aplicación. Este módulo utiliza el sistema Unix y el módulo de servidor de disponibilidad.

El tercer módulo, el Gestor del Hardware (HM), maneja in-situ las funciones de gestión del hardware de la plataforma continuando la supervisión de sus componentes hardware. Está basado en el módulo gestor de disponibilidad.

El Gestor de Software (SM) gestiona la plataforma y la aplicación software, ambas durante el desarrollo y, más tarde, en casa del cliente. Proporciona varios escenarios para el cambio on-line de la versión del software (sin interrupción del servicio) y para la gestión del mantenimiento. Dependiendo del tipo de actualización del software (cambio de interface, cambio de plataforma o configuración del disco, etc.), hay disponibles dos procedimientos: cambio de versión "menor" o "mayor" (para más detalles, ver apartado "Gestión de Software").

Además de estos módulos, los cuales están diseñados para dotar al sistema Unix de la misma calidad de servicio que la plataformas propiedad de Alcatel, Nectar ofrece módulos opcionales que realizan servicios genéricos de Telecomunicación, tales como el Canal de Señalización Nº 7, una puerta de comunicación sobre TCP/IP y la administración de la infraestructura.

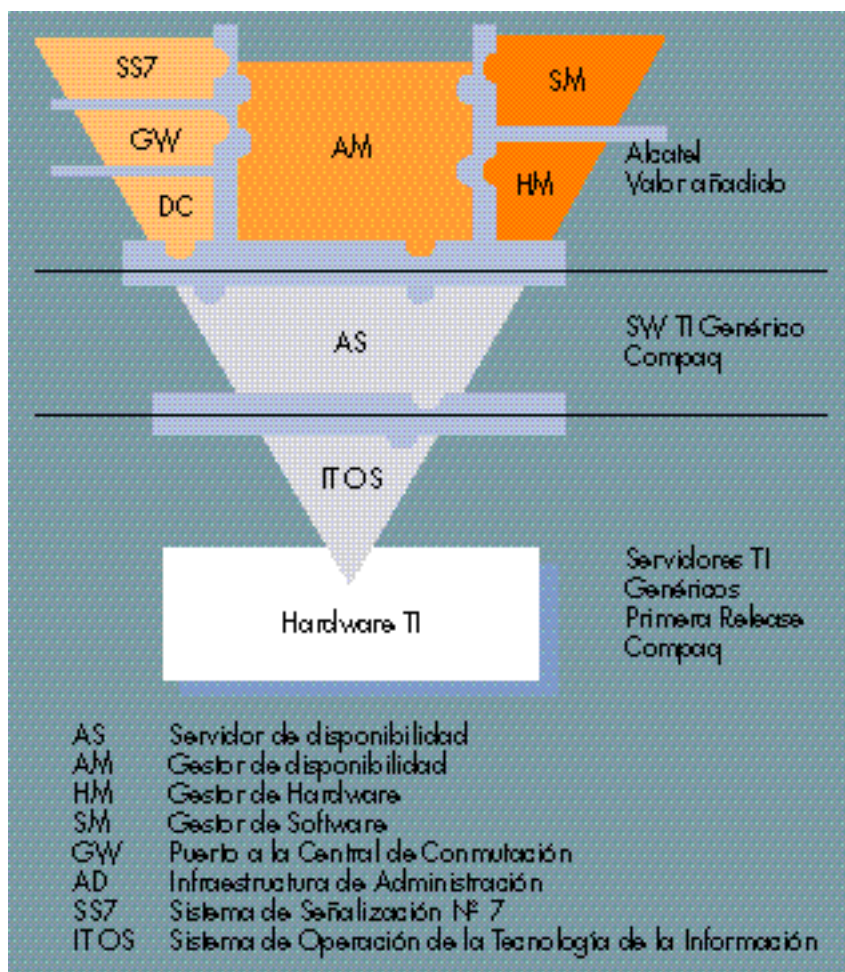


Figura 3 - Configuración de la Plataforma.

El canal de señalización N° 7 (Sistema de Señalización N° 7) proporciona las funciones normalizadas del sistema de señalización N° 7 de banda estrecha y banda ancha (Modo de Transferencia Asíncrono; ATM). Está basado en el sistema de protección Nectar para proporcionar la calidad de servicio y la distribución requeridas por la red inteligente o aplicaciones de conmutación ATM.

El servicio genérico de comunicación TCP/IP ofrece una puerta (Puerta a conmutador GW) que enlaza la plataforma Nectar a las centrales de conmutación de Alcatel (Alcatel 1000 E10, Alcatel 1000 S12 y Alcatel 1000 BBX). Esta puerta proporciona medios de comunicación para separar la configuración y la gestión de la aplicación soportada por Nectar de la configuración del software de conmutación con el cual interactúa.

Finalmente, el tercer servicio proporciona una infraestructura de administración Q3 y un agente de gestión de la plataforma Q3 (módulo de Administración; AD). Esta infraestructura facilita el desarrollo y soporte de los agentes de administración de aplicaciones de acuerdo a los estándares de la línea de productos de gestión de red de Alcatel. Para realizar esto, utiliza la cadena de herramientas de gestión extendida C++ OSI (COMET) desarrollada y mantenida por la línea de productos.

Modelo de Ejecución Nectar

Gestión de Aplicaciones Distribuidas y Calidad de Servicio

El servicio ininterrumpido ofrecido por las aplicaciones Nectar es realizado por la distribución de los procesos Unix que lleva a cabo una aplicación con el objetivo de asegurar la continuidad del servicio ante la eventualidad de los errores de software o hardware que puedan afectar a uno de los procesos. Es importante hacer notar que para cumplir con la especificación mientras se utilizan servidores fuera de la plataforma, el soporte básico Unix debe mejorarse con servicios de software que puedan

gestionar, supervisar y reconfigurar el software de aplicación encapsulado en la plataforma ante la eventualidad de un fallo.

Modos de protección

La gestión del software y el servicio de mantenimiento ayuda a encontrar el criterio de continuidad de servicio a través de los modos de protección descritos a continuación.

Cuando se definen estos mecanismos de protección, Alcatel tiene en cuenta la necesidad esencial de transparencia en la seguridad que es implementada para el programador de los módulos de software de la aplicación; la protección es controlada por el administrador que define la configuración de los datos de la aplicación. En la práctica, al programador solo le concierne la semántica del módulo software y no necesita tener en cuenta los modos de operación tales como activo/espera o compartición de carga.

Modo activo/espera: El proceso maneja el procesamiento (modo activo); un proceso de espera está listo para hacerse cargo del tratamiento si fallase el proceso actual (Figura 4). Los datos son sincronizados periódicamente de forma que el tratamiento puede ser seguido sin interrupción.

Modo compartición de carga: Los procesos son compartidos entre los servidores de la plataforma como se muestra en la Figura 5; en una configuración con N servidores, cada uno maneja una enésima parte del proceso. Si ocurre un fallo, la carga es redistribuida entre los servidores que todavía permanecen operativos.

Sistema de gestión de datos

El Gestor de Datos (DM) proporciona acceso en tiempo real a los datos, mientras asegura la integridad y persistencia de esos datos. Las propiedades que ofrece este servicio son en la práctica una combinación de:

- Propiedades ACID (A= Atomicidad, C= Consistencia, I= Aislamiento, D= Durabilidad) del sistema de gestión de la base de datos normalizada:
 - A: Atomicidad de operaciones (transacciones de "todo o nada") donde todos o ninguno de los cambios hechos durante la transacción son aplicados a los datos.
 - C: El acceso a los datos es consistente (un concepto que está estrechamente relacionado al concepto "Aislamiento" descrito a continuación)
 - I: Aislamiento entre operaciones paralelas (visto desde el exterior, cada acontecimiento como si las opera-

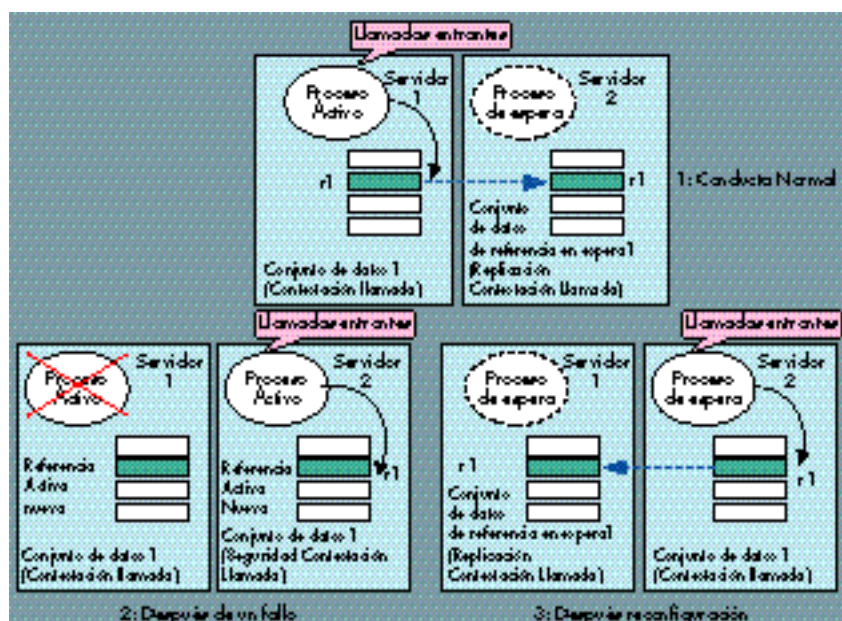


Figura 4 - Protección típica activo/espera.

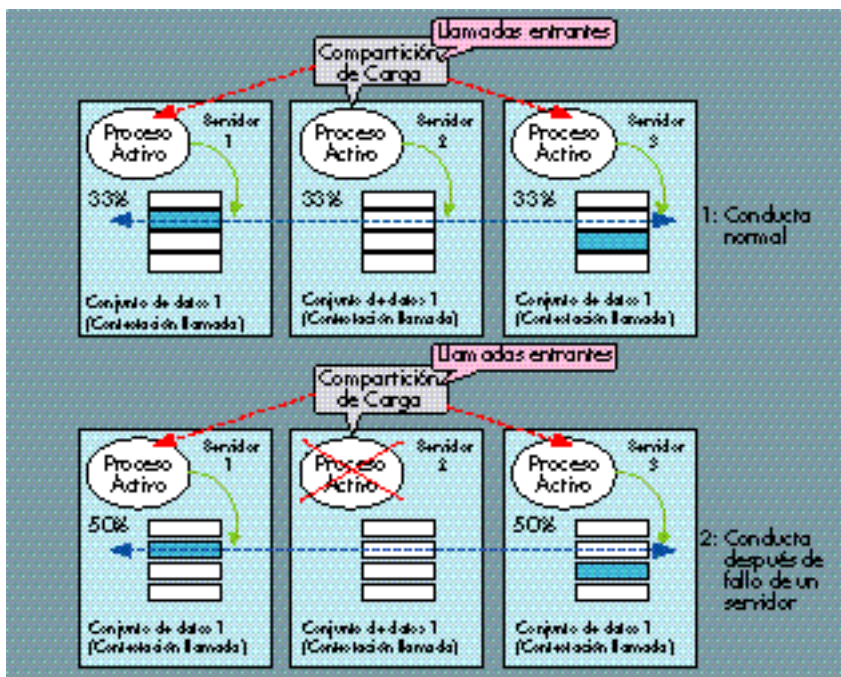


Figura 5 - Ejemplo de protección de compartición de carga.

ciones hubieran sido realizadas en serie).

- D: Durabilidad garantizada de los datos; esta función, por ejemplo, es esencial para una aplicación de dispensación de efectivo, asegurando que el cliente recibe su dinero sin temor a que lo figurado en el registro pueda producirle pérdida.
- Un servicio que proporciona un potente (tiempo real) acceso a un gran volumen de datos estructurados en un equipo distribuido.

Los datos son organizados en conjuntos de datos, cada uno consistente de un número de registros. Para permitir en tiempo real el acceso a los datos (algunas décimas de microsegundos), los datos residen en la memoria principal de la máquina.

Los datos son accedidos indistintamente en modo lectura y/o escritura dentro de una transacción o en aislamiento (solamente una lectura o primera escritura es ejecutada en una operación). La selección del registro accedido puede ser directa (número del cursor) o secuencial, con o sin la llave de acceso a registros; el campo que representa la clave en el registro debe ser igual al valor especificado de la clave

(por ejemplo, número de abonado); la misma clave puede ser primaria o secundaria.

La gestión de los datos Nectar también ofrece un servicio de transacción que permite el acceso atómico (todo o nada) a los datos. Una transacción tiene una fase de apertura, una fase de acceso a los datos y una fase de cierre. Cuando se abre la transacción, un identificador único de transacción (específico para la transacción y único en espacio y tiempo) se suministra al usuario; cualquier acceso de datos dentro de esta transacción utiliza este identificador. Ningún cambio hecho durante la transacción puede ser visto desde el exterior hasta que la transacción esté cerrada. Las modificaciones son o bien todas aplicadas (*cumplidas*) o bien todas canceladas (*fracasadas*) cuando la transacción es cerrada.

En un camino que es transparente al usuario, la gestión de datos asegura que los datos en los conjuntos de referencia de datos sean consistentes con los datos en los conjuntos de *replicación* de datos utilizados para seguridad (comenzar de nuevo los procesos después de fallos basados en el concepto de un conjunto activo de datos de referencia y un conjunto de datos de referencia en espera) o

compartición de carga (N nodos conteniendo procesos que acceden a los mismos datos replicados). Las *replicaciones* son actualizadas al final de una transacción o al final de una escritura por simple acceso.

Si es necesario, los datos pueden (mediante configuración) ser salvados como copia de seguridad en discos de back up. La actualización de los ficheros de copia de seguridad es o explícita (i) cuando se solicita una copia de seguridad de un conjunto específico de datos, o implícita (ii) cada vez que los datos son modificados.

Un usuario puede, si lo requiere, ser notificado de cualquier cambio que afecte al conjunto de datos bien que vengan de fuera o de una transacción (desencañamiento).

La **Figura 6** muestra un caso típico en el cual la aplicación está utilizando dos conjuntos de datos protegidos en la memoria y distribuidos sobre la plataforma. El conjunto de *registros de Datos 1* es también grabado en el disco (en el caso de datos semipermanentes), el cual es a su vez protegido por copia en un disco espejo. Esta figura muestra la ejecución de una transacción escrita en cada conjunto de datos. Aquí la gestión de datos es responsable de la actualización atómica de todas las copias de los conjuntos de datos cuando la transacción se cierra.

Gestión de Configuración

La configuración es gestionada por dos componentes: el módulo de gestión y supervisión del hardware (gestor de hardware) y el módulo de gestión de la configuración del software (gestor de software).

Gestión de hardware

El módulo de gestión de hardware explora continuamente las operaciones de los componentes de la plataforma hardware. Si se detecta un fallo, la unidad afectada es aislada inmediatamente y es identificada la unidad de reemplazamiento del hardware defectuoso.

Además de la robustez del software avanzado que Nectar aporta al sistema Unix, un diseño dedicado de bastidor que comporta elementos normalizados

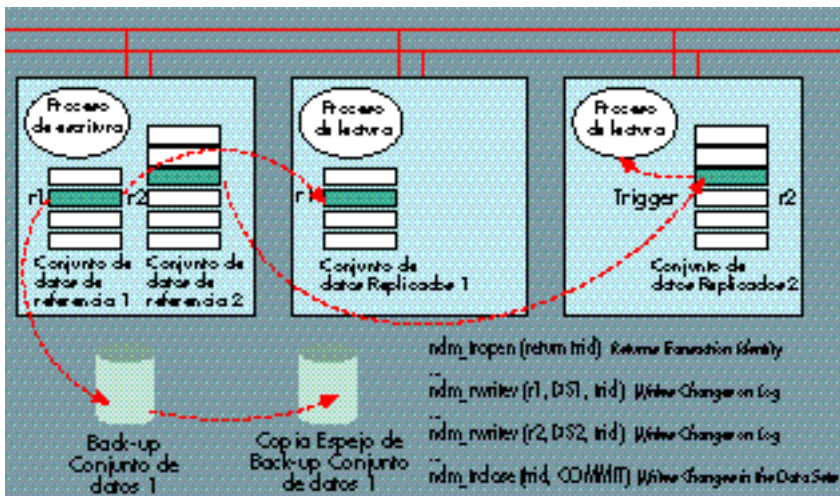


Figura 6 - Utilización típica de la gestión de datos.

de telecomunicación, ha sido desarrollado en colaboración con Compaq, uno de los mayores fabricantes de ordenadores. Este sistema de bastidores proporciona a los clientes la misma calidad que nuestros sistemas propietarios.

Además, el sistema de bastidores proporciona las interfaces necesarios para supervisar los componentes del hardware que suplen las unidades reemplazadas.

Gestión de software

El gestor de software proporciona soporte para mejorar o corregir el mantenimiento in-situ de la aplicación, mientras que el servicio de mantenimiento de software observa y supervisa la conducta del software en casa de los clientes. El gestor de software por sí mismo proporciona las herramientas necesarias para instalar un nuevo software y cambiar las versiones de software in-situ.

Dos tipos de cambios de versión en línea están disponibles. El primero, llamado cambio de versión *menor*, es usado cuando se arregla la instalación de la aplicación y produce errores in-situ. La mejora principal está basada en los mecanismos de protección y reconfiguración ante la eventualidad de que se produzcan fallos y trae consigo a su vez el reemplazamiento de varios módulos.

El segundo tipo, llamado cambio de versión *mayor*, se aplica cuando hay cambios en la aplicación (cambio de interface, adición de nuevos componentes, etc.). El principio de mejora del softwa-

re está basado en la división lógica de la plataforma en dos partes: la primera mitad de la plataforma continúa soportando la versión N de software, mientras que la versión N+1 está instalada en la segunda mitad. La mejora del software es ejecutada por conmutación de tráfico a la segunda mitad de la plataforma cuando la versión N+1 está conectada y funcionando.

Este servicio significa que todo el software de aplicación y de la plataforma puede ser reemplazado in-situ sin interrupción del servicio. También permite el retorno automático a la vieja versión de la instalación por fallo de la nueva versión.

■ Canal de Señalización Nº 7

La plataforma Nectar proporciona a la red SS7 un conjunto de protocolos de acceso: protocolo de Mensaje de la Parte de Referencia (MTP), Conexión de Señalización y Parte de Control (SCCP) y capas de la Parte de Aplicación de Capacidades de Transacción. Este sistema genérico de señalización Nº 7 a su vez utiliza las funciones de protección de la plataforma Nectar para proporcionar una arquitectura flexible de forma que pueda ser dimensionada la capacidad de conexión de la red, por una parte, para adaptarse a las necesidades de los elementos de la red (arquitectura distribuida) y, por

otra, para proporcionar servicio ininterrumpido.

Las Figuras 7 y 8 muestran el software y la arquitectura hardware del servicio de señalización Nº 7 de banda estrecha y banda ancha, respectivamente, de Nectar.

La arquitectura que muestra la Figura 7 es para una configuración que utiliza un front end de servidores SS7 (o procesadores front end) la cual soporta las más altas capas del protocolo (MTP, SCCP) y que funciona como un número de pares activo/espera. La aplicación es distribuida sobre los servidores de aplicación, corriendo en modo activo/espera y/o en modo compartición de carga. Estos servicios presentan las capas superiores del protocolo SS7: SCCP y TCAP.

En el caso de banda ancha, la arquitectura mostrada en la Figura 8 representa una configuración con servidores de señalización nº 7 soportando los niveles más bajos de la Capa de Adaptación de la Señalización ATM (SAAL) y la Capa de Transporte de Mensajes (MTP) en el modo de compartición de carga. En la banda estrecha, la aplicación se distribuye entre el extremo de la parte final de los servidores del procesador funcionando en activo/espera y/o modo de compartición de carga. Estos servidores manejan las capas superiores del protocolo SS7 (RDSI Parte Usuario; ISUP).

■ Administración de la Infraestructura

El servicio de administración de Nectar proporciona la infraestructura de administración en la propia plataforma, así como las funciones para el desarrollo de las aplicaciones de los agentes de administración a nivel elemental de red. Para hacer esto, Nectar soporta el conjunto de herramientas COMET (ver el artículo de Uszynski y Wardé en esta Revista) desarrollada por la línea de productos de gestión de red de Alcatel. Los servicios genéricos de administración también toman parte de este servicio: anotación, notificación de acontecimientos, alarma de sobrecarga, transferencia de ficheros, etc.

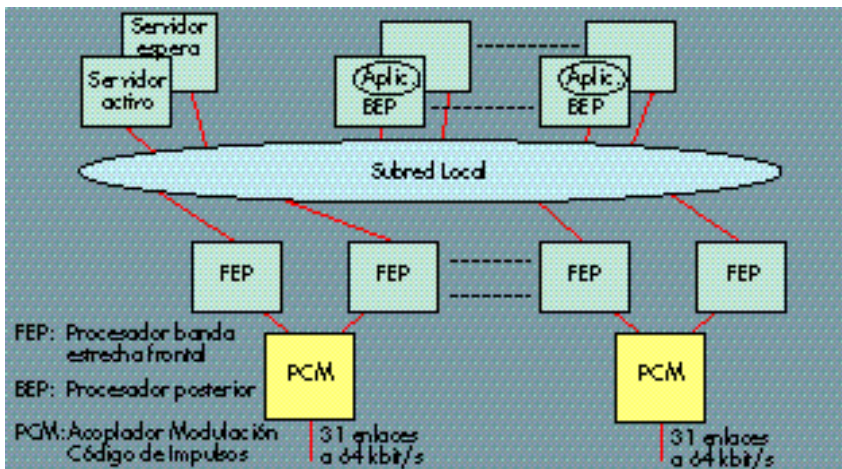


Figura 7 - Canal de señalización Nº 7 de banda estrecha.

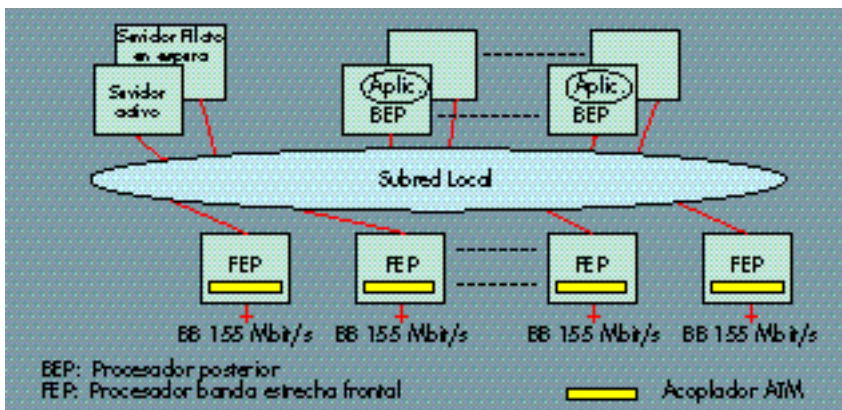


Figura 8 - Canal de Señalización nº7 de banda ancha.

Este servicio de administración está basado en un desarrollo metodológico orientado a objetos que cumple con las directrices de los estándares de gestión de elementos de red de la Red de Gestión de Telecomunicaciones (TMN)/Interconexión de Sistemas Abiertos (OSI), e incluye un amplio conjunto de protocolos Gestión Común de los Elementos de Servicio de la Información (CMI-SE)/Transferencia, Acceso y Gestión de ficheros (FTAM) y protocolos Q3 para X.25, RDSI, ATM y otras conexiones de red.

La **Figura 9** muestra la arquitectura software del servicio de administración Q3.

Aunque provee a la plataforma de la estructura de administración Q3, Nectar no fuerza a la aplicación para utilizar la administración normalizada Q3. Es también posible utilizar otros productos de la gama Compaq para Unix,

tales como la Gestión de Protocolo de Red/nica (SNMP) o CORBA para escribir los agentes de administración de las aplicaciones.

■ Beneficios de Nectar

Rendimiento

La **Tabla 1** resume el comportamiento básico de la plataforma Nectar en términos de continuidad de servicio, comparado con un sistema normalizado Unix.

Nectar distribuye en tiempo real los accesos a la ejecución de la gestión de los datos como se muestra en la **Tabla 2**. Estas medidas están basadas en los procesadores Compaq Alpha 21164, con 500 MHz de CPU. (14 Spec Int 95; esto representa la ejecución de un proceso basando la unidad de medida en el fun-

cionamiento de un conjunto de programas de referencia).

Disponibilidad

La configuración típica de una aplicación soportada por la plataforma Nectar contiene un conjunto de servidores enlazados conjuntamente a una LAN. La capacidad total de una configuración de estas características, en términos de potencia bruta, es la potencia de los servidores multiplicada por el número de servidores en la configuración. La configuración teórica máxima de una plataforma Nectar está relacionada con los siguientes parámetros: capacidad de la LAN protegida que interconecta las estaciones y conformidad con los estándares de disponibilidad.

Para su LAN interna, la plataforma Nectar utiliza tecnología normalizada de TI (esto conlleva una extrema rapidez en los cambios debido a que el mercado es sumamente innovador y dinámico).

En el presente caso, implica una red que utiliza concentradores activos aplicables Ethernet funcionando a 100 Mbit/s. Utilizando esta tecnología, pueden apilarse cuatro hubs de 24 puertos, proporcionando una configuración teórica máxima de 96 servidores.

Con respecto a la conformidad con los estándares de disponibilidad, la plataforma Nectar (como todas las plataformas de telecomunicación) está diseñada para ser tolerante a fallos. Sin embargo, se acepta que fallos simultáneos en dos de los servidores de la plataforma resultarán en una indisponibilidad. Consecuentemente, la configuración debe ser proyectada para asegurar que la probabilidad de fallos simultáneos en dos servidores sea menor que 5×10^{-6} (objetivo de indisponibilidad).

- Ante el evento de un fallo de hardware: en el peor de los casos, uno cada cinco años para un servidor de TI. Los servidores Digital AS800 están especificados en 25.000 FIT (Fallos En Tiempo) o aproximadamente 40.000 horas de operación entre fallos de hardware (Tiempo Medio Entre Fallos; MTBF). El Tiempo Medio Para Reparación (MTTR) garantizado son cuatro horas, dando un porcentaje de indisponibili-

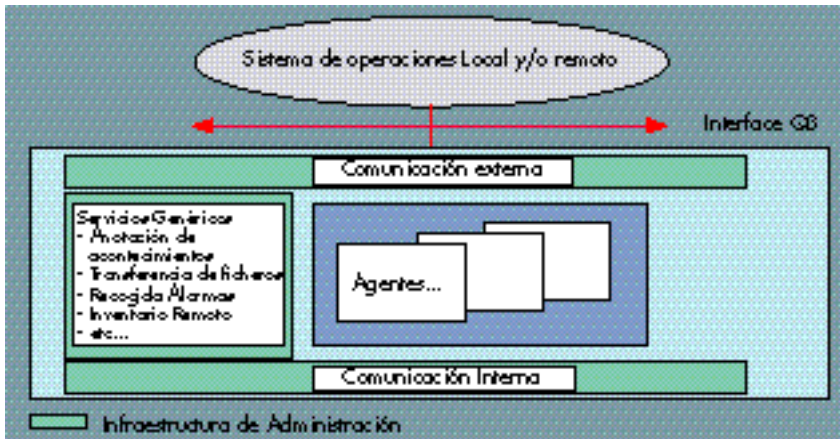


Figura 9 - Infraestructura de Administración.

dad resultante de los fallos de hardware de: (MTTR/MTBF) el cual es 4 horas/40.000 horas = 10^{-4} .

- Ante el evento de un fallo de software, lo normalmente aceptado es un acontecimiento cada tres meses con un tiempo de arranque garantizado de menos de cinco minutos con una garantía de tiempo de parada (T) o un porcentaje de indisponibilidad resultante de los fallos de software de:
 - T/N2 (en donde N² es el número de minutos de tiempo de operación entre fallos de software); aquí el número medio de días en tres meses se

toma como 92, de forma que: el porcentaje de indisponibilidad es 5 min./ $(92 \times 24 \times 60 \text{ min.}) = 3,6 \times 10^{-5}$.

Por todo ello, la indisponibilidad global de un servidor por todas las causas es:

- $MTTR/MTBF + T/N2 = 10^{-4} + 3,6 \times 10^{-5} = 1,36 \times 10^{-4}$

La probabilidad de ocurrencia de fallos simultáneos en dos de los servidores de la plataforma que conduzca a un fallo total es por esto el producto de la probabilidad de un fallo ocurriendo en un

servidor cuando el otro está ya indisponible. Sea n el número de servidores, por tanto:

- $[(MTTR/MT(BF) + (T/N2)] \times \{(n-1) \times [(MTTR/MTBF) + (T/N2)]\} = 1,36 \times 10^{-4} \times [(n-1) \times (1,36 \times 10^{-4})]$

dicho lo cual, para conservarlo dentro del objetivo de disponibilidad de 5×10^{-6} , significaría que el sistema debería tener una configuración de más de n=271 servidores; Esto se aleja del máximo de configuraciones proyectadas.

Tomando la capacidad de la red como un factor de limitación, el rendimiento máximo de una configuración Nectar es actualmente de $96 \times 55 = 5.280$ SpecInt95 (referencia: procesador Alpha 21264 en 55 SpecInt 95).

En comparación con la generación previa de plataformas de Alcatel, esto representa aproximadamente 100 veces el rendimiento de una configuración Alcatel 8360.

■ Configuración Hardware

Alcatel y Compaq han colaborado en el diseño del bastidor (ver Figura 10) para los servidores instalados en las centrales de conmutación o las plataformas de

| | Sistemas Unix estándar | Sistema Unix robusto mediante Nectar |
|---|------------------------|--|
| Tiempo para reanudar una operación después de un fallo de software | Varios segundos | 4 ms |
| Tiempo para reanudar una operación después de un fallo de software con recuperación del contexto de datos | No | 4 ms |
| Tiempo para reanudar una operación después de un fallo en LAN Ethernet | Varios segundos | No se interrumpe el servicio (protección Ethernet LAN) |
| Tiempo para reanudar una operación después de una fallo del servidor | Decenas de segundos | 200 ms |

Tabla 1 - Rendimiento básico de la plataforma Nectar.

| Operaciones | Acceso no protegido (µs) | Acceso protegido (µs) |
|--------------------------|--------------------------|-----------------------|
| Lectura de un registro | 15 | 15 |
| Escritura de un registro | 40 | 80 |

Tabla 2 - Rendimiento del gestor de datos distribuido en tiempo real de Nectar.

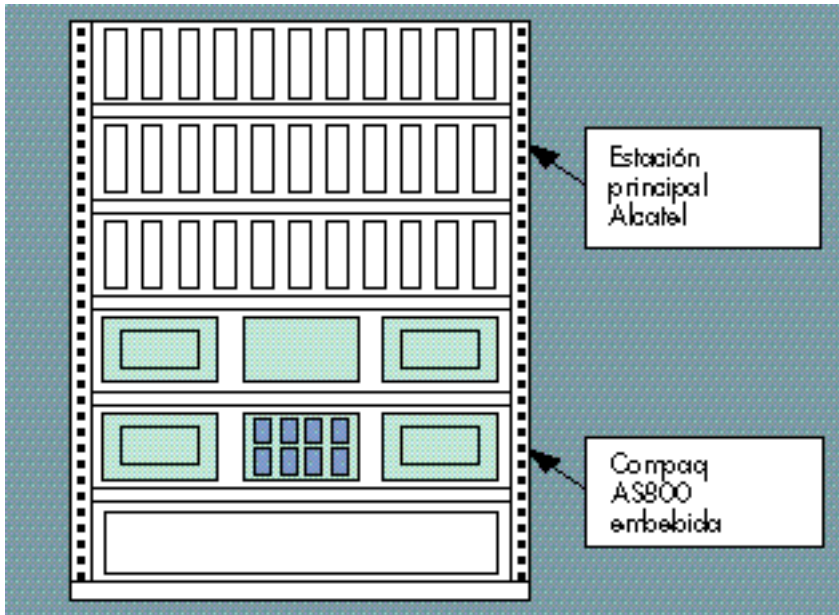


Figura 10 - Disposición del bastidor Alcatel para un servidor Compaq.

control (Alcatel 1000/BBX, GPRS). Este diseño persigue los siguientes objetivos:

- Los servidores Compaq AS800 pueden ser instalados en los bastidores de Alcatel.
- El producto cumple con las normas de telecomunicación en medio ambiente, radiación, temperatura, gestión de hardware, etc.
- Ofrece la facilidad de reemplazar de unidades en campo.
- Garantiza un producto de larga vida e intervalos de mantenimiento próximos a las actuales normas para equipos de telecomunicación; este no es el caso con la oferta TI normalizada.
- Precio de coste comparable al del hardware normalizado TI.

■ Conclusiones

La plataforma Nectar es parte de la estrategia de desarrollo de servidores de valor añadido. El propósito de esta estrategia es permitir a los componentes fuera del soporte ser utilizados en todos nuestros productos de telecomunicación fijos, móviles y de red de datos.

En resumen, la plataforma Nectar proporciona productos fuera del soporte con la continuidad de servicio y requisitos de ejecución en tiempo real requeridos por los servicios de telecomunicación. En consecuencia, aúna las telecomunicación y las TI utilizando lo mejor de cada uno de ambos mundos.

Philippe Richard es Director del Grupo de Plataformas en la División de Negocio de Conmutación de Alcatel, en Velizy, Francia

Dominique Rasseneur es Director del Programa GPRS en la División de Negocio de Conmutación de Alcatel (E10) en Vélizy, Francia.

ESTRATEGIA DE LA PLATAFORMA DE SOFTWARE PARA GESTIÓN DE RED AVANZADA

M. USZYNSKI
P. WARDÉ

El rápido desarrollo de las soluciones de gestión de red requiere una plataforma de software potente, flexible y abierta.

■ Introducción

Hoy día, la gestión de red es un componente básico de cualquier red de telecomunicación, tanto para los operadores como para los fabricantes.

Los operadores de red tienen que hacer frente al desafío de la globalización, liberalización e incremento de la competencia en los mercados, que son en definitiva las fuerzas conductoras de los actuales negocios en el sector de las telecomunicaciones.

La globalización implica a menudo la fusión de redes y su mantenimiento en interoperación. También supone un fuerte impacto en los ciclos de vida de los productos. El resultado es que la mayoría de las redes de telecomunicación son construidas como una malla de elementos de red individuales, para la cual los operadores requieren una capacidad global de gestión. Tal capacidad está basada en un sistema integrado de gestión de red capaz de administrar redes de telecomunicación que aglutinan una variedad de tecnologías (por ejemplo, conmutación, transmisión, acceso, móviles) y proporcionan una amplia gama de servicios avanzados. Los operadores de redes esperan soluciones flexibles de gestión de red adaptadas a la configuración y tamaño de sus redes y de sus organizaciones internas, tanto si son centralizadas como distribuidas.

La liberalización y creciente competencia en el mercado conducen a mejoras significativas en calidad de servicio y velocidad de aprovisionamiento, aspectos

necesarios para reducir los costes de operación. Soluciones eficaces de gestión de red son la clave que capacita la automatización de las tareas complejas y repetitivas relacionadas con el funcionamiento de los equipos y la automatización del mantenimiento, supervisión de la red, calidad del servicio de mantenimiento y asistencia al cliente.

En consecuencia, la gestión de red potente se convierte rápidamente en parte integral del modelo de negocio de un sistema de telecomunicación.

Los fabricantes de redes necesitan herramientas adecuadas para construir y suministrar aplicaciones de gestión de red que administren los requerimientos de los clientes. Estas herramientas deben soportar el desarrollo rápido, consistente y a bajo coste de aplicaciones que a menudo son complejas.

Los costes de desarrollo y el tiempo de puesta en el mercado son minimizados por la reutilización de los componentes de software y la limitación de los riesgos en el desarrollo del software. La reutilización es un objetivo clave en el proceso de desarrollo de diferentes aplicaciones, al identificar aquello que es común en el amplio espectro de los requisitos utilizados en las diferentes áreas de gestión.

Es a menudo necesario integrar nuevos equipamientos con los sistemas ya existentes en un entorno multiproveedor. Esto requiere soluciones abiertas y la capacidad de adaptación del sistema de gestión de acuerdo con el crecimiento y evolución de la red. Se necesita un soporte adecuado para garantizar la

adaptación suave de las soluciones suministradas.

De cara a las fuertes presiones competitivas del mercado abierto de las telecomunicaciones, los suministradores de equipos deben optimizar sus inversiones. Los factores clave para mantenerse en cabeza incluyen costes reducidos de desarrollo, tiempo de ajuste al mercado y riesgos de desarrollo.

Este artículo describe ALMAP, la Plataforma de Gestión de Red de Alcatel que ha sido diseñada para proporcionar soluciones de gestión de red integradas y de alta funcionalidad para acelerar el desarrollo de las aplicaciones software. Primero, hablaremos de la plataforma ALMAP, su arquitectura y características, presentando los beneficios del proceso de desarrollo asociados con el entorno de la plataforma. El éxito de ALMAP es un ejemplo de la consistencia que ha sido mantenida entre la arquitectura, los componentes de software, el conjunto de herramientas y la definición de proceso.

■ Plataforma Software ALMAP

Arquitectura de Referencia

La arquitectura ALMAP es también utilizada como la arquitectura de referencia para todos los productos basados en ALMAP. Conforme a normas de arquitectura de la Gestión de la Red de Telecomunicaciones (TMN) garantiza la flexibilidad requerida de un sistema de

gestión. En particular, combina la apertura hacia sistemas externos con la disponibilidad para integrar sistemas heredados en la TMN, preservando así la inversión previa. La capacidad de adaptación y la disponibilidad para evolucionar le permiten adaptarse de acuerdo al crecimiento y evolución de las redes, reduciendo el mantenimiento futuro y los costes de migración.

La arquitectura ALMAP logra todo esto aplicando esos principios a todos los componentes básicos de los sistemas globales. Además, ALMAP proporciona un conjunto de herramientas potente para el rápido desarrollo de las aplicaciones de gestión de red que cumplen con la arquitectura TMN. Los componentes principales de la arquitectura ALMAP se discuten a continuación.

Arquitectura en capas lógicas

TMN es una arquitectura en capas en la cual cada una de ellas trata diferentes aspectos de gestión de la red (**Figura 1**)

- Capa de elementos: enfocada hacia el equipo de red.
- Capa de red: trata la infraestructura y los recursos lógicos de red.

- Capa de servicios: dirigida hacia los servicios de red.
- Capa de negocios (\$): enfocada sobre las economías de red.

La gestión de cada capa está ligada al funcionamiento de las capas inferiores, de forma que la ejecución de las capas más altas puede estar restringida por el soporte proporcionado por las capas más inferiores. La base que soporta la pirámide determina la calidad y cantidad de la información de gestión en la capa superior de negocios. ALMAP ejecuta la gestión jerárquica TMN, proporcionando soporte extensivo a todas las capas TMN.

Arquitectura basada en componentes

Las aplicaciones ALMAP están divididas en componentes software, cada uno de los cuales es responsable de una tarea específica. Estos componentes, que están diseñados para integrarse sin fisuras con la gestión de aplicaciones de red, abarcan un conjunto de funciones bien definidas que ofrecen una aplicación real de servicios. Cada componente incluye un conjunto de interfaces externas, algunas de las cuales son capaces de inte-

roperar con componentes externos, no pertenecientes a ALMA.

Enfoque orientado a objetos

ALMAP es absolutamente una aplicación orientada a objetos que proporciona soporte completo para el enfoque de orientación de objetos de los estándares TMN. Los recursos de sistema TMN son modelados como *objetos gestionados* en donde un objeto gestionado es una entidad simple que contiene tanto el código como los datos que controlan los recursos de la gestión de red. Consiste en un conjunto de atributos que le caracteriza, acciones que pueden ser ejecutadas y notificaciones que pueden ser emitidas.

Modelo agente-gestor

Los componentes de ALMAP y sus aplicaciones TMN están organizados conforme al modelo de agente-gestor TMN (**Figura 2**) como se define en la norma TMN UIT-T M.3010. Cada componente de la aplicación actúa como un gestor y/o como un agente.

La jerarquía completamente ordenada de gestores y agentes garantiza la permanencia de los sistemas TMN:

- Un agente es un componente del sistema TMN que representa un conjunto de *objetos gestionados* con el propósito de supervisar la gestión. Recibe y solicita servicios de uno o más gestores para operaciones definidas por aquellos objetos gestionados: creación o supresión, recuperación o modificación de atributos y desencadenamiento de acciones. También envía notificaciones cuando un recurso gestionado cambia de estado o detecta un error.
- Un gestor es un componente del sistema distribuido TMN el cual controla y maneja los agentes de red. Actuando en nombre del operador, puede enviar operaciones de gestión a los agentes de red y reportar acontecimientos y alarmas recibidas de los agentes de red.

Objetos gestionados

Los objetos gestionados -estando la gestión compartida de conocimientos entre gestores y agentes- son organizados en

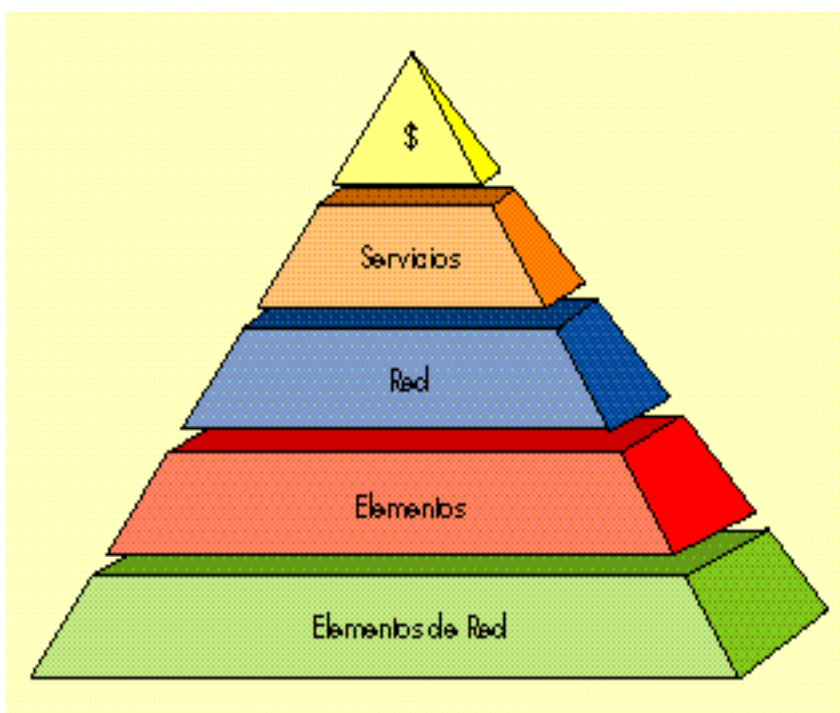


Figura 1 - Arquitectura de gestión en capas.

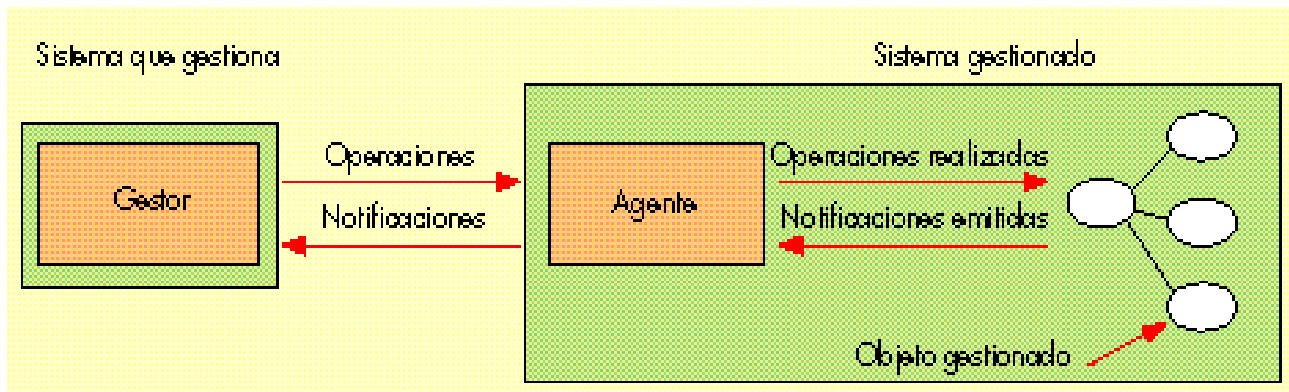


Figura 2 - Modelo Agente/Gestor.

un directorio de estructura jerárquica ramificada conocido como Información de la Gestión Ramificada (MIT). Un gestor puede repartir operaciones sobre objetos individuales en este directorio e incluso en objetos múltiples por especificación de un ramal (campo) y selección de criterio (filtros). El conjunto de objetos gestionados actualmente, contenidos en el agente, forman la llamada Base de Información de la Gestión (MIB).

Separación de la presentación y gestión de datos

ALMAP y sus aplicaciones están diseñadas para separar claramente la presentación de componentes de los componentes de gestión de los datos. La presentación de componentes son aplicaciones frontales que actúan típicamente como gestores. La gestión de datos es realizada por componentes que combinan los roles de gestor y agente.

Este enfoque está en línea con los conceptos de la Unión Internacional de Telecomunicaciones (UIT) conocido como Función del Sistema de Operación (OSF) y Función de Estación de Trabajo (WSF). Garantiza que la información de gestión es consistente a través de la distribución del sistema TMN, manteniendo una simple referencia de la información y no permitiendo duplicaciones.

La separación entre las funciones de presentación de las funciones de manipulación de datos también contribuye a la configuración flexible del sistema, a la distribución de la presentación y funciones de gestión de datos a

través de TMN, y a la configuración del sistema de operación de acuerdo al número de operadores y al tamaño de la red gestionada. Este concepto agente-gestor hereda los beneficios de la definición cliente/servidor.

El flujo de datos entre los componentes de presentación y de gestión de datos es limitado a los datos de aplicación, mientras que el procesamiento de datos de la interface de usuario es delegado a los clientes locales.

La presentación de gestores múltiples puede interactuar recíprocamente con gestores de datos a través de redes de área amplia, aún con una limitada comunicación de ancho de banda.

El interface entre la capa de presentación y la de aplicación es bien definida y se deriva de un modelo de información formal, permitiendo funciones de presentación a medida a clientes para ser implementadas. Esto podría ser necesario para tener en cuenta los requisitos de la operación de interface de clientes.

Interfaces abiertos de gestión

Dentro de la arquitectura ALMAP se especifican interfaces abiertos entre gestores y agentes. Estos interfaces están definidos en la Guía para la Definición de Objetos Gestionados (GDMO) y en la notación Abstract Syntax Notation 1 (ASN.1); esto, junto con el uso del Protocolo Común de Servicios de Gestión de Información (CMIS) para la gestión de la conmutación de información entre gestores y agentes, promueve la interoperabilidad y el interfuncionamiento entre componentes del Sis-

tema TMN, mientras sigue siendo independiente de la implementación interna.

Cualquier elemento de red o sistema de gestión con un interface Q3 por encima de un Sistema Abierto de Interconexión/Protocolo Común de Servicios de Gestión de Información (OSI/CMIP) puede integrarse en tal entorno, pero no todos los sistemas pueden soportarlo. Dispositivos de mediación o funciones de adaptación-Q pueden proporcionar una interface normalizada para estos "sistemas heredados". Ellos pueden adaptar la gestión de la comunicación para interfaces Qx no normalizados.

Navegación entre aplicaciones

Las capacidades de navegación entre aplicaciones está prevista vía un sistema de mensajería. Son usadas extensamente en ALMAP para proporcionar servicios de valor añadido al nivel de interface de usuario, proporcionando un camino sencillo para añadir al usuario la integración del interface y capacidades de navegación para aplicaciones residiendo en diferentes capas gestionadas (por ejemplo, de visión de red a visión de equipo, visión de alarmas o visión de rendimientos en escenarios típicos de localización de fallos).

Infraestructura de comunicaciones distribuidas.

La gestión distribuida se proporciona por varios caminos: a través de la utilización del Gestión Distribuida (DM) HP OpenView y el estándar de la industria interface Protocolo de Ges-

ción X/Open (XMP), o utilizando el entorno proporcionado como parte de la Gestión Ampliada de la Cadena de Herramientas (COMET) C++ OSI de ALMAP. Esta cadena de herramientas proporciona inicialmente soporte para la comunicación basada en CMIS y está siendo mejorada para soportar la comunicación Interface Descripción de Lenguaje/Objeto Común Arquitectura Demanda Agente (IDL/CORBA).

Entorno de Desarrollo TMN

COMET es un framework TMN para el rápido desarrollo del modelo de información basado en agentes y gestores utilizando el lenguaje C++. COMET reduce significativamente el tiempo de desarrollo mediante librerías C++ que realizan la parte genérica de la aplicación TMN. COMET puede generar código C++ de las especificaciones de interface TMN expresadas en la notación GDMO y ASN.1. Esta cadena de herramientas no sólo proporciona servicios básicos de plataforma (por ejemplo, comunicación, manejo de errores y configuración), sino que también asegura una codificación homogénea y

consistente de las aplicaciones TMN. Para disimular la complejidad de los interfaces OSI CMISE, COMET permite al programador poner todo su esfuerzo y atención en lo específico de la gestión de aplicación de la red. El desarrollo de los dispositivos de mediación es facilitado por un número de servicios básicos, aumenta notablemente la posibilidad de crear una virtual Base de Gestión de la Información (MIB) en el dispositivo de mediación el cual permite a un gestor acceder a los objetivos de un agente remoto.

Los servicios de persistencia pueden ser utilizados sobre una base de datos Orientada a Objetos o sobre cualquier otro sistema de persistencia.

Agente Común TMN/Gestor de Servicios

COMET proporciona al programador un compilador para generación automática del código C++ desde las especificaciones GDMO y ASN.1 (**Figura 3**). Genera automáticamente el código de una aplicación de gestor vía:

- Un núcleo de gestor que lleva a cabo objetivos genéricos normalizados con

poderes de comportamiento dentro de la aplicación del gestor: creación, supresión y modificación de Discriminadores de Acontecimientos de Transporte (EFD), mecanismos para envío de requerimientos CMIP y confirmaciones de las respuestas a los requerimientos, etc.

- Un compilador de generación de Código C++ para el acceso a objetos que dependen de la especificación GDMO con el fin de gestionar los objetos en el lado agente.

Para las aplicaciones agente, COMET proporciona:

- Un núcleo de agente con los servicios genéricos de un agente: mapeado entre CMIS y gestión de objetos, implementación del árbol de nombres, competencia automática y soporte de filtrado, respuestas varias a las peticiones CMIS, etc.
- Compilador que genera el código agente C++, el cual depende de la especificación GDMO de un gestor. El agente es generado con un comportamiento por defecto, el cual puede ser llevado a cabo a medida.

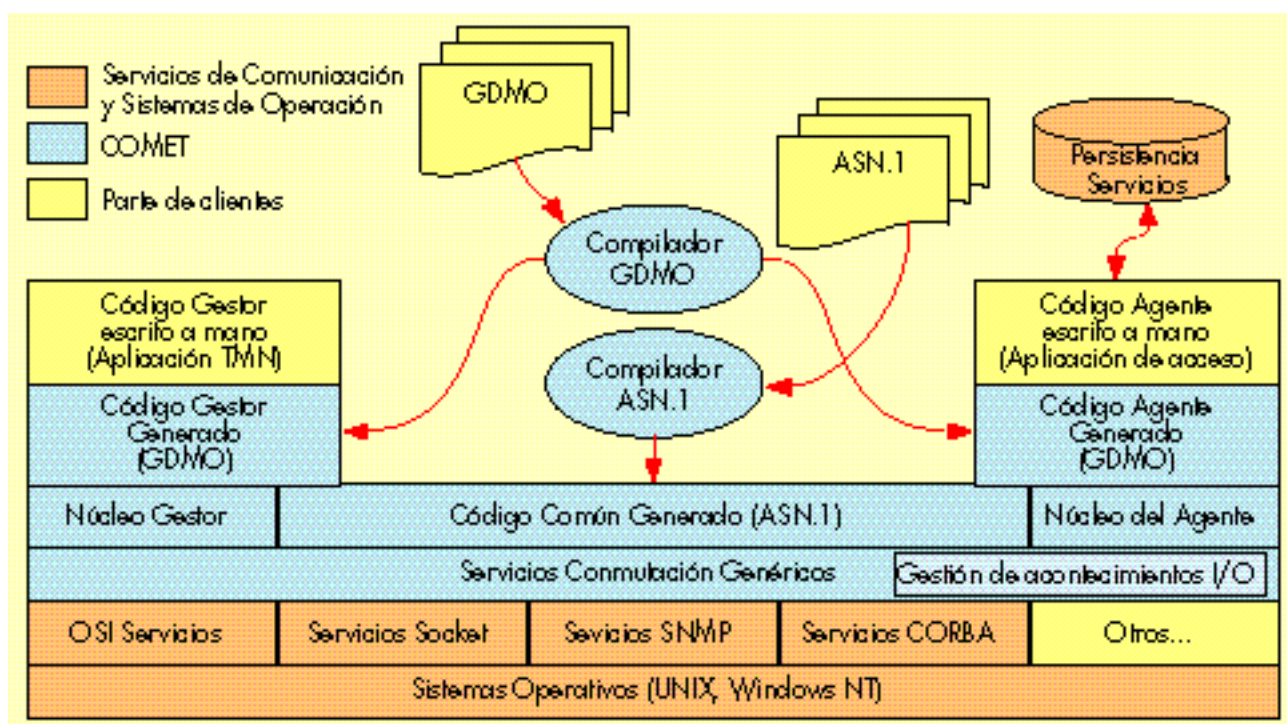


Figura 3 - Aplicación COMET de generación de código.

Servicios de comunicación

COMET proporciona librerías C++ que ocultan la utilización de los servicios CMIS. Así el programador no tiene que escribir cualquier código relacionado con la comunicación CMIS. Sin embargo, el módulo de comunicación puede integrar varios protocolos: CMIS/CMIP, Conexión/CMIP, Internet Inter ORB (Objetivo, Demanda, Agente) Protocolo (IOP), Protocolo Único de Gestión de Red (SNMP), etc.

Dependiendo de la relación gestor-agente, el programador puede escoger varias implementaciones de los servicios CMIS vía un conjunto de enlaces de comunicaciones, los cuales pueden ser enlaces distribuidos con localización transparente de objetos o enlaces de igual-a-igual.

Servicios de gestión procesos de entrada/salida.

Una aplicación TMN consiste en un conjunto de procesos que pueden ser un gestor, un agente o ambos. COMET proporciona todos los servicios necesarios para gestionar el proceso Entrada/Salida (I/O). Esta gestión está basada en el principio de un bucle principal que puede procesar un conjunto de acontecimientos (descriptores):

- Acontecimientos entrantes de comunicaciones, vía todas las clases de canales integrados.
- Interface Gráfica de Usuario (GUI) de acontecimientos o mensajes de aplicación.
- Reloj de expiración de acontecimientos.

Los trabajos de usuario pueden ser programados por su ejecución cuando no se reciben acontecimientos. Una rutina de respuesta a llamada puede asociarse con un acontecimiento, de forma que se activa el proceso apropiado cuando aquel acontecimiento es recibido.

Servicios básicos de plataforma

COMET proporciona librerías C++ que permiten que los servicios básicos de plataforma sean manejados de una forma homogénea y orientada a objeto. Estos servicios conciernen a:

- Servicios opcionales de persistencia para la aplicación TMN de objetos

GDMO/ASN.1, árbol de nombres y transacciones de aplicación en una Base de Datos Orientada a Objetos (OODB), parecida a un almacén de objetos. COMET también proporciona una Interface abierta de Programación de Aplicación (API), permitiendo la persistencia en otras clases de bases de datos (Sistemas de Gestión de Base de Datos Relacionales; ficheros planos, etc.).

- Definición de los parámetros de configuración de la aplicación TMN; nombre y valor.
- Varios niveles de trazabilidad, tales como manejo de errores, actividades de procesos y descubrir acontecimientos en las comunicaciones. El destino de salida para el trazado (consola o ficheros) y el trazado de filtros pueden ser configurados dinámicamente de acuerdo con una máquina o nombre del ejecutable. El trazado puede ser habilitado o incapacitado, en cualquier momento, por cualquier nivel/entidad del sistema TMN.
- Mecanismo genérico para manejo de error de software.

Componentes Software Genéricos

ALMAP incluye un conjunto de aplicaciones genéricas que proporcionan funciones comunes para aplicaciones y sistemas TMN. Estas aplicaciones son genéricas, es decir, que no dependen de un dominio específico de aplicación y pueden ser adaptadas a los clientes e integradas en un período de tiempo para una aplicación TMN:

- *Gestión de topología de la red:* Proporciona supervisión coherente y eficaz de los elementos de gestión de la red (inventario de elementos de red, gestión de los mapas físicos de red, supervisión de los elementos de red, manejo de enlaces).
- *Supervisión de alarmas:* Recoge y presenta en tiempo real acontecimientos de alarmas (**Figura 4**). Proporciona detección rápida de fallos de red y mejora el tiempo de reacción a las anomalías. La supervisión de alarma proporciona todos los servicios relativos al manejo de alarmas en un sistema distribuido TMN rela-

cionando acontecimientos de alarma, apuntando dichos acontecimientos y manejando la alarma activa (incluyendo el registro de perturbaciones), historia de las alarmas, etc.

- *Gestión de acontecimientos y anotación (ELM):* Una herramienta de administración central gráfica la cual proporciona un interface de usuario adecuado para gestionar distribución de acontecimientos y anotación, asegurando que las notificaciones gestionadas llegan donde se necesitan y son propiamente archivadas en anotaciones. La distribución de acontecimientos es realizada por el Discriminador de Transporte de Acontecimientos (EFD) el cual especifica el criterio del transporte del acontecimiento. Además, el ELM hace posible la presentación de toda clase de acontecimientos de un conjunto dado de agentes. La creación dinámica EFD captura agentes supervisados y presenta acontecimientos sobre la marcha. La gestión de acontecimientos y anotaciones realiza la función de gestión del acontecimiento y la función de control de la anotación definida en las normas UIT-T X.734 y X.735.
- *Gestión distribuida del sistema:* Este sistema global de mecanismo de defensa es responsable de la supervisión de la gestión del proceso; aumenta la estabilidad y disponibilidad del sistema. Está encargado de controlar el proceso de ejecución en varias máquinas (HP, Sun, o WNT) en un sistema distribuido TMN. Esta tarea abarca la aplicación de los procesos de comienzo, monitorización y parada, así como defiende al sistema TMN contra procesos de accidentes.
- *Gestión de la seguridad:* La administración centralizada de rendimientos, accesos correctos y perfiles de usuario, ayuda a proteger el sistema contra daños intencionados o no intencionados. Esta focalizado hacia los aspectos de seguridad del sistema TMN:
 - Especifica, almacena y distribuye el acceso de la información de control a través del sistema TMN.
 - Perfila la GUI del sistema TMN conforme a los accesos correctos de usuario.

ascurum : Alarm Sublist : NE-101

| Name | | | | | | | Total |
|----------|-------|-------|---------|--------|-------|------|-------|
| NE-101 | | | | | | | 17 |
| 2 | 3 | 0 | 12 | 0 | 9 | 17 | |
| Critical | Major | Minor | Warning | Indet. | Clear | NACK | |

| Perceived Severity | Friendly Name | Event Date & Time | Event Type | Reserve Status | Clearing Status | Ack. Status | Correlated Notification Flag | Repetition Counter |
|--------------------|---------------|---------------------|--------------------|----------------|-----------------|-------------|------------------------------|--------------------|
| CRITICAL | NE-101 | 1997/09/23 09:07:06 | PROCESSING ERROR | NRSV | NCLR | NACK | NO | 0 |
| CRITICAL | NE-101 | 1997/09/22 12:35:29 | PROCESSING ERROR | NRSV | CLR | NACK | NO | 3 |
| MAJOR | NE-101 | 1997/09/23 09:07:06 | PROCESSING ERROR | NRSV | NCLR | NACK | NO | 0 |
| MAJOR | NE-101 | 1997/09/22 12:35:30 | ENVIRONMENTAL | NRSV | CLR | NACK | NO | 3 |
| MAJOR | NE-101 | 1997/09/22 12:35:29 | PROCESSING ERROR | NRSV | CLR | NACK | NO | 3 |
| WARNING | NE-101 | 1997/09/23 09:07:06 | QUALITY OF SERVICE | NRSV | NCLR | NACK | NO | 0 |
| WARNING | NE-101 | 1997/09/23 09:07:06 | QUALITY OF SERVICE | NRSV | NCLR | NACK | NO | 0 |
| WARNING | NE-101 | 1997/09/23 09:07:06 | COMMUNICATIONS | NRSV | NCLR | NACK | NO | 0 |
| WARNING | NE-101 | 1997/09/23 09:07:06 | COMMUNICATIONS | NRSV | NCLR | NACK | NO | 0 |

Selected : 0

Figura 4 - Presentación actual de alarmas.

- Controla el acceso a los recursos del sistema por las aplicaciones TMN.
- Proporciona el medio de enviar al exterior alarmas de seguridad.

Integración Framework

El framework de integración de ALMAP proporciona un conjunto de servicios integrados en una plataforma abierta y realizaciones extremo-a-extremo de soluciones de gestión en un entorno multi-vendedor y multi-tecnología. Complementariamente a la definición del componente genérico, disponible a nivel básico, la integración framework proporciona un sistema integrado en el tiempo de ejecución, el cual puede utilizarse como un producto completo de núcleo, acelerando así el desarrollo del producto final y el ciclo de integración.

Como se muestra en la **Figura 5**, la integración framework en ALMAP ofrece servicios dedicados a la integración de aplicaciones de gestión de red, en el cual varias tecnologías de equipos de red, varios protocolos de equipos y equipos de diferentes vendedores necesitan ser integrados en un nivel de operación de red. Los servicios proporcionados incluyen:

- Agentes genéricos proxy para la integración de aplicaciones heredadas (por ejemplo, gestores de elementos).
- Gestión de fallos integrada con registro de perturbaciones.
- Correlación y filtrado de alarmas.
- Gestión del Perfil de Asignación de la Importancia de Alarmas (ASAP)
- Topología, gestión de enlaces y gráficos.
- Gestión de activos e inventarios.
- Gestión de acontecimientos y herencias.
- Gestión de seguridad.
- Gestión de las funciones del sistema.

Aplicaciones Añadidas

Además de proporcionar los componentes y características descritas anteriormente, ALMAP es abierto y puede integrar aplicaciones opcionales añadidas para soportar funciones complementarias (registro de perturbaciones, gestión del flujo del trabajo, sistemas de inventario, gestiones de ejecución, etc.).

■ Procesos de Desarrollo e Ingeniería de Software

Los beneficios de una aplicación de gestión de red basada en ALMAP van más

allá del conjunto de características obtenido por la utilización de componentes genéricos TMN. La arquitectura de referencia, su entorno TMN de desarrollo y su modelo de realización orientado a objetos son elementos clave que, junto con un proceso de desarrollo de software bien definido, permiten el rápido desarrollo de aplicaciones de gestión de red.

Esta sección está enfocada a los procesos de desarrollo de software aplicados al dominio TMN y a las aplicaciones basadas en ALMAP. El proceso de software también tiene en cuenta las restricciones de la organización Alcatel y su capacidad de suministrador global dentro de una organización internacional. En particular, su capacidad para el desarrollo distribuido de software en sus centros de competencia de tecnología de telecomunicación, así como con la necesidad de producir componentes comunes o facilidades para producir diferentes productos para diferentes mercados, e incluso para diferentes clientes.

Este proceso de desarrollo de software, el cual ha sido refinado y mejorado desde que fue implementado originalmente en 1995, junto con el desa-

rollo de la plataforma de software ALMAP y el sistema de gestión Alcatel 1353 SH para los equipos de transmisión, está caracterizado por:

- *Gestión de requisitos*, que gestiona los requisitos de definición de productos para su validación y seguimiento a través de las diferentes fases del proyecto, componentes de la arquitectura y actividades.
- *Desarrollo iterativo*, el cual es definido por un sistema progresivo de construcción a través de diferentes pasos de integración y validación de un conjunto de productos. Esto mejora el control de los riesgos del proyecto, proporciona visibilidad externa al departamento de marketing (incluso a clientes), ayuda a reaccionar y a enfrentarse con cambios externos de requisitos, mejora la estabilidad de la arquitectura global del sistema y la ejecución de la validación de productos y permite procedimientos y herramientas de industrialización de productos con anticipación. Una versión del producto global es así dividido en iteración de productos, desarrollo principal de hitos, los cuales pueden ser ellos mismos divididos en los llamados frecuentemente expedición y montaje de paquetes (cada dos meses por ejemplo). Estas iteraciones y paquetes son planificados con antici-

pación y no aparecen durante el desarrollo. Cada iteración es planificada en detalle y refinada al principio de la iteración.

- *Análisis y diseño orientado a objetos*, el cual se realiza en varias vías. Primero, ambas interfaces externa e interna entre componentes de software son descritas utilizando la definición de lenguaje GDMO/ASN.1 de objeto orientado y especifica el comportamiento de los llamados objetos gestionados. Puesto que la cadena de herramientas desarrolladas por COMET es utilizada para ambos lados, gestor y agente, los modelos de realización son estrictamente derivados de la compilación de resultados. Segunda, la metodología de diseño Booch (inicialmente) y más recientemente el Modelo Unificado de Lenguaje (UML), proporcionan directrices de metodología, conceptos comunes y notaciones que pueden ser utilizadas por sistemas de equipos de especificaciones y equipos de desarrollo para refinar el comportamiento del sistema y detallar la elección de diseño.
- *Arquitectura del sistema de referencia*, que es muy importante puesto que proporciona un modelo único de arquitectura, el cual es independiente de la aplicación detallada del modelo. Una arquitectura

específica de producto será derivada del modelo de referencia solamente si hay requisitos específicos. En este camino, el trabajo de arquitectura está muy ajustado y la fase arquitectural es reducida al mínimo. Los errores introducidos durante la fase inicial de arquitectura son el mayor coste a corregir cuando son descubiertos durante la integración o fase de validación. Básicamente, la arquitectura del producto en una probada arquitectura de referencia disminuye el riesgo de errores. La arquitectura del sistema ALMAP de referencia presenta los tópicos siguientes:

- Descomposición de componentes siguiendo el concepto TMN agente gestor, en particular la división entre los datos y la gestión de los componentes de la información y la presentación de los componentes al usuario;
 - Distribución del Sistema, permitiendo transparencia respecto a la localización de componentes;
 - Gestión de datos, incluyendo persistencia;
 - Apertura de interfaces externos con compatibilidad hacia atrás de una versión a otra;
 - Navegación entre aplicaciones, permitiendo la navegación entre la presentación de componentes en el nivel de interface de usuario;
 - Plataforma básica de hardware y software, con un amplio y consistente conjunto de hardware y software de ejecución "runtime" e integrando los productos de terceras partes para sistemas basados en Unix y PC;
 - Topología de sistemas con diversos hardware, desde la configuración del servidor a la estación de trabajo distribuido y configuraciones PC;
 - Proceso de errores, incluyendo gestión de las causas potenciales de errores externos y errores internos del software;
 - Organización de ficheros de ejecución, permitiendo la instalación del sistema, aumentando la recuperación y la coexistencia de diferentes aplicaciones.
- *Utilización extensiva de la generación de código*, utilizando la cadena

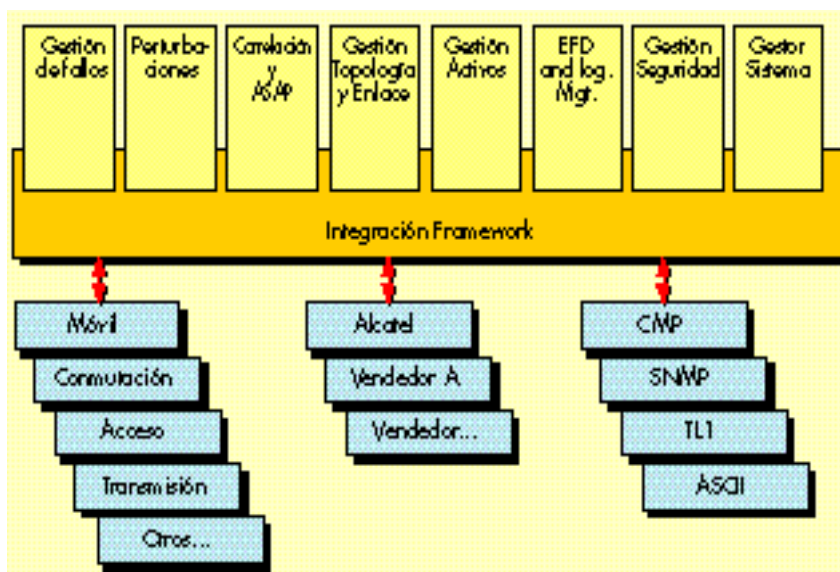


Figura 5 - Integración framework en ALMAP.

de herramientas COMET, con un número significativo de componentes gestor y agente siendo generado de la definición formal de la interfaz GDMO/ASN.1.

- *Organización y automatización de las pruebas*, de la unidad de prueba para integración y validación de la prueba, incluyendo pruebas funcionales y de sistema. El sistema de pruebas incluye realización, defensa y pruebas de dimensionado; si es posible, son realizadas durante las primeras iteraciones del ciclo de desarrollo. El pruebas de sistema y las pruebas de regresión son automatizadas utilizando agentes de testadores, bien por terceras partes o generadas desde la definición de interface GDMO/ASN.1 utilizando la cadena de herramientas COMET.
- *Desarrollo distribuido paralelo y geográfico* permitiendo a las aplicaciones de dominio específico ser desarrolladas allí donde se encuentra la mejor experiencia profesional. El desarrollo de productos lleva un tiempo que puede ser reducido por identificación de partes independientes del producto final, tales como partes de elementos específicos de red de un sistema de gestión de elementos diseñado para gestionar varios tipos de elementos de red. La capa de gestión de elementos identifica partes comunes de los sistemas de gestión de elementos; diversas aplicaciones en el campo específico de las telecomunicaciones (por ejemplo, transmisión terrestre o submarina, acceso, conmutación) pueden ser desarrolladas en paralelo.

El desarrollo en paralelo de partes independientes de productos puede ser iniciado bajo diferentes situaciones, por ejemplo, próximo a donde está siendo desarrollada la tecnología de elementos de red. El proceso de desarrollo identifica entonces las funciones de los diversos sitios de desarrollo, en particular, una unidad de desarrollo de núcleo y unidades de desarrollo local, con claras responsabilidades repartidas en cada una de ellas, y una consolidación global de planificación y medición de proyectos.

- *Gestión de configuración* que soporta el desarrollo iterativo y la organización de desarrollo distribuido. La gestión de configuración no es sólo la herramienta de software más apropiada para controlar y gestionar versiones de software, sino también (y principalmente) la metodología común compartida, conocida y utilizada por todos los actores durante las diversas fases de software: desarrollo, funcionalidades del prototipo en contextos separados de desarrollo, integración, corrección de defectos, producción, mantenimiento después de la entrega, etc.
- *Gestión del cambio* que es utilizada para gestionar el impacto de los cambios -informes de fallos o peticiones que requieren cambios- a través de las fases y grupos del proyecto, de la especificación y arquitectura para desarrollo, integración y validación. El proceso de gestión de cambio es también distribuido para hacer frente a la organización del proyecto: está unido con la gestión de prueba y con la gestión de configuración.
- *Estimación del proyecto*, planificación y seguimiento, con estimación global y detallada del proyecto y metodología de seguimiento. El seguimiento global se hace utilizando la herramienta y metodología Métrica Cualitativa de Software (QSM). Los detalles de seguimiento y cumplimiento en el nivel de equipo utilizan la herramienta Project Management Workbench (PMW), permitiendo la planificación y seguimiento detallado, así como la consolidación al nivel de gestión de proyecto.

■ Conclusiones, Próximos pasos y Oportunidades

Este artículo ha descrito las funciones clave de la plataforma software ALMAP y el proceso de desarrollo de muchas aplicaciones basadas en ALMAP, en paralelo con el desarrollo de la propia plataforma. El proceso de maduración del desarrollo y los principios arquitecturales han acelerado el desarrollo de las aplicaciones complejas de gestión de red.

Como ejemplo, el desarrollo del sistema de gestión de elementos Alcatel 1353 SH para equipos de transmisión ha validado plenamente los conceptos. La **Figura 6** muestra la división del código (totalizan más de 600.000 líneas de código) para el Alcatel 1352 SH versión 3.0 (diseñado para gestionar multiplexores SDH de inserción/extracción) entre los componentes genéricos ALMAP, código generado COMET, código que es común al sistema típico de gestión de elementos (parte de la iniciativa Core EML), y el código específico para gestión de equipos SDH.

Desde el punto de vista de la plataforma, la evolución está ya lista en cierta medida en una arquitectura basada en CORBA para la distribución de aplicaciones y como una norma surgida para interfaces externos. ALMAP será así una plataforma mixta, permitiendo las comunicaciones CMIS/CMIP y IDL/CORBA, tanto interna como externamente. Otra evolución es hacia el lenguaje de programación Java, permitiendo no sólo una mejor portabilidad, sino también mejoras en productividad. Inicialmente, Java está siendo utilizado para usar presentación de componentes, pero posteriormente pueden utilizarse de manera más amplia. Desde el punto de vista de proceso de desarrollo y del entorno, el impacto de la introducción de Java deberá ser tenido en cuenta en todos los niveles del proceso. Una posterior automatización de algunas fases desarrollo o prueba podrá ser llevada a cabo.

La plataforma ALMAP está siendo completada en los diferentes niveles de la escala de gestión TMN (elementos, red, servicio, negocios). En particular, la plataforma completa la iniciativa Core EML en la gestión de los niveles de elementos para proporcionar presentación de gestión de elementos comunes. El objetivo va más allá de una plataforma hacia la normalización de aplicaciones de la gestión en el nivel TMN dentro de Alcatel, para permitir conexión y funcionamiento de aplicaciones de dominio específico dentro de un núcleo de productos único e integración sin cambios de dos aplicaciones de gestión de ele-

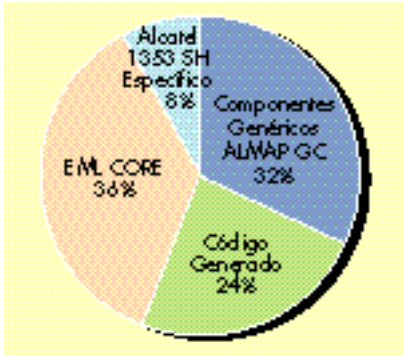


Figura 6 - Desglose de código para Alcatel 1353 SH versión 3.0.

mentos para el beneficio de los clientes Alcatel.

En el nivel de gestión de red, la integración de la estructura ALMAP proporciona un tiempo de funcionamiento de la estructura para la construcción integrada de las aplicaciones de gestión de la red. Más generalmente, la plataforma de software proporciona a Alcatel todas las facilidades necesarias para crecer y concentrarse en un nivel de gestión de servicio el cual ofrece nuevos y mejores servicios a los clientes.

Marc Uszynski es Director de la Unidad de Gestión de Red Corporativa de Alcatel en la División de Negocio de Sistemas de Transmisión, en Marcoussis, Francia.

Paul Wardé es Gestor de la línea de productos de la plataforma ALMAP y los programas de la gestión de red del núcleo en la Unidad de Gestión de Red, en Marcoussis, Francia.

NUEVOS SERVICIOS DE INFORMACIÓN MULTIMEDIA

J-D. CHILLET
P. VIGNARD

Nuevas estructuras de datos y tecnologías de acceso han estimulado el desarrollo de nuevos servicios cercanos a los usuarios.

■ Introducción

Internet es uno de los más importantes desarrollos de los últimos años, y marca el momento en donde la sociedad entra definitivamente en la era de la información. Su éxito está basado principalmente en la disponibilidad de tecnologías que tienen suficiente madurez y han sido normalizadas y, sobre todo, en la existencia de aplicaciones con las funciones necesarias y facilidad de uso, particularmente la World Wide Web (WWW).

En esta nueva situación altamente dinámica, la distinción tradicional entre los mundos de las telecomunicaciones, la tecnología de la información y los medios audio visuales se hace confusa. Los usuarios están buscando servicios innovadores que pueden ser usados sin ninguna limitación de tiempo y lugar, y que puedan manejar los volúmenes enormes de información multimedia que ya están disponibles.

El crecimiento en volumen y el número siempre creciente de fuentes de información está convirtiendo el acceso a la información en una batalla real. Los usuarios dedican cada vez más tiempo a buscar y conseguir información. Como víctima de su propio éxito, se está disminuyendo la calidad de transporte de Internet dada la necesidad de fiabilidad y seguridad. Los tiempos de respuesta y de descarga están llegando a ser prohibitivos.

De esta forma, surge la cuestión de cómo identificar nuevos métodos y técnicas para cubrir los deseos de los usua-

rios de información y transportar con eficacia la información desde el proveedor hasta el usuario final. A causa de que los conceptos de acceso de información se complementan unos a otros, las tecnologías de la estructura de datos en conjunción con nuevas tecnologías de acceso (particularmente las conocidas como "empujar/tirar", push/pull) están proporcionando una solución al problema y representan una oportunidad mayor para el desarrollo de nuevas aplicaciones.

■ La Era de la Información

Internet está considerado actualmente como la tecnología más importante para entrar en la Era de la Información. Su velocidad de crecimiento a lo largo del mundo se mide en meses, mas que en años. El mercado de Internet se expande con incrementos anuales de casi un 40% y el número de usuarios conectados se elevará de los 108 millones, en 1998, a cerca de 400 millones en el 2001.

El espectro de aplicaciones es muy variado, las más conocidas incluyen correo electrónico, transferencia de ficheros, boletines de información y grupos de discusión, y la Web, un sistema distribuido hipermedia (proporcionado información a través de contenidos multimedia) que funciona en modo cliente/servidor sobre Internet. Utilizando la Web, la información puede estar disponible en forma de documentos hipermedia, combinando texto, imágenes, sonido y vídeo.

A través de Internet, la integración de tecnologías suficientemente maduras proporciona a los usuarios un acceso más rápido y ajustado a la información que necesitan para la toma de decisiones y la realización de acciones más eficaces.

■ Información Exacta en el Tiempo y Lugar Exactos

Perseguir un objetivo, bien sea en la vida privada o en la profesional, implica que debemos contar con la información exacta para conseguirlo, en otras palabras, para poder decidir sobre las mejores acciones a desarrollar y para escoger los recursos más apropiados. Es también interesante hacer notar la dualidad entre la información y los recursos (hardware, tiempo, energía): cuanto antes obtengamos la información exacta, menos recursos consumiremos en lograr nuestro objetivo. Sin la información apropiada, la utilización de los recursos puede hacerse tan grande que la consecución del objetivo llegue a ser una ilusión.

Actualmente la Web es uno de los caminos más fáciles y menos demandados para buscar la información, puesto que proporciona acceso a un espacio virtual infinito de información, mucho más grande de lo que somos capaces de manejar. Sin embargo, disponer de mucha información es casi tan contraproducente como no disponer de ella en absoluto, y mucho más si no contamos con los medios para

procesarla. Varios estudios realizados concuerdan con las siguientes cifras: como media, un navegador de la Web gasta el 70% de su tiempo en búsqueda y descarga de información en su ordenador personal, un 25% leyendo información innecesaria y sólo un 5% leyendo información útil.

Las tecnologías de “empujar/tirar” para el acceso y la estructuración de los datos están diseñadas para invertir esta tendencia y permitir a los usuarios encaminarse directamente a la información que requieren, en cualquier momento, en cualquier parte, con un mínimo esfuerzo y con el menor retraso. La energía y el tiempo ahorrados en este proceso puede rentabilizarse simplemente utilizando la información.

■ Modelo de Acceso Empujar/Tirar

Empujar (push) y tirar (pull) son principalmente conceptos de acceso y distribución de los datos. Estas tecnologías posibilitan soluciones tangibles y operacionales para ser implementadas de forma que puedan ser aplicados los conceptos asociados.

El modelo básico conceptual es el modelo cliente/servidor (Figura 1). El servidor atiende las demandas del cliente que está buscando un servicio y envía, en sentido contrario, la respuesta apropiada.

Modelo “Tirar”

En el modelo “tirar” (Figura 2), el usuario busca la información cuando (y solamente cuando) es requerida, y la utiliza cuando la ha encontrado; varias aplicaciones interactivas (principalmente la Web) operan con este principio.

Los motores de búsqueda proporcionan una ayuda inestimable en esta cuestión. Con todo, es necesario decir que el proceso se encuentra aún en una fase primitiva (búsqueda por palabra clave). Sin embargo, la importancia decisiva de la estructuración y calidad de los contenidos de datos para su eficaz análisis, visualización y almacenaje análisis debe reconocerse.

Modelo “Empujar”

En el modelo “empujar” (Figura 3), la información es seleccionada de acuerdo al criterio especificado por el usuario, y

es dirigida hacia él cuando está disponible. El concepto es justamente el opuesto al comentado en el punto anterior.

Así, cuando un usuario registra una fuente de información, el o ella esta-

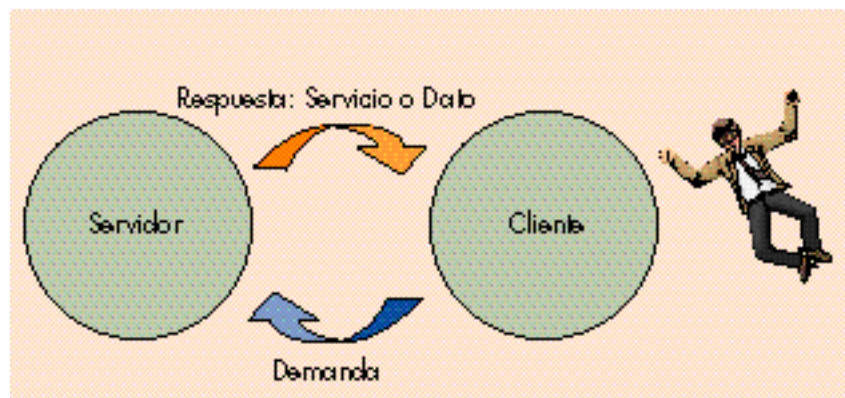


Figura 1- Modelo cliente/servidor.

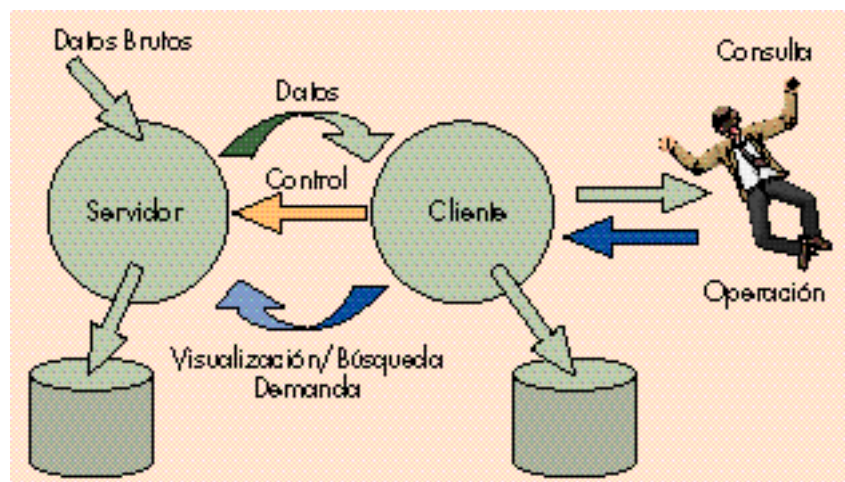


Figura 2 - Modelo de datos “tirar”.

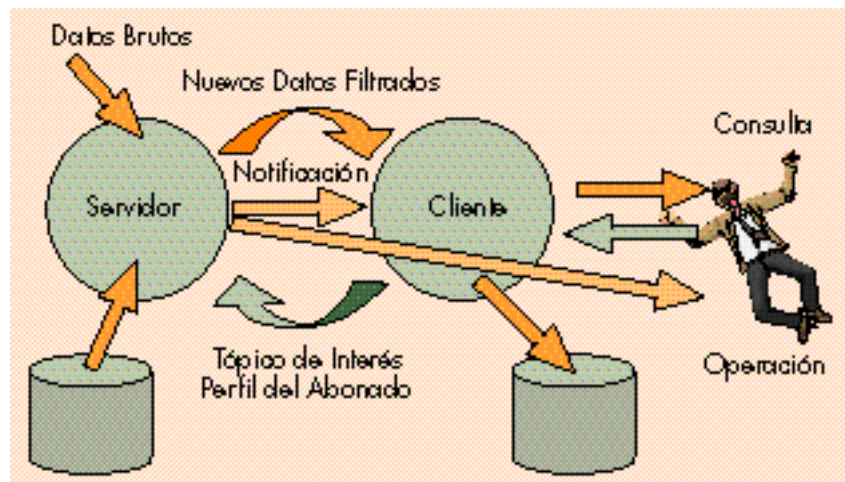


Figura 3 - Modelo de datos “empujar”.

blece un canal de comunicación entre el proveedor de la información y él mismo. En consecuencia, la información demandada es expedida utilizando el método especificado por el usuario: en tiempo real, fuera de línea o bajo demanda. Uno de los principales valores añadidos descansa en el hecho de que el usuario es automáticamente notificado de cuándo ha llegado una información. Como consecuencia de que el usuario ha seleccionado claramente ciertas opciones, preferencias o tareas, la información suministrada es completamente personalizada.

Por analogía, el modelo empujar (push) es más parecido a los periódicos y revistas adquiridos mediante suscripción, y menos parecido a la televisión de lo que podría pensarse en un principio.

Limitaciones de Empujar: el Complementario Tirar

Nosotros podemos ver que el modelo “empujar” es muy atractivo cuando el usuario desea que los datos demandados estén disponibles y sean actualizados automáticamente. Sin embargo, es menos conveniente cuando el usuario tiene que intervenir para controlar el proceso o cuando solamente se desea el acceso ocasional a una fuente de información. En este caso, uno se vuel-

ve al modelo “tirar”, el cual está mejor ajustado a este tipo de uso.

Parece que los sistemas y aplicaciones más eficaces combinarán una juiciosa mezcla de ambos métodos. El usuario es guiado y se le ofrecen las opciones de recepción automática de información o de tomar el control en cualquier momento para influir en el próximo acontecimiento.

■ Contenido Multimedia

Las tecnologías empujar/tirar (push/pull) para el acceso de la información están basadas en la realización e integración de tecnologías básicas, muchas de las cuales derivan del mundo Internet (ver **Figura 4**). A pesar de ello, se pueden identificar tecnologías que son especialmente indicadas para el entorno “empujar”, tales como la definición de canales de información los cuales, entre otras cosas, proporcionan un camino para la descripción de la estructura lógica de las cadenas de información y la frecuencia de las actualizaciones de la información.

En este contexto, las tecnologías utilizadas para describir el contenido y determinar como es procesado ocupa una posición especial. Representan un valor añadido muy alto para los servicios de información multimedia y particularmente para los servicios móviles.

Descripción de Contenidos

El contenido se estructura utilizando un lenguaje que describa la composición lógica del documento. Este es visto como una unidad compuesta que contiene todos los objetos básicos (texto, imágenes, sonido, etc.), los cuales son almacenados en ficheros electrónicos. Cada servidor de datos soporta esta estructura de documento y la proporciona a los clientes.

La presentación normalizada de datos es aún el HyperText Markup Language (HTML), un lenguaje de hipertextos de descripción de datos que es actualmente ineludible sobre la Web. Los documentos que utilizan simultáneamente HTML definen su propio contenido (texto, imágenes, etc.) y los atributos de presentación. Sin embargo, no resulta convenientemente ajustado para definir documentos complejos con abundantes formatos. En particular, el HTML no capacita a los datos para estar totalmente disociados de su formato de presentación. Las últimas incorporaciones a este estándar - víctimas del deseo de apertura que ha conducido este trabajo- han introducido desafortunadas inconsistencias e incompatibilidades.

Consciente de estas limitaciones, el World Wide Web Consortium (W3C), la organización que lidera los cambios en los estándares de la Web, ha empezado a definir un formato más avanzado conocido como Lenguaje de Alto Nivel Extensible (XML, eXtensible Markup Language). Es un meta-lenguaje, es decir, un lenguaje para definir otros lenguajes, tales como contenidos de formato, pero que no contiene información sobre como los datos son presentados en el documento. Esta presentación de información es guardada en un fichero de estilo hoja Excel, la cual es separada del contenido.

Formato para describir los canales de “empuje” de datos

El Formato de Definición de Canal (CDF, Channel Definition Format) es un fichero con un formato de definición de datos para ser suministrados a un cliente por el método “empujar”. Propuesto por Microsoft, el formato XML

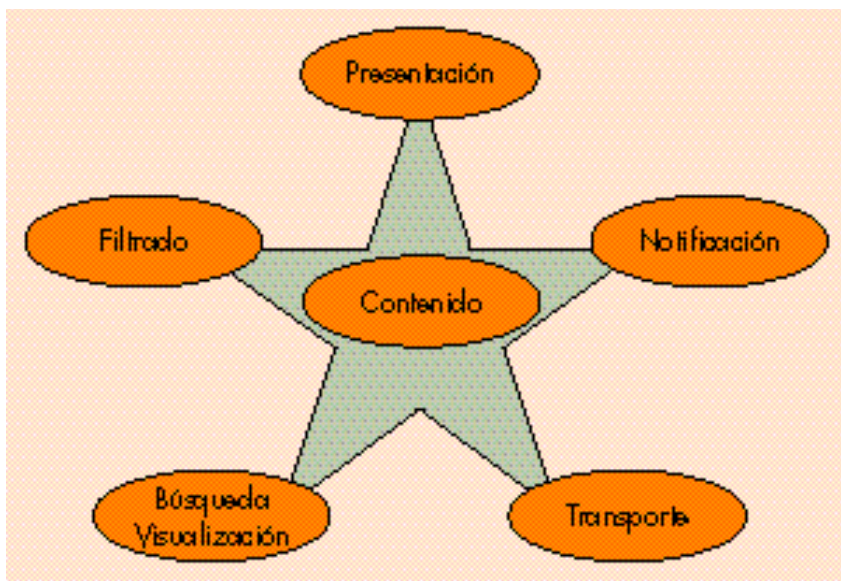


Figura 4 - Dominio de la tecnología de datos básica “empujar/tirar”.


```

<HDMLVERSION=3.0 TTL=0>
<DISPLAY>
<ACTION LABEL=Del ICON=132 TYPE=SOFT1 TASK=NOOP>
<ACTION LABEL=Learn ICON=104 TYPE=SOFT2 TASK=NOOP>
<ACTION LABEL=Summ ICON=145 TYPE=ACCEPT TASK=NOOP>
<WRAP>
Stocks ended higher as Wall Street got a late boost from upbeat
corporate earnings.
<BR>
The Dow Jones industrial average was up 82.65 points, or 0.9 percent, at
9,203.32.
<BR>
But in the broader market, declining issues edged out advances 1,519 to
1,440 on active volume of 722 million shares on the New York Stock
Exchange.
</DISPLAY>
</HDML>

```

Figura 5 - Descripción típica del contenido utilizando HDML.

añade un fichero de definición conteniendo toda la información que define el canal; esta es su principal ventaja para el enfoque empujar datos. Ejemplos de ello son la identificación de los contenidos del canal (lista de documentos disponible), cómo son actualizados frecuentemente y la división en subcanales lógicos (diferentes tópicos dentro de un canal dado).

El fichero CDF es cargado por el usuario al abonarse para posteriormente cada vez que se acceda al canal pueda tener conocimiento de su definición y tener en cuenta cualquier cambio.

Formatos de contenido para móviles (itinerancia)

El Dispositivo Manejado de Lenguaje de Alto Nivel (HDML) es una clase de lenguaje para describir el envío del contenido a los terminales que tienen mejor presentación, entrada y limitaciones de recursos (potencia, memoria, procesador, etc.) o limitaciones de ancho de banda (terminales móviles en particular). Un ejemplo se muestra en la **Figura 5**.

Es similar y completamente compatible con HTML. También la traducción de un documento HTML a un documento HDML es una forma rápida y elegante de adaptar el contenido de un lugar de la Web (servidor soportando un ramal de estructuras de páginas

HTML) para utilización con, por ejemplo, un terminal móvil, teléfono o buscapersonas digital (PDA). En este camino, los proveedores de servicio y de contenidos pueden aumentar significativamente sus ingresos promoviendo sus existentes sitios Web después de la transferencia a la comunidad de usuarios móviles.

■ Ejemplo Práctico: Empujar la Información sobre un Terminal Móvil de Bolsillo

La división de Servicios de Red de Alcatel (NSD) y el Centro Corporativo de Investigación (CRC) han colaborado en el estudio y diseño de una aplicación que mejora el tipo empujar (push) del servicio de acceso de información, basado en las tecnologías normalizadas discutidas anteriormente. Esta también basado en la interconexión entre las redes Internet y el Sistema Global de Comunicaciones Móviles (GSM).

Este servicio proporciona noticias importantes relacionadas con los tópicos de interés para los abonados descritos en sus perfiles. Cuando se añade un abono con el proveedor de servicio, el usuario especifica su interés y el tipo de servicio requerido (notificación, consulta simple, resúmenes, transferencia, tipo de presentación, etc.). Esta

información constituye el perfil del usuario de forma que el servicio pueda ser exactamente personalizado (concepto de uno-a-uno). La información se accede y mira directamente desde un Alcatel One Touch Pocket. La **Figura 6** muestra la disposición básica, mientras la **Figura 7** ilustra sobre la arquitectura de conjunto.

La originalidad de este servicio está basada en:

- Notificación de la posición del usuario móvil.
- Potencia de filtración de los datos brutos, cuya profundidad en el filtrado puede variarse fácilmente de acuerdo con el número de filtros usados (creando una cadena de valor añadido).
- Personalización del servicio y contenido (concepto uno-a-uno).
- Información directamente accesible por el teléfono móvil.
- Contenido adaptado a la petición del terminal receptor: PC, móvil o Web-Touch™ One.

Una evaluación operacional de este servicio se está preparando por uno de los mejores operadores franceses de telefonía móvil.

Los principios que limitan las funciones principales de servicio son brevemente explicadas a continuación.

Información de aumento de la carga bruta

La información básica es recogida por el abonado para el contenido del proveedor (utilizando el método empujar), el cual puede ser una agencia de prensa como la Agencia France Press o Reuters. Los datos originales o brutos llegan como un caudal de datos continuos o intermitentes en el formato del proveedor.

Filtrado

Utiliza las funciones de notificación, una de las formas de añadir valor a un servicio de información.

El filtrado implica el análisis del caudal de datos originales o brutos recibidos para separar la información en cada toma y valorar la importancia que tiene para los requisitos del abonado (descrito en el perfil de usuario). Este proceso consiste principalmente en un análisis

léxico y sintáctico, el cual puede ser complementado por un análisis semántico para filtrar la información requerida.

Numerosas funciones pueden ser utilizadas para mejorar las funciones

básicas de filtrado y por eso representa un valor añadido en este nivel, basado en la plena potencia adicional procesada.

Esto incluye:

- Resumen, proporcionando varios grados de refinamiento del contenido.
- Transferencia on-line.
- Mejorar la información obtenida integrándola y resumiéndola a partir de varios puntos o fuentes de información.



Figura 6 - Principio general de la información "empujar" en móviles.

Notificación

Una vez que la información relevante ha sido identificada, el abonado necesita ser notificado de su existencia. Esto se hace utilizando el Servicio de Mensajes Cortos (SMS) porque el objetivo es una población de usuarios móviles equipados con terminales GSM. Sin embargo, la notificación también puede ser proporcionada por otros medios, tales como correo electrónico, teléfono o fax.

Entrega de la información.

El contenido de valor añadido se hace disponible a los usuarios en un lugar

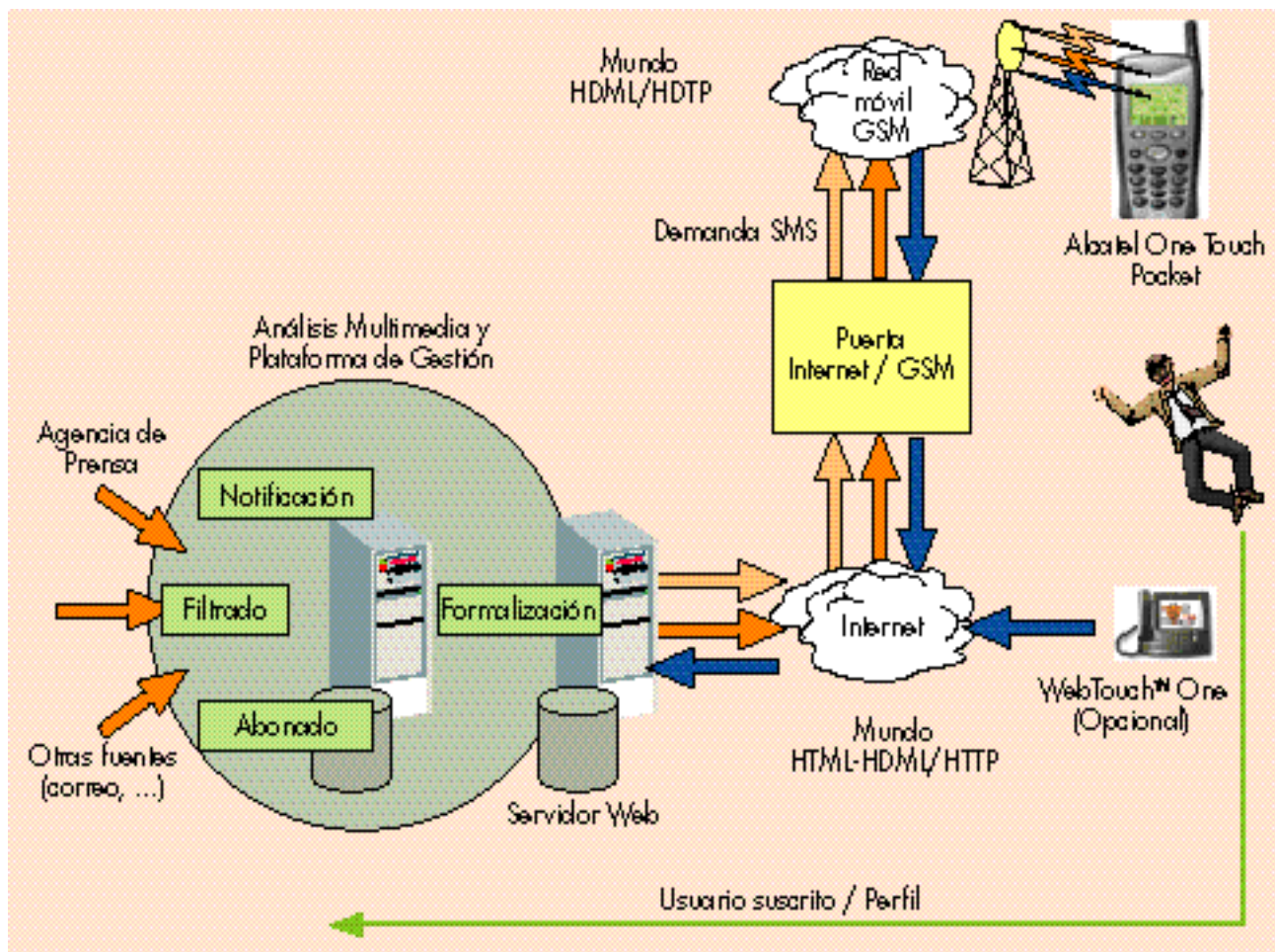


Figura 7 - Arquitectura conjunta del sistema de información "empujar" en móviles.

HDTP - Protocolo de Manejo Dispositivo de Transporte

HTTP - Protocolo de Transferencia Hipertexto

especial de la Web, al cual puede ser accedido a través de la red GSM vía un puerto Internet. También puede ser accedido directamente por un PC utilizando la visualización de los terminales/teléfonos WebTouch™ One. El contenido es entonces adaptado para seguir al terminal que consulta.

Lectura de la información

Este entorno posibilita que los datos sean leídos mediante un Alcatel One Touch Pocket, como se muestra en la **Figura 8**. El usuario accede a los servidores de la Web



Figura 8 - Presentación en un teléfono móvil.

y entra directamente en Internet desde un teléfono móvil. Además, también es factible un servicio complementario de conversión texto/voz.

■ El Futuro: Multimedia y Semántica

No solamente hablamos de un volumen creciente de información disponible, sino que al mismo tiempo dicha información se vuelve a incrementar en textos, sonido, imágenes y vídeo, con orientación multimedia, que a menudo son utilizados en el mismo documento. Actualmente, aunque las técnicas para representar y encontrar la información están relativamente maduras como es el acceso de documento sólo de textos, aquellas que se relacionan con imágenes y vídeos están en muchos casos todavía en la etapa de prototipo de laboratorio.

El análisis del significado de la información del contenido, es el mejor camino de valorar su importancia para el usuario. En adición el primer nivel de análisis (léxico y sintaxis para texto), el análisis de semántica, aunque complejo (particularmente para imágenes y vídeo), puede utilizarse más exactamente para valorar y personalizar los servicios asociados.

Tanto en el modo “empujar”, como en el “tirar”, la descripción de la estructura del contenido es fundamental para el desarrollo de los servicios multimedia. La indisponibilidad de búsquedas inteligentes, de archivo e indexación basadas en el contenido multimedia es un obstáculo para el desarrollo de tales servicios, particularmente en línea.

Entornos de creación de contenidos con Hipervideo.

Para preparar los futuros servicios de acceso a información, tanto si se usan los métodos “empujar” como “tirar”, dos subsistemas de prototipos imagen/vídeo han sido realizados por el Centro de Investigación Corporativa de Alcatel (CRC) en colaboración con dos de los mejores laboratorios públicos de investigación en Francia (INRIA y ENSCachan). Cada uno de estos siste-

mas constituye un entorno hipervideo (analógico con hipertexto), como muestra la **Figura 9**.

El primero es para la creación estructurada de vídeo, que es vídeo enriquecido por información en su contenido. Diferentes funciones están disponibles para extraer firmas, clasificaciones, análisis de movimientos y resúmenes.

El segundo es para la creación de vídeo activable. Un usuario puede identificar un objeto conocido en el vídeo y activar un proceso especial relativo a ese objeto: por ejemplo, para obtener información adicional o un cambio continuo a través del vídeo a otra secuencia mostrando el mismo objeto e incluso un cambio continuo a otro seleccionado con el contenido multimedia.

Tales tecnologías pueden ser utilizadas para crear nuevos y más potentes servicios para cambios, búsqueda y difusión de información multimedia.

■ Conclusiones

El mercado para los servicios de información, particularmente multimedia, fomentará a aquellos capaces de satisfacer eficazmente las demandas fundamentales dirigidas al consumidor para acceso a la información en cualquier tiempo y en cualquier parte.

Los modelos empujar/tirar y las tecnologías asociadas están ya siendo utilizadas para desarrollar soluciones eficaces con alto valor añadido para cumplir las necesidades de los usuarios. Varios de estos productos son ofrecidos por suministradores propietarios, pero los actuales trabajos de normalización se dirigen a vislumbrar soluciones de trabajo interoperativas que finalmente conseguirán un crecimiento muy fuerte del mercado con valores estimados de billones de dólares.

Las soluciones de acceso completan la información multimedia, basadas en aplicaciones igual que las discutidas en este artículo, debiendo permitir a los proveedores de servicios ofrecer a sus clientes servicios atractivos y de alto valor añadido, los cuales son cada vez más y más demandados.



Figura 9 - Entorno hipervideo para caracterización del contenido multimedia.

■ Referencias

- 1 C. Bonnet, J-F Macary: "Tecnologías de Empujar", Eyrolles, Octubre 1997.
- 2 T. Van Landegem, M. Jadoul: "Offering Services: A survival Guide for the Information Age", Revista de Telecomunicaciones 1^{er} Trimestre 1998, Páginas 6-12.

Jean-Denis Chillet es Jefe de Proyecto para proyectos de Información y Redes en el departamento de Software del Centro de Investigación, en Marcoussis, Francia.

Philippe Vignard es Director de la Unidad de Arquitectura de Redes en la línea de productos de soluciones de Red en la División de Servicios de Red, en Vélizy, Francia.

MEJORA DE LOS SERVICIOS SUPLEMENTARIOS DEL ALCATEL 1000 S12, USANDO JAVA Y TECNOLOGÍA INTERNET

Y. DEPRE
H. VANDERSTRAETEN
W. VAN LEEKWIJCK

Java y otras tecnologías Internet pueden mejorar los servicios suplementarios existentes y facilitar el desarrollo de sus contrapartidas en Internet.

■ Introducción

La liberalización y el incremento resultante de la competitividad están forzando a los proveedores de servicios de telecomunicaciones a buscar nuevos caminos para poder diferenciar sus servicios de los que ofrecen sus competidores. Un arma importante dentro de la batalla por los clientes es la provisión de servicios avanzados, útiles y fácilmente accesibles respaldados por un sistema de gestión de los mismos. Java, y la tecnología Web en general, pueden sumar un valor añadido sustancial a los servicios de telecomunicación en términos de: interfaces mejoradas para los servicios existentes o extensiones de los mismos y servicios completamente nuevos. Además, cada vez hay más información acerca de como Java reduce los costes totales en el desarrollo, la entrega y el mantenimiento de los componentes software.

Alcatel 1000 S12

El Sistema Alcatel 1000 S12 se basa en una única arquitectura con un control totalmente distribuido. Las principales características son:

- Conmutación altamente modular.
- Elementos de control conectados a la central.
- Uso de los mismos componentes para todos los rangos de centrales y aplicaciones.
- Comunicación por mensajes entre módulos.

- Estructura software modular.

Este sistema está en servicio en muchos países a través del mundo (cerca de 12.000 centrales y más de 130 millones de terminales). Como resultado, se ha equipado con un amplio rango de facilidades y servicios suplementarios, incluyendo los servicios de identificación del número, llamada en espera, terminación de llamada y desvío de llamada.

El núcleo del Alcatel 1000 S12 ha sido adaptado para efectuar tareas de gestión de la llamada en tiempo real. Las tareas de procesamiento de la información no-en tiempo real pueden efectuarse en un elemento de control basado en UNIX (UNIX-CE) conectado a la central, resultando una arquitectura abierta en la que tanto el hardware como el software dedicado pueden trabajar conjuntamente con los paquetes software y hardware comerciales. Las primeras aplicaciones ejecutándose en plataformas UNIX son para el procesamiento posterior de datos masivos (por ejemplo, registros de facturación), para el mantenimiento de aplicaciones software y aplicaciones de Internet. Así el UNIX-CE es un buen punto de entrada del Alcatel 1000 S12 para soportar el acceso de servicios de conmutación desde terminales habilitados con Java.

Acceso a los Servicios de Conmutación

Actualmente, el acceso a los servicios de conmutación esta constreñido por la

limitada capacidad de interacción de los terminales de usuario. Las teclas del teléfono tradicional pueden usarse para configurar algunos servicios suplementarios, pero imponen severas restricciones según el tipo de interacción y la cantidad de datos que el usuario final puede manipular. Como ejemplos están la activación y desactivación del desvío de la llamada, y la introducción del número al que se desea desviar la llamada. Hasta ahora, los intentos de hacer estos terminales más inteligentes y fáciles han involucrado el uso de hardware y software propietario.

Java e Internet

Java y las tecnologías Internet relacionadas han mostrado recientemente su considerable potencial para reducir la dependencia de los terminales de usuario sobre la tecnología propietaria, introduciendo un entorno independiente de la plataforma. Esto promete un modelo de "escribir una sola vez, ejecutar en cualquier lugar". La emergencia de un amplio abanico de nuevos dispositivos que posibilitan Java, tales como redes de ordenadores, teléfonos Web, el Sistema Global para Comunicaciones Móviles (GSM, Global System for Mobile Communication), y las televisiones Web, muestran que Java está en la senda de convertirse en la plataforma número uno para desarrollar software para clientes. Su mayor beneficio es que el software sólo necesita diseñarse y mantenerse para una única plataforma. Adicionalmente, y debi-

do a su estrecha integración con Internet, los componentes Java pueden almacenarse en un servidor central y desplegarse inmediatamente hacia los terminales de usuario según sus necesidades.

■ Oportunidades

La omnipresente plataforma Java, con su soporte de carga dinámica de componentes software, crea un enorme rango de oportunidades para suministrar servicios de usuario. Dichas oportunidades pueden dividirse en dos clases principales: gestión de servicio de cliente y servicios Internet de valor añadido.

Gestión de Servicio del Cliente

La Gestión de Servicio del Cliente (CSM, Customer Service Management) ofrece un acceso amigable, posibilitando al usuario configurar y monitorizar los servicios suplementarios suscritos por él. Cada usuario es capaz de ver sus datos de configuración y efectuar

un limitado conjunto de operaciones para modificar estos datos.

Actualmente, los datos de un servicio suplementario de un usuario pueden alterarse por operador a través de una consola o un centro completo de conmutación de servicio, o por el usuario mismo a través de las teclas de su teléfono (**Figura 1**). La habilidad del usuario para cambiar los datos está claramente muy limitada y los procedimientos para llevarlo a cabo no son para nada amigables.

Esta situación puede mejorarse notablemente si se usa Java u otras tecnologías de Internet. La introducción de un servidor central conectado tanto a Internet como a las centrales permite a los usuarios configurar sus servicios suplementarios a través del navegador de Web. La Interfaz Gráfica de Usuario (GUI, Graphical User Interface) puede ofrecer una ayuda sensible al contexto, un soporte avanzado de navegación, un "mira y prueba" y otras características que mejoran significativamente la interacción del usuario.

Además, los usuarios pueden conectarse al servidor central y modificar los

datos de configuración de su servicio desde cualquier dispositivo de Internet que permita Java. Esto significa que no sólo pueden usar su ordenador personal para configurar sus servicios telefónicos, sino también cualquier otro dispositivo Internet, tal como el Internet Screenphone de Alcatel, el cual combina las ventajas de un teléfono familiar con las capacidades gráficas de un PC (**Figura 2**).

Otra ventaja de conectar un servidor central a Internet es que el acceso a los datos de los servicios suplementarios es totalmente universal, desde el punto de vista del usuario. De hecho, ya que el único requisito es el que el usuario tenga acceso a Internet, es posible configurar los datos desde cualquier terminal remoto, el cual no necesita conectarse a la central local a través de la línea telefónica del usuario. Por ejemplo, esto permite a un usuario configurar los servicios suplementarios para su casa desde la oficina.

Desde el punto de vista del operador, un interfaz accesible y amigable para configurar los servicios de conmutación, incrementará el uso de los servicios existentes obteniendo así mayores beneficios.

Servicios de Valor Añadido de Internet

Los Servicios de Valor Añadido de Internet (IVAS, Internet Value-Added Services) no sólo extienden los existentes en conmutación con la contrapartida de Internet, sino que también permiten definir nuevos servicios. La extensión de la llamada en espera en Internet parece ser un IVAS muy interesante. Hoy en día, un usuario cuando navega por Internet está ocupando una línea analógica durante horas. Típicamente una sesión de Internet dura un tiempo largo, durante el cual el servicio de llamada en espera no puede usarse ya que los pulsos que manda a la línea afectarían al funcionamiento del módem. Sin embargo, la introducción de un servidor central conectado a Internet posibilitaría al usuario ser avisado de la llegada de cualquier llamada a través del navegador, incluyendo la información del llamante. Podría

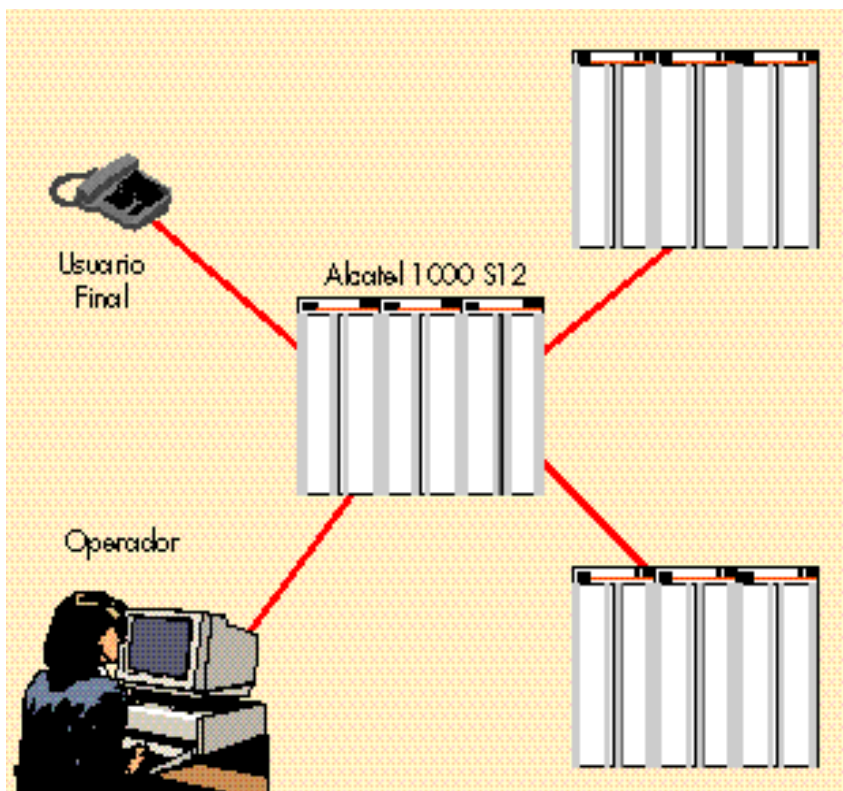


Figura 1 - Configuración tradicional de los servicios suplementarios.

entonces enviar una locución específica al llamante, conectar la llamada entrante al sistema de correo por voz o incluso (si el terminal tiene las caracte-

terísticas oportunas) contestar a la llamada directamente desde el PC usando el protocolo IP (IP, Internet Protocol) sobre voz (**Figura 3**).

Estas extensiones de los servicios existentes desde Internet generarían beneficios considerables para el operador, no sólo por el nuevo servicio, sino

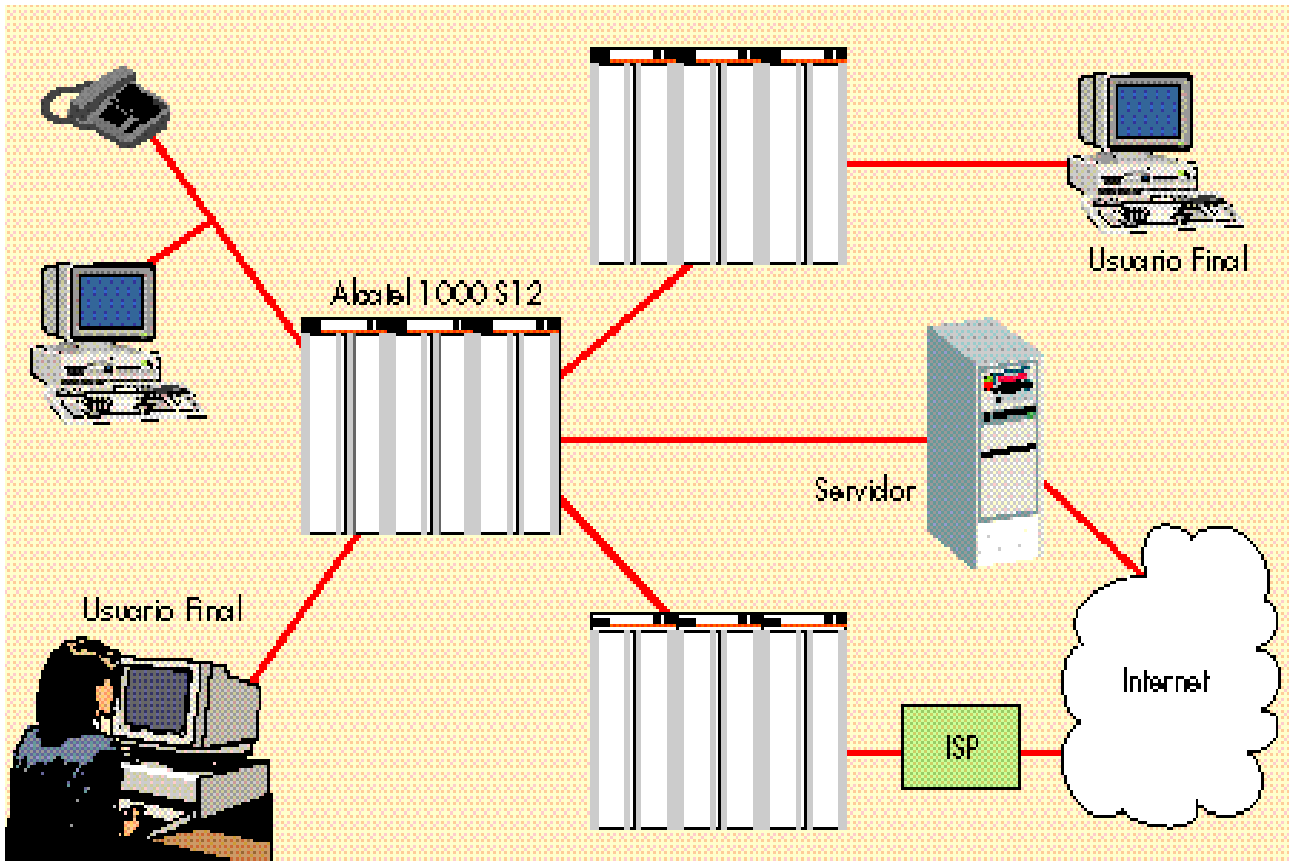


Figura 2 - Configuración de los servicios suplementarios usando Internet.

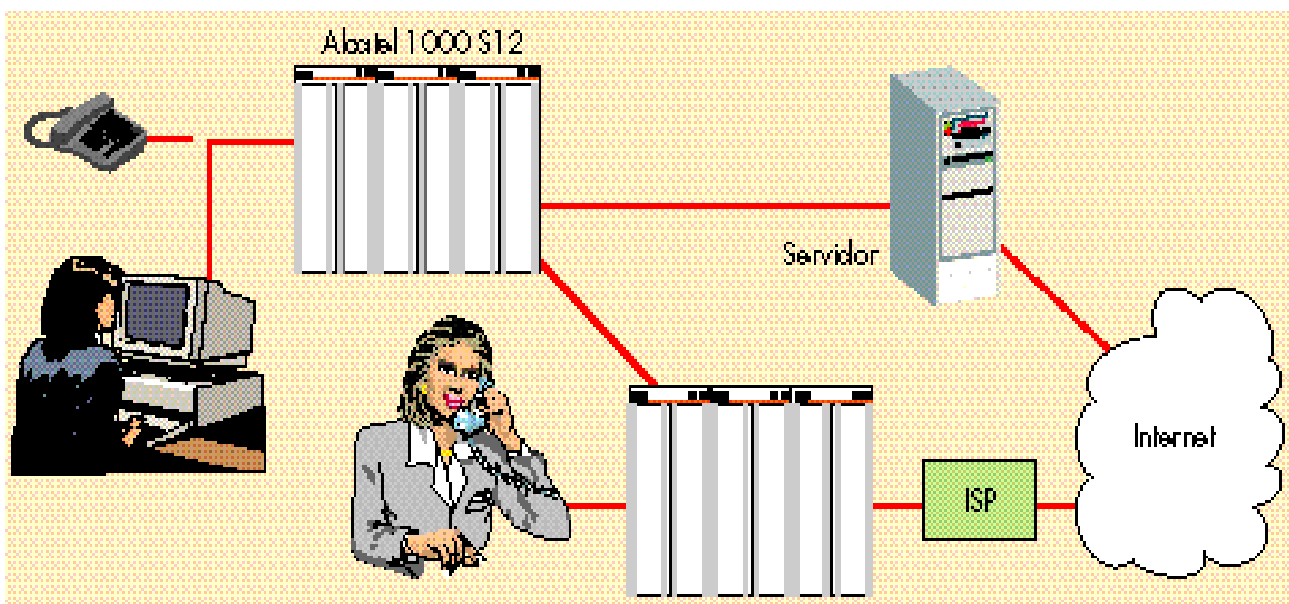


Figura 3 - Extensión de la llamada en espera a través de Internet.

también porque reduciría el volumen de llamadas incompletas.

En ambas clases de servicios, Java y las otras tecnologías Web son las que posibilitan la rápida provisión de servicios amigables, útiles y fácilmente accesibles.

■ Extensión del Prototipo del Alcatel 1000 S12

Para valorar el impacto y los beneficios de Java en el caso de los servicios de conmutación del Alcatel 1000 S12, se estableció un proyecto conjunto entre el departamento de Sistemas del Alcatel 1000 S12 y el Centro Corporativo de Investigación de Alcatel para desarrollar un prototipo de la extensión del Alcatel 1000 S12.

Objetivos

El prototipo tuvo dos objetivos:

- Ensayar la interacción del Alcatel 1000 S12 con los terminales Java, investigando el uso de Java para configurar los servicios suplementarios del Alcatel 1000 S12 (CSM) y extender los servicios ya existentes con la contrapartida de Internet (IVAS).
- Demostrar que la plataforma del Alcatel 1000 S12 es abierta para las nuevas

tecnologías software, tales como Java a través del UNIX-CE, desarrollando una interfaz Java para la base de datos del Alcatel 1000 S12 y un manejador puro en Java para los mensajes.

Servicios Elegidos

Los servicios elegidos para el prototipo fueron:

- *Gestión de Servicio del Cliente:* Activación y desactivación de la llamada en espera; activación y desactivación y aprovisionamiento del número del servicio de la llamada a destino fijo; y la activación y desactivación y entrada del número para las tres variantes del servicio de desvío de llamada (desvío incondicional, desvío cuando está ocupado y desvío cuando no hay contestación)
- *Servicios de Valor Añadido de Internet:* Notificación en la Web de la llamada, en la cual el usuario que está navegando por Internet a través de una línea analógica recibe una notificación de la llegada de una llamada no de Internet, incluyendo la información del llamante.

Arquitectura

Esta nueva clase de servicios está basado en una arquitectura multicapa

que hace fácil la introducción de nuevos servicios y soporta una configuración flexible, un balance avanzado de carga y una variedad de dispositivos de Internet (**Figura 4**).

Conmutador Alcatel 1000 S12

El conmutador Alcatel 1000 S12, controlando todos los datos relacionados con el usuario para los servicios suplementarios, está potenciado con el UNIX-CE el cual ofrece una interfaz abierta y extensible para la conmutación. En el prototipo, las APIs existentes están envueltos con las APIs con Java, promoviendo el uso de Java en los servidores y en el UNIX-CE. De esta manera, puede explotarse la potencia de Java para el desarrollo rápido del software y sus capacidades de carga dinámica con el fin de introducir servicios nuevos o mejorados con un tiempo de corte cero para la central.

Servidor Web

Se usa un servidor estándar que suministra al usuario paginas HTML (HTML, HyperText Markup Language) y applets para el control de los servicios suplementarios.

Servidor de aplicaciones

El servidor de aplicaciones actúa como un servidor central, suministrando la interfaz entre Internet y los servicios de

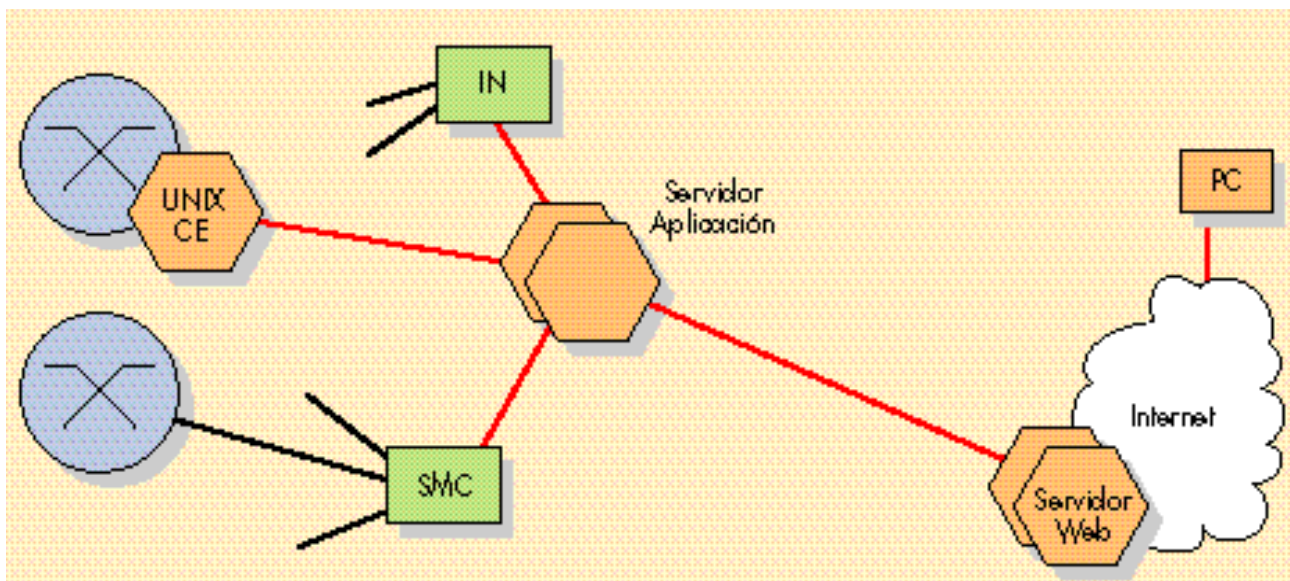


Figura 4 - Arquitectura general para los nuevos servicios.

conmutación. Dependiendo del número de usuarios, puede consistir en una máquina servidora única o un conjunto de servidores cooperantes trabajando en el modo de carga compartida. Alternativamente, es posible mover parte del software del servidor de aplicaciones a los UNIX-CE conectados directamente a la central. Debido a su naturaleza modular, es posible también conectar el servidor de aplicaciones a la red inteligente de Alcatel (RI), para la configuración de sus servicios, o al SMC (Switch Management Center) de Alcatel. Esto ofrece un único punto para todos los servicios relacionados con Internet y hace que la implementación de un servicio en particular (de conmutación o de RI) sea completamente transparente desde el punto de vista del usuario. Además, se pueden extender los servidores con una interfaz para el operador y para el sistema de facturación, ofreciendo al usuario facilidades avanzadas tales como poder visualizar directamente la facturación, suscribir automáticamente nuevos servicios, hacer el pago electrónico de servicios de telecomunicación, comercio electrónico, etc.

Software cliente

Consta de un conjunto de componentes Java de servicios, cada uno con una interfaz gráfica y algún servicio inteligente. En el caso de los servicios de configuración, la parte inteligente es la responsable de la comprobación inicial de errores y asegura la consistencia entre la selección diferente de servicios. Moviendo parte de la lógica de los servicios al terminal, se reduce la carga en los servidores de aplicaciones y también se garantiza una rápida respuesta ante el caso de entradas erróneas o incompletas por parte del usuario.

Estos componentes Java de servicio se cargan dinámicamente al terminal de usuario cuando se necesitan y no imponen ningún requisito o preinstalación software, salvo el navegador con Java habilitado.

En el prototipo, la modularidad del software permitió combinarse el UNIX-CE, el servidor de aplicaciones y el servidor Web en una única máquina

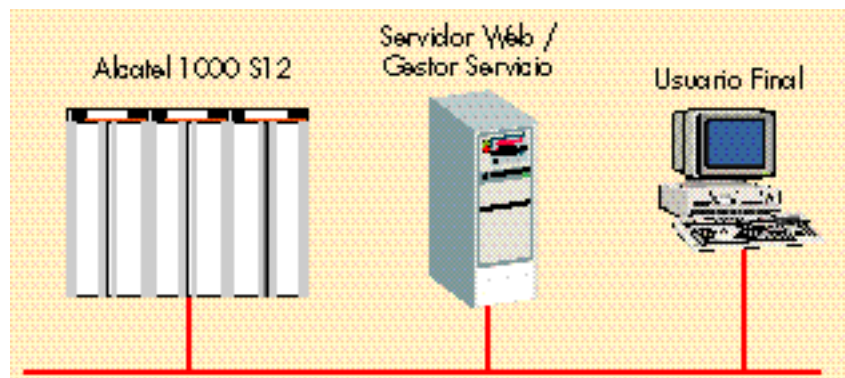


Figura 5 - Configuración del prototipo de Extensión del Alcatel 1000 S12.

física denominada “Gestor del Servicio” (Figura 5).

Escenario 1: Configuración de los Servicios Suplementarios

Las Figuras 6 y 7 muestran, respectivamente, las pantallas de conexión y de configuración del servicio, mientras

que las Figuras 8 y 9 ilustran un escenario detallado para la configuración de los servicios suplementarios:

- Después de conectarse a Internet a través de su proveedor de Internet (ISP, Internet Service Provider), el usuario dirige el navegador hacia el Web del operador.



Figura 6 - Pantalla de conexión.

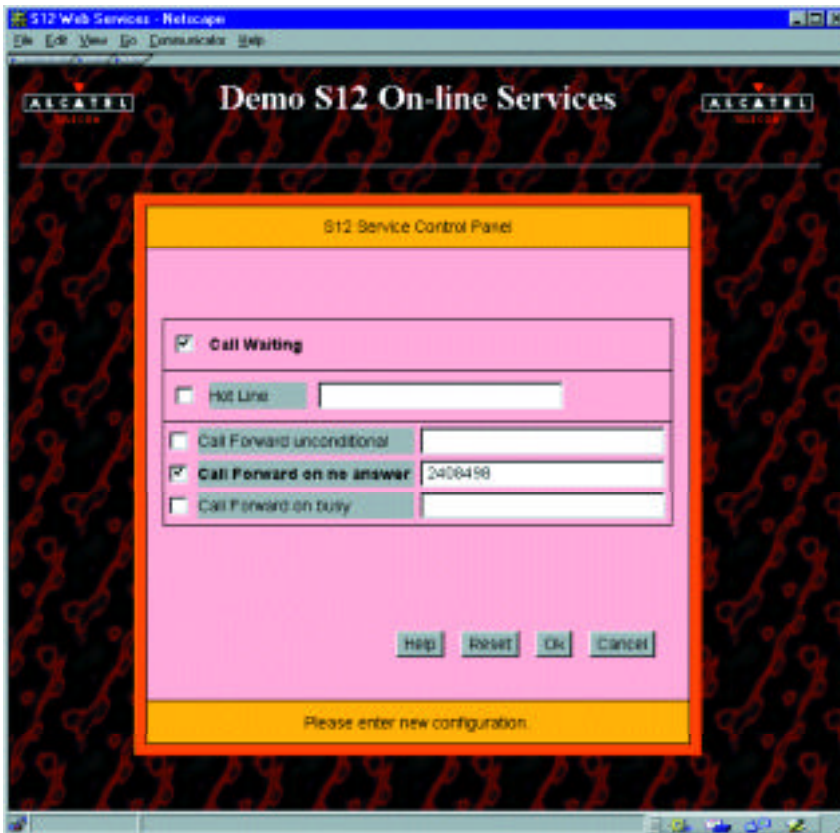


Figura 7 - Pantalla de configuración del servicio.

- El usuario introduce su código y su clave de identificación, los cuales son cifrados antes de ser enviados al gestor del servicio.
- El gestor del servicio verifica la identificación del usuario (código y clave) y la convierte mediante una transformación en un número de directorio.
- El gestor del servicio se conecta a la central para recuperar los datos de los servicios suplementarios asociados con dicho número de directorio.
- La central envía un informe con toda la información relevante al gestor del servicio, donde es analizado.
- Los datos relevantes son redirigidos al applet del usuario final.
- El usuario puede ver la configuración actual y efectuar modificaciones que son enviadas al gestor del servicio (**Figura 7**).
- El gestor del servicio genera los comandos adecuados para modificar los datos en la central.
- La central devuelve un informe notificando que los cambios han sido modificados con éxito en la base de datos.
- El informe es redirigido al applet.
- El usuario o bien se desconecta explícitamente o simplemente sale del navegador.

Escenario 2: Notificación de Llamada en Web

La **Figura 10** nos muestra la pantalla de usuario, mientras que las **Figuras 11 y 12** nos muestran el escenario detallado para este servicio. Los cuatro primeros pasos son idénticos a los expuestos en el escenario 1:

- Después de conectarse a Internet a través de su proveedor de Internet (ISP, Internet Service Provider), el usuario dirige el navegador hacia el Web del operador.
- Desde el servidor de Web se carga una página HTML en el navegador, conteniendo el applet inicial de Java. El usuario ve una página de bienvenida y una pantalla de conexión.
- El usuario introduce su código y su clave de identificación, los cuales son cifrados antes de ser enviados al gestor del servicio.

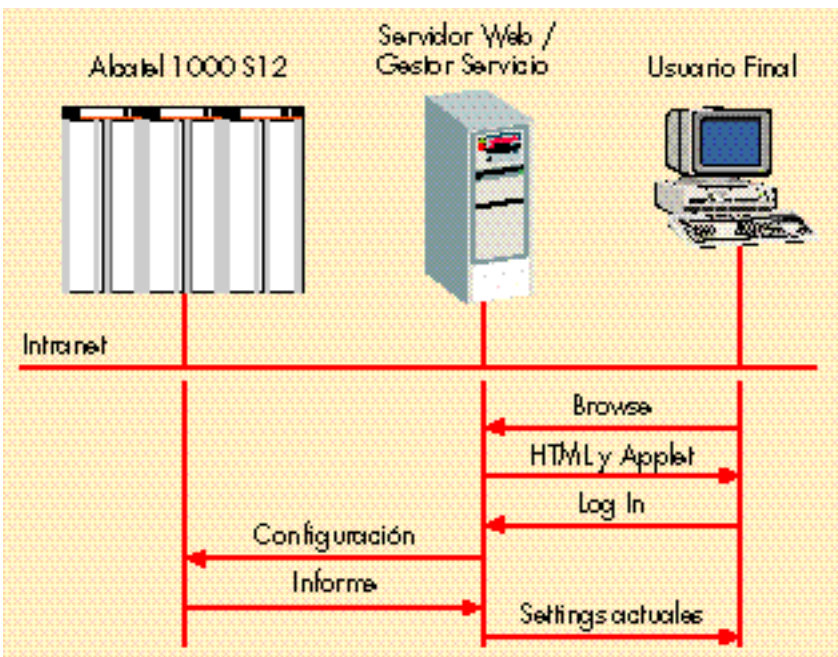


Figura 8 - Escenario 1.

- Desde el servidor de Web se carga una página HTML en el navegador conteniendo el applet inicial de Java.

El usuario ve una página de bienvenida y una pantalla de conexión (**Figura 6**).

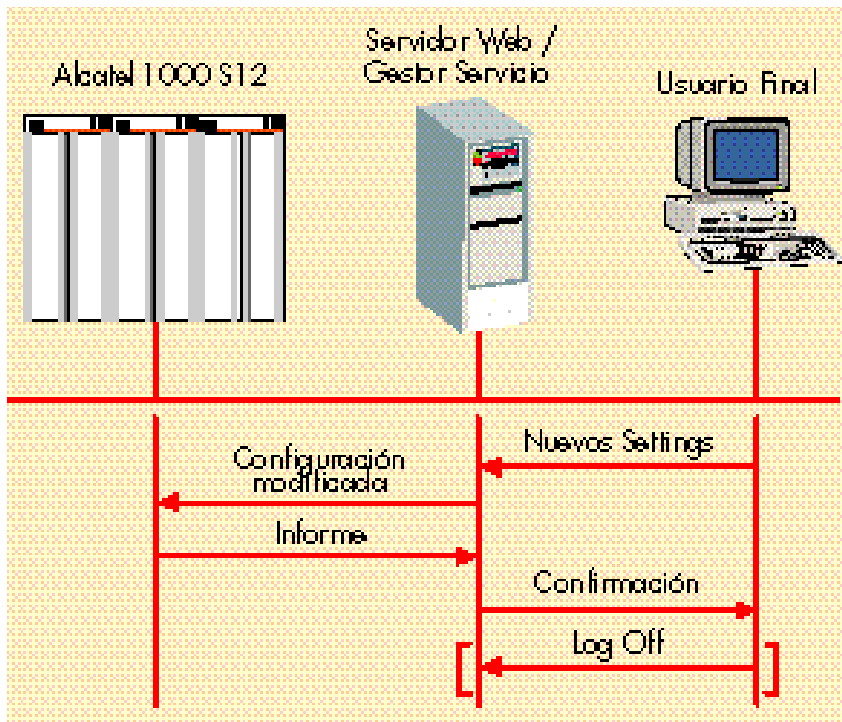


Figura 9 - Escenario 1 (continuación).

- El gestor del servicio verifica la identificación del usuario (código y clave) y la convierte mediante una transformación en un número de directorio.
- El gestor del servicio activa el servicio de notificación de la llamada en Web. A partir de este momento, el usuario puede recibir notificaciones instantáneas de llamadas entrantes a través de este servicio.
- El usuario comienza visualizando en Internet (o efectuando algunas actividades locales, pero conservando el navegador y la conexión a Internet activos).
- Algún tiempo después, una llamada telefónica normal llega al usuario.
- La central envía un mensaje Alcatel 1000 S12 al gestor del servicio indicando el usuario llamante y llamado.
- El gestor del servicio examina el mensaje y envía la información relevante a través de Internet y el applet de Java actualmente ejecutándose en el navegador del usuario.
- El usuario ve una ventana visualizando el numero de directorio del llamante (ver **Figura 10**). Es el usuario el que tiene que realizar la acción apropiada. Puede simplemente igno-

rar la notificación, anotarla y devolver la llamada más tarde o terminar la sesión inmediatamente.

- De nuevo, el usuario puede explícitamente desconectarse o salir del nave-

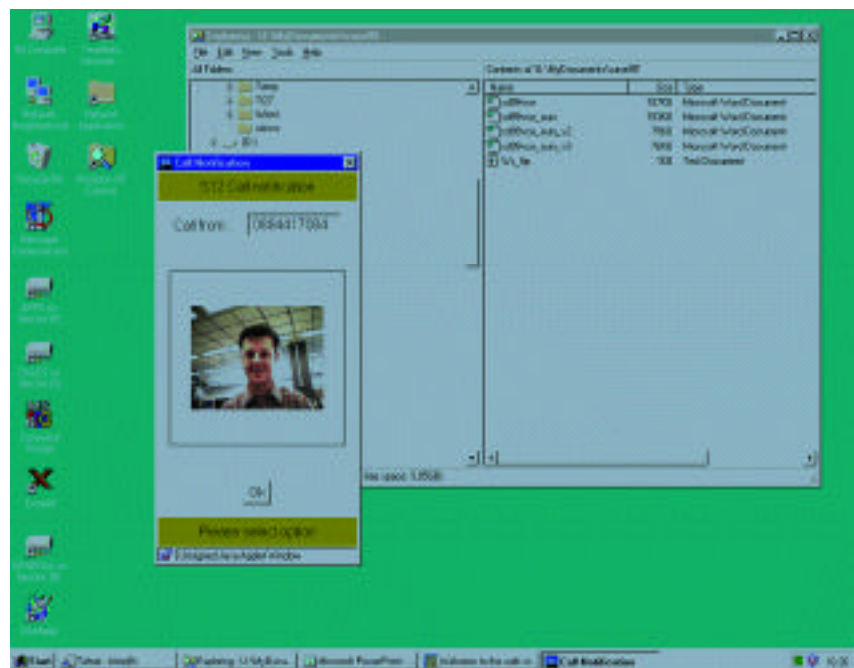


Figura 10 - Servicio de notificación de llamada en Web (en la pantalla mostrada aquí, la única opción es <OK>).

gador para finalizar la sesión. El gestor del servicio desactiva entonces el servicio de notificación de llamadas en Web.

Resultados

Las opiniones de varios operadores acerca del prototipo nos han mostrado que existe un notable interés para acceder a los servicios suplementarios a través de Internet, y especialmente al servicio de notificación de llamada. Además, desde el punto de vista del desarrollo y puesta en marcha, se ha demostrado que Java presenta claras ventajas tanto para el usuario final como para el software del servidor.

■ Aspectos de Seguridad

Está claro que la posibilidad de que los usuarios controlen sus servicios de telecomunicaciones desde Internet, haciendo uso del IVAS, o incluso efectuando pagos, requiere una robusta arquitectura de seguridad. Los requisitos incluyen la identificación del usuario y del gestor del servicio, el cifrado de la información que se intercambia entre el usuario y el gestor y el uso de

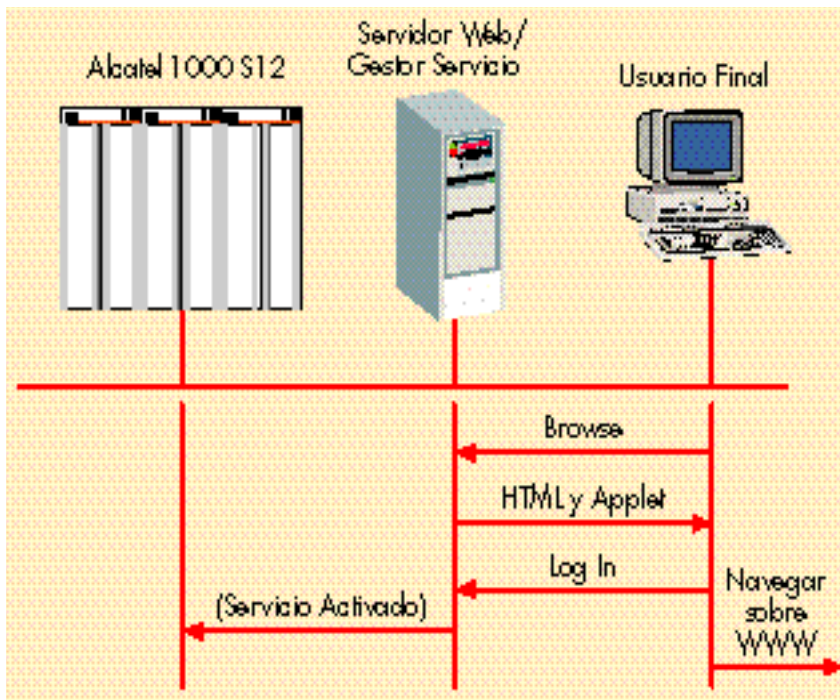


Figura 11 - Escenario 2.

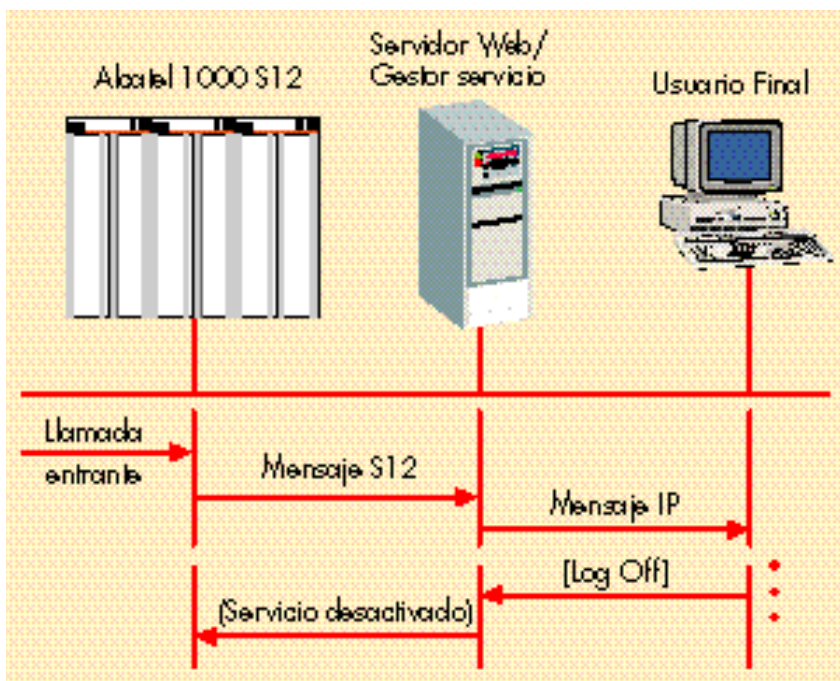


Figura 12 - Escenario 2 (continuación).

firmas digitales para asegurar la ausencia de repudiación del origen de una llamada (esto es, el llamante no puede negar que ha efectuado la llamada). Significa que existe una necesidad para

un entorno de seguridad que posibilite la identificación, la transferencia segura de datos, la gestión de la certificación y los eventos de seguridad en la conexión.

■ Trabajo Posterior

Las posibles extensiones del prototipo pueden incluir la posibilidad de que el usuario reaccione de diferente manera a la notificación de la llamada. Como ejemplos están: desviar la llamada a otro número, conectar el llamante al sistema de correo por voz, enviar una locución específica al llamante y la integración de voz sobre IP de tal manera que el usuario pueda aceptar la llamada desde el ordenador.

Las investigaciones van también encaminadas al uso de otras tecnologías avanzadas para la entrega de servicios suplementarios de forma efectiva y fácil. La aplicación de agentes móviles e inteligentes, el lenguaje XML (eXtensible Markup Language), Java "beans" y la integración de la arquitectura CORBA (Common Object Request Broker Architecture) están siendo evaluadas en detalle para determinar la mejor manera de introducirlas en la línea del producto del Alcatel 1000 S12.

■ Conclusiones

Se ha demostrado el carácter abierto del Alcatel 1000 S12 para las aplicaciones Java, como también se ha demostrado que el uso de Java y otras tecnologías de Internet pueden mejorar significativamente la accesibilidad y el fácil uso de los servicios suplementarios existentes. Adicionalmente, la extensión de los servicios existentes con una contrapartida Internet ofrecerá un amplio abanico de nuevos servicios que apuntan al rápido crecimiento del mercado de Internet. Se espera que en un futuro muy próximo, más y más servicios se extenderán a Internet para ofrecer un paquete de telecomunicaciones totalmente integrado para los usuarios de una manera flexible y fácil.

■ Reconocimientos

Los autores agradecen a todos aquellos que han contribuido al desarrollo del prototipo, especialmente a Stefan Custers del departamento de Sistemas del

Alcatel 1000 S12 de la División de Conmutación.

■ Bibliografía

- 1 Java Technology (<http://java.sun.com>)
- 2 Alcatel Telecom Switching Products (<http://www.alcatel.com/telecom/mbd/products/groups/switching/>).

Yves Depré es el responsable del Grupo de Arquitectura del Alcatel 1000 S12 de la División de Conmutación, en Amberes, Bélgica.

Hans Vanderstraeten es Jefe de Proyecto del Proyecto de Despliegue de Servicio dentro del Departamento Software del Centro de Investigación de Alcatel, en Amberes, Bélgica.

Werner Van Leekwijck es un Ingeniero CASE trabajando en Service Componentware dentro del Departamento Software del Centro de Investigación de Alcatel, en Amberes, Bélgica.

ENTORNO DE CREACIÓN DE SERVICIOS: FLEXIBILIDAD, APERTURA Y EVOLUCIÓN

M. GENETTE

La evolución del Entorno de Creación de Servicios de Alcatel beneficiará a los suministradores de servicios en términos de apertura y rapidez en la creación de servicios.

■ Introducción

El concepto de Red Inteligente (RI) nació a finales de los años ochenta con el objetivo de proporcionar un nuevo rango total de servicios de red. Un objetivo importante fue extender la capacidad para proporcionar servicios a un conjunto más amplio de organizaciones, además de a los suministradores de RI y, en particular, a los suministradores de servicios. Detrás de esta aproximación estaba la necesidad de ofrecer extensas facilidades para la personalización de los servicios.

Alcatel ha cumplido con todos estos requisitos desarrollando un potente Entorno de Creación de Servicios (SCE) que se encuentra en uso desde 1993, comenzando con el suministro de una RI para Singapore Telecom. Los servicios ofrecidos al departamento de internacional de Singapore Telecom incluyen: Servicio Alternativo de Facturación (ABS), que es una familia de servicios de tarjetas de crédito virtuales; Servicio Avanzado de Llamada Gratuita (AFS), incluyendo el kiosco y servicios de valor añadido y la familia de servicios de Llamada Gratuita; y la Red Privada Virtual Internacional (IVPN). Estas son las tres familias fundamentales de servicios de RI.

La **Figura 1** muestra el actual mercado mundial de los productos de RI de Alcatel

■ Enseñanzas Provenientes de la Introducción del SCE

La primera entrega del SCE de Alcatel mejoró significativamente el suministro de servicios, particularmente en lo

relativo a los costes de desarrollo (que fueron reducidos por un factor de seis) y los tiempos de desarrollo (que se redujeron en un tercio). Este fue inicialmente el resultado de explotar completamente las dos facilidades siguientes:

- *Reutilización del software:* El SCE puede utilizarse para crear la lógica del servicio (por ejemplo, scripts) por medio del ensamblaje de bloques software predefinidos, conocidos como Bloques Funcionales Independientes del Servicio (SIB). En la primera implantación del SCE un análisis de la reutilización demostró que el 47% de los SIBs fueron utilizados en el conjunto de las tres familias de servicios relacionadas anteriormente, mientras que además otro 22% del total de SIBs se usaron en dos servicios; otro 31% lo fue en sólo un servicio.
- *Generación del Código Software:* El SCE contiene un generador de código software, que produce el código de la lógica del servicio. La generación automática tiene la gran ventaja de eliminar la intervención humana, por lo que se reducen los tiempos de desarrollo y se incrementa la productividad.

Un área susceptible de mejora, en comparación con la primera versión del SCE, estaba en proporcionar capacidad a los clientes para producir sus propios servicios. La principal razón para ello era la estructura del SCE de

Alcatel, que está basado en dos bloques principales:

- *Entorno de Creación de los Servicios* (o alto nivel del SCE): Utilizado para la creación de la lógica de los servicios por medio del ensamblaje de los SIBs.
- *Entorno de Desarrollo de los Servicios* (SDE): Utilizado para el desarrollo de los bloques software (SIBs y objetos) que son usados por el SCE para crear un servicio.

Para crear un servicio completo, el diseñador del servicio tenía que utilizar el SDE para desarrollar nuevos SIBs que podrían correr sobre el Punto de Control del Servicio (SCP) de Alcatel. En la primera implantación, el SCP corría sobre una máquina propietaria de Alcatel con su propio sistema operativo. Se requería una mayor apertura para permitir a los clientes el desarrollar sus propios SIBs utilizando el SDE.

Sin embargo, los clientes utilizaban totalmente el SCE de alto nivel y podían utilizar un conjunto predefinido de SIBs para implantar sus servicios. Naturalmente, desde el comienzo el SCE de Alcatel era distinto al ofrecido por la competencia soportando todos los componentes software del servicio de una forma integrada, lo que significaba un elemento fundamental para el éxito de la Red Inteligente de Alcatel.

Los componentes soportados por el SCE son los siguientes:

- *Lógica del Servicio:* Programas que tratan el control de la llamada y la



Figura 1 - Actuales mercados del equipamiento de RI Alcatel.

oferta de los servicios (corre en el SCP)

- *Gestión del Servicio*: Programas que tratan con el modelo de datos de los servicios y las capacidades asociadas de gestión de los servicios (corre sobre el Punto de Gestión del Servicio, SMP).
- *Gestión del Servicio. Interfaz Gráfica del Usuario* (GUI): Aplicación de cliente que proporciona al operador del abonado/suministrador del servicio una interfaz gráfica de usuario amigable que facilita la gestión

de los servicios en términos de suministro de datos de servicio, obtención de estadísticas, gestión de alarmas, etc.

■ Evolución del Entorno de Creación de Servicios

Desde su primera introducción, Alcatel se ha esforzado constantemente en la mejora del SCE en cuatro áreas principales y con el objetivo de satisfacer los requisitos expresados por los clientes.

Evolución Tecnológica

La primera razón para la evolución fue la necesidad de hacer frente a la propia evolución tecnológica y la tendencia del mercado hacia el uso del Sistema Operativo (SO) UNIX. El SCE ha evolucionado para adaptarse a los cambios del mercado.

La Release 0 del SCE producía servicios para una plataforma de RI (SCP y SMP) basada en el Alcatel 8300. Con la serie de Releases 1, el SCE proporciona una función SMP que corre sobre

una plataforma hardware comercial bajo UNIX. Con las series de Releases 2, la función SCP producida por el SCE corre bajo UNIX utilizando una base de datos comercial estándar.

El núcleo del SCE ha sido adaptado para soportar el nuevo hardware y software. Sin embargo es importante hacer notar que el concepto SCE y la interfaz de usuario no han cambiado. La mejora se realizó sobre el generador de código de diferentes elementos de los SCE/SDE. En vez de producir programas en código C y utilizar SO primitivos para el Alcatel 8300, el SCE actualmente produce programas en código C++ utilizando primitivos UNIX. Las herramientas de producción, que forman parte de SCE, se han adaptado para hacer frente al nuevo hardware, la nueva arquitectura y al entorno de desarrollo software UNIX. La principal ventaja de esta evolución ha sido la de mantener el mismo principio de diseño de los servicios. Consecuentemente, la fuente (script) gráfica de un servicio en la Release 0 del SCE es básicamente la misma en el contexto de la Release 2 del SCE.

Procesos Software

Una segunda fuerza impulsora, detrás de la evolución, fue la necesidad de mejorar el proceso software asociado con el desarrollo de los servicios para poder reducir el tiempo de desarrollo de los servicios.

En lo relativo al proceso software, la pequeña curva de aprendizaje resultante de la facilidad de uso del SCE reduce los costes de desarrollo por un factor de 2 entre las Releases 0 y 1. El tiempo de desarrollo para el desarrollo de los servicios se redujo a seis meses. No obstante, para la mejora de la calidad del software y para manejar la explosión de diferentes versiones de los servicios, se integraron tres mejoras fundamentales en el SCE:

- En la Release 1, el emulador consta de dos partes, una que emula el comportamiento de la red, y otra que emula el Alcatel 8300 OS. El objetivo era correr el servicio sobre una estación UNIX. En la Release 2, esta

herramienta se integró dentro del SCE para proporcionar a los diseñadores de los servicios la capacidad de probarlos desde su propio puesto de trabajo de una forma integrada. En la Release 2, la red está emulada sobre una verdadera cola de señalización del Sistema de Señalización N° 7 (SS7), aunque los servicios corren sobre la plataforma final. La **Figura 2** muestra la interfaz del diseñador de servicios para la emulación de la red.

- La segunda mejora introducida en el SCE está relacionada con la capacidad de crear y mantener una base de datos de pruebas y en el suministro de métodos para la ejecución automática de pruebas, así como para el análisis de los resultados correspondientes (**Figura 3**). Esto permite realizar pruebas de regresión totales ejecutadas durante la noche cuando se evoluciona el servicio.
- La tercera mejora está relacionada con la integración dentro de la cadena

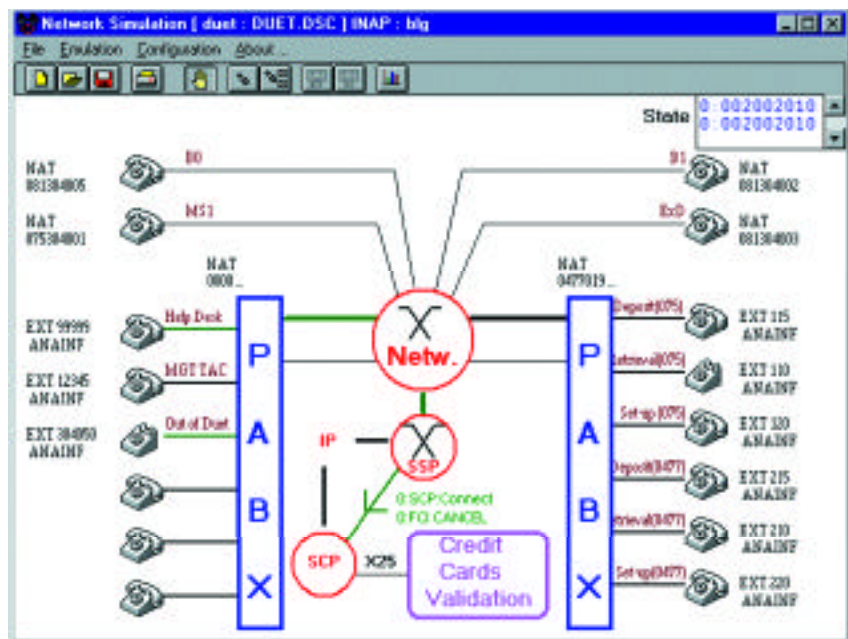


Figura 2 - Interfaz del diseñador del servicio que permite la emulación de la red en relación con el servicio de RI; capacidades totales de traza se proporcionan a través de esta amigable interfaz.

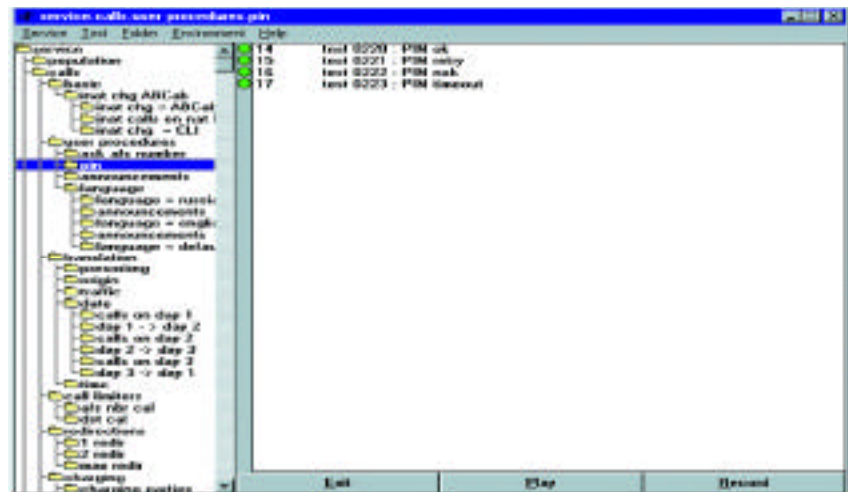


Figura 3 - Base de datos de pruebas asociada con un servicio. Las pruebas detalladas de las facilidades se muestran en la pantalla junto con sus estados (verde, naranja, y rojo).

de herramientas de una base de datos histórica que permite trazar cualquier modificación que se haya realizado sobre un componente software. Esta mejora resuelve el problema relacionado con la reutilización del software asociado con los cada vez más rápidos ciclos de evolución del software.

El resultado de estas mejoras fundamentales del SCE complementadas con el programa de Mejora de los Procesos Software (SPI) hace posible el alcanzar un tiempo de desarrollo de T0+4 (entrega al cliente cuatro meses después de que la especificación se haya congelado)

Independencia de la Red

La tercera razón para la mejora del SCE fue la evolución de la propia red. Los principales impulsores de esta mejora -la evolución del protocolo de la Interfaz de Aplicación de la Red Inteligente (INAP), y la generalización de la función del Punto de Conmutación del Servicio (SSP) en una red compuesta de conmutadores de diferentes suministradores son requeridos por el SCE para enfrentarse a redes heterogéneas.

La forma en que el concepto de SIB ha sido implantado alcanza uno de los objetivos principales del SCE, que consiste en la verdadera independencia de estos bloques de software de las redes heterogéneas que los soportan. Naturalmente, el SCE puede estar asociado con diferentes librerías de SIBs relacionadas con diferentes comportamientos de los conmutadores y diferentes versiones del protocolo INAP. En muchas redes, el SCE de Alcatel permite que los servicios tengan interfaces simultáneamente con diferentes conmutadores, adaptando la lógica del servicio al comportamiento del conmutador de una forma oculta al usuario del servicio. Desde el punto de vista de aprovisionamiento del servicio, el servicio es único y gestionado centralmente por el SMP.

Apertura

La cuarta evolución está relacionada con el requisito del mercado para una

integración más cerrada del sistema de RI con los actuales sistemas heterogéneos de información del cliente: Atención al Cliente y Servicio de Facturación (CCBS), y Sistemas de Soporte de las Operaciones (OSS). Básicamente, esto requiere una interfaz SMP abierta. Se ha cumplido con este requisito extendiendo la capacidad de la interfaz SMP con:

- *Capacidad de Interfaz con la Web:* El SCE ha sido adaptado para producir, durante el ciclo de desarrollo del servicio, aplicaciones JAVA que corren el GUI de la gestión del servicio para el cliente del SMP.
- *Una Interfaz CORBA definida por IDL:* El SCE genera también la definición de la interfaz SMP en un formato IDL estándar junto con el asociada aplicación de servidor Common Object Request Broker Architecture (CORBA). Esto proporciona al cliente acceso a una Interfaz de Programación de Aplicaciones (API), permitiendo al SMP estar totalmente integrado con el sistema de información del cliente por medio del desarrollo de un cliente CORBA.

La **Figura 4** muestra la arquitectura del Object Request Broker (ORB) para la RI.

Además, para alinear el modelo de datos del servicio (**Figura 5**) con el modelo de datos del servicio de información, el SCE incluye una herramienta

de modelado de datos que permite al diseñador del servicio importar/exportar la parte del modelo de datos relacionada con el servicio.

■ Nuevas Oportunidades, Nueva Evolución

Cuando crece el mercado de los servicios y se incrementan las expectativas del usuario en relación con las capacidades de los servicios, el SCE se enfrenta a nuevos desafíos:

- La arquitectura de RI está siendo cada vez más utilizada para controlar la Red Telefónica Conmutada Pública (PSTN) y el tráfico de la Red Móvil Terrestre Pública (PLMN). Esto representa un enorme crecimiento de la RI, creando una fuerte presión en la arquitectura actual en términos de rendimiento de gestión (suministro de estadísticas) y acceso a la base de datos, lo que llevará a una importante evolución de su arquitectura.
- El uso cada vez mayor del Protocolo Internet (IP) para la comunicación de datos y la rápida expansión de Internet requiere que se extienda la aplicabilidad de la arquitectura de RI.
- El volumen de servicios que tienen que suministrarse se incrementa al aumentar el número de suministradores de servicios. Los nuevos sumi-

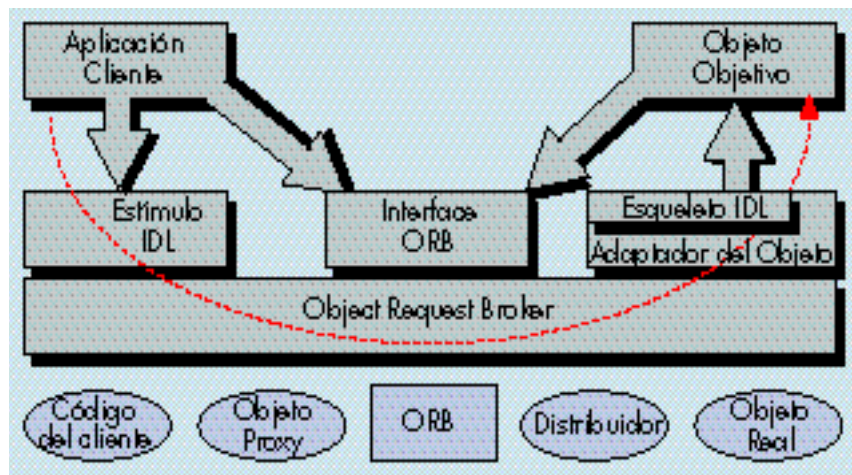


Figura 4 - Arquitectura ORB para la RI, el servidor forma parte del suministro de la RI; se proporciona un API estándar para los desarrollos del cliente.

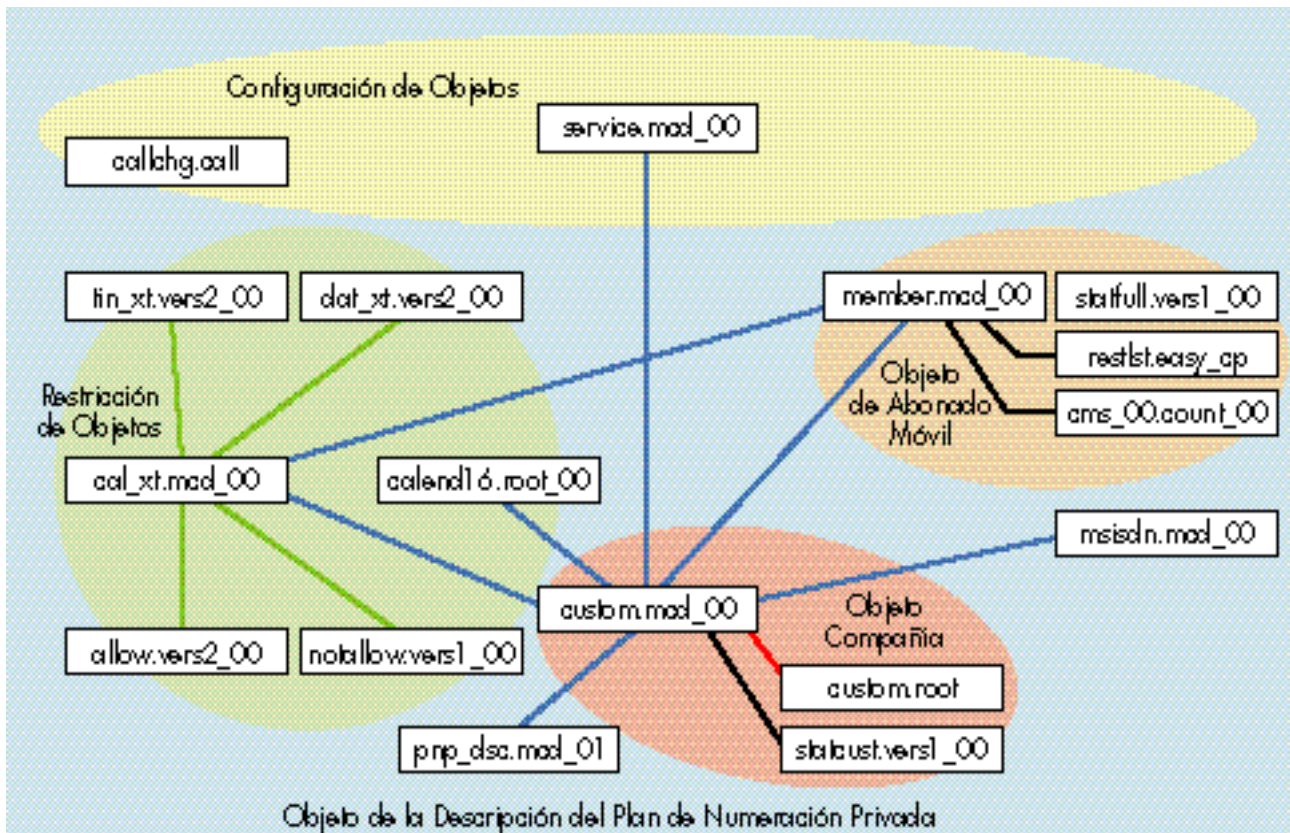


Figura 5 - Ejemplo del modelo de datos de un servicio.

nistradores de servicios son más agresivos en relación con los servicios y el tiempo de disponibilidad de estos en el mercado, máxime cuando ellos representan una parte vital de su oferta de servicio.

- La explosión de la distribución de inteligencia en diferentes puntos de la red y para cada punto de comunicación dentro de la nueva arquitectura de la red.

Estos desafíos hacen necesario la convalidación y revisión del concepto de RI en los siguientes aspectos.

Nueva Arquitectura

Varias mejoras de la arquitectura ayudarán a aliviar las limitaciones actuales:

- *Distribución de la función SMP:* Para resolver las limitaciones en la actividad de gestión causadas por la expansión de la función SMP, esta función puede distribuirse entre varios nodos hardware.

- *El concepto SDP:* Actualmente emergiendo desde la estandarización, el concepto de Punto de Datos de Servicio (SDP) permite la separación de la lógica del servicio de los datos (Figura 6). Este enfoque supera las limitaciones del acceso a la base de datos permitiendo que ésta pueda ser distribuida entre los diferentes nodos. El direccionamiento del correcto SDP se lleva a cabo por el SCP en el que corre la lógica del servicio

Nuevo Modelo de Desarrollo

Para poder ser capaz de desarrollar el gran número de servicios que se le requieren a Alcatel como líder de la IR, la productividad software necesita incrementarse nuevamente. El enfoque de funcionalidad dirigida por datos (data driven) se ha desarrollado para ello. La separación de la lógica del servicio de los datos del servicio es el objetivo de la arquitectura SDP. El nuevo modelo de desarrollo separa los datos software de la máquina que correrá usando estos datos. Es-

ta separación permitirá una mejor reutilización del código que era principalmente independiente de las interfaces externas. En este modelo, la máquina seguirá siendo la misma independientemente de cual sea la interfaz, solamente los datos software evolucionarán. Los SIBs se convertirán en una máquina genérica y en datos software específicos.

Uno puede imaginar que un paso fundamental a través de la personalización ("data driven" versus desarrollo) se ha realizado. Es posible alcanzar el objetivo de permitir a los clientes que ellos mismos desarrollen completamente los servicios si este nuevo modelo es extendido a servicios "data driven".

Jerarquía de los Servicios

Una jerarquía de los servicios tiene que ser introducida para permitir que la inteligencia pueda distribuirse sobre las distintas partes de la red (terminal cliente-servidor). Este es un enfoque natural para alcanzar el concepto SCE de reutilización: siendo el SCE una herramienta que permite que la

inteligencia pueda construirse por medio del ensamblaje de bloques software, solamente la definición de estos bloques tiene que evolucionar. Inicialmente se definió un bloque como un SIB que implanta una fun-

ción; seguidamente el bloque fue extendido a una "macro SIB" que implanta una macro compuesta de varias funciones. La última evolución extiende la definición del bloque a la lógica completa del servicio. El

centro de comunicaciones evolucionará hacia procedimientos de comunicaciones claramente separados desde el transporte de mensajes. En la fase de producción, el diseñador decide cómo y dónde los servicios tienen que ser implantados, y cómo será transportada la comunicación.

Nueva Aplicabilidad

Actualmente, el principal activo de Alcatel SCE es el de poder soportar aplicaciones independientemente del protocolo de interfaz y, por tanto, abriendo el camino para una utilización más extensiva del SCE. Esto amplía el uso del SCE para servicios que no están basados en el protocolo INAP. Consecuentemente el potente SCE puede ahora utilizarse para el desarrollo de cualquier dispositivo de mediación o cualquier aplicación de servidor, como se ilustra en la **Figura 7**. En el pasado, la evolución de la plataforma de RI hacia la plataforma de nodo de servicios extendió los servicios de la RI a servicios de voz. Ahora el mismo tipo de evolución extenderá los servicios de la RI a servicios de datos. La integración de estas plataformas hará posible el desarrollo de servicios que utilicen voz y capacidades IP, con la considerable ventaja de que ambos mundos se unirán dentro de un todo integrado.

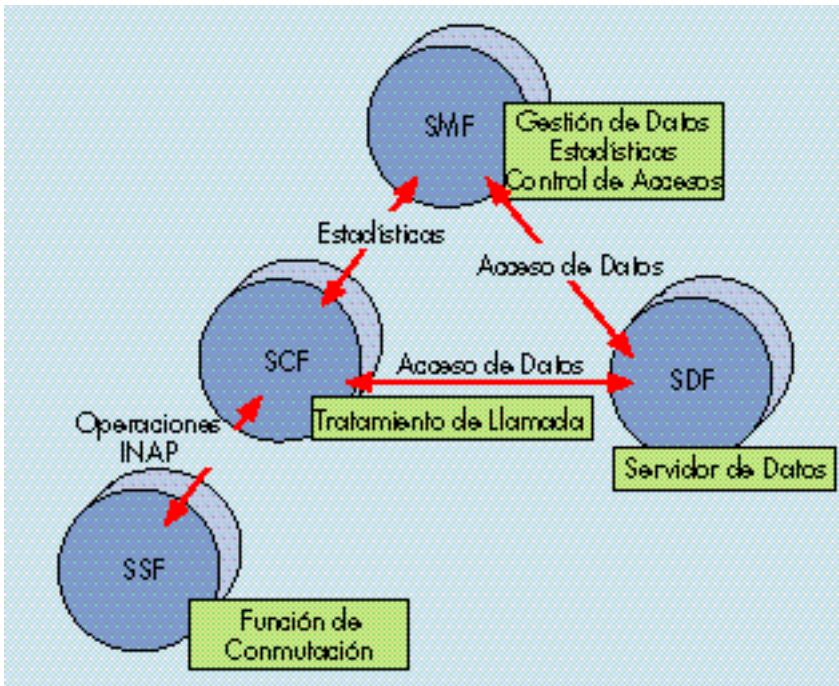


Figura 6 - Nueva arquitectura de RI; el movimiento de la base de datos del abonado a un nodo extendido permite que la lógica trate con bases de datos muy grandes e incluso compartir la base de datos con otros SCFs.

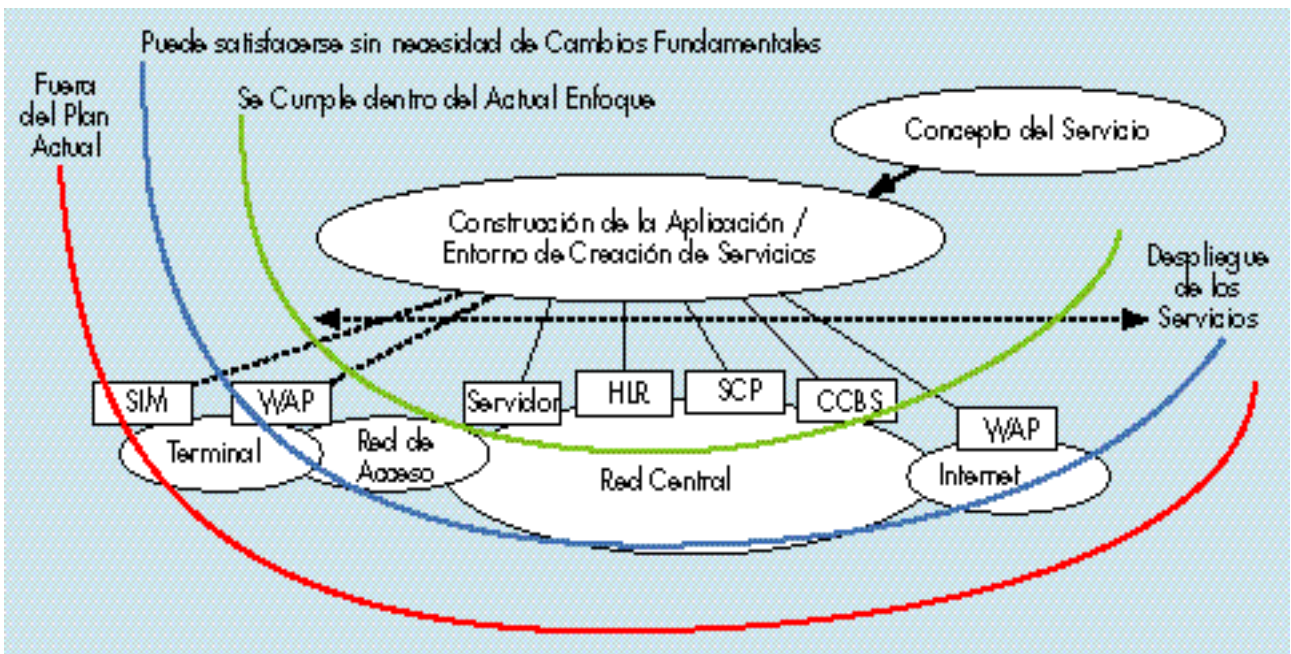


Figura 7 - El proceso software soportado por el SCE podría usarse para un rango más amplio de desarrollo de aplicaciones; este es el principal objetivo de la evolución (HLR: Home Location Register, SIM:Subscriber Identity Module, WAP: Wireless Access Protocol).

■ Conclusiones

La implantación del SCE hizo que Alcatel se convirtiera en un líder mundial en la tecnología de las redes inteligentes. Sin embargo, hoy en día, el mercado de la RI atrae a cada vez más y más clientes tanto desde el mundo de la tecnología de la información como de las compañías de soft-

ware. En vista de estos cambios, es esencial para Alcatel la mejora de su SCE para poder cumplir con los nuevos requisitos de mercado si la compañía quiere mantener su liderazgo técnico en este campo. Alcatel lo está haciendo de una forma coherente, aprovechando totalmente los activos existentes e integrándolos con las tecnologías venideras.

Michel Genette es el responsable de las estrategias tecnológicas dentro de la Línea de Producto de las Redes Inteligentes, en Namur, Bélgica.

ARQUITECTURAS SOFTWARE, LÍNEAS DE PRODUCTO Y ENTORNOS DE TRABAJO

G. DONNAN
J. JOURDAN

El rápido desarrollo del software, la reutilización de componentes y las soluciones software a medida están incrementando la importancia de las arquitecturas software en las ingenierías de producto.

■ Introducción

Un aspecto fundamental que sostiene el creciente papel que están adquiriendo en las telecomunicaciones los productos basados en el software, es el potencial para un rápido desarrollo y una eficaz explotación de la gran masa de software que constituye la más rica de las características de dichos productos. Lo más importante es el creciente solape de los mercados de las telecomunicaciones y de la información, que está creando posibilidades para soluciones comunes, así como tecnologías convergentes en ambos sectores en las áreas de los servicios, aplicaciones avanzadas, sistemas y plataformas. Conforme la competitividad de los mercados incrementa la presión sobre los costes de desarrollo y los tiempos de entrega, la eficiencia de los procesos y los métodos y técnicas de producción del software se convierten en un factor competitivo de primer orden.

Estas consideraciones, entre otras, se están combinando para promover el desarrollo de productos software basados en el ensamblaje, como una respuesta a las demandas comerciales y técnicas de los mercados de alta tecnología. La ingeniería de la arquitectura, la reutilización de los componentes software, las aplicaciones middleware y los entornos de trabajo de alto rendimiento, que utilizan el paradigma de tratamiento informático de los objetos distribuidos, son las herramientas tecnológicas predominantes en la actualidad. Todas ellas ayudan a conseguir altos niveles de

productividad software y grandes economías de escala, ofreciendo, al mismo tiempo, rapidez y una personalización rentable, para satisfacer las necesidades de los diferentes mercados, un requisito cada vez más importante en el sector de las telecomunicaciones.

■ Oportunidades

Los productos de telecomunicaciones basados en el software, especialmente los servicios y las aplicaciones avanzadas, se enfrentan a importantes desafíos críticos de negocio en términos de eficacia, tiempo de desarrollo y rápida evolución, como respuesta a los cambios en los requisitos de mercado

tanto para los grandes como para los pequeños sistemas basados en el software y con valiosas características internas incorporadas.

En este espectro del desarrollo software que va desde “la programación en grande” hasta “la programación en pequeño”, la arquitectura está emergiendo como el factor dominante en la definición de la estructura de referencia a través de la cual las crecientes inversiones en el software serán reutilizables, remodeladas y repetidamente explotadas, como parte de un escenario de producto integrado en el cual el middleware y la plataforma de servicios están continuamente evolucionando.

Los enfoques hacia la producción de software avanzado han madurado hasta

■ Antecedentes

El Alcatel/Thomson-CSF Common Research Laboratory, dedicado a la arquitectura software y a los entornos de trabajo, fue formalmente establecido en septiembre de 1998 e, inicialmente, desarrolla un programa técnico de trabajo de tres años. Entre los aspectos tratados por el Common Research Laboratory se encuentran aquellos concernientes a la eficacia, rapidez y tasa de desarrollo de los grandes sistemas basados en software. Típicamente, el laboratorio investigará arquitecturas, enfoques sobre componentes, tecnologías de personalización y mecanismos de plataformas para soportar sistemas distribuidos en tiempo real.

El programa está estructurado en dos partes principales. Una parte está dirigida a la definición y a la construcción de arquitecturas de líneas de producto, mientras que la otra está estudiando la definición y realización de un entorno de trabajo capaz de soportar sistemas basados en componentes que requieren distribución, tiempo real, y finalmente, garantías de continuidad del servicio.

el punto de que el rápido desarrollo del software basado en la reutilización de componentes para suministrar soluciones software específicas, está ahora aplicándose activamente. Los beneficios proporcionados por este enfoque pueden ampliarse con el uso complementario de entornos de trabajo -una agregación de middleware especializado (tercera parte específica y genérica)- que explotan diversos sistemas operativos comerciales. Ellos ofrecen un grado adicional de diversidad y flexibilidad en la explotación de las características de la plataforma.

La capacidad para soportar el paradigma cliente/servidor, en sistemas cada vez más distribuidos, también requiere que el software se encuentre en forma de componentes para facilitar su ensamblaje basado en la arquitectura. en los productos y su despliegue flexible en sistemas y plataformas heterogéneos y de diferentes tamaños.

En el software de telecomunicaciones, este desafío típicamente implica la explotación de arquitecturas multiescala, de componentes con características múltiples, de tecnologías middleware distribuidas, robustas y en tiempo real,

y de plataformas en tiempo real de alta disponibilidad. Más que nunca, la necesidad de soluciones de ingeniería personalizadas a gran escala utilizando componentes genéricos modificables, basadas en diversas tecnologías y posiblemente incluso diferentes soluciones, es fundamental.

Este artículo estudia la arquitectura y su uso en la construcción de familias de líneas de producto. Además, se ocupa de la definición y suministro de un marco de trabajo capaz de soportar sistemas basados en componentes que requieran distribución, tiempo real y, finalmente, garantías de continuidad del servicio.

■ Arquitectura, Desarrollo Línea de Productos

Muchos productos de éxito están basados en el concepto de Línea de Producto, que puede definirse como un grupo de productos que comparten un conjunto común de características gestionadas que satisfacen las necesidades de un mercado en particular (en lo concerniente a tamaño, evolución tec-

nológica, requisitos, etc.). Conforme la importancia de la tecnología aumenta, la mayoría de estos productos llegan a convertirse en información intensiva.

La estructuración de un sistema en una familia de productos permite que el esfuerzo de diseño sea compartido por toda la familia, ayudando a contrarrestar, de esta manera, el impacto de una complejidad siempre creciente. Esto ayuda a sostener la tasa de evolución del producto y su introducción en nuevos mercados.

Hasta la fecha, se ha desarrollado una madura tecnología de ingeniería software para crear, primeramente, productos individuales, y por tanto los actuales modelos de proceso no son los adecuados para el desarrollo de líneas de producto. La idea que hay detrás de un enfoque de línea de producto (**Figura 1**) es la construcción de un nuevo producto (sistema, aplicación, etc.) desde un conjunto común de activos definidos durante el desarrollo de productos previos pertenecientes a la misma familia. Un activo software podría definirse como una descripción de una solución parcial al problema total que se plantea, éste podría ser un

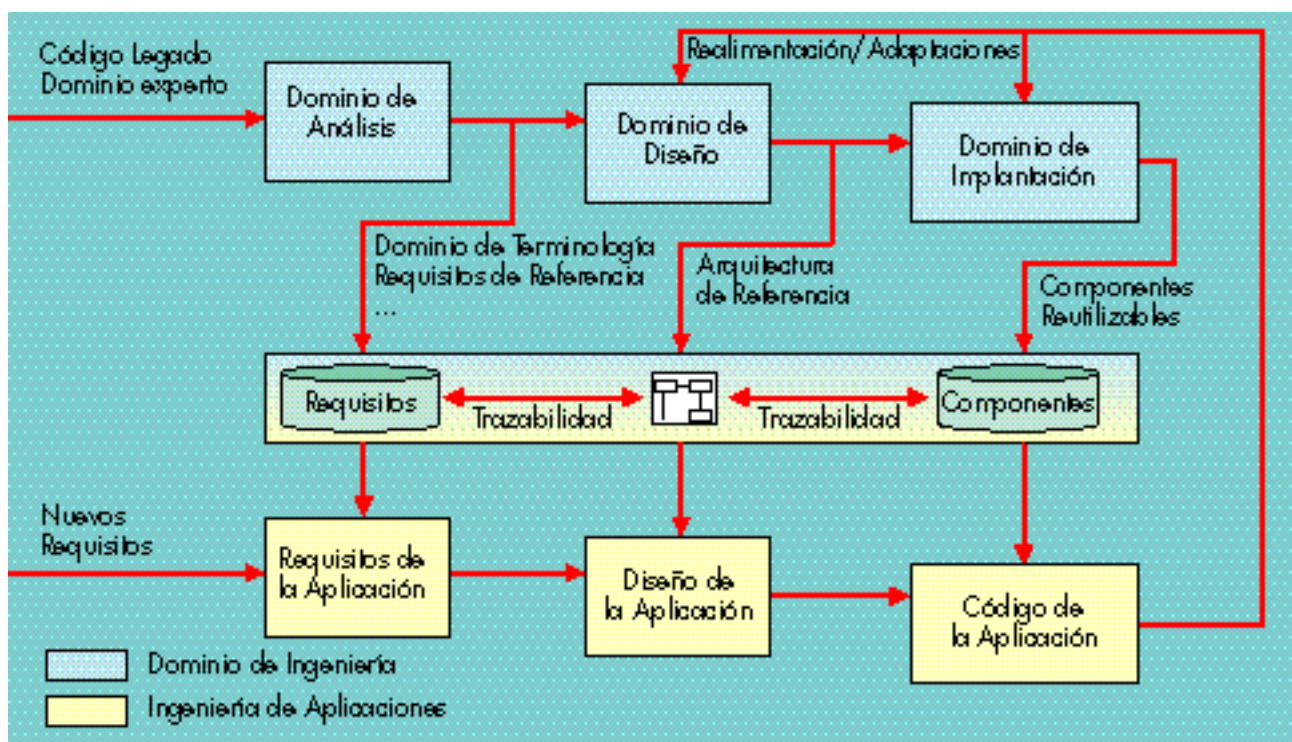


Figura 1 - Ingeniería de Línea de Producto.

componente o requisitos conocidos o elementos de diseño que un arquitecto/ingeniero ha utilizado previamente para construir o modificar un producto software.

Una línea de producto define dos estructuras de ingeniería diferenciadas:

- *Ingeniería de dominios*: Responsable del diseño y gestión del dominio de los activos, incluyendo la Ingeniería de Familias de Aplicaciones, Arquitectura de Referencia, e Ingeniería de Sistemas de Componentes
- *Ingeniería de Aplicaciones*: Responsable del desarrollo de un nuevo producto utilizando el dominio de los activos. Incluye la Ingeniería de Sistemas de Aplicaciones.

Las principales tareas de esta actividad se describen a continuación.

Análisis y Modelado de y para las Familias de Productos

La definición de los activos de los sistemas existentes es una tarea fundamental que requiere el descubrimiento y representación de una gran parte del producto y conocimiento de ingeniería. Para la búsqueda de la información requerida se necesita una fase de análisis en profundidad y un equipo multidisciplinar. Esto es imprescindible para poder recuperar todo el conocimiento existente acerca de la familia de producto (técnica, fiabilidad, flexibilidad, etc.) y construir la arquitectura de la familia de producto que pueda formar la base de productos altamente competitivos en una versión múltiple, en un entorno de ingeniería de frecuentes entregas. Diferentes análisis tienen que realizarse: análisis del dominio, análisis de arquitectura, análisis de aspectos y análisis del sistema.

Definición y Descripción de Líneas de Productos

Aunque en los últimos años se ha hecho la luz considerablemente sobre los elementos técnicos y metodológicos para abarcar las líneas de producto, una respuesta integrada se encuentra todavía en sus inicios. Un objetivo fun-

damental es el de proporcionar un paquete tecnológico, que incluya los procesos, métodos y técnicas necesarias para construir los diferentes dominios de activos (requisitos, arquitectura de las líneas de productos, componentes) utilizados en una línea de producto. El diseño de una línea de producto implica la identificación de la forma de construir la arquitectura, de las características comunes y de las variables de todos los activos para poder explotarlos durante la fase de personalización.

Derivación de un Nuevo Producto, Variantes

Las variantes del producto a liberar tienen que construirse para satisfacer los requisitos específicos de funcionalidad o calidad. En particular, las selecciones tienen que realizarse considerando los componentes que constituirán el sistema de aplicación, los parámetros (propiedades, atributos) de los verdaderos componentes genéricos ejecutables, las interfaces que soportan los aspectos de calidad y, finalmente, la arquitectura de producto individual que implante las características específicas de un producto (pero basándose en la arquitectura de referencia genérica).

En algunos enfoques esto se llama ingeniería del sistema de aplicación. Ésta utiliza un proceso que incluye la captura de requisitos, análisis de robustez, diseño, implantación, pruebas y empaquetado del sistema.

Pruebas de la Arquitectura de la Línea de Producto

Las pruebas del enfoque de la arquitectura implican el análisis del dominio para elaborar la arquitectura de referencia: identificación de modelos de arquitectura en el dominio concernido, evaluación de la arquitectura actual versus otros modelos propuestos, identificación de la flexibilidad de la arquitectura frente a cambios y la provisión de herramientas para la captura de técnicas racionales y la documentación de la arquitectura y sus variaciones utilizando el Unified Modeling Language (UML).

La capacidad de trazar los requisitos de referencia versus la arquitectura de referencia (especificación de los enlaces entre un requisito de referencia y un elemento de arquitectura, particularmente mapeando las necesidades de un mercado en concreto) es una parte imprescindible del enfoque total, de tal manera, que todas las restricciones del diseño y su pertinencia para satisfacer los requisitos del producto puedan ser verificadas en su totalidad.

Se pueden citar algunas pruebas importantes como una indicación de las potenciales ganancias industriales. Durante los últimos cinco años, el Departamento de Defensa de los Estados Unidos ha organizado varios programas intensivos con el objetivo de institucionalizar un enfoque de línea de producto para sistemas de software-intensivo.

Programas software, tales como DSSA y STARTS atrajeron a las mayores compañías implicadas en las áreas civiles y de defensa. Estos programas han hecho posible reducir costes, mejorar la planificación e incrementar la calidad de los sistemas de software intensivo.

Por ejemplo, los proyectos de demostración de STARTS han mostrado la importancia de alterar el enfoque en el desarrollo software, pasando desde los sistemas clásicos a los sistemas de líneas de productos. Este proyecto tiene como objetivos reducir los tiempos de desarrollo y los costes por un factor de cuatro en los campos de los sistemas de control y comandos, aviónica, inteligencia electrónica, sistemas de telecomunicaciones y sistemas de simulación.

Para cuantificar los potenciales beneficios, la **Figura 2** muestra el retorno de la inversión como resultado de la adopción del concepto basado en la reutilización de la arquitectura en un enfoque de línea de producto después del desarrollo de la aplicación T44 de entrenamiento aéreo de la Navy (este experimento fue conducido por Boeing Inc., para la Marina de los EE.UU.). Para mantenerse competitivos en el dominio de negocio aplicable al enfoque de línea de producto, todos aquellos implicados requerirán estrategias similares de reducción de costes.

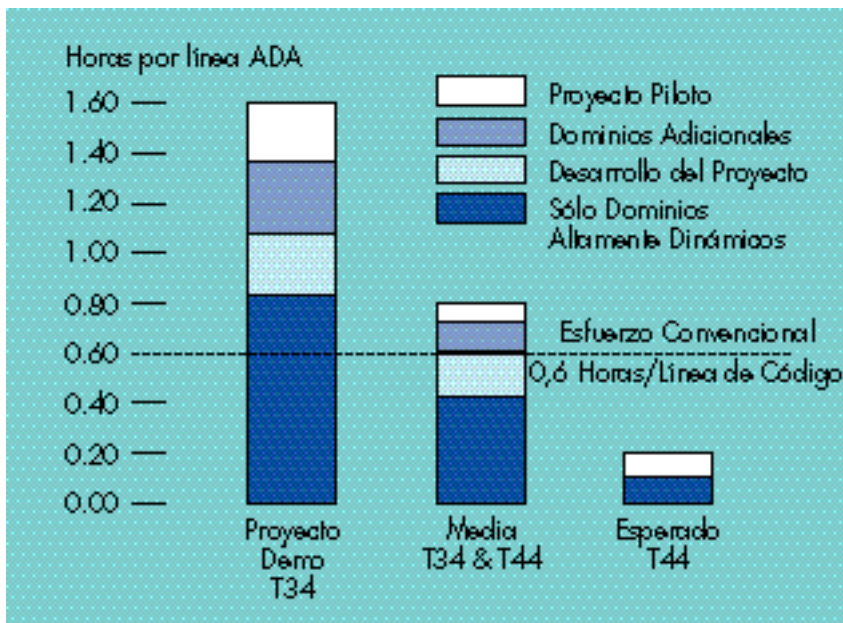


Figura 2 - Programa STARTS: Retorno de la inversión de Boeing.

■ Entorno de Trabajo para las Aplicaciones Distribuidas y en Tiempo Real

El objetivo fundamental del entorno de trabajo es el de ofrecer soporte de ejecución proporcionando componentes middleware basados en la tecnología Object Request Broker Architecture (CORBA) y en conceptos independientes de la plataforma, junto con herramientas de ingeniería que puedan transformar rápidamente componentes a nivel de fuente en componentes a nivel de ejecución y, en consecuencia, desplegarlos eficientemente sobre plataformas comerciales y propietarias. Tal plataforma está diseñándose para permitir una experiencia directa e industrial sobre la tecnología (que se encuentra estable a nivel de implantación) y que es transferible a productos que se mueven sobre arquitecturas software basadas en componentes (incluyendo la explotación de la actividad de las líneas de producto). El modelo UML se utilizará como un punto de entrada común para el concepto fundamental de entorno de trabajo y para las herramientas de diseño de arquitectura, validación de arquitectura, simulación del rendimiento y facilidades de despliegue.

Herramientas de Despliegue y de Validación de la Arquitectura

El UML es un estándar de hecho para el software de análisis y diseño basado en componentes y orientado a objetos. La definición de la arquitectura, el diseño de aplicaciones y el despliegue de los componentes requiere de esquemas y herramientas dedicadas al software distribuido en tiempo real. Una normalización del UML y una extensión de las herramientas UML para generación de código, generación de pruebas ejecutables y validación de la arquitectura (funcional y no funcional), será definida en colaboración con institutos de investigación externos. Esto incrementará de manera significativa las capacidades de automatismo del entorno de trabajo, reduciendo los ciclos de vida de desarrollo y armonizando los niveles de calidad del software.

El Entorno de Trabajo: Modelo y Tecnología

El enfoque del entorno de trabajo se beneficia del potencial del UML (y de su extensión en tiempo real el UML++) para la integración de la aplicación y de su entorno a un nivel más alto de abstracción de lo que se hace normalmente, haciendo de esta manera

que el software tenga menos posibilidades de contener errores y consecuentemente mejorando su calidad intrínseca. Se enlaza fuertemente el soporte a la ejecución (la plataforma) a las etapas de diseño y desarrollo y a las herramientas asociadas (la cadena de herramientas de ingeniería software en ordenadores de la plataforma). El objetivo es simplificar el proceso de desarrollo por medio de herramientas y conceptos de forma que se ajusten a algunas de las propiedades requeridas por las aplicaciones. El entorno de trabajo se basa automáticamente en las herramientas del UML para generar porciones de código que pueden derivarse sistemáticamente desde el formalismo UML. Además, hace un amplio uso del CORBA y su integración con el UML para resolver los aspectos relacionados con la portabilidad, distribución, apertura y heterogeneidad. También permite a la plataforma la explotación de componentes comerciales -Components Off The Shelf (COTS)- siempre que tengan las características requeridas.

El entorno de trabajo utiliza el UML para la descripción de componentes y aplicaciones. Añade extensiones al UML para ampliar su capacidad en los dominios del tiempo real y en el de la expresión de propiedades formales, permitiendo la verificación de estas propiedades en una fase posterior. Hasta donde es posible, todos los aspectos de una aplicación se especifican por medio de la utilización del entorno de trabajo UML y sus extensiones, creando por tanto un modelo de información de objetos. El objetivo final es el uso de este modelo de información para la realización de las siguientes tareas:

- Validación de la arquitectura utilizando simulación.
- Generación de código y despliegue en soporte de la ejecución.
- Asistencia en la generación de programas de pruebas y validación en línea.
- Aspectos de configuración y despliegue del sistema completo, especialmente los requisitos de ejecución en tiempo real de las aplicaciones.

El proceso de desarrollo del entorno de trabajo utiliza un conjunto de herramientas para modelar las aplicaciones desde diferentes perspectivas utilizando el UML. Algunas de estas herramientas se usan para generar código soporte que será incluido en la aplicación final conjuntamente con los servicios de la plataforma. Además del soporte al modelado y diseño, el entorno de trabajo proporciona servicios de aplicación y la plataforma soporte. Estos proporcionan programadores de aplicaciones con dominios con propiedades y servicios específicos que pueden integrarse fácilmente en el diseño de la aplicación utilizando construcciones UML ya existentes. La **Figura 3** muestra los principales componentes de la plataforma del entorno de trabajo y las relaciones en la configuración que soportan el proceso de desarrollo.

La plataforma del entorno de trabajo proporciona las funciones básicas y los servicios requeridos en las complejas aplicaciones distribuidas. Estas aplicaciones pueden clasificarse en varias clases:

- Modelo de objetos y servicios.
- Configuración.
- Fiabilidad.
- Evolución en línea.

Modelo de objetos y servicios

Una aplicación del entorno de trabajo está compuesta de un conjunto de objetos que tienen asociadas unas propiedades específicas. Los objetos pueden ser estáticos o dinámicos. Los objetos estáticos se crean durante la iniciación de la aplicación, mientras que los objetos dinámicos son dinámicamente creados y destruidos por la aplicación de acuerdo con sus necesidades.

La creación de los objetos puede realizarse transparente y remotamente para satisfacer los requisitos de capacidad definidos en la configuración de la aplicación. Los objetos pueden ser preasignados para evitar sobrecargas en la creación de objetos en periodos críticos (por ejemplo, cuando hay fuertes requisitos de tiempo real). La preasignación de objetos está determinada por la configuración de la aplicación. Puede trazarse el ciclo de vida de cada objeto del entorno de trabajo.

La transparencia de la distribución y de la arquitectura es la principal característica que CORBA aporta a las aplicaciones distribuidas, permitiendo a los programadores acceder a los objetos independientemente del lugar en donde hayan sido asignados. Específicamente, esto permite que el desarrollo de la aplicación pueda separarse del despliegue actual de los componentes en el sistema. Por tanto el sistema puede cambiar, reconfigurarse tanta veces como sea requerido, y los componentes pueden moverse de una máquina a otra sin que el código de la aplicación se vea afectado. En parte esto se debe a que la descripción de las interfaces (en Interface Description Language IDL) es independiente de la

definición del objeto (en Object Definition Language ODL). El entorno de trabajo añade el concepto de transparencia de carga equilibrada, es decir que la creación de objetos respeta los umbrales de creación y que los objetos pueden crearse en diferentes cápsulas (unidad de ejecución de la plataforma) si fuera necesario. Esta facilidad es opcional y controlable por un componente de supervisión dinámica.

Los servicios del entorno de trabajo constituyen el valor añadido del enfoque para las aplicaciones. Los actuales servicios cubren los aspectos y conceptos básicos, pero el número de servicios disponibles se incrementará y expandirá en dominios de aplicaciones específicas. Los actuales servicios del

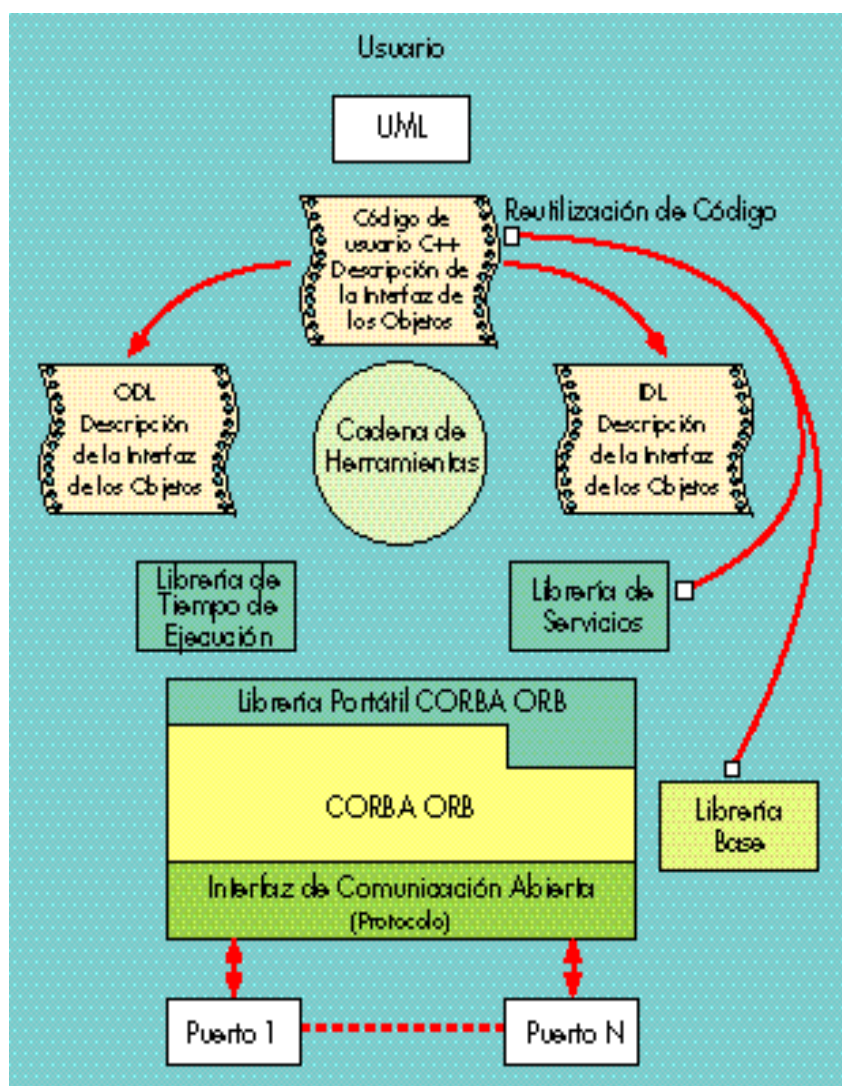


Figura 3 - Resumen del Entorno de trabajo.

ORB - Object Request Broker.

entorno de trabajo están dirigidos a las siguientes áreas:

- aplicaciones de configuración.
- aplicaciones de supervisión.
- equilibrio de carga y preasignación.
- nombramiento de objetos.
- ciclo de vida de los objetos y limpieza del sistema.
- persistencia de los objetos.
- configuración en tiempo-real.
- evolución en-línea.
- direccionamiento lógico.

Varios servicios tratan el almacenamiento de los datos y se utilizan por el servicio de tolerancia de fallos. Hay un servicio de persistencia de objetos para tratar con los aspectos relacionados con el almacenaje de los objetos y un servicio de almacenaje fiable que asegura que los datos pueden hacerse fiablemente persistentes con un rendimiento aceptable.

Configuración

La configuración es un proceso que mapea un conjunto de componentes sobre una aplicación. Trata con todos los aspectos de la aplicación de configuración: naturaleza, nombre y propiedades de los objetos de la aplicación. El proceso de configuración se lleva parcialmente a cabo utilizando herramientas UML, y parte utilizando herramientas especializadas de visualización. En la práctica, la configuración actual se realiza por medio de un fichero de configuración que reúne la información más importante desde diferentes fuentes antes de la fase de generación de código.

La estructura de la aplicación es descrita independientemente de la implantación de los componentes, lo que permite el desacoplo de los dos pasos y una reconfiguración más sencilla. Esta estructura se extrae desde una síntesis de la descripción UML de la aplicación y se integra mediante herramientas de configuración especializadas. Se identifican dos niveles de configuración. La aplicación de configuración representa el mapeo de los componentes sobre unidades de ejecución (cápsulas) de las diferentes máquinas que componen el sistema, mientras

que la plataforma de la configuración corresponde a la configuración de las máquinas físicas en el sistema, del tiempo de ejecución del entorno de trabajo y del middleware que constituye el soporte distribuido para la aplicación.

Fiabilidad

La fiabilidad del software puede incrementarse de varias formas por el entorno de trabajo durante las diferentes fases de la producción. Primeramente, durante la fase de diseño se divide el software entre componentes bien encapsulados con claras interfaces; un objetivo fundamental del enfoque en la línea de producto. Este comportamiento está fuertemente recomendado por la especificación.

En segundo lugar, se especifican las propiedades adicionales de los objetos, algunas relacionadas con propiedades de la arquitectura (tolerancia a los fallos, persistencia), otras relacionadas con la ejecución (tiempo real, por ejemplo), y otras con propiedades funcionales. Las propiedades funcionales ayudan a las herramientas automáticas a verificar que el comportamiento de los componentes está de acuerdo con las especificaciones. La verificación puede realizarse durante el desarrollo utilizando bien métodos formales (aplicados a componentes específicamente identificados) o por simulación. Las propiedades de los objetos pueden también utilizarse para la generación de muestras de prueba para validar los componentes antes de que sean ensamblados.

En tercer lugar, un porcentaje de la verificación de las propiedades puede realizarse en tiempo de ejecución para mejorar la cobertura de la fase de pruebas del desarrollo. Las características de la plataforma también incrementan la fiabilidad del entorno de trabajo reduciendo las peligrosas tareas de creación y eliminación de objetos.

Finalmente, cuando la aplicación diseñada e implantada está corriendo, un sistema de trazado permite al programador trazar la ejecución de la aplicación de una forma eficaz y no intrusiva. Esto es útil no sólo para la puesta a punto de la aplicación, si no que también es un potente mecanismo

para seguir el comportamiento de la ejecución (especialmente fallos de funcionamiento) cuando ésta se despliega. Cuando los fallos de funcionamiento del software han sido resueltos, las correcciones tienen que realizarse sobre sistemas ya instalados que se encuentran en servicio. Esto presenta la necesidad de tener la capacidad de instalar componentes y alterar la configuración del sistema en tiempo de ejecución: a esto se le llama evolución en-línea.

Evolución en línea

El dominio de aplicación del entorno de trabajo está compuesto de sistemas que ya se encuentran en la base instalada. Estos sistemas no pueden ser retirados para mejorar el software, bien por que sus requisitos de disponibilidad son incompatibles con tales retrasos o bien por que son totalmente inaccesibles (como el caso de sistemas de satélites). Otra justificación para la evolución en-línea es la de que los requisitos y las necesidades pueden variar frecuentemente, por lo que las mejoras del software ya no son un evento excepcional y el coste y las consecuencias producidas sobre los servicios pueden evitarse más fácilmente.

Por estas razones, el entorno de trabajo incluye capacidades para la evolución en línea. Se pueden cargar los módulos software y realizarse acciones de reconfiguración. Está integrado con sistema de tolerancia de fallos, y es posible arrancar réplicas de los objetos dinámicamente; uno puede arrancar réplicas de una nueva versión y, posteriormente, destruir la versión antigua utilizando la herramienta de supervisión. Los mecanismos de tolerancia de fallos aseguran que los cambios son transparentes a los clientes de los componentes reemplazados. Por ejemplo, en una operación normal siempre que el entorno de trabajo detecta que ha fallado un objeto activo, otro objeto réplica del primero es elegido como el nuevo activo y se recoge el último estado estable almacenado perteneciente al objeto previamente activo. Esta clase de mecanismos pueden cubrir algunos objetivos de tolerancia de fallos.

Visión de la tecnología del entorno de trabajo

El entorno de trabajo confía en CORBA por razones de transporte y heterogeneidad. Existen actualmente implantaciones para ORBACUS y VisiBroker en UNIX y Windows NT. Estas aplicaciones hacen una extensa utilización de la arquitectura CORBA e imponen requisitos y nuevas especificaciones sobre una subsiguiente evolución de CORBA. Esto ha llevado a la implicación directa en el proceso de estandarización de CORBA de CORBA mínimo, el tiempo real de CORBA y la tolerancia de fallos de CORBA.

■ Conclusiones

Este artículo trata sobre un objetivo fundamental del Alcatel/Thomson Common Laboratory establecido en Septiembre de 1998. Un objetivo de este

Laboratorio es el de demostrar la viabilidad de las arquitecturas de línea de producto y su rápida transferencia a entornos de trabajo distribuidos, en tiempo real, y de alta disponibilidad capaces de soportar las aplicaciones típicas que se encuentran en los catálogos de productos de ambas compañías. Los tipos de entornos de trabajo que se están investigando promueven el importante papel de la arquitectura, los criterios de simulación dirigido por los requisitos, las restricciones de pruebas y el ciclo de vida de los componentes en aras de conseguir la disponibilidad de un cerrado, pero flexible, enlace entre línea de producto, la ingeniería basada en los componentes y el entorno de trabajo que los soporta.

Los objetivos centrales del Common Laboratory son los de proporcionar soluciones industriales de calidad para explotaciones rentables de técnicas eficientes de producciones software ade-

cuadas para líneas de productos, y el rápido despliegue de este eficiente software en un entorno de trabajo construido para ajustarse de la mejor forma posible a los mecanismos y componentes específicos y comerciales.

Gary Donnan es responsable del proyecto Architecture, Components and Frameworks en el Alcatel Corporate Research Center de Marcoussis, Francia

Jean Jourdan es responsable del grupo Architecture and System Technology en Thomson - CSF Central Research Laboratory, en Courbeville, Francia.

TECNOLOGÍA DE AGENTES Y SUS APLICACIONES

F. CARREZ

Los agentes son una nueva y prometedora tecnología que revolucionará el mundo de las TI.

■ Introducción

La tecnología de agentes es un prometedora y progresivamente creciente nueva tecnología [1 a 6] en el campo de la tecnología de la información (TI). Reúne varias tecnologías, como las aplicaciones de código de móviles y totalmente distribuidas, que tienen el objetivo común de actuar autónomamente en nombre de otra entidad (por ejemplo, una persona). Existen diferentes razones para la proliferación de agentes y de las tecnologías asociadas. En primer lugar, la generalización de Internet da la oportunidad y proporciona los medios técnicos para crear aplicaciones y servicios totalmente distribuidos (por ejemplo, el comercio electrónico, la obtención de información). Segundo, la programación y la conceptualización basadas en agentes es una metáfora para tratar otros muchos problemas. Tercero, los seres humanos siempre han estado interesados en la delegación de tareas (robots) y en recibir ayuda (Personal Digital Assistant, secretarías electrónicas). Finalmente, muchas técnicas de Inteligencia Artificial Distribuida (DAI) ya han madurado y se pueden utilizar para desarrollar servicios más inteligentes y sofisticados.

Este artículo muestra las diferentes tecnologías y filosofías de agentes, particularmente los agentes móviles e inteligentes. Una parte importante del artículo se dedica a las aplicaciones basadas en agentes existentes y posibles, en particular en el campo de las telecomunicaciones.

■ ¿Qué es un Agente?

¿Qué es exactamente un agente? Incluso esta pregunta directa no tiene una respuesta sencilla que pueda ser aceptada por toda la comunidad de agentes. Casi todos los investigadores tienen su propia definición de un agente, en gran parte debido a que los agentes pueden diferir básicamente según el campo donde se apliquen.

Se deben tener en cuenta algunos aspectos filosóficos no menores cuando se definen los agentes. ¿Qué es lo que realmente hace un agente? ¿Actúan sólo los agentes en nuestro nombre? ¿Debemos confiar en ellos? ¿Estamos preparados para darles nuestro PIN (número de identificación personal) para realizar una tarea en la red? ¿Queremos que sean totalmente autónomos, dirigidos, o sólo esclavos? ¿Queremos que aprendan de nosotros y de su entorno?

Las respuestas a estas sencillas preguntas llevan naturalmente a diferentes definiciones de lo que es un agente. No obstante, la mayoría de las definiciones están de acuerdo en que una agente es un programa de software que actúa en nombre de otra entidad (donde una "entidad" puede ser una persona u otro programa de software) y que posee cierta "autonomía". Además de estas dos importantes características de delegación y autonomía, un agente puede disfrutar de otras propiedades o facilidades tales como racionalidad, proactividad, reactividad, adaptabilidad, planificación,

capacidad social, razonamiento, confianza, modelado de usuario o entorno, base de conocimientos y movilidad. Esta claro que un agente no tiene que tener todas estas características, pero al depender de sus características el agente caerá en una de las categorías que se describe más adelante.

La mayoría de los términos anteriormente citados provienen del campo de la DAI, que puede ser considerado como el principal origen de la así denominada tecnología "inteligente" de agentes. Sin embargo, la DAI no es la única fuente de la tecnología de agentes. Otra clase de agente, el conocido como Agente Móvil, ha surgido últimamente: se puede considerar como una etapa por encima de la programación orientada a objetos clásica. La tecnología de agentes móviles ofrece un atractivo nuevo paradigma en el que la movilidad de código ocupa un lugar importante. Estos agentes son en general muy diferentes de los agentes inteligentes.

Para clarificar los diferentes tipos de tecnologías de agentes y su propósito, puede ser de ayuda hacer una taxonomía (categorización) de los agentes. En un principio, se puede hacer una distinción entre agentes móviles y agentes inteligentes, sobre todo porque provienen de diferentes campos y difieren en las tecnologías y técnicas implicadas. Otra razón natural para hacer esta distinción es que los agentes inteligentes están aún, en general, en fase de investigación, mientras que ya se

dispone de plataformas de agentes móviles y se utilizan mucho más en programas de investigación industrial. Pero cuando se lee la literatura existente está claro que este sencillo enfoque no es ni satisfactorio ni suficiente.

■ Una Taxonomía de los Agentes

Se puede hacer una taxonomía más detallada de los agentes basándose en los siguientes criterios: inteligencia, autonomía, racionalidad, movilidad, comunicación y agencia.

Vamos a empezar introduciendo estos términos:

- *Inteligencia*: es bastante difícil de definir con precisión. Cuando se aplica a los agentes, se podría definir la inteligencia como algunas capacidades de alto nivel, como la capacidad del agente para razonar sobre su conocimiento, aprender e inferir nuevos conocimientos, y planificar las acciones consecuentes para alcanzar sus objetivos. En este caso, la inteligencia tiene el mismo sentido que el término “inteligencia artificial”.
- *Autonomía*: Existen diferentes formas de entender el término “autonomía”. En el sentido más común (absoluto), la autonomía se refiere a lo impredecible del comportamiento del agente: cuanto más impredecible sea, más autonomía aparecerá. Sin embargo, la autonomía absoluta no se aplica muy bien a los agentes ya que, de acuerdo con nuestra mínima definición, un agente tiene que servir para algunos propósitos que se la limitan. Un tipo más útil de autonomía es la “autonomía social”, que se refiere a la autonomía del agente cuando se pone en una sociedad de agentes. No obstante, unas relaciones más sociales, como la cooperación, coordinación y compromiso, limitan al agente. Finalmente, la ejecución de la autonomía se refiere a la capacidad del agente para elegir y llevar a cabo la acción adecuada en el momento oportuno (técnicas de

planificación). De hecho, al diseñar un agente, se tiene que encontrar un fino equilibrio entre autonomía y servidumbre. En lo que queda de artículo, el mayor grado de autonomía se alcanzará por los “agentes de objetivos dirigidos”. Un nivel inferior puede consistir en un agente que represente a un usuario, como el bien conocido PDA (Asistente Personal Digital).

- *Racionalidad*: Se aplica esencialmente a los agentes autónomos e inteligentes. Los agentes racionales tienden a elegir y realizar acciones que maximicen su esperada utilidad en función de sus percepciones y de nuestro propio conocimiento. La racionalidad también implica que la acción elegida sea consistente con los deseos y creencias del agente.
- *Movilidad*: Es la capacidad que tiene un agente para iniciar su ejecución en cualquier sitio, e irse a otra posición (llevándose datos y códigos) donde continuar su ejecución. De hecho, la “movilidad” tiene dividida a toda la comunidad de agentes en dos escuelas: la de los que piensan que la movilidad no es un aspecto esencial (la comunidad de multiagentes DAI, esencialmente académicos) y la de los que claman que los agentes móviles son el futuro de los agentes (comunidad de programadores orientados a objetos). Como sucede en todo, la verdad está en un punto intermedio. Si parece claro que algo que puede ser hecho con los agentes móviles, puede a priori hacerse con técnicas de programación convencionales, se trata de un hecho que la programación basada en agentes móviles trae un nuevo paradigma que realza la flexibilidad y eficacia del diseño y ejecución de las aplicaciones distribuidas, y reduce el ancho de banda de red requerido (ver los apartados “Agentes Móviles” y “Aplicaciones” para más detalles).
- *Comunicación*: Es un aspecto crucial para un agente, que, por definición, tiene que comunicarse durante su vida bien con la entidad en nombre de la que actúa, bien con los otros agentes con los cuales necesi-

ta colaborar. El nivel de interacción/comunicación depende en gran medida del nivel de conocimiento del agente, que se mueve entre “datos”, “información” y “conocimiento”. Los datos se corresponden con el nivel más básico, que puede ser el contenido de una variable, un archivo, etc. La información está mucho más estructurada y puede consistir en la descripción de parte de un equipo de telecomunicación, un documento XML (eXtensible Markup Language) [XML] como un formulario de pedido (en aplicaciones de agentes de comercio electrónico), o un correo electrónico (con datos asociados). El conocimiento consiste en información estructurada y reglas lógicas. Se pueden asociar con cada nivel de información los lenguajes de comunicación y protocolos adecuados. Por ejemplo, un agente puede usar FTP (Protocolo de Transferencia de Ficheros) para enviar un archivo, o llamar al método de otro agente para obtener o transmitir información. El HTTP (HyperText Transfer Protocol) se puede utilizar para transmitir descripciones XML. Finalmente, lenguajes basados en la Speech Acts Theory [7, 8], como KQML/KIF (Knowledge Query and Manipulation Language/Knowledge Interchange Format) [9, 10, KQML, KIF], el lenguaje de comunicación de agentes FIPA (Formation for Intelligent Physical Agents) pueden ser usados para intercambiar información.

Una sencilla taxonomía de los agentes basada en agentes móviles, guiones móviles y agentes inteligentes se presenta más adelante. Los guiones móviles y los agentes móviles, ampliamente usados dentro de la industria, se consideran en los dos apartados siguientes. Los agentes inteligentes (o multiagentes), que están en una avanzada fase de investigación, se tratan al final; están abiertos a prometedoras nuevas perspectivas de futuro.

Agentes Móviles

Los agentes móviles [1, 3, 11, 12] son

programas software (generalmente pequeños) que tienen la capacidad de iniciar su ejecución en un punto (ordenador), suspender la ejecución, ir a otro punto de la red, continuar la ejecución, etc.

Si echamos una ojeada a la definición “mínima”, la autonomía de un agente móvil consiste esencialmente en “autonomía espacial”. Esto quiere decir que el agente es “libre” de ir a otro punto de la red; se puede ejecutar sin necesidad de un enlace o de ser controlado desde la posición original. En general, los agentes móviles no disfrutan de la propiedad de “autonomía absoluta” (ver apartado de “Agente Inteligente”).

Llamadas a procedimientos remotos frente a los agentes

Los agentes móviles se pueden contemplar como una solución a los problemas surgidos en la computación distribuida en un entorno heterogéneo. Son, en particular, una alternativa natural [12] del clásico paradigma

“cliente/servidor”, como se muestra en la **Figura 1**.

Consideremos una tarea particular de un cliente que necesita de algunos servicios remotos (N) para que se realice. En el caso del modelo cliente/servidor, el cliente tiene que acceder a N servicios remotos utilizando una secuencia de RPCs (llamadas a procedimientos remotos), consistiendo cada una de ellas en una petición y una respuesta a uno de los servicios necesarios. Consecuentemente, la red tiene que transportar $2N$ mensajes.

La tecnología de agentes móviles propone una interesante alternativa a este paradigma. En este caso, el cliente consiste en un agente que se mueve a través de la red directamente hasta la posición del servidor, en donde accede localmente a los servicios requeridos. Todas las interacciones entre el cliente y los servicios se hacen localmente; el agente puede volver a su posición inicial tan pronto finalice su tarea.

Existen numerosas ventajas al utili-

zar la tecnología de agentes en este caso. Primero, el cliente y el servidor no necesitan acordar un protocolo de comunicaciones. Es suficiente para el agente cliente conocer la interfaz suministrada por el servidor. Segundo, sólo la información “interesante” transita por la red (es decir, el propio agente lleva los resultados de su tarea). Consecuentemente, la utilización de la tecnología de agentes móviles requiere de una menor anchura de banda de red. Tercero, no se necesita mantener un enlace físico entre el cliente y el servicio durante la operación, sólo se requiere durante el tránsito del agente. Finalmente, los agentes traen mucha flexibilidad en términos de integración y desarrollo software.

Conceptos subyacentes

Para permitir que las plataformas de agentes móviles interoperen, se han introducido algunos conceptos generales, que están siendo normalizados por el OMG (Grupo de Gestión de Objetos). Además, el OMG define como se pue-

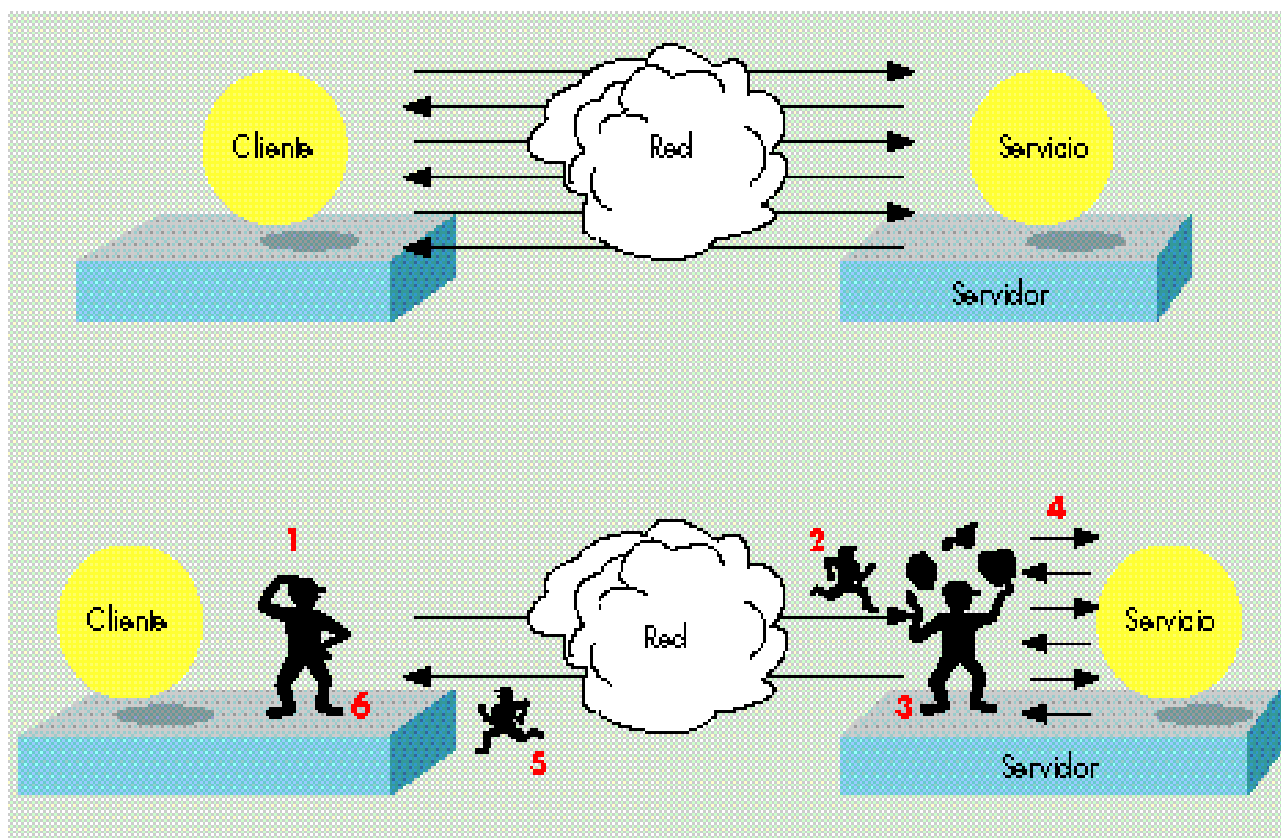


Figura 1 - Llamadas a procedimientos remotos frente a los agentes.

den soportar la movilidad de agentes y la invocación de sus plataformas de agentes subyacentes encima del CORBA (Arquitectura de Negociación de Petición de Objetos Comunes) [OMG]. Este apartado proporciona algo de información sobre estos conceptos. Sin embargo, el lector puede referirse, para más detalles, al MASIF (Mobile Agent Systems Interoperability Facility) del OMG [13].

La **Figura 2** muestra la mayoría de los conceptos definidos por la recomendación MASIF. Hay que resaltar que estos conceptos ya existen en la primera plataforma de agentes TELES-CRIPT [General Magic].

Un sistema de agentes es una plataforma que tiene la capacidad de crear, nombrar, ejecutar, despachar y terminar agentes. Durante su recorrido, un agente pasa por entornos de ejecución llamados "sitios". Un sitio es un contexto dentro del sistema de agentes en el cual el agente se puede ejecutar.

Adicionalmente, un sitio proporciona servicios de seguridad tales como el control de acceso, a los recursos del sistema de agentes. El concepto de "región" se utiliza para agrupar algunos sitios que están bajo la misma autoridad. La transferencia del agente se realiza por serialización y de-serialización del código del agente, es decir, la transformación de un objeto en un flujo, y viceversa. Finalmente, la mayoría de las facilidades de alto nivel proporcionadas por el sistema de agentes y los sitios se basan en servicios CORBA.

La movilidad de código saca a relucir, e incluso empeora, un problema que ya existía con las aplicaciones clásicas de software: la seguridad. La conexión entre los agentes móviles y virus es fácil de establecer. En particular, es de importancia primordial el ser capaz de autenticar un agente entrante y la entidad en nombre de la cual actúa, para controlar el acceso

del agente a los recursos físicos visitados, la integridad del agente y garantizar otras propiedades, como el no-repudio (el agente no puede negar que tiene algo que hacer) y la confidencialidad. Otros problemas surgen en el caso de los agentes de saltos múltiples (más de dos saltos), para los que aún no existe una solución. Hay que hacer notar que la mayoría de las soluciones técnicas existentes para agentes móviles se basan en soluciones de seguridad creadas o en CORBA o en Java. En el caso de applets Java, la seguridad generalmente se asegura por el browser que solo acepta applets firmados o un canal seguro para comunicarse con el sitio madre. Un ejemplo es el SSL (Secure Sockets Layer).

Plataformas de agentes móviles

Aunque el mercado de las plataformas de agentes está cambiando rápidamente, ya se dispone de varias plataformas de agentes móviles fiables para utilizar

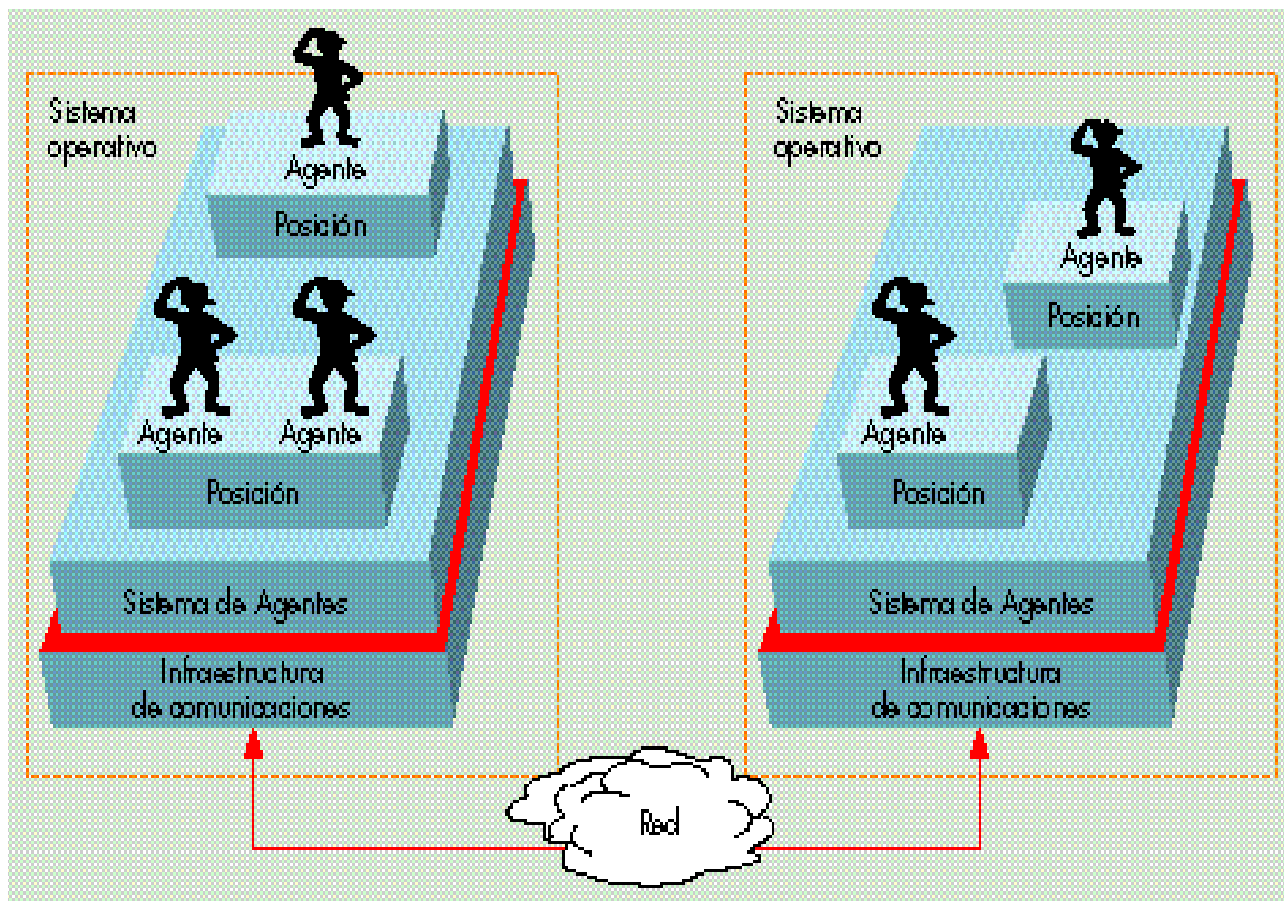


Figura 2 - Conceptos de agentes móviles como se definen en la recomendación MASIF.

en un contexto industrial: Aglets Work-Bench de IBM Corp., Odyssey de General Magic [Odyssey] (el sucesor de Telescript), Grasshopper [14] (la única plataforma de acuerdo con MASIF [IKV++]) y Voyager [15, ObjectSpace]. Todas estas plataformas son 100% Java, por lo que las aplicaciones basadas en agentes heredan la independencia de las plataformas Java. Otra consecuencia es que se necesita el JDK (kit de desarrollo de Java) para ejecutar el sistema de agentes y los agentes. La utilización de plataformas de agentes basadas en Java es un aspecto clave cuando se trata de entornos heterogéneos multivendedor.

Guiones Móviles

Un guión móvil es un programa software, escrito en un programa de guiones, que se puede ejecutar remotamente. Existen varios tipos de guiones móviles, el más conocido es el applet Java que se ejecuta en las sesiones de Netscape o Explorer en Internet. Con más precisión, un "applet" es un programa Java (generalmente pequeño) que se incluye en una página HTML (HyperText Markup Language). Cuando se marca la página HTML en el browser, se cargan automáticamente desde el sitio remoto los applets referenciados, utilizando el protocolo HTTP, y se ejecutan en el browser.

Otros tipos de guiones móviles, los llamados agentes XML, consisten en guiones que están incluidos dentro de una página XML que se pueden cargar y ejecutar por cualquier plataforma que permita XML. Se espera que el XML sea el futuro estándar para la Web, sustituyendo al actual estándar HTML y permitiendo una información más estructurada (ver [XML]).

En esta fase, se puede hacer una distinción entre guiones móviles y agentes móviles. Un guión móvil se ejecuta remotamente, pero no tiene un "billete de vuelta" a su sitio original; además, no es autónomo.

El resultado del "applet" (por ejemplo, información de usuario, información de seguridad como un código PIN) se puede devolver, sólo utilizando un mecanismo de socket. Por el

contrario, un agente móvil puede desplazarse en uno o incluso múltiples saltos, y (posiblemente) volver a su posición original.

■ Aplicaciones Avanzadas de Telecomunicaciones

La tecnología de agentes se puede aplicar en muchos aspectos de las telecomunicaciones, tales como la gestión de red, las redes inteligentes y los servicios on-line [16].

Áreas Potenciales

Gestión de red

La mayoría de las áreas de gestión de red de configuración de faltas, contabilidad, prestaciones y seguridad (FCAPS) se pueden cubrir por aplicaciones basadas en agentes. Así, por ejemplo, se pueden usar para propagar una configuración por una RGT (Red de Gestión de Telecomunicaciones), para realizar trabajos de auditoría, para correlacionar alarmas distribuidas, para verificar la consistencia de una base de datos distribuida, para llevar a cabo diagnósticos de fallos de servicio y de sistema, para planificar acciones curativas, etc.

Gestión de servicios

Los sistemas multiagente se pueden usar para proveer dinámicamente servicios de telecomunicaciones, tales como la videoconferencia en una red privada virtual (ver la aplicación "Gestión y Aprovechamiento de Redes" de FIPA[FIPA/TC7] y el proyecto FACTS ACTS).

MIAMI, otro proyecto ACTS [MIAMI], tiene por objetivo proporcionar una plataforma que permita crear y gestionar "Empresas Virtuales". Una empresa virtual permite que sus miembros se proporcionen servicios sobre recursos de telecomunicaciones compartidos, según un "contrato". La plataforma, en particular, oculta a los compañeros la complejidad y variedad de los recursos usados negociando con los diferentes proveedores de servicios de telecomunicaciones sobre la base de la mejor relación calidad/precio.

Red Inteligente

En la actualidad se consideran a las redes inteligentes (RI) como la solución tanto para el rápido aprovisionamiento como para la creación de servicios de telecomunicación basados en conmutación. La arquitectura de RI separa claramente las partes de control y conmutación del servicio. La parte de conmutación se sitúa en el SSP (Punto de Conmutación de Servicios) dentro del conmutador, mientras que los datos y la lógica del servicio lo están en plataformas hardware especializadas, conocidas como SCP (Puntos de Control de Servicios), fuera del conmutador. Se accede a la lógica del servicio usando llamadas a procedimientos remotos. Como resultado de esta arquitectura, el SCP puede ser un cuello de botella que disminuya la eficacia de la distribución de servicios de RI.

La tecnología de agentes móviles se puede usar para mejorar la arquitectura de RI al introducir la noción de servicio móvil de RI, llevando el servicio (inteligencia) directamente al cliente. Los servicios de RI basados en agentes son también una forma más flexible y potente de desplegar nuevos servicios [17].

Aplicaciones de Telecomunicaciones en Alcatel

Alcatel ha estado estudiando todas las tecnologías de agentes (agentes móviles e inteligentes, agentes XML). Aquí se detallan dos de estos estudios.

Atención Inteligente a Clientes

La arquitectura GSM (Global System for Mobile Communication) es enorme, compleja y abarca una amplia variedad de componentes que, si fallan, podrían causar problemas desde el punto de vista del usuario: llamadas perdidas, pobre calidad de voz, indisponibilidad de red, etc. Además, eventos externos, como el mal tiempo o una anormal gran cantidad de personas en una zona particular (causada por una congestión de tráfico, una feria, un concierto, etc.), pueden ser la causa indirecta de un problema. Por ejemplo, demasiadas llamadas GSM

en un momento dado en un sitio pueden originar la congestión de celdas (a pesar del amplio despliegue de equipo de radio) y evitar que se puedan hacer nuevas llamadas.

Cuando nos enfrentamos a un problema técnico, como la incapacidad de hacer una llamada, un usuario no suele tener otra alternativa que volver a intentarlo de forma repetida. Lo que un usuario espera recibir razonablemente de su operador móvil se reduce generalmente a su forma más sencilla: el usuario llama a un servicio de hot-line gratis, cuenta su problema, y se le indica que debe hacer. Una posible consecuencia es que el usuario cambie a otro operador, esperando un mejor o más eficaz servicio.

El ICC (Atención Inteligente a Clientes) busca mejorar la respuesta de los operadores a sus clientes, como se muestra en la **Figura 3**, donde un usuario consulta un problema técnico, lo describe a posteriori mediante un

applet en su browser favorito, y recibe (casi en tiempo real) una explicación y un posible remedio al problema.

El principal objetivo del ICC es explicar la causa de un problema al cliente una forma inteligible. La **Figura 3** muestra los componentes involucrados en el proceso de “búsqueda de una explicación”. Las fuentes de información de la Gestión de Red (NM) son:

- Todos los datos (alarmas, contadores, estados, indicadores de calidad de servicio, etc.) proporcionados por el Centro de Operaciones y Gestión (OMC/R) responsable de la gestión de la parte de radio de la red.
- Los datos (por ejemplo, qué celda gestiona las llamadas en una posición particular; mapa de los niveles de recepción) recogidos durante la fase RNP (Planificación de Redes de Radio).

Cierta información que está fuera del alcance de la gestión red, como la información meteorológica, la del tráfico rodado y la de los eventos sociales (mediante páginas HTML), también se puede usar para consolidar el diagnóstico realizado sobre la base de la información de la gestión de red. La arquitectura del ICC se basa en gran medida en la utilización de agentes móviles e inteligentes.

Los agentes móviles (fácilmente reconocibles) son los responsables de alcanzar y calcular localmente los datos en los nodos web (por ejemplo, información meteorológica y de tráfico) o en mapas enormes en el servidor RNP. Los agentes inteligentes analizan los datos de gestión de red directamente en su fuente: el OMC/R. En este caso, el agente inteligente consiste en un sistema móvil experto que aplica reglas “expertas” a los datos disponibles para llevar a cabo un diagnóstico preliminar.

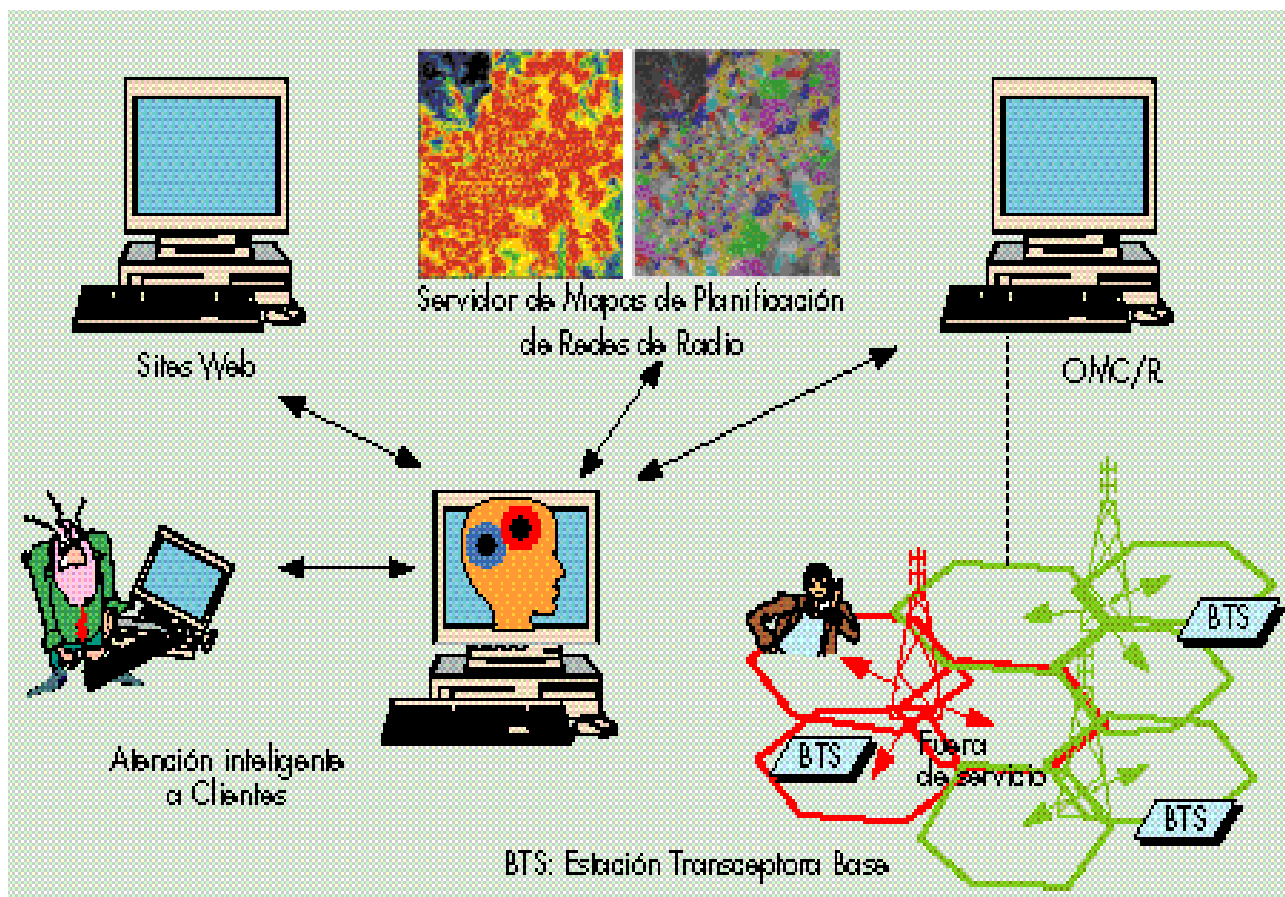


Figura 3 - Componentes involucrados en la atención inteligente a los clientes.

Los posibles problemas (por ejemplo, llamadas perdidas o indisponibilidad de la red de radio) se encuentran como resultado de la congestión de celdas, problemas de transferencia, un defecto del equipo o un agujero en el área de cobertura. Este diagnóstico inicial se consolida usando la información de gestión de red adicional.

Finalmente, se presenta la "explicación" al cliente en una forma apropiada como: "Nuestras investigaciones han mostrado que esta área aún no está, desafortunadamente, cubierta, por lo que le pedimos excusas. Sin embargo, usted puede establecer su llamada 500 metros al este de su posición actual", o "Nuestras investigaciones han mostrado que, debido a congestiones de tráfico, usted se encuentra en una celda congestionada. Su petición será pronto tenida en cuenta." Desde un punto de vista técnico, las ventajas son una reducción en el uso del ancho de banda al eliminar la transmisión de datos (por ejemplo, alarmas, mapas), una mejor flexibilidad y una independencia de plataformas, y un menor coste de integración.

Negociación de los servicios

El proyecto de plataforma multiagentes para servicios on-line (MAPOS) fue llevado a cabo por el centro de investigación corporativo de Alcatel situado en Amberes, en 1997. Su objetivo fue desarrollar una arquitectura para la negociación de servicios, en el que todas las partes involucradas se representan por un agente. En esta arquitectura (**Figura 4**) un agente, llamado broker (negociador), representa un cliente que está potencialmente interesado en un servicio particular. Su papel es ocultar al usuario la complejidad de la red (en términos de ofertas disponibles), y negociar con los diferentes proveedores que le pertenecen. El broker posee algo de información sobre el cliente, como sus preferencias, presupuesto y perfil. Esta arquitectura asume que muchos proveedores de servicios son capaces de proporcionar el servicio pedido al cliente.

Cuando el cliente pide un servicio particular, el broker negociará el servicio con los diferentes proveedores para

obtener la mejor relación calidad/precio basada en los datos personales del cliente.

Ya que un proveedor de servicios puede depender de muchos componentes de servicio, el precio resultante será una combinación de los componentes del servicio constituyentes. La tarificación del servicio se calcula dinámicamente de acuerdo con la calidad de servicio ofrecida. El proceso de negociación se obtiene usando KQML/XML como lenguaje de comunicación entre el agente broker y los agentes proveedores de servicios. El prototipo se ha escrito en Java.

■ Otras Aplicaciones Basadas en Agentes

Otras aplicaciones fuera del puro campo de las telecomunicaciones ya se están beneficiando de la tecnología de agentes. Algunas de ellas serán muy importantes en el futuro y se utilizarán

cada vez más en la vida cotidiana. Este es ciertamente el caso del comercio electrónico y de los PDAs.

Comercio electrónico

El comercio electrónico ya es una realidad. Cada vez más empresas se conectan a Internet para comprar o vender productos o bienes. Sin embargo, el proceso no está todavía automatizado y requiere la intervención humana. Con la aparición de las tecnologías de agentes, las cosas serán mucho más automatizadas. Los agentes pueden viajar por la red en nombre del comprador para encontrar el mejor trato. Si todas las partes involucradas están de acuerdo (incluso la red) es incluso posible para el agente llevar el número de la tarjeta de crédito del comprador directamente al vendedor. Sin embargo, esto requiere un alto nivel de confidencialidad que sólo se garantizaría usando técnicas criptográficas.

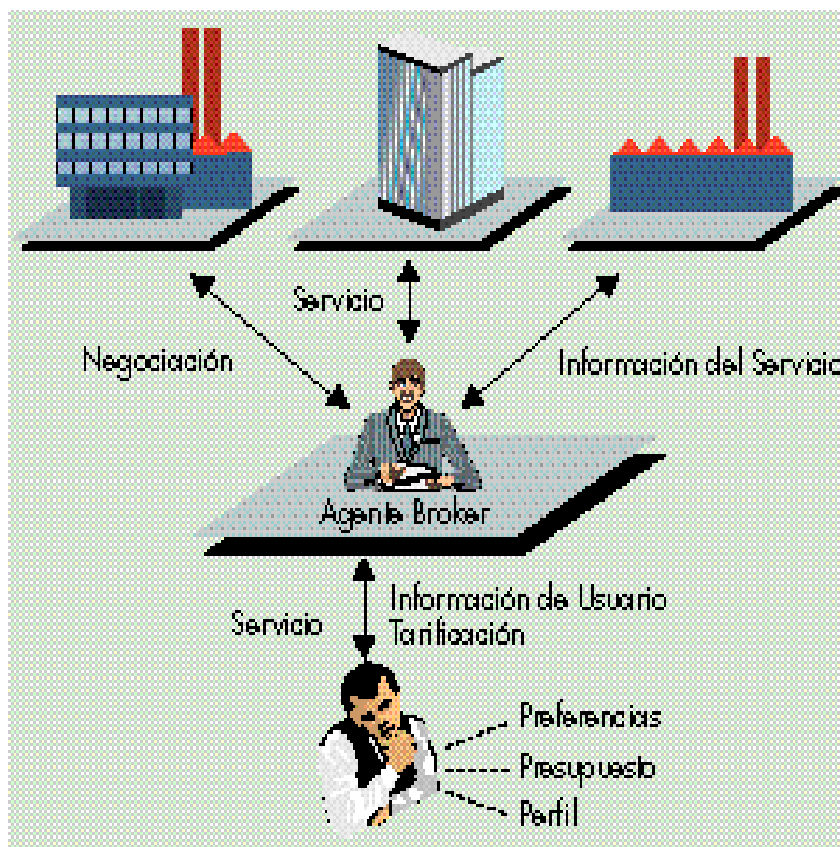


Figura 4 - Arquitectura MAPOS.

PDA

El PDA (Personal Digital Assistant) es la clase más amplia de la aplicación de agentes. Su objetivo es simplificar la vida cotidiana de las personas. Diferentes clases de aplicaciones se pueden asimilar a los PDAs.

Asistente de viajes personal

Un asistente de viajes personal (PTA) organiza viajes en nombre del usuario. Esta clase de agentes se equipa con un modelo de usuario, ya que se necesita información sobre el usuario para el proceso de reserva. El viaje se organiza completamente, es decir, el agente del usuario:

- contacta con una compañía ferroviaria para conseguir un billete, si el usuario necesita
- coger un tren desde su lugar de origen hasta el aeropuerto;
- contacta con las líneas aéreas para hacer una reserva;
- encarga un taxi o alquila un coche en el lugar de destino;
- y, finalmente, reserva una habitación en un hotel.

Para realizar todo el proceso de forma correcta, el agente debe saber todo sobre su cliente: si fuma o no, primera clase o turista, presupuesto, y todo tipo de información sobre líneas aéreas, hoteles, etc. El PTA también se encarga de negociar el mejor servicio al mejor precio. Ya existen dichas aplicaciones y están siendo desplegadas. Una de ellas, conocida como MOTIV (de Siemens, Benz y otras empresas) ya funciona en tres estados (landers) de Alemania; también proporciona guía de coches (usando el GPS, sistema de posicionamiento global) y un buscador de aparcamiento.

Hay que hacer notar que el modelado del usuario es un aspecto muy importante y complejo de las aplicaciones basadas en agentes, en particular, las interfaces inteligentes (o adaptativas) y el asistente personal. Este aspecto debería ser tratado por el FIPA durante 1999.

Secretarías electrónicas

Esta clase de agentes realiza tareas

básicas, como la organización de reuniones. Teniendo en cuenta las preferencias y peticiones de sus propietarios, el agente contacta los agentes de los convocados para determinar una hora que satisfaga a todos los participantes. El agente también puede negociar dónde se va a realizar la reunión, o tiene en cuenta otros requisitos como buscar el mínimo coste de la reunión o el número de personas que tienen que viajar.

Otras aplicaciones incluyen el filtrado adaptativo automático o el reencaminamiento de los correos electrónicos según la agenda, disponibilidad o carga de trabajo del propietario.

Obtención de información

La enorme cantidad de información disponible en la World Wide Web y la necesidad de usar un URL (Localizador Uniforme de Recursos) para localizarla, están haciendo cada vez más difícil el acceder rápidamente al servicio o información necesarios. Esto es verdad en particular para la gente que accede a Internet sin una enseñanza previa. En realidad, existe una falta de amigabilidad cuando se accede a Internet sin un objetivo particular en mente.

Los motores de búsqueda y los agentes que obtienen información se han creado para llenar este vacío.

Un agente que obtiene información consiste en un agente que viaja por la red buscando la información requerida, teniendo en cuenta el criterio de búsqueda. Por ejemplo, se puede pedir a un agente buscar en la web una librería que pueda suministrarle cierto libro por menos de 20 dólares (incluyendo envío) en 48 horas.

Esta clase de aplicación es esencial para la práctica eficaz y amigable del comercio electrónico.

Control de Plataforma

Los sistemas multiagente tienen muchas aplicaciones en el campo del "control". Por ejemplo, OASIS es una prueba de campo de un sistema de control de tráfico aéreo basado en multiagente BDI en el aeropuerto de Sidney. Teniendo en cuenta los aviones pendientes de aterrizar y diferentes factores externos, el sistema planifica el orden de las lle-

gadas; cualquier retraso se negocia directamente con los aviones (representados por agentes).

Se ha desarrollado un sistema multiagente para controlar un acelerador de partículas de elevada energía como parte del proyecto europeo ARCHON; se está usando en el laboratorio del CERN. Este sistema se basa en el GRATE (Generic Rules and Agent Model Testbed Environment) desarrollado durante el proyecto ARCHON [18]. Otra prueba de campo de un sistema para gestionar el transporte eléctrico también se basa en la plataforma ARCHON.

Otros Campos

Muchos otros campos están comenzando a aprovecharse de la tecnología de agentes, en particular: la vida artificial [19, 20], la robótica, los bancos, las finanzas, los flujos de trabajo, la fabricación, la educación, la salud, y el ocio [21].

■ Agentes Inteligentes

Esta clase de agentes proviene del campo de la DAI. Los agentes inteligentes disfrutan de facilidades como la planificación, el aprendizaje, el razonamiento y la cooperación. Un sistema multiagentes se refiere a un grupo de agentes inteligentes, pertenecientes a una así llamada sociedad de agentes, que trabajan y cooperan para alcanzar un objetivo común.

Existen dos escuelas de pensamiento que se refieren a los agentes inteligentes: la deliberativa y la reactiva. Esta última escuela clama porque los agentes que participan en un sistema inteligente de multiagentes globalmente no tengan que ser inteligentes de manera individual [5]. Por ello, los agentes reactivos son agentes cooperativos que poseen mecanismos que les permiten reaccionar frente a los eventos entrantes, sin tener en cuenta ni mecanismos de planificación ni objetivos explícitos.

En la primera, cada agente deliberativo posee su propia base de conocimiento que contiene toda la informa-

ción y experiencia necesaria para realizar su tarea y gestionar todas sus interacciones con los otros agentes y su entorno. Los agentes deliberativos son también sistemas de intención en los que poseen objetivos y planes para conseguir sus fines. Al cooperar con otros agentes, deben negociar y resolver los posibles conflictos. En general, los agentes deliberativos se diseñan utilizando una arquitectura BDI (Belief, Desire, Intent) que permite modelar los agentes y diseñarlos de acuerdo a su conocimiento, objetivos y planes.

El problema de comunicación es crucial dentro de un sistema multiagente, ya que la teoría de telecomunicaciones subyacente tiene que capturar toda la sutileza de los mensajes intercambiados. En particular, tiene que capturar totalmente las intenciones de los agentes. Las teorías de comunicaciones a la moda usadas en los sistemas multiagente consisten generalmente en adaptaciones de la bien conocida teoría Speech Act de J.L. Austin [7] y J.R. Searle [8]. La teoría Speech Act es una filosofía de comunicación entre seres humanos, que dice que “hablar es actuar”. En particular, esta teoría distingue tres aspectos cuando emite una sentencia:

- *Locución*, que consiste en la emisión de sentencias físicas que se refieren a la producción de sentencias usando una gramática y un vocabulario.
- *Ilocución*, que se refiere al propio acto que realiza el orador frente al oyente.
- *Perlocución*, que captura todos los efectos (directos o indirectos) que el acto puede inducir en las actitudes mentales del oyente.

En la teoría Speech Act, la ilocución es capturada por lo así llamados “performativos”. Dentro del inglés, Austin ha identificado un gran número de performativos (verbos ingleses) que tienen una fuerza ilocucionaria. Entre ellos están: desear, pedir, confirmar, asertar, prometer.

Siguiendo la teoría Speech Act, el ACL (inter-Agent Communication Languages) propone una colección de per-

formativos que se pueden usar, según el contexto de trabajo del agente (por ejemplo, la negociación), para crear “actos comunicativos”. Los ACL KQML y FIPA son dos lenguajes de comunicación usados normalmente por la comunidad de multiagentes.

El KQML ofrece un conjunto grande de performativos y es fácil de usar. Sin embargo, tiene varios inconvenientes: algunos performativos no tienen utilidad y son incluso redundantes; otros faltan, como los que se refieren a la obligación. En conjunto, el KQML no tiene una semántica formal asociada; consecuentemente, el utilizar performativos puede llevar a la equivocación o incluso a la ambigüedad.

Para remediar estos problemas, el consorcio FIPA ha especificado una semántica formal (una lógica modal de primer orden) basada en las actitudes mentales del agente (confianza, incertidumbre, preferencia, deseo, etc.). Con más precisión, cada performativo se asocia con:

- Qué se asume sobre el emisor antes de enviar el performativo (requisito previo).
- Qué efecto se espera en el receptor desde el punto de vista del emisor (efecto racional).

Para más detalles sobre la semántica de los performativos ver [22, 23, FIPA]. Los performativos son sólo un aspecto del acto comunicativo. El segundo consiste en el contenido del mensaje. Por ejemplo, en la frase: “Te pido que compres todas las existencias de Alcatel”, “yo” y “tu” son agentes financieros, “pides” es el performativo; el sujeto de la petición, llamado “contenido”, es “compres todo...” En el caso de dos agentes interactivos, como “yo” y “tu”, el principal problema es compartir un vocabulario común y un conjunto de conceptos (“comprar”, “todos”, “existencias”, “Alcatel”) que les permita entenderse. Este conjunto común se llama una ontología (compartida) [KS]. Las ontologías existen (o debían existir) para cualquier campo que permita los agentes, como el comercio electrónico, los viajes, la química, y la gestión de red. Definir una ontología no

es una cosa fácil, y necesita generalmente la experiencia de un tercero (por ejemplo, un experto financiero en el caso del comercio electrónico): la lista de conceptos y relaciones entre los conceptos debe ser completa y clara. Para más detalles sobre cómo crear ontologías de dominios ver los lenguajes como KIF [10], Ontolingua [Ontolingua] y XLM/Resource Description Framework (RDF) [XML/RDF].

Permitir que los agentes inteligentes interoperen no es una tarea fácil ya que existen diferentes tecnologías heterogéneas y propietarias. Además algunos aspectos de los agentes inteligentes se tienen que normalizar para permitir y facilitar su despliegue. La FIPA tiene como objetivo asegurar un alto grado de interoperatividad entre agentes diseñados por diferentes fabricantes. El ámbito de FIPA incluye los agentes y las interfaces de los diferentes componentes en el entorno con el que el agente tiene que interactuar (es decir, gente, otros agentes, software sin agentes y el mundo físico). Otros asuntos como el servicio de ontología y la seguridad, también se contemplan. Como regla general, sólo está normalizado el comportamiento externo y no el trabajo interno, que se deja a la creatividad del diseñador. En 1997 y 1998 se liberaron dos especificaciones diferentes. Se pueden encontrar más detalles en FIPA URL [FIPA].

■ Conclusiones

La tecnología de agentes es mucho más que “otra tecnología”; de acuerdo con muchos expertos, esta tecnología revolucionaría el mundo de las telecomunicaciones, las aplicaciones de redes y los servicios. Es particularmente apropiada para aprovechar el mercado electrónico y el alto grado de interconectividad disponible en todo el mundo. Proporciona un potente paradigma natural para diseñar aplicaciones totalmente distribuidas como el comercio electrónico, la gestión de red, los servicios on-line, las redes privadas virtuales y los asistentes personales, por nombrar solo unas pocas.

Sin embargo, para permitir y facilitar el despliegue global de las aplicaciones ba-

sadas en agentes, aún se tienen que resolver algunos problemas. Primero, los agentes tienen que ser seguros debido a que los usuarios tienen que confiar en ellos. Además, la seguridad será un aspecto técnico crucial en el futuro. Segundo, la tecnología, o al menos algunos de sus aspectos, aún tiene que ser normalizada para permitir a los agentes interoperar y comprenderse. Los trabajos de FIPA y OMG en este campo, durante los últimos años, han sido importantes.

Finalmente, existen una serie de preguntas relativas al impacto social y a la ética de los agentes inteligentes. ¿Cuál es o cuál será la exacta naturaleza de la relación entre un agente y su propietario? ¿Qué agente será demasiado dependiente del software? ¿Introducirá la obtención de información y el filtrado de agentes la censura o afectará a nuestra privacidad? ¿Puede la emergencia de la inteligencia ser considerada como un renacimiento del síndrome HAL? ¿Cómo caerán dentro de los marcos legales existentes? ¿Se aplicarán a las aplicaciones basadas en agentes (por ejemplo, las transacciones del comercio electrónico)? ¿Y, sin ser lo último, la generalización de las aplicaciones de ordenador/asistidas por agentes eliminarán las relaciones humanas?

Todas estas preguntas tienen que ser respondidas, haciéndose necesario un consenso antes de que los agentes inteligentes se puedan usar de forma segura y provechosa.

■ Reconocimientos

Agradecemos a los miembros del Centro de Investigación Corporativo de Alcatel, en Amberes, y en particular a B. Bauwens, su aportación sobre el proyecto MAPOS.

■ Bibliografía

- 1 M. J. Bradshaw (ed): "Software Agents", MIT Press and AAAL Press.
- 2 J. Ferber: "Les systèmes Multi-Agents", InterEditions, 1995.
- 3 C. Guilfoyle, E. Warner: "Intelligent Agents: the New Revolution in Software", OVUM Report, 1994.
- 4 N. R. Jennings, M. J. Woolridge (eds): "Agent Technology: Foundation, Applications and Markets", Springer and Unicom, 1997.
- 5 M. Minsky: "The Society of Mind", InterEditions, 1985.
- 6 "Agents and Multi-Agents Systems: Themes, Approaches and Challenges", Capítulo 1 de "Readings in Agents", Morgan Kaufmann.
- 7 J. L. Austin: "How To Do things With Words" Clarendon Press, 1962.
- 8 J. R. Searle: "Speech Acts", Cambridge University Press, 1969.
- 9 T. Finin, R. Fritzson, D. McKay, R. McEntire: "KQML as an agent Communication Language", Proceedings of the 3rd International Conference on Information and Knowledge Management, ACM Press.
- 10 M. R. Genezereth: "Knowledge Interchange Format", Proceedings of the 2nd International Conference on the Principles of Knowledge Representation and Reasoning.
- 11 D. Chess, C. Harrison, A. Kershenbaum: "Mobile Agents: Are They a Good Idea?", IBM Report RC19887 (88465), 1995. Disponible en www.research.ibm.com/xv-d953-mobag-ps.
- 12 J. White: "Mobile Agents White Paper", General Magic, 1996 (página HTML: www.genmagic.com).
- 13 "Mobile Agent System Inter-operability Facilities Specification (MASIF)", GMD Fokus and IBM, OMG TC Document orbos/97-10-05.
- 14 "An Intelligent Mobile Agent Platform: Grasshopper", IKV++ (página HTML: www.ikv.de).
- 15 "Voyager Core Package Technical Overview", ObjectSpace Inc, 1997.
- 16 R. Weihmayer, Velthuijsen: "Intelligent Agents in Telecommunications, Appears in [4], páginas 203-217.
- 17 T. Magedanz, R. Popescu-Zeletin: "Towards Intelligence on Demand-on the Impacts of Intelligent Agents on IN", Proceedings of the 4th International Conference on Intelligent Networks, 1996, páginas 30-35.
- 18 N. R. Jennings, M. J. Woolridge: "Application of Intelligent Agents", Appears in [4], páginas 3-28.
- 19 C. G. Langton (ed): "Artificial Life, an Overview", MIT Press, 1995.
- 20 P. Maes: "Modeling Adaptive Autonomous Agents", appears in [19], páginas 135-162.
- 21 "Creature 2" by CyberLife Technology.
- 22 P. R. Cohen and H. J. Levesque: "Intention is Choice with Commitment", Artificial Intelligence, volumen 41, páginas 213-261.
- 23 A. S. Rao, M. P. Georgeff: "Modeling Rational Agents with-in BDI-Architecture", Readings in Agents, página 317, Morgan Kaufmann.

■ Información en la web

- [FIPA]: www.fipa.org
- [General Magic]: www.genmagic.com
- [IKV++]: www.ikv.de
- [KIF]: www.cs.umbc.edu/agents/kse/kif
- [KQML]: www.cs.umbc.edu/agents/kse/kqml
- [KS]: logic.stanford.edu/sharing/knowledge.html
- [MIAMI]: www.fokus.gmd.de/research/cc/IMA/MIAMI
- [ObjectSpace]: www.objecspace.com
- [Odyssey]: www.genmagic.com/technology/mobile_agent.html
- [OMG]: www.omg.org
- [Ontolingua]:

François Carrez está a cargo de las actividades de agentes dentro del proyecto Software/Network Application en el Centro de Investigación Corporativo de Alcatel, en Marcoussis, Francia

GLOSARIO

■ API

Interfaz de Programa de Aplicación: método prescrito por un programa de aplicación por el cual otro programa de aplicación puede hacer una llamada al primer programa.

■ Applet

Un applet es un programa Java que se puede incluir en una página web, de igual forma que sucede con la inclusión de una imagen, Cuando se emplea un browser de web para ver una página que contiene un applet Java, el código del applet se transfiere al sistema del usuario y es ejecutado por el browser.

■ Cliente-Servidor

Cliente/servidor es una arquitectura informática que involucra procesos de clientes pidiendo servicios de los procesos del servidor.

■ CMIP/CMIS

Common Management Information Protocol: un protocolo de gestión de red basado en el modelo de comunicación OSI (Interconexión de Sistemas Abiertos). El relativo a Common Management Information Services (CMIS) define servicios para acceder a la información sobre objetos o dispositivos de red, cómo controlarlos, y cómo recibir sus informes de estado.

■ CORBA

Arquitectura de Negociación de Petición de Objetos Comunes: arquitectura y especificación normalizadas OMG (Object Management Group) para procesar objetos distribuidos. Las implementaciones CORBA proporcionan el middleware que permite escribir objetos en diferentes lenguajes y distribuirlos en máquinas diferentes que trabajan conjuntamente de forma transparente.

■ Computación Distribuida

Distribución de la lógica de aplicaciones y negocios a lo largo de múltiples platafor-

mas de proceso. La computación distribuida implica que el proceso se realice en más de un procesador para que se complete una transacción.

■ Framework

Conjunto de clases software orientadas a objetos que abarcan un diseño abstracto para soluciones a una familia de problemas relacionados, y soportan la reutilización con una mayor granularidad que las clases. Los frameworks son colecciones de objetos, servicios y protocolos reutilizables y colaborativos.

■ Hypermedia

Un documento hypermedia se compone de una serie de segmentos de medios (por ejemplo., texto, gráficos, sonido, animación) conectados por enlaces que ofrecen al lector diferentes caminos por los cuales navegar a través de la información disponible.

■ IDL

Lenguaje de Definición de Interfaz: lenguaje de especificación usado para definir las capacidades de un servicio distribuido CORBA junto con un conjunto común de tipos de datos para interactuar con dicho servicio distribuido.

■ Interoperatividad

Capacidad de los componentes y sistemas software heterogéneos para funcionar conjuntamente, sin un esfuerzo especial por parte del programador. Los productos pueden lograr la interoperatividad con otros productos por adhesión a las normas de interfaces publicadas y/o por el uso de un "broker" (negociador) de servicios que puede convertir una interfaz de productor en otra interfaz de productos sobre la marcha.

■ Middleware

Programa intermedio: Capa software que media entre los programas de aplicación y el sistema operativo.

■ OSI

Interconexión de Sistemas Abiertos: una descripción normalizada o "modelo de referencia" sobre como se deben transmitir los mensajes entre dos puntos de una red de telecomunicaciones. Su objetivo es guiar a los que implementan los productos para que éstos trabajen de manera consistente con otros productos. El modelo de referencia define siete niveles de funciones que tienen lugar en cada extremo de la comunicación.

■ TCP/IP

Transmission Control Protocol/Internet Protocol: el protocolo de comunicación básico de Internet. TCP/IP es un programa de dos capas: la capa superior TCP gestiona el desensamblado de un mensaje o archivo en paquetes más pequeños que se transmiten por Internet siendo recibidos por una capa TCP que reensambla los paquetes para convertirlo en el mensaje original. La capa inferior IP trata la parte de dirección de cada paquete para obtener el destino correcto.

■ UML

Lenguaje de Modelado Unificado: norma de facto para el análisis y diseño del software orientado a objetos. El UML es un lenguaje notacional de propósito general para especificar y visualizar el software complejo, especialmente los grandes proyectos orientados a objetos. Se basa en los previos métodos notacionales como el OMT (Técnica de Modelado de Objetos) de G. Booch y J. Rumbaugh, y en casos de utilización de I. Jacobson. El UML se está desarrollando bajo los auspicios del OMG.

ABREVIATURAS EN ESTE NÚMERO

A

| | |
|-------|--------------------------------------|
| ABS | Servicio de tarificación alternativa |
| ACL | Lenguajes de Comunicación de Agentes |
| AD | Módulo de Administración |
| AFS | Advanced Freephone Service |
| AL | Vida Artificial |
| AM | Gestión de Disponibilidad |
| API | Interfaz de Programa de Aplicación |
| AS | Servidor de Disponibilidad |
| ASAP | Perfil Asignación Severidad Alarmas |
| ASN.1 | Sintaxis de Notación Abstracta 1 |
| ATM | Modo de Transferencia Asíncrono |

B

| | |
|-----|-----------------------------|
| BDI | Belief, Desire, Intent |
| BSS | Subsistema de Estación Base |

C

| | |
|-------|--|
| CCBS | Servicio de Tarificación y Atención al Cliente |
| CDF | Formato de Definición de Canal |
| CMIP | Common Management Information Protocol |
| CMIS | Common Management Information Services |
| CMM | Capability Maturity Model |
| COMET | C++ OSI Management Extended Toolchain |
| CORBA | Arquitectura de Negociación de Petición de Objetos Comunes |
| COTS | Components Off The Shelf |
| CRC | Centro de Investigación Corporativo |
| CSM | Gestión de Servicios de Clientes |
| CUG | Grupo Cerrado de Usuarios |

D

| | |
|-----|-------------------------------------|
| DAI | Inteligencia Artificial Distribuida |
| DM | Gestor de Datos |
| DSM | Gestión Distribuida de Sistemas |

E

| | |
|-----|-------------------------------------|
| ECT | Transferencia Explícita de Llamadas |
| EFD | Event Forwarding Discriminator |
| ELM | Event and Log Management |

F

| | |
|-------|---|
| FCAPS | Fault Configuration, Accounting, Performance and Security |
| FIT | Failures In Time |
| FPA | Análisis de Punto de Función |
| FTP | Protocolo de Transferencia de Ficheros |

G

| | |
|-------|--|
| GDMO | Guía para Definir Objetos Gestionados |
| GPRS | Global Packet Radio Service |
| GRATE | Generic Rules and Agent Model Test bed Environment |
| GSM | Global System for Mobile Communication |
| GUI | Interfaz Gráfico de Usuario |

H

| | |
|------|---------------------------------|
| HDML | Handheld Device Markup Language |
| HM | Gestión del Hardware |
| HTML | Hyper Text Markup Language |
| HTTP | Hyper Text Transfer Protocol |

I

| | |
|------|---|
| I/O | Input/Output |
| IA | Ingoing Access |
| ICaB | Incoming Call Barrier |
| ICC | Atención Inteligente de Clientes |
| IDL | Lenguaje de Definición de Interfaz |
| IN | Red Inteligente |
| INAP | Intelligent Network Application Interface |
| IP | Protocolo Internet |
| ISDN | Integrated Services Digital Network |
| ISP | Proveedor de Servicios Internet |
| ISUP | Parte de Usuario RDSI |
| IT | Tecnología de la Información |
| IVAS | Servicios de Valor Añadido de Internet |
| IVPN | Red Internacional Privada Virtual |

J

| | |
|-----|----------------------|
| JDK | Java Development Kit |
|-----|----------------------|

K

| | |
|-----|------------------------------|
| KIF | Knowledge Interchange Format |
|-----|------------------------------|

| | |
|------|---|
| KQML | Knowledge Query and Manipulation Language |
|------|---|

L

| | |
|-----|-------------------|
| LAN | Red de Área Local |
| LP | Larch Prover |

M

| | |
|-------|---|
| MAPOS | Plataforma Multi-Agentes para Servicios On-line |
| MASIF | Mobile Agent Systems Interoperability Facility |
| MCD | Mobile Communication Division |
| MIB | Base de Información de Gestión |
| MIT | Árbol de Información de Gestión |
| MTBF | Tiempo Medio entre Fallos |
| MTP | Parte de Transferencia de Mensajes |
| MTTR | Tiempo Medio de Reparación |

N

| | |
|-----|---------------------------|
| NM | Gestión de Red |
| NSD | Network Services Division |

O

| | |
|-------|------------------------------------|
| O&M | Operaciones y Mantenimiento |
| OA | Outgoing Access |
| OCaB | Outgoing Call Barrier |
| ODL | Lenguaje de Definición de Objetos |
| OMC/R | Centro de Operaciones y Gestión |
| OMG | Grupo de Gestión de Objetos |
| OMT | Técnica de Modelado de Objetos |
| OODB | Base de Datos Orientada a Objetos |
| OS | Sistema Operativo |
| OSF | Operation System Function |
| OSI | Interconexión de Sistemas Abiertos |
| OSS | Operations Support Systems |

P

| | |
|-----|-----------------------------------|
| PDA | Personal Digital Assistant |
| PIN | Número de Identificación Personal |
| PMW | Project Management Workbench |
| PTA | Personal Travel Assistant |

Q

| | |
|-----|------------------------------|
| QSM | Qualitative Software Metrics |
|-----|------------------------------|

R

RNP Radio Network Planning
RPC Llamada a Procedimiento Remoto

S

SAAL Signaling ATM Adaptation Layer
SCCP Parte de Control y Conexión de Señalización
SCP Punto de Control de Servicio
SCE Entorno de Creación de Servicios
SDE Entorno de Desarrollo de Servicios
SDP Service Data Point
SEC Gestión de Seguridad
SEPG Software Engineering Project Groups
SIB Service Independent Building Block
SM Gestor de Software

SMC Centro de Gestión de Conmutación
SMP Punto de Gestión de Servicios
SMS Servicio de Mensajes Cortos
SNMP Protocolo Simple de Gestión de Red
SPI Mejora del Proceso Software
SS7 Sistema de Señalización N° 7
SSD Switching System Division
SSL Secure Socket Layer
SSP Punto de Conmutación de Servicios

T

TCAP Transaction Capabilities Application Part
TCP Protocolo de Control de Transmisión
TMN Red de Gestión de las Telecomunicaciones

U

UML Unified Modeling Language
UNIX-CE Elemento de Control basado en UNIX
URL Localizador Uniforme de Recursos
USST Usage Specification and Specification Testing

W

W3C World Wide Web Consortium
WSF Workstation Function
WWW World Wide Web

X

XML eXtensible Markup Language

OFICINAS EDITORIALES

Cualquier asunto relacionado con las distintas ediciones de la Revista de Telecomunicaciones de Alcatel deberá dirigirse al director adecuado. Las peticiones de suscripción deben enviarse por fax o por correo electrónico.

■ EDICIÓN INGLESA:

Mike Deason
Alcatel Telecommunications Review
Alcatel
54, rue La Boétie
75382 Paris Cedex 08
France
Tel.: 33 (0)1 40 76 13 48
Fax: 33 (0)1 40 76 14 26
E-mail: (ver edición francesa)

■ EDICIÓN FRANCESA:

Catherine Camus
Revue des Télécommunications
d'Alcatel
Alcatel
54, rue La Boétie
75382 Paris Cedex 08
France
Tel.: 33 (0)1 40 76 13 48
Fax: 33 (0)1 40 76 14 26
E-mail: catherine.camus@alcatel.fr

■ EDICIÓN ALEMANA:

Andreas Ortelt
Alcatel Telecom Rundschau
Alcatel
Department ZOE/FP
70430 Stuttgart - Germany
Tel.: (49) 711 821 446 90
Fax: (49) 711 821 460 55
E-mail: A.Ortelt@stgl.sel.alcatel.de

■ EDICIÓN ESPAÑOLA:

Gustavo Arroyo
Revista de Telecomunicaciones
de Alcatel
Alcatel
Ramirez de Prado 5
28045 Madrid - Spain
Tel.: (34-1) 330 49 06
Fax: (34-1) 330 50 41
E-mail: gustavo@alcatel.es

■ EDICIÓN CHINA:

Isabelle Liu
Alcatel China
Beijing Representative Office
2nd & 3rd Floor Landmark Building
8 North Dongsanhuan Road
Chaoyang District
Beijing 100004
P.R. China
Tel: 86 10 65924670
Fax: 86 21 65064265 /
65073784
E-mail: isabelle.Liu@alcatel.com.hk

El próximo número, a publicar durante el 2º Trimestre de 1999, tratará sobre "El Cambio Revolucionario desde los Circuitos a los Paquetes".