

**GESTIÓN DE SERVICIOS PARA REDES INTELIGENTES
UTILIZANDO EL SISTEMA DE SEÑALIZACIÓN No. 7**

Ing. Héctor Fabio Jaramillo Ordóñez

Director: Mag. Emigdio Andrés Lara Silva

Asesores:

P.Hd. José Fernán Martínez

P.Hd. Deni Librado Torres

**POPAYAN
UNIVERSIDAD DEL CAUCA
INSTITUTO DE POSTGRADOS EN ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN TELEMÁTICA
2004**

**GESTIÓN DE SERVICIOS PARA REDES INTELIGENTES
UTILIZANDO EL SISTEMA DE SEÑALIZACIÓN No. 7**

Ing. Hector Fabio Jaramillo Ordóñez

Director: Mag. Emigdio Andrés Lara Silva

Asesores:

P.Hd. Jose Fernan Martinez

P.Hd. Deni Librado Torres

**Monografía de tesis para optar al
Título de Magíster en Telemática**

**POPAYAN
UNIVERSIDAD DEL CAUCA
INSTITUTO DE POSTGRADOS EN ELECTRÓNICA Y
TELECOMUNICACIONES
MAESTRÍA EN TELEMÁTICA
2004**

Nota de aceptación

Jurado

Jurado

Ciudad y fecha (día, mes, año)_____

AGRADECIMIENTOS

Este es un breve espacio para poder expresar mi gratitud a todos los seres que me han aportado de alguna manera para alcanzar esta tan anhelada meta, por supuesto que el principal y más importante agradecimiento es para esa fuerza mística e indefinible a la que suelo llamar DIOS, quien con su amor infinito permitió que muchas personas me acompañaran y ayudaran en este proceso; a las personas que agradezco en especial son:

Mi Hija, Natalia Jaramillo Callejas, quien ha sido mi motor de impulso desde que llegó a mi vida, a mi madre, Nubia Lucia Ordóñez Campos, quien con su amor y buen ejemplo me ha enseñado los caminos rectos de la vida y a Andrea Viviana Yanza Hurtado, quien fue mi principal motivo de alegrías y mi mejor compañía en los momentos difíciles.

También quiero agradecer a mi director y asesores quienes me han enseñado mucho y me han colaborado en cada parte del proyecto para hacerlo una exitosa realidad, y de manera muy significativa deseo agradecer a los jóvenes: Ana Cristina Moya, Ricardo Andrés Vallecilla, Mahdi Safa, Mario Fernando Henao y Diego Fernando Miranda que han trabajado juiciosa e incansablemente para implementar los diferentes módulos de este proyecto como verificación de los planteamientos realizados en él, a ellos en especial los señalo por ser unos excelentes ingenieros y aún mejores personas.

Héctor Fabio Jaramillo

TABLA DE CONTENIDO

| | Pág. |
|--|-------------|
| 1 INTRODUCCIÓN | 1 |
| 2 ENTORNO DEL PROYECTO..... | 4 |
| 2.1 ANTECEDENTES..... | 4 |
| 2.1.2 Objetivo general..... | 9 |
| 2.1.3 Objetivos específicos | 9 |
| 3 TECNOLOGIAS HABILITADORAS | 11 |
| 3.1 MANEJADORES DE TRAMA Y MICROPROCESADORES PARA COMUNICACIONES | 11 |
| 3.1.1 Descripción del ambiente Hardware/Software de trabajo..... | 12 |
| 3.2 SOFTWARE BASADO EN COMPONENTES..... | 13 |
| 3.2.1 COMPONENTES [comp-def 2002] | 13 |
| 3.3 JAVA MANAGEMENT EXTENSIONS (JMX) | 18 |
| 3.3.1 Gestión de redes basada en componentes CORBA..... | 18 |
| 3.3.2 Gestión de Redes Basada en componentes Java | 19 |
| 3.4 GESTION EN TIEMPO REAL | 22 |
| 4 ARQUITECTURA SMART III Y SU GESTION BASADA EN SS7..... | 24 |
| 4.1 ARQUITECTURA SMART III..... | 24 |
| 4.2 DESCRIPCIÓN FUNCIONAL DE LA ARQUITECTURA..... | 26 |
| 4.2.1 SCP (Service Control Point) | 26 |
| 4.2.2 SSP (Service Switching Point)..... | 34 |
| 4.2.3 SMS (Service Management System) | 45 |
| 4.2.4 Características generales de operación del sistema implementado..... | 46 |

| | | |
|-------|---|----|
| 5 | EL MODELO DE CONSTRUCCION DE SOLUCIONES PARA SISTEMAS EMPOTRADOS PARA COMUNICACIONES | 48 |
| 5.1 | METODOLOGÍA | 48 |
| 5.1.1 | MCS (modelo para la construcción de soluciones)..... | 48 |
| 5.2 | TÉCNICAS DE CODISEÑO ELECTRÓNICO APLICADAS A SISTEMAS DE TELECOMUNICACIONES | 51 |
| 5.2.1 | Requerimientos de especificación..... | 52 |
| 5.2.2 | Diseño estructurado de sistemas heterogéneos (codiseño) | 54 |
| 5.2.3 | Java como lenguaje de especificación para sistemas CTI..... | 62 |
| 5.3 | MODELO DE CONSTRUCCIÓN DE SOLUCIONES CON CODISEÑO (MCS-Co)..... | 64 |
| 6 | TRABAJOS DE GRADO VINCULADOS AL PROYECTO Y VALIDACIÓN DE LA ARQUITECTURA | 66 |
| 6.1 | VALIDACIÓN DEL MODELO DE GESTIÓN SOPORTADA EN MENSAJES DE SEÑALIZACIÓN NO. 7 | 66 |
| 6.2 | VALIDACIÓN DE LA PLATAFORMA PARA PRESTACIÓN DE UN SERVICIO MULTIMEDIAL GESTIONABLE CON SS7..... | 70 |
| 6.3 | SISTEMA DE GESTIÓN INTEGRADA PARA LOS SERVICIOS DE SMART III..... | 73 |
| 7 | CONCLUSIONES | 76 |
| 8 | TRABAJOS FUTUROS | 78 |
| 9 | REFERENCIAS Y BIBLIOGRAFIA..... | 79 |
| | ANEXO 1: Certificaciones de pruebas contra centrales reales de la plataforma SMART III | 83 |

TABLA DE ILUSTRACIONES

| | Pág. |
|-------------|---|
| Figura 2.1 | Tecnologías de soporte..... 8 |
| Figura 3.1 | Arquitectura <i>hardware</i> del sistema 12 |
| Figura 3.2 | Agente JAVA de gestión dinámica 20 |
| Figura 4.1 | Modelo de despliegue del proyecto SMART III..... 25 |
| Figura 4.2 | Arquitectura funcional 27 |
| Figura 4.3 | Modelo computacional del VoD (vista del usuario) 29 |
| Figura 4.4 | Modelo computacional de VoD (vista del administrador)..... 30 |
| Figura 4.5 | Interfaz JMX para acceso al servicio 31 |
| Figura 4.6 | Modelo computacional total de VoD 34 |
| Figura 4.7 | División Funcional del Sistema 35 |
| Figura 4.8 | Formato de trama HDLC 36 |
| Figura 4.9 | Ambiente de trabajo ISUP 37 |
| Figura 4.10 | Diagrama en bloques funcionales subsistema ISUP 37 |
| Figura 4.11 | Componentes Lógica ISUP 39 |
| Figura 4.12 | Procedimiento ISUP-LOOPBACK..... 40 |
| Figura 4.13 | Arquitectura del sistema de gestión basado en SS7 a través de JMX. 44 |
| Figura 5.1 | Macro componentes del modelo 49 |
| Tabla 5.1 | Estructura de referencia para descripción del Sistema-Solución 49 |
| Figura 5.2 | Visión general del modelo de proceso de desarrollo 50 |
| Tabla 5.2 | Descripción de las fases de referencia 50 |
| Figura 5.3 | Fases del diseño estructurado Top-Down para co-diseño..... 54 |
| Figura 5.4 | Esquema de verificación funcional 55 |
| Figura 5.5 | División Funcional En Bloques 56 |

| | | |
|-------------|--|----|
| Figura 5.6 | Clasificación de los PLDs | 60 |
| Tabla 5.3 | Algoritmos de particionamiento en Co-diseño | 62 |
| Figura 5.7 | Metodología de codiseño propuesta para sistemas CTI..... | 63 |
| Tabla 5.4 | Estructura de referencia para descripción del Sistema-Solución | 64 |
| Tabla 5.4 | Estructura de referencia para descripción del Sistema-Solución (continuación) | 65 |
| Figura 5.8 | Visión general del modelo de proceso de desarrollo | 65 |
| Figura 6.1 | Panel de control de llamadas del SSP | 67 |
| Figura 6.2 | Interfaz de configuración del SSP..... | 68 |
| Figura 6.3 | Interfaz JMX en el web para SSP | 69 |
| Figura 6.4 | Presentación de inicio del servicio..... | 70 |
| Figura 6.5 | Interfaz de verificación de usuario..... | 71 |
| Figura 6.6 | Cuadro de despliegue de las películas..... | 71 |
| Figura 6.7 | Interfaz de control de reproducción | 71 |
| Figura 6.8 | Interfaz para búsqueda de usuarios del servicio..... | 72 |
| Figura 6.9 | Interfaz para edición de suscripción de usuarios del servicio | 72 |
| Figura 6.10 | Interfaz gráfica para Suscribir un nuevo usuario Buscar/Editar | 72 |
| Figura 6.11 | Interfaz gráfica para Buscar la lista de videos existentes..... | 73 |
| Figura 6.12 | Interfaz gráfica para Agregar un nuevo video..... | 73 |
| Figura 6.13 | Interfaz para registro de nuevos servicios..... | 74 |
| Figura 6.14 | Interfaz de monitoreo de conexiones y solicitud de estadísticas..... | 75 |

1 INTRODUCCIÓN

Dentro de las líneas de investigación del Departamento de Telemática, de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, siempre ha existido un marcado interés por las aplicaciones para servicios a través de telefonía convencional, tanto así que desde allí fue como surgió el grupo de investigación con sus primeros proyectos relacionados con digitalización de centrales telefónicas electromecánicas; hoy en día los retos han cambiado para los investigadores de los servicios telefónicos y las nuevas tendencias en comunicación de datos a través de redes como Internet han volcado los esfuerzos sobre las redes telefónicas para ofrecer servicios comparativos y competitivos con respecto a los que ofrece Internet.

El Departamento de Telemática cultiva y se fortalece en las nuevas tendencias de comunicación a través de las redes de datos, pero también mantiene su compromiso con la comunidad para facilitar la prestación de más y mejores servicios de telefonía, es por tal motivación por la que se ha desarrollado la plataforma SMART (*Sistema Modular para Aplicaciones de Redes inteligentes y Telemáticas*), constituyéndose en una plataforma de bajo costo para la prestación de servicios básicos de Red Inteligente (RI); sin embargo hoy en día se debe pensar más allá para mantener la competitividad en el mercado de las telecomunicaciones, sin tener que realizar enormes inversiones de dinero, es por esto que la fase III del proyecto SMART se orienta a la ampliación de la plataforma para la prestación de servicios avanzados de telecomunicaciones a través de la red telefónica pública conmutada (RTPC o PSTN - Public Switched Telephonic Network) y que a su vez facilite la gestión que es necesaria para mantener la calidad del servicio (QoS – Quality of Service) ofrecido y los controles de tarificación del mismo.

La ampliación propuesta para la plataforma existente se desarrolló con la financiación al proyecto SMART III por parte de la Vice Rectoría de Investigaciones de la Universidad del Cauca (VRI-Unicauca), con su programa “*Apoyo a proyectos en el marco de maestrías y doctorados*”, y dentro de este proyecto de investigación se enmarca el presente trabajo de maestría, el cual recoge los objetivos del proyecto y se constituye en sí mismo en el proyecto SMART III como tal, dentro del cual se financiaron y desarrollaron los diferentes módulos constitutivos de la plataforma actualizada a través de diferentes trabajos de grado desarrollados dentro del programa de ingeniería electrónica y de telecomunicaciones.

Con el desarrollo del proyecto SMART III se pretendía generar un producto comercializable que facilitara la competitividad en el mercado de las telecomunicaciones, para así aumentar la oferta y calidad de servicios, fomentando la baja en los costos de acceso para el usuario común; también se buscó crear una arquitectura de referencia para la integración de los niveles de la gestión de servicios y la gestión de red a través de la utilización del sistema de señalización telefónico No7 (SS7) como protocolo para monitoreo y gestión de recursos que nos permita mantener la QoS y actuar sobre ella en tiempo real.

Por todo lo anterior se identifica el presente trabajo con el proyecto SMART III, y a lo largo del documento se hará referencia a uno y al otro de manera indistinta, como dos formas de llamar un mismo objeto.

SMART III es actualmente una plataforma para prestación de servicios avanzados de telecomunicaciones para usuarios de la RTPC que puede monitorear, procesar y modificar los mensajes de señalización No7. para obtener control sobre la operación de la red sobre la que se transportan los servicios de la plataforma, de forma tal que además de la prestación de los servicios se pueden crear nuevos componentes gestionables para aumentar las capacidades de control del SSP sobre la red.

Para el presente documento se ha organizado la información de manera que facilite al lector el seguimiento del proceso de planteamiento, desarrollo y verificación de los productos del proyecto, a través de los capítulos que constituyen la monografía de la siguiente manera:

En el capítulo 1, titulado entorno del proyecto, se encuentra una reseña acerca de las condiciones actuales de los servicios multimediales sobre redes de telefonía fija y los sistemas de gestión que los cobijan.

En el capítulo 2, titulado tecnologías habilitadoras, se exponen las diversas tecnologías hardware y software que fueron utilizadas durante este proyecto, a la vez que se indican los motivos de su selección.

El capítulo 3, que se titula arquitectura para gestión de servicios usando señalización No 7, está explicada la arquitectura de gestión propuesta, con cada uno de sus módulos constitutivos.

El capítulo 4, que se titula metodología para desarrollo de sistemas empotrados para telecomunicaciones, contiene la metodología propuesta, y usada durante el desarrollo de este proyecto, donde se plantean algunas extensiones al modelo de construcción de soluciones para usarlo como metodología para desarrollos *hardware/software* en telecomunicaciones.

En el capítulo 5, titulado proyectos de verificación y validación, se reseñan los trabajos de grado vinculados al proyecto SMART III, a través de los cuales se desarrolló la implementación de cada uno de los módulos básicos del sistema para su verificación operativa y validación.

El penúltimo capítulo, titulado conclusiones, contiene las conclusiones propias del desarrollo del proyecto y la propuesta arquitectónica.

Por último, el capítulo final con título: trabajos futuros, incluye el planteamiento de extensiones y evoluciones de la plataforma para solucionar los nuevos retos de los servicios multimediales en casa.

De esta forma se da inicio a este documento que pretende ser una referencia comparativa para futuros desarrollos de sistemas empotrados en telecomunicaciones y en especial los orientados a la gestión de servicios.

Universitariamente,

Héctor Fabio Jaramillo

2 ENTORNO DEL PROYECTO

2.1 ANTECEDENTES

Las TIC (Tecnologías de la Información y las Comunicaciones) en el mundo han sido uno de los negocios más importantes y con mayor crecimiento en las últimas décadas; en Latino América se ha mantenido un crecimiento que ha estimulado la inversión en el sector, y en Colombia en los últimos 5 años se ha presentado un crecimiento bastante alto pero que presenta una diferencia de aproximadamente 5.5 puntos porcentuales con respecto al comportamiento de toda Latino América [CCIT 2003]; ya que la mayor parte del crecimiento en Colombia se ha presentado en el sector de las comunicaciones móviles, sin embargo el sector de las telecomunicaciones fijas ha crecido y reportado ganancias constantemente pero sólo para la telefonía local, donde las empresas prácticamente manejan un monopolio en cada región del país, y cada día las empresas más grandes como ETB y EPM abarcan más mercado y las pequeñas operadoras tienden a desaparecer. Hoy por hoy incluso las grandes compañías de telefonía fija han comenzado a ver reducidos sus ingresos a causa de la competencia que plantean los operadores móviles que ofrecen más y algunas veces mejores servicios que su contraparte fija, es por esta clase de situaciones que muchos expertos ya aseguran lo que hace tiempo se venía venir: “El mercado de las telecomunicaciones se orienta al cliente y se dirige a través de servicios”.

Una de las áreas del mercado donde aún se mantiene un alto nivel de competencia, dando la oportunidad de penetrar en él y poder subsistir es el sector de los servicios de valor agregado, y es en este sector donde los servicios basados en Redes Inteligentes [IEC 2000] [ITU-T 1997a] pueden marcar la diferencia, por lo cual se hacía necesario crear un soporte arquitectónico que brinde facilidades para la prestación y gestión de estos servicios, pero que a su vez sea de bajo costo para poder ser adquirido por las pequeñas empresas que pretenden mantenerse o incursionar en el mundo de los servicios, es allí donde las TIC pueden ofrecer soporte a este tipo de desarrollos que propendan por el mejoramiento y ampliación de los servicios para redes fijas, manteniendo bajos costos de inversión para las mismas.

En el mundo de los servicios a través de redes inteligentes se han mantenido dos niveles de gestión básica (inspirados en la gestión TMN), a saber, la gestión de los servicios y la gestión de la red; lo cual ha creado un sistema de gestión dividido y de alto costo que limita su difusión entre las pequeñas y medianas industrias de las

telecomunicaciones que pretenden incursionar en este mundo, por lo cual se hacia necesario, además de las condiciones de competitividad antes mencionadas, que exista dentro del soporte arquitectónico de los servicios unas capacidades de gestión de red y servicios de manera integral y económica.

Actualmente en Colombia las empresas medianas y pequeñas del sector de las telecomunicaciones fijas cuentan con un mercado que temporalmente les permite tener una solvencia económica suficiente para mantenerse en competencia, dado que manejan un poco más de 2'500.000 del total de 9'000.000 aprox. de líneas telefónicas activas que existen en el mercado nacional (repartidas entre TELECOM, ETB y EPM) [DNP 2003] [CRT 2003]. Sin embargo las oportunidades de negocio que surgen con la implementación de nuevos servicios utilizando Redes Inteligentes (RI) se les escapan de las manos debido a los altos costos de adquisición y mantenimiento de los equipos suministrados por los grandes proveedores de sistemas de telecomunicaciones. Esta situación les restringe la explotación que pueden hacer de la red y poco a poco los desplaza fuera del mercado por no tener capacidades para prestación de nuevos servicios.

Dadas las actuales tendencias de convergencia tecnológica y las múltiples ventajas que se presentan al integrar las fortalezas de diferentes tecnologías, existe la posibilidad de construir servicios interdominio (entre otros) que rompan las fronteras de las redes de telecomunicaciones y permitan el intercambio de información de múltiples formas. El paradigma actual, para el logro de este objetivo de integración de los servicios de telecomunicaciones, es la arquitectura de Red Inteligente basada en componentes [OMG 2002] y usando TINA [TINA-C 1996a] [TINA-C 1996b] como modelo de implementación, con capacidad para ofrecer servicios de valor agregado universales a través de Internet y de la Red Telefónica Pública Conmutada (RTPC).

Con el Proyecto **SMART**¹, que ha contado con financiación de convocatorias anteriores de la VRI, el Grupo de Ingeniería Telemática ha estado buscando la estructuración e integración de una solución abierta y de bajo costo para la implementación y prestación de servicios de Red Inteligente y Servicios Avanzados de Telecomunicaciones, que sea asequible a las pequeñas y medianas empresas operadoras de telecomunicaciones, permitiéndoles mantener su competitividad en el mercado nacional de las telecomunicaciones.

Uno de los principales inconvenientes que ha tenido el proyecto SMART es la ausencia de un módulo para utilizar el Sistema de Señalización No.7 (SS7), el cual es aplicado en monitorización y control de las llamadas con fines de tarificación y gestión [ITU-T 1997b]. Para su incorporación en la plataforma SMART se requiere un módulo que integre de manera adecuada circuitos especializados de telecomunicaciones con programas de manejo de protocolos. Los módulos comerciales que se conocen (Hughes SS7 Protocol Stack², SpiderSS³) tienen costos prohibitivos para los recursos disponibles en el proyecto, y las iniciativas que se

¹ SMART III, Sistema Modular para Aplicaciones en Redes Inteligentes y Telemática fase III

² http://www.hssworld.com/protocolstacks/ss7/ss7_home.htm

³ <http://www.artesyncp.com/products/Spider-SS7.html>

vienen adelantando a nivel internacional para obtener una solución de dominio público (OpenSS7⁴) aún no han avanzado lo suficiente para resolver las necesidades del proyecto. En las fases anteriores, el proyecto había optado por una doble estrategia consistente por una parte en utilizar el sistema de señalización R2 Digital, de menor capacidad pero de más fácil implementación, y por otra intentar el desarrollo del módulo SS7 a partir de productos que había obtenido el proyecto CTD (Concentrador Telefónico Digital) en 1994; sin embargo, aún no se han logrado resultados satisfactorios, y hace cuatro años entró en vigencia una norma del Ministerio de Comunicaciones [Resolución 087 de 1997, ARTICULO 6.15. OBLIGACION DE LOS OPERADORES DE TPBC DE ATENDER LAS NORMAS TECNICAS OFICIALES PARA LAS REDES] exigiendo el uso de SS7 en toda la red nacional de telecomunicaciones. Frente a esta situación, se planteó como hipótesis que es posible apropiarse la tecnología requerida para construir un módulo empotrado [LOPEZ 1998] para SS7 utilizando los circuitos más avanzados especializados en telecomunicaciones, tales como el QUADFALC y el C165UTAH [Infineon -2002] de Infineon, e integrar dicho módulo a la plataforma básica de servicios RI de SMART.

La superposición de la RI sobre la RTPC requiere contar con módulos SS7 para gestionar las llamadas, ya que la norma nacional exige la operación con este tipo de señalización para todas las centrales telefónicas del área urbana. Inicialmente se tenían tres alternativas para cumplir este requisito en SMART, a saber:

Utilización de tarjetas CTI para procesamiento de SS7, las cuales son de fácil manejo, pero su principal inconveniente es el costo de adquisición debido al cobro que realizan por el programa que maneja la pila de protocolos de SS7.

Utilización de microprocesadores y microcontroladores que posean la pila del protocolo HDLC, el cual podría ser modificado para asimilar las funciones del SS7 hasta el MTP3. Esta pila también tiene un costo un poco elevado para los presupuestos disponibles, pero el principal problema radica en la necesidad de modificar el microcódigo de HDLC, lo cual conllevaría mucho tiempo y esfuerzo de desarrollo.

La última alternativa, y que se consideró como la posibilidad más favorable, es emplear procesadores especializados para comunicaciones, entre los cuales existen varios que incorporan la pila del SS7 hasta el MTP2 (familia Power Quicc, 68360 de Motorola y QUADFalc de Infineon, entre otros) lo cual reduce los tiempos de trabajo para generar el producto modular completo, además nos permite dedicarnos sólo a la implementación del *software* para procesamiento y generación de mensajes MTP3(básicos) e ISUP, ya que todos los mensajes de bajo nivel serían manejados por los microprocesadores seleccionados con sólo configurarlos. Una de las mejores ventajas que nos ofrece esta alternativa de desarrollo es el proceso de afianzamiento de las técnicas para desarrollo de sistemas de gestión, entre otras aplicaciones basadas en SS7.

⁴ <http://www.openss7.org>

Con cualquiera de las alternativas anteriores se debía garantizar que el sistema podría establecer, controlar y liberar cualquier llamada de voz o datos que utilizara SS7, además, a través de los mensajes intercambiados con SS7 poder generar estadísticas de operación de la red y el servicio, así como aprovechar capacidades básicas de gestión de la llamada para realizar reasignación de recursos de llamada en curso, para controlar la calidad del servicio prestado a través de RI-RDSI, y generar datos de inicio y finalización de llamada, número de llamadas fallidas, número de llamadas exitosas y otras estadísticas de operación y tarificación.

El uso de SS7 como sistema de transporte y control para gestión de llamadas no es algo nuevo, pero si es algo que hasta ahora se ha comenzado a implementar de manera formal (aunque estaba previsto en las recomendaciones Q.75x de la ITU-T) debido a la coyuntura tecnológica que ahora se vive y que permite realizar este tipo de desarrollos; estas tecnologías y concepciones que han dado viabilidad a un proyecto como este se pueden resumir en la figura 2.1, donde se aprecian las tecnologías, estándares y trabajos relacionados que brindan soporte a los objetivos de SMART III.

También cabe indicar que el uso de microprocesadores/microcontroladores especializados para comunicaciones abrirá muchas nuevas posibilidades para investigación y desarrollo dentro de la Facultad de Ingeniería Electrónica y Telecomunicaciones, ya que este tipo de trabajos se había relegado por los altos costos de implementación, pero actualmente se puede acceder a la tecnología para diseñar, implementar y verificar tales tipos de sistemas con bajos costos de inversión, y es con este proyecto que se retomará ese camino que ofrece muchas posibilidades para los distintos grupos de la FIET y afianzará la preparación de nuestros egresados en el conocimiento y manejo del SS7.

Un sistema como el desarrollado para la plataforma SMART III constituyó un reto por la gran cantidad de tecnologías que comprende y el impacto que puede generar a nivel social y comercial en el ámbito de los servicios de telecomunicaciones; inclusive ya se están viendo esfuerzos similares por realizar gestión a través del SS7 como es el caso de Net(s) CCS de Tekno Telecom [TEKNO 2003] pero aún con capacidades limitadas al monitoreo de los equipos de red.

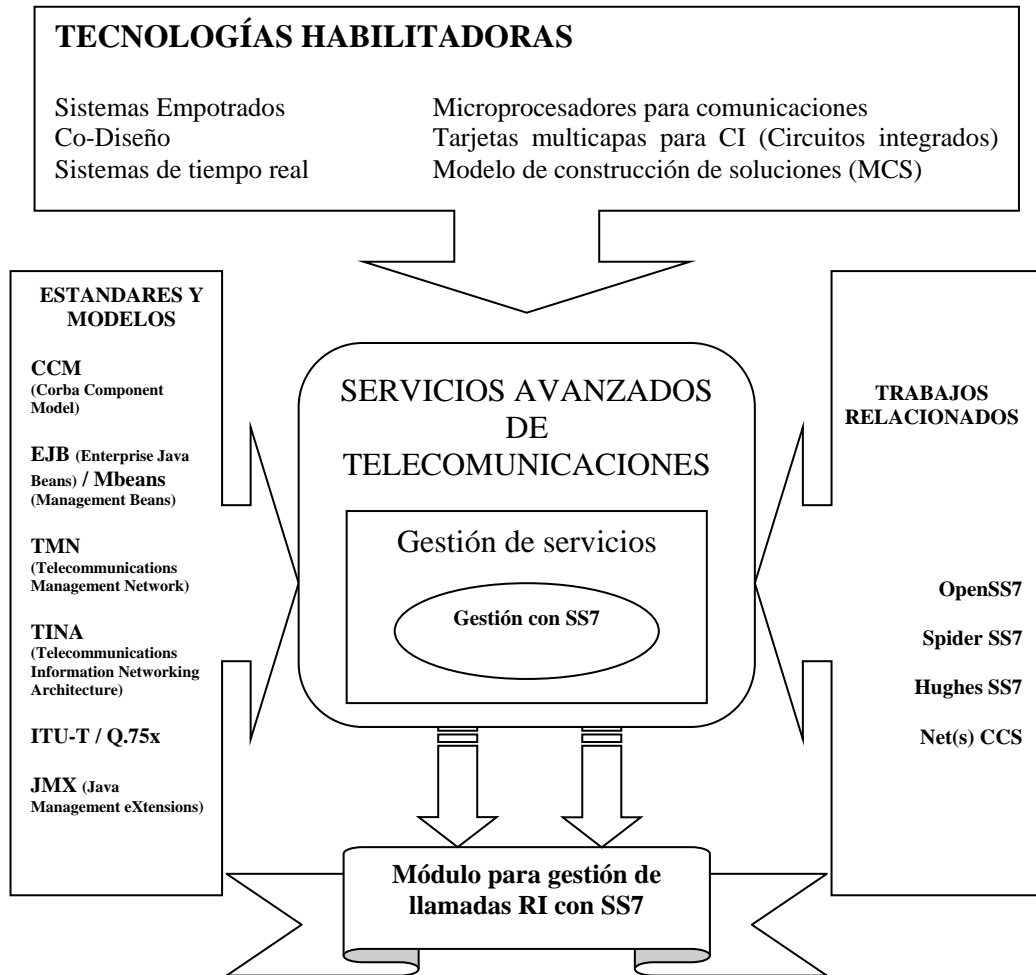


Figura 2.1 Tecnologías de soporte

Utilizar el SS7 como un protocolo de gestión ha sido un interés generado en varias ocasiones en el pasado, incluso por el ITEC de Telecom en Colombia, pero con resultados incompletos debido a las tecnologías de soporte con las que contaban; actualmente existen gran cantidad de tecnologías de interacción *hardware/software* que facilitan el acceso a los objetivos del proyecto SMART III, entre estas tecnología se puede mencionar de manera especial a JMX (Java Management eXtensions) con la cual se puede acceder a las funciones de gestión en tiempo real del hardware y comunicarnos directamente con el software, teniendo un sistema distribuido y con soporte para tiempo real efectivo; aunque sobre esta misma línea se está moviendo RT-CORBA, este no fue adoptado como tecnología de desarrollo dado su necesidad de un entorno de ejecución que no hace parte de él mismo, situación que no se presenta con JMX soportado con RTJ (Real Time Java), cuyo entorno de operación es provisto por Sun y TimeSys.

Algo a resaltar es el interés que hubo en este proyecto de crear una arquitectura para gestionar red y servicios de manera integrada, de forma que se puedan modificar las condiciones de operación sobre la red (o una llamada en particular como es el caso) para mantener la calidad de un servicio, o modificar las condiciones de prestación del servicio para adaptarse a la operación de la red de transporte, además de poder monitorear el comportamiento de dichas llamadas y modificaciones para generar toda la información estadística pertinente para evaluar la operación del sistema como un todo.

También se contó con la ventaja de tener la facilidad de contar con dispositivos como el QUADFalc de Infineon el cual brinda un soporte completo para los niveles básicos del MTP de SS7 a un precio mucho menor que cualquier otro componente que realice las mismas funciones.

En general se ha visto que existen muchas soluciones completas en el mercado, pero a precios muy altos, o soluciones con precios más accesibles pero con capacidades limitadas, por esto este proyecto propende por la creación de una solución completa a precios razonables y que instaure una arquitectura de referencia para el desarrollo de aplicaciones similares e implementación de servicios avanzados de telecomunicaciones sobre RI que sean gestionables a través de SS7.

Dentro del proyecto SMART III se plantearon varios objetivos que a través del desarrollo se han podido satisfacer e incluso sobrepasar en algunos puntos, todo esto gracias al apoyo de las empresas (operadores y proveedores) del sector de las telecomunicaciones, tales como: Nortel Networks, Ericsson de Colombia, Siemens, Telecom, Emcali y Transtel, y con el apoyo y esfuerzo del equipo de trabajo del proyecto SMART III, estos objetivos planteados fueron:

2.1.2 Objetivo general

Diseñar e implementar una tarjeta impresa para manejo de SS7 hasta el nivel ISUP, y su *interfaz* de control de forma que permita gestionar la llamada básica y los servicios RI de SMART, utilizando tecnologías para gestión en tiempo real a través de la señalización No. 7.

2.1.3 Objetivos específicos

Los objetivos específicos han sido distribuidos en 5 áreas para delimitar mejor su enfoque y alcance, a saber:

EVALUACIÓN

A partir de las actuales arquitecturas propuestas como solución para soportar y gestionar servicios avanzados de telecomunicaciones, crear una arquitectura que ofrezca soluciones equivalentes pero a menor costo, probando el modelo de construcción de soluciones y las metodologías de Co-Diseño para la caracterización del sistema empotrado para comunicaciones que servirá como plataforma.

FORMALIZACIÓN

Diseñar una arquitectura de referencia para la gestión de servicios y gestión de red utilizando las capacidades de SS7, aplicada a los servicios soportados sobre una Red Inteligente, de forma que se pueda crear una integración real de estos dos niveles de gestión a través de un protocolo único y universal.

Generar un modelo básico para el desarrollo de proyectos en Telemática que impliquen creación de *hardware* y *software* en un mismo módulo utilizando técnicas de Co-Diseño.

APLICACIÓN

Crear un módulo empotrado que pueda monitorear y gestionar señalización # 7 para ofrecer capacidades de gestión de llamada y servicio para los servicios de una Red Inteligente.

VALIDACIÓN

Gestionar la llamada básica al sistema SMART a través de SS7 para soportar funciones de re-enrutamiento y facilitar las tareas de tasación del servicio.

Gestionar la calidad de un servicio avanzado de telecomunicaciones (multimedial) a través de SS7 utilizando una arquitectura abierta para gestión.

DIVULGACIÓN

Presentar los avances relevantes del proyecto y sus resultados finales en artículos para congresos y revistas.

A través del documento y en especial en el capítulo 6 se podrá ver claramente como se cubrieron cada uno de estos aspectos del proyecto y sus resultados finales.

3 TECNOLOGIAS HABILITADORAS

3.1 MANEJADORES DE TRAMA Y MICROPROCESADORES PARA COMUNICACIONES

Durante las fases previas del proyecto SMART, y aún en los inicios de la fase III, uno de los principales obstáculos era el tener una solución para el manejo de los niveles bajos de la pila de protocolos de SS7, esta solución debía ofrecer un completo manejo de las restricciones temporales que imponen el MTP1 y MTP2 para garantizar el establecimiento y sostenimiento del enlace mientras fuera necesario.

En el pasado, e incluso en algunas aproximaciones actuales, se trató de soportar las tareas de tiempo real sobre la lógica del PC utilizando S.O.T.R. (Sistemas Operativos de Tiempo Real), pero lastimosamente esas aproximaciones son bastante difíciles y requieren un equipamiento muy especial en el PC (casi como usar CTI) para poder mantener la operatividad del protocolo contra una central telefónica real; por lo anterior se decidió utilizar un dispositivo empotrado que pudiera soportar las restricciones temporales y funcionara como la base necesaria de la pila de protocolos para descansar la lógica de los niveles superiores sobre equipos con S.O.T.R.

Dentro de la gama de posibilidades que se analizaron dentro del proyecto SMART III se tuvieron en cuenta microprocesadores (uP) para comunicaciones de diferentes proveedores, y en especial las capacidades que ofrecen los productos de Motorola, tal como el 68360, 68260 y otros derivados de la serie PowerQuick II, se constituían como la mejor opción; sin embargo esta posibilidad tuvo que descartarse debido a las dificultades de poder adquirir un sistema para desarrollo de prototipos sobre el cual se pudiera trabajar el desarrollo del sistema, además cada uno de dichos uPs estaba cerca de los 300 dólares americanos y el presupuesto del proyecto debía tratar de optimizarse para adquirir no sólo los uPs sino también su entorno de desarrollo y tarjetas de prototipado.

La solución final fue utilizar un microcontrolador (uC) para comunicaciones que en conjunto con un manejador de trama (framer) nos podía ofrecer las características equivalentes a un buen uP de comunicaciones, a esta descripción se ajustó el framer QuadFalc 22554 y el uC C166 UTAH, los cuales no sobrepasaron los 50 dólares en conjunto y contaba con un completísimo sistema de desarrollo y tarjetas de prototipados a bajo costo; curiosamente después de mucho trámites para realizar la

adquisición de dichos equipos y componentes se tomó la decisión de contactar directamente a los fabricantes de estos dispositivos, la empresa Infineon que es una subsidiaria de Siemens y ellos al enterarse de los objetivos del proyecto y sus posibles aplicaciones decidieron donar el equipo para la Universidad, el cual se constituyó en el corazón y base del éxito de este proyecto.

3.1.1 Descripción del ambiente Hardware/Software de trabajo

El ambiente *hardware* de desarrollo utilizado para obtener las funcionalidades de la Parte de Transferencia de Mensajes (MTP) del protocolo SS7 se basa en la tarjeta para prueba de prototipos Easy22554 orientada a la prueba de un framer especializado en telecomunicaciones que funciona como interfaz de línea y permite la obtención de las tramas codificadas en HDB3 provenientes de un E1 estándar [Easy25554-2001]. Sus funciones se encuentran divididas en una serie de bloques funcionales que interactúan a altas velocidades sobre un uC orientado a sistemas RDSI el cual cuenta con periféricos especializados como HDLC, interfaz de puertos seriales, interfaces IOM-2, y trabaja a una velocidad de 30 MHz suficientes para responder adecuadamente a los requerimientos de velocidad exigidos por el protocolo como se muestra en la figura 3.1

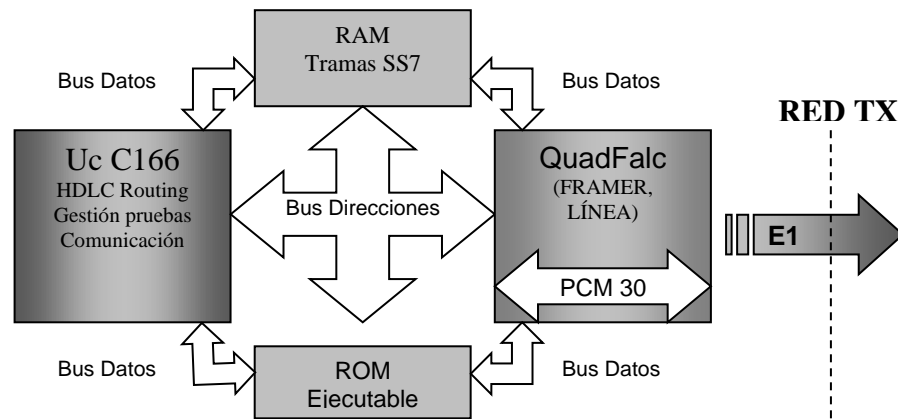


Figura 3.1 Arquitectura *hardware* del sistema

La arquitectura *hardware* esta dividida en bloques funcionales de acuerdo a la carga de procesamiento que deben manejar y las restricciones temporales que deben cumplir en ellos, tal como lo especifican las técnicas de Codiseño para sistemas de este tipo. Consiste de un uC con expansión de memoria ROM (1MB) y RAM (1 MB) lo cual es suficiente para el manejo de un enlace de señalización. Su principal componente es un ASIC o circuito integrado especializado, el cual es la interfaz con el mundo externo (Los E1 de TX) y realiza las labores más críticas en el tiempo. Su funcionamiento es sintetizado en *hardware* y aunque el uso de estos circuitos no permite que el sistema sea muy evolutivo, su justificación de uso esta en su gran velocidad, eficiencia y versatilidad, lo que reduce considerablemente los tiempos de desarrollo. Estos componentes principales están conectados por un bus de direcciones

multiplexado en 16 segmentos de 64 kbytes y un bus de datos de 16 bits el cual transporta todo el flujo de información entre los componentes. El control del bus de datos esta centralizado totalmente sobre el uC, el cual es el maestro y el responsable del control de toda la información interna, mientras el framer trabaja en modo esclavo y se encarga del manejo y tratamiento de información PCM de acuerdo a las configuraciones hechas sobre él por el microcontrolador.

3.2 SOFTWARE BASADO EN COMPONENTES

Otro de los habilitadores de la solución alcanzada en SMART III fue la maduración y conocimiento de las tecnologías de componentes, los cuales son un producto indirecto de los conceptos de SIB (Service Independent Block) propios de los servicios de redes inteligentes.

Anteriormente la implementación de un servicio debía realizarse a través de los componentes reutilizables para servicios de RI, conocidos como SIB, lo cual dificultaba bastante los procesos de desarrollo e integración de servicios que requirieran la modificación de los SIBs existentes o la creación de uno nuevo, es allí donde las tecnologías de componentes ha abierto un nuevo espacio para el desarrollo de servicios independientemente del lenguaje de implementación y la plataforma de soporte, a continuación se presentan algunos de los conceptos más relevantes acerca de esto y cómo sus características se amoldan perfectamente a los intereses de desarrollo de servicios dentro del proyecto SMART III.

3.2.1 COMPONENTES [comp-def 2002]

“Un componente es una unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos que puede ser desarrollado, adquirido e incorporado a un sistema compuesto con otros componentes desarrollados de forma independiente, en tiempo y espacio” Clemens Szyperski.

3.2.1.1 Características

Los componentes son unidades de despliegue o implementación, por lo tanto no corren dentro del sistema, es decir, se ejecutan dentro de contenedores que interfazan las comunicaciones con otros componentes para crear la lógica del sistema solución. Cada componente tiene una interfaz pública, la cual define sus responsabilidades. Para crear soluciones los componentes son ensamblados, la interacción entre éstos se realiza a través de estándares de comunicación.

3.2.1.2 Beneficios

Técnicos: Como los componentes se encuentran limitados, los desarrolladores se pueden concentrar en un pequeño conjunto de funcionalidades, generando una mayor productividad al no tener que preocuparse por aspectos irrelevantes.

Para el negocio: El desarrollo paralelo y el desarrollo externo de recursos, generan un rápido desarrollo. Aunque hay un gran trabajo en localizar componentes adecuados y un mayor esfuerzo al integrarlos para generar soluciones, este es menor al necesario en la construcción de todo un sistema desde cero.

Económicos: El costo de construir un sistema a través de componentes se ha ido reduciendo a medida que el mercado de estos se ha incrementado, ya que ha creado competitividad y por lo tanto los precios han disminuido.

3.2.1.3 Diferencias entre objetos y componentes

Los componentes se extienden por composición y no por herencia.

Los objetos se enfocan en entidades conceptuales mientras que los componentes están basados en cosas físicas.

Los componentes y los objetos son dependientes del lenguaje y de *interfaces* separadas de la implementación, pero los componentes dependen también del contexto.

Los componentes son más pragmáticos, lo que impone una disciplina menos rigurosa para los desarrolladores, y más genéricos, lo que permite la creación de un mercado de componentes.

3.2.1.4 Desarrollo basado en Componentes (CBO - Component-Based Development)

Este es el proceso de construir aplicaciones con componentes reusables, disminuyendo el tiempo y el costo del desarrollo. Es una extensión de la programación orientada a objetos e incluye la filosofía “*reuse antes de comprar, compre antes de construir*”.

En el desarrollo basado en componentes, se utilizan tanto componentes desarrollados al interior de la organización, como componentes de mercado abierto. Los desarrolladores escriben los componentes adicionales y el código que integra todos los componentes. Estos nuevos componentes que se crean son almacenados para que otros desarrolladores puedan utilizar las funcionalidades que se han creado, lo que finalmente disminuye el costo de desarrollo.

A nivel mundial las tecnologías *software* basadas en componentes se han organizado alrededor de tres grandes modelos que constituyen la regencia cuando se trata de trabajar con componentes, estos modelos [soft-comp 2000] son: Modelo de Objetos de Componentes (COM - Component Object Model) de Microsoft [comp-micro 2003], Java Beans y Enterprise Java Beans (EJB) de Sun Microsystems [comp-sun 2003] y el Modelo de componentes de CORBA (CCM - CORBA Component Model) del OMG (Object Management Group, Inc.) [comp-corba 2002].

Microsoft [Comp-micro 2003]

El COM de Microsoft se ha convertido en uno de los más populares estándares en la industria. Microsoft envolvió este estándar en el Modelo de Componentes de Objetos Distribuidos (DCOM) que soporta componentes y aplicaciones distribuidas. Ese

modelo básico continua evolucionando con COM+, el cual entre otras cosas, soporta un nuevo nombrado y servicio de carga balanceada.

El modelo de componentes Microsoft es a la vez flexible y restringido. Es flexible porque soporta una variedad de lenguajes de programación (Java, C++, Visual Basic) pero son restringidos porque sólo corren sobre plataformas Windows. No obstante, esta tecnología de componentes soporta un gran y creciente mercado de componentes, y es considerado por muchos conocedores como el mejor modelo de componentes pese a sus restricciones de entorno de ejecución.

Mientras que CORBA fue propuesta por el OMG con la meta explícita de crear un sistema de objetos distribuidos, COM emergió por otras y variadas razones. Microsoft estaba intentando construir un sistema que permitiera a documentos de Microsoft Office de diferentes tipos ser enlazados y embebidos (empotrados) entre ellos para crear un “documento compuesto” llamado objeto de enlace empotrado (OLE). Esto requirió la definición de comunicaciones entre objetos (o interacciones) independientes de su aplicación y descritas solamente en términos de interfaces.

Mientras tanto Visual Basic (VB) introdujo Extensiones de Visual Basic (VBXs) para habilitar desarrollos y crear mejores interfaces gráficas de usuarios (GUIs), por ejemplo, dibujando elementos de GUI como los botones. Para resolver el problema de independencia de la implementación, los VBXs fueron encapsulados y adoptaron principios de objetos, pero añadieron el elemento adicional de empaquetar para realizar una distribución independiente y adaptable a el despliegue de cualquier sistema del que pudiera formar parte; la idea clave detrás del modelo extensible fue que terceros podían crear estas VBXs. Siguiendo principios de objetos, conformaron un modelo de componentes y pudieron ser empaquetados y distribuidos en forma binaria. Esta fue la primera vez que un modelo práctico soportaba la creación de componentes binarios para el reuso.

Posteriormente se generalizaron VBXs para otros ambientes y OCXs renombrados, fue allí cuando Microsoft reunió los modelos de los programas VBX/OCX y OLE bajo un nuevo modelo de objetos abstracto llamado COM (Component Object Model). COM se convirtió en una completa arquitectura de sistemas de componentes distribuidos con el soporte de la distribución de componentes binarios y de despliegue.

Sun microsystems [comp-sun 2003]

La Sun Microsystems y vendedores aliados han desarrollado dos especificaciones de componentes: Java Beans y Enterprise Java Beans. Ambos son promovidos como extensiones del lenguaje de programación Java y sin embargo proporcionan una solución de programación con plataforma neutral, que está íntimamente ligada al lenguaje de Java, a continuación se expone un poco acerca de estos modelos de componentes.

Los Java Beans, son una arquitectura y modelo de software basado en componentes de plataforma neutral para construir usando componentes dinámicos Java; mezcla las

ventajas del desarrollo de software basado en componentes con el poder y la flexibilidad de la plataforma de red Java. Provee mecanismos para interfaces públicas, manejadores de eventos, persistencia, control de esquema y apoyo para la construcción de la aplicación.

En resumen, un Java Bean es, "*un componente de software reutilizable que puede ser manipulado visualmente por una herramienta de construcción*". Bajo esta premisa se puede entender como "componente": elementos GUI tipo botones o barras de desplazamiento, visualizadores de bases de datos, e incluso aplicaciones más sofisticadas tipo procesadores de texto u hojas de cálculo; y así mismo se entiende como "herramienta de construcción": un entorno de desarrollo visual, un editor de textos (se podría incluir un componente Beans dentro de nuestros documentos) y, por supuesto, un editor de páginas Web (para integrar los componentes Beans con la Internet).

Los Enterprise Java Beans (EJB) son un estándar de infraestructura en el lado del servidor que fue presentado en 1998 y ha evolucionado con apoyo de selectos colaboradores de Sun, es una de las muchas tecnologías que constituyen la especificación Java 2 Enterprise Edition (J2EE) y tiene la intención de difundir un multi-sistema de proveedor neutral para construir aplicaciones utilizando el lenguaje de programación Java. Define una infraestructura en tiempo de ejecución que incluye soporte de transacciones, seguridad y persistencia al igual que un modelo contenedor que el programador reconoce independientemente de con cual compilador ha sido desplegado. Tiene las mismas ventajas del CCM (Corba Component Model), con la diferencia de que es una especificación solamente Java, EJB es también el lenguaje de mapeo para componentes CORBA.

OMG [comp-corba 2002]

La OMG, creadora de la especificación CORBA2, habilitada en 1996, define una infraestructura de *middleware* que puede ser utilizada para integrar componentes. Un número de productos han sido producidos e implementados con la especificación CORBA2, incluyendo Inprise VisiBroker e IONA Orbix. Esta especificación también define el protocolo de Internet Inter-ORB (IIOP) que esta siendo usado ampliamente como un protocolo subyacente en EJB y el método de invocación remota RMI que soporta interoperabilidad entre diferentes servidores de aplicaciones.

La especificación CORBA3 es una extensión de CORBA2 y cuenta con una arquitectura de componentes llamada CORBA components model (CCM). Las características más relevantes de este modelo son:

- Un ambiente contenedor que empaqueta transacciones, seguridad, persistencia y provee una interfaz de resolución de eventos.
- Integración con EJB.
- Un formato de software distribuido que habilita el desarrollo del mercado de software CORBA components.

Una meta importante en CORBA3 es influenciar las habilidades de los programadores comerciales para que estos puedan producir aplicaciones de negocio que adquieran estos atributos automáticamente.

El CCM es un modelo de componentes del lado del servidor para construir y desarrollar aplicaciones CORBA; es muy similar a Enterprise Java Beans (EJB) porque utiliza un patrón de diseño estándar lo cual facilita su uso, permitiendo que una gran cantidad de código sea generado para reuso de forma que las funciones del sistema sean implementadas por el proveedor del contenedor (quien ofrece el entorno de ejecución del componente), en lugar del desarrollador de la aplicación.

CCM forma parte del CORBA 3.0 y sus componentes extienden el modelo de objetos de CORBA, lo que significa que los componentes son un nuevo meta-tipo en CORBA que extiende al meta-tipo “*interfaz*”, heredando todas sus características e introduciendo otras nuevas. Los componentes, favorecen y aconsejan la composición en vez de la herencia. La herencia hace que una clase quede ligada en el diagrama de herencia y de clases sin exponer de forma explícita cuales son los requisitos de esa clase (otras clases que necesita, etc.), lo cual dificulta su reuso y portabilidad.

CCM está basado en EJB (Enterprise Java Beans) y en COM (Component Object Model), y esto especialmente visible en la estandarización de:

- Los servicios ofrecidos a las implementaciones de componentes.
- Despliegue (a través de un servidor de componentes).
- Implementación de múltiples interfaces (como en COM).
- Todo esto ofrecido a través de una interfaz “simple”.
- Además, no está limitado ni a Java (como EJB) ni a Windows (como COM), por lo que sigue al 100% la filosofía CORBA.
- Admite clientes que no entienden de componentes (toda la tradición CORBA anterior), con lo que se permite una migración cómoda y sin saltos hacia los componentes de CORBA.

CCM (o CORBA 3) añade ciertas características a CORBA, que son:

- El modelo de componentes, incluyendo extensiones del IDL, del IR (*Interfaz Repository*) y del ORB.
- El modelo de contenedor.
- El framework de implementación de componentes (CIF, Component Implementation Framework).
- El nuevo lenguaje de descripción de implementaciones de componentes (CIDL, Component Implementation Description Language).
- El modelo de empaquetado y de despliegue.
- Especifica un contenedor basado en CORBA para albergar EJBs. Esto es esencial para proveer dos caminos de interoperabilidad entre EJBs en Java y CCMs en cualquier lenguaje de programación CORBA.

Con todos los modelos de componentes presentados, se tuvo una ardua labor de selección dadas las ventajas y dificultades que ofrecían cada una de ellas, y aunque el

modelo más recomendado a seguir sería el CCM, este fue descartado debido al poco soporte que actualmente ofrece para aplicaciones prácticas que trabajen utilizando tiempo real.

Es por lo anteriormente expuesto que el modelo de componentes de Java fue el adoptado por el proyecto, no sólo por la interacción natural que ofrecía para trabajar con RTJ (Real Time Java) sino por su facilidad para exponer sus interfaces de gestión a través de JMX, lo cual se constituyó en una verdadera pasarela de intercomunicación entre los diferentes módulos del sistema tanto para la invocación de métodos como para procesos de gestión en tiempo real sobre el protocolo de señalización No. 7, ahora se presentará un poco más acerca de las mencionadas extensiones para gestión de Java.

3.3 JAVA MANAGEMENT EXTENSIONS (JMX)

3.3.1 Gestión de redes basada en componentes CORBA

[gest-corba 1999]

CORBA ha cobrado gran importancia en la gestión de redes debido a la tendencia a utilizar una sola arquitectura para el desarrollo, uso y gestión de un sistema distribuido; por lo anterior no existe una diferencia fundamental entre las aplicaciones de gestión y cualquier otra aplicación distribuida.

Para modelar datos de usuario o datos de gestión se utilizan los mismos métodos y la mayoría de las herramientas de desarrollo de sistemas distribuidos sirven para la implementación de aplicaciones de gestión, y en cualquier desarrollo se utilizan los mismos medios de comunicación, con todo lo cual se reducen costos, se aumenta la eficiencia y eficacia, y se hace más sencillo el mantenimiento.

Las facilidades CORBA para el área de gestión de sistemas permiten la posibilidad de integrar los servicios de gestión en la OMA (Object Management Architecture); mientras tanto estos y nuevos servicios son especificados, principalmente, por el X/Open Systems Management Working Group (XoTGsysMan), aunque es a partir de CORBA2 donde se empezó a incluir protocolos inter-ORB para ofrecer interoperabilidad total, lo cual es esencial para una gestión integrada.

Para el desarrollo de clientes para gestión se utilizan las invocaciones dinámicas a través del repositorio de interfaces, aunque conllevan una mayor complejidad en la implementación de los clientes, pero esto nos ofrece una gran flexibilidad lo cual es una gran ventaja al trabajar con sistemas de gestión ya que se puede utilizar un mismo cliente para acceder a nuevos servidores, como podría ser el caso para acceso a MIBs (Management Information Base) definidas en tiempo de compilación y MIBs definidas en tiempo de ejecución, todas accedidas por el mismo cliente a través de invocaciones dinámicas.

En CORBA las aplicaciones de gestión usan clientes con solicitudes asincrónicas (o sincrónicas retrasadas) para poder ejecutar otras tareas sobre el mismo hilo mientras la solicitud es atendida y respondida por el servidor; pero esto conlleva la mayor restricción para crear sistemas de gestión basados en CORBA con restricciones de Tiempo Real, ya que los tiempos de respuesta a dichos tipos de invocaciones no pueden ser determinísticos.

Al final CORBA básicamente soporta la intercomunicación de componentes de una aplicación de gestión. El modelo de gestión con CORBA está orientado al nivel de servicios y de red, para la gestión de elementos de red utiliza interfaces Q3 (dominio OSI) siguiendo las especificaciones del grupo JIDM (Joint Inter-Domain Management), y así poder comunicarse con otras plataformas TMN (Telecommunication Management Network).

El principal medio de comunicación para soportar los reportes de eventos sobre un objeto (o componente) en CORBA es el servicio de eventos que posee dos filosofías de operación para el envío de los mensajes de manera asíncrona, a saber:

- Push model: Donde el proveedor (servidor-elemento) genera un mensaje hacia el consumidor (cliente-gestor) cuando se presenta un evento.
- Pull model: Donde el consumidor solicita al proveedor la notificación de los eventos ocurridos.

Para realizar esta comunicación se utilizan los canales de eventos para que los objetos (consumidor y proveedor) puedan estar desacoplados; los canales en sí mismos son objetos que según sea necesario hacen las veces de proveedor o consumidor virtual para su contraparte.

3.3.2 Gestión de Redes Basada en componentes Java

[jdmk-sun 2003]

El Java Dynamic Management Kit (JDMK), es la herramienta de Java que permite a los desarrolladores crear rápidamente agentes inteligentes basados en Java Management Extensions (JMX). El poder del JMX Framework es que soporta un protocolo de acceso múltiple para gestionar información residente en el agente.

Utilizando el JDMK se puede desarrollar un agente SNMP basado en una MIB específica. El agente resultante SNMP puede inicialmente ser desplegado como un agente tradicional SNMP y ser actualizado fácilmente utilizando la flexibilidad de la tecnología JDMK. Cuando un agente basado en JDMK es gestionado por un gestor SNMP, este gestiona dinámicamente capacidades de permanencia opacando al gestor SNMP, sin embargo, pueden fácilmente desarrollarse nuevas aplicaciones utilizando el software JDMK, integrando los dos mundos.

3.3.2.1 Servicios de Gestión Dinámica

El agente inteligente reside en una librería de servicios de gestión, incluidas en el JDMK. Los servicios proveen funcionalidades como persistencia, monitoreo, programación, descubrimiento, carga y lanzamiento de MBeans, filtraje y servicio de proxy. Los desarrolladores pueden adicionar servicios de gestión a un agente conectándolo en el MBean de su propio servicio.

Los MBeans pueden ser también descargados de un servidor web y conectados a un agente en cualquier momento, independiente de la aplicación de gestión. Carga dinámica de clases significa que los servicios futuros y aplicaciones pueden ser cargados en ejecución y sin tiempo de reposo una vez que han sido desarrollados en la red.

3.3.2.2 Multiprotocolo de gestión de red

Las aplicaciones de gestión confían en un API de gestión para proveer una interfaz estándar a los agentes y permanecer independientes del protocolo. Los adaptadores de protocolo y conectores implementan esta API y se comunican con agentes a través de un protocolo de red establecido. El JDMK provee adaptación a la mayoría de los protocolos (HTTP/TCP o HTTP/SSL, RMI y SNMP) para ser desarrollados en cualquier entorno de red.

Adicionalmente JDMK provee una herramienta SNMP que permite generar MBeans que soportan MIBs comunes. Esto permite integrar la nueva tecnología basada en agente en infraestructuras de redes existentes, incluyendo la mayoría de los productos de gestión basados en SNMP

3.3.2.3 Beneficios para desarrolladores

- Rápida integración de los componentes de gestión a sistemas existentes, aplicaciones y dispositivos de red.
- Compatibilidad con cualquier herramienta JavaBeans.
- Desarrollos MBean y herramientas de depuración incluyendo un compilador SNMP y MIB.
- Idénticos agentes y APIs de gestión, las cuales simplifican diseños y permiten reuso de código.
- Escalabilidad dinámica, permite que los servicios sean desplegados como fueron desarrollados.

En general, gracias a la gran cantidad de ventajas ofrecidas por JMX, se decidió trabajar sobre esta arquitectura y utilizar su API para la creación de sistemas de gestión dinámica, donde el agente desarrollado para la gestión sobre SMART III puede descubrir dinámicamente las características gestionables de los componentes del sistema, esto es pensado para la ampliación de las capacidades de gestión del protocolo SS7 dentro de la plataforma y para incluir las características gestionables de los diferentes servicios que puede soportarse sobre la plataforma. Además es

relevante mencionar nuevamente la perfecta integración que nos ofrece JMX, por su desarrollo en Java, con las características de RTJ para poder realizar gestión del servicio y de la red en tiempo real.

3.4 GESTION EN TIEMPO REAL

El concepto de gestión en tiempo real ha sido un poco mitificado gracias a la división que presenta la gestión en la pirámide de TMN (Telecommunications Management Network), ya que a la mayoría de los servicios se les realiza gestión independientemente de la gestión de red, paradigma que ha creado un mundo dual donde la gestión del servicio se ha caracterizado por ser en tiempo promedio y la gestión de red ha sido en tiempo real.

Una de las premisas de SMART III es poder realizar gestión del servicio, en especial su control de calidad de servicio (QoS – Quality of Service) , a través de la gestión de parámetros de la red, como por ejemplo el tipo de compresión de datos sobre la línea, o el número de canales activos en un mismo enlace de comunicación.

Es por lo anterior que en el proyecto se tuvo la necesidad de buscar una solución sólida que brindara una salida robusta para nuestras necesidades de integración entre la gestión de servicio y la gestión de red; la gestión de la red está centrada en las capacidades que nos ofrecen los mensajes de nivel ISUP para controlar los parámetros de la red que soporta un servicio, pero dichas capacidades deben ser controladas y atendidas con restricciones de tiempo real para no afectar la operación del sistema como tal y en general los equipos de la PSTN que interactúan con la plataforma.

El primer intento fue el trabajo de gestión en tiempo real a través de las capacidades de gestión de CORBA y sus características para comunicación en tiempo real; los canales desacoplados que ofrece CORBA2 nos servían perfectamente para cumplir las necesidades de comunicación con el sistema de gestión del servicio, pero no nos permitían determinar los tiempos de respuesta para su actividad sobre la gestión de red. Desde este punto se trabajó en el estudio de las características de CORBA3 para atender invocaciones en tiempo real a través de canales dedicados, sin embargo el simple hecho de compilación e instalación de la herramienta de desarrollo para este tipo de aplicaciones se constituyó en un verdadero obstáculo que se debía haber superado de alguna manera sino hubiera surgido una nueva alternativa, RTJ.

RTJ es una especificación relativamente nueva, cuya implementación aún está en fase de maduración, dado que a pesar de estar diseñado para funcionar como soporte para aplicaciones de tiempo real, aún sigue siendo una derivación de Java2 y trata de mantener su lema “WORA” (Write Once, Run Anywhere) lo cual ha generado algunos problemas un poco complicados cuando se trata de migrar el *hardware* de sistemas de tiempo real duro; a pesar de lo anterior esta plataforma resultó ser un excelente aliado para realizar todas las comunicaciones entre elementos de Java y el

S.O.T.R., además que los problemas anteriormente mencionados pueden ser eliminados al escribir el código para una arquitectura de PC predeterminada, por su puesto el WORA deja de existir, pero entra a funcionar la eficiencia y seguridad de un H.R.T.S. (Hard Real Time System) como se necesitaba en el proyecto.

4 ARQUITECTURA SMART III Y SU GESTION BASADA EN SS7

4.1 ARQUITECTURA SMART III

La arquitectura SMART III es el resultado de un arduo trabajo de diseño *hardware* y *software* donde se actualizó la plataforma SMART para suplir las necesidades del mercado de las telecomunicaciones y en especial las del cliente en relación a los servicios que puede acceder a través de su línea telefónica.

El diseño de dicha arquitectura se desarrolló basado en el modelo de despliegue que se presenta en la figura 4.1, en ella se pueden apreciar los componentes macro para la prestación de un servicio, ellos son:

- SSP: El cual es el punto de conmutación del servicio y que en este gráfico corresponde a una central telefónica con líneas RDSI y señalización No7.
- USER: El cual es el equipo terminal del usuario que accede al servicio ofrecido.
- IP-SCP3: Este es el componente que contiene la lógica del servicio a prestar y se constituye de los equipos del proveedor de tercer orden del servicio.
- SCP1-SMS: Este componente corresponde al proveedor de servicios de primer orden y contiene la lógica del servicio para acceso a múltiples servicios dentro del mismo dominio (el dominio SMART III en este caso).
- SSP-SS7: Este último componente es el corazón del modelo de despliegue y se constituye en el sistema empotrado que permite monitorear, procesar e insertar mensajes de señalización No.7 para poder mantener el control sobre las llamadas RI de los usuarios de sus servicios.

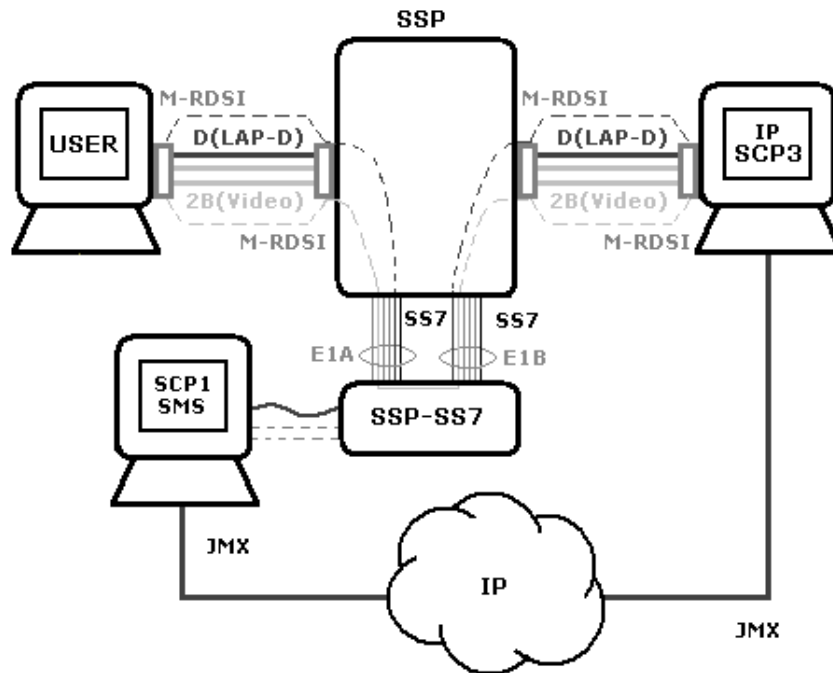


Figura 4.1 Modelo de despliegue del proyecto SMART III

El presente modelo de despliegue cubre los aspectos básicos de prestación de un servicio multimedial cualquiera para un cliente con acceso a la PSTN a través de una línea RDSI, como se presenta a continuación:

El usuario inicia el servicio marcando un número de RI conocido como el número del proveedor del servicio, este es interpretado por la central telefónica, la cual se encarga de re-enrutar la información de esta llamada al módulo de señalización y control (SSP-SS7), una vez iniciada la comunicación se puede iniciar la tarificación de acceso al servicio.

El SSP-SS7 se encarga de establecer la comunicación directa entre el usuario y el módulo de gestión (SCP1 SMS), por medio del servicio suplementario de RDSI denominado “señalización usuario a usuario” (UUS), obteniendo la información que transmite el usuario por el canal D, de esta manera el usuario puede comunicarse con el módulo de gestión, enviando mensajes de tipo UUS3 los cuales son recibidos por el módulo de gestión a través del módulo de señalización y control. De esta manera el usuario puede autenticarse con el módulo de gestión y recibir la lista de servicios a los cuales puede acceder de acuerdo a su perfil, cuando el usuario seleccione el servicio, el módulo de gestión se comunica con el módulo proveedor de servicio (SCP3) a través de JMX, realizando una validación del usuario y obteniendo la lista de opciones a las cuales puede acceder el usuario de acuerdo a su perfil, y así se inicia la tarificación del servicio propiamente dicho.

Cuando el usuario selecciona la opción que desea, se finaliza la comunicación establecida entre el usuario y el proveedor de servicios y entre el usuario y el módulo de gestión (aunque esto no es perceptible por el usuario), y el proveedor de servicio

se encarga de inicializar una nueva llamada al usuario para el envío de la información de servicio, para que sea reproducida en el lado del usuario. Así que se establece una nueva comunicación, donde se reestablece la comunicación entre el usuario y el módulo de gestión, ya sea con 1 o 2 canales de voz, de acuerdo al perfil del usuario y se inicia el envío de la información de servicio desde el servidor hacia el usuario, esto se hace necesario para poder mantener el control y monitoreo de la señalización ISUP que se utiliza en dicha comunicación y sobre ella realizar las diferentes actividades de gestión requeridas.

Durante la prestación del servicio, el usuario podrá solicitar gestionar el ancho de banda de la conexión a través del canal D, esta información llega al módulo proveedor del servicio, el cual se encarga de realizar la solicitud específica hecha por el usuario. Si la solicitud es de gestionar el ancho de banda, el proveedor del servicio se encargará de hacer la solicitud de otro canal para la comunicación o de terminar alguno de los canales ya solicitados.

Cuando el usuario desee terminar el uso del servicio, le envía esta solicitud al proveedor del servicio, el cuál se encargará de liberar los canales que se encontraban asignados para la comunicación, liberar todos los recursos y notificar al sistema de gestión para así terminar el proceso de tarificación del servicio.

Además de este proceso básico y correcto existen otras variantes donde el usuario puede descartar el uso del servicio, o incluso que se presenten fallas al momento de conectar al proveedor del servicio, lo cual no genera costos para el usuario pero si alimenta las bases de datos estadísticas de la RI para mantener esos datos disponibles para el suscriptor (proveedor del servicio).

4.2 DESCRIPCIÓN FUNCIONAL DE LA ARQUITECTURA

La arquitectura funcional de SMART III está soportada enteramente por el modelo para RI y por la filosofía de componentes de Sun, los EJB, aunque esto genera dependencia del lenguaje, sólo fue una aproximación a la implementación, pero cualquier otro desarrollador podría haberlo hecho con C++ u otros lenguajes y modelos de diseño, siempre y cuando mantengan las interfaces JMX para la interacción de los diferentes módulos, estos módulos funcionales se pueden apreciar en la figura 4.2.

4.2.1 SCP (Service Control Point)

Punto de Control del Servicio: Contiene la lógica de los servicios de RI, tiene una capacidad de procesamiento en tiempo real y una disponibilidad comparable a la de una central telefónica. Debido a su demanda en la red requiere la capacidad de manejar volúmenes de tráfico.

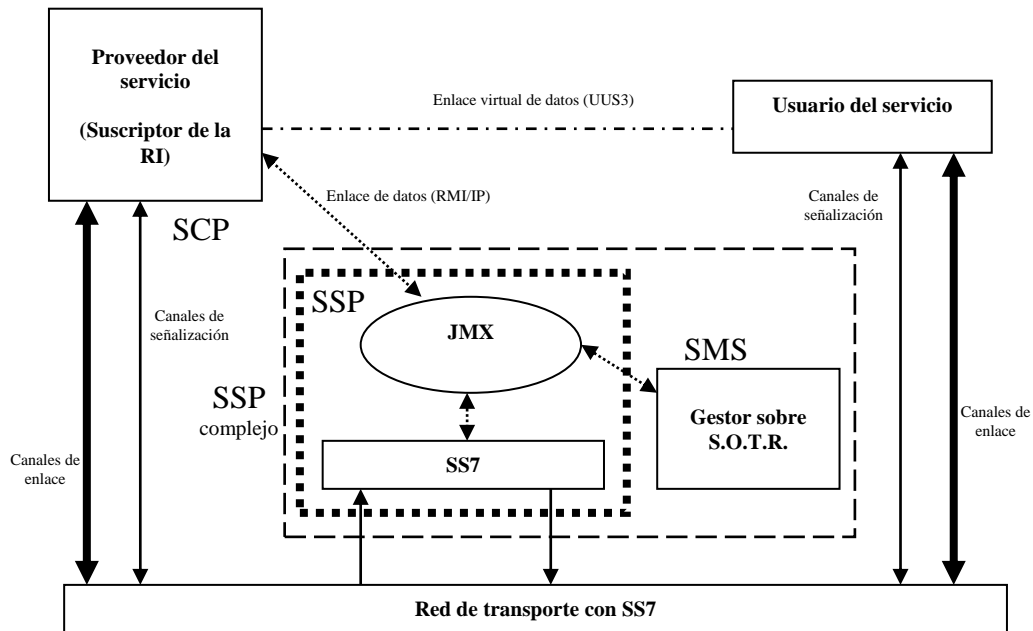


Figura 4.2 Arquitectura funcional

Este componente funcional además de cumplir con las características propias del servicio debe contar con una interfaz JMX para exponer sus características gestionables por parte del SMS, de forma que dichas capacidades puedan ser descubiertas de forma dinámica por el gestor de la plataforma. Es necesario además que el SCP contenga la lógica e interfaces *hardware* para capturar, procesar y generar mensajes UUS que sirvan como mecanismo de comunicación en tiempo real entre el suscriptor de la RI y el usuario del servicio.

Ahora se expondrán un poco más a fondo las capacidades para manejo de UUS que debe poseer el SCP:

4.2.1.1 Señalización de usuario a usuario

[UUS-rec]

El servicio suplementario señalización de usuario a usuario (UUS, user to user signalling), permite a un usuario de la RDSI enviar a otro usuario RDSI y recibir de éste una cantidad limitada de información por el canal de señalización, en asociación con una llamada al otro usuario RDSI ⁵.

⁵ Estos procedimientos son aplicables a la transferencia de información de usuario a usuario (UUI, user-to-user information) en asociación con un servicio de telecomunicación con conmutación de circuitos solamente

El servicio suplementario UUS es aplicable a usuarios que están conectados a una red por un acceso básico o por un acceso a velocidad primaria.

El servicio suplementario UUS permite al usuario enviar a otra interfaz usuario-red y recibir de ésta una cantidad limitada de información generada por el usuario. Esta información es transferida transparentemente (es decir, sin modificación del contenido) a través de la red. Normalmente, la red no interpretará esta información ni efectuará ninguna acción con respecto a la misma.

El usuario puede transferir información de usuario a usuario (UUI) en diferentes fases de la llamada, dependiendo del servicio o servicios a los cuales está suscrito. Estos servicios son:

Servicio 1

Se puede enviar y recibir UUI durante el origen y la terminación de las llamadas.

En el caso de una configuración punto a multipunto en el usuario llamado, se aplicarán las siguientes restricciones a la transferencia de UUI:

- En el sentido hacia adelante, el usuario llamante puede enviar UUI solamente como parte de la petición para originar una llamada, o como parte de la petición para terminar la llamada. Si el usuario llamante termina la llamada antes de que se establezca la conexión, sólo se enviará la UUI a los terminales que han indicado que el usuario ha sido informado de la llamada.
- En el sentido hacia atrás, el usuario llamado puede enviar UUI al usuario llamante con la indicación de que el usuario llamado está siendo informado de la llamada. Será responsabilidad del usuario llamado evitar el conflicto. El usuario llamado puede enviar también UUI cuando responde a la llamada.
- Si el usuario llamado rechaza la llamada y la red recibe un solo rechazo del usuario llamado, se enviará al usuario llamante el motivo del rechazo y la UUI acompañante (si la hubiere).
- Si la red recibe múltiples rechazos del usuario llamado, con la misma razón o razones diferentes para el rechazo, la red seleccionará el primer rechazo recibido, con prioridad más alta, basado en el motivo o motivos del rechazo. El motivo del rechazo y la UUI acompañante (si la hubiere) del rechazo seleccionado se enviarán al usuario llamante.

Servicio 2

Se puede enviar y recibir UUI después que el usuario llamante ha recibido una indicación de que el usuario llamado está siendo informado de la llamada y antes del establecimiento de la conexión. La UUI enviada por el usuario llamante antes de recibir la aceptación de la llamada por el usuario llamado, se enviará, como una opción del proveedor del servicio, al usuario llamado después que la llamada ha sido establecida.

El servicio es aplicable solamente si existe una configuración punto a punto en el lado del usuario llamado.

Servicio 3

Se puede enviar y recibir UUI solamente mientras se establece la conexión. Se imponen también limitaciones al número de mensajes por unidad de tiempo para el servicio 3. El control de flujo de cada sentido funciona independientemente.

Los servicios 1, 2 y 3 permiten la transmisión de 128 octetos por mensaje⁶.

Como mecanismo de verificación, parte del proyecto se centró en el desarrollo de un servicio multimedial para controlar sus características de QoS y su control básico de llamada utilizando la señalización No.7, este servicio fue el VoD (Video on Demand – Video por demanda), el cual se basó en una variante de TINA, su modelo computacional se aprecia en la figura 4.3:

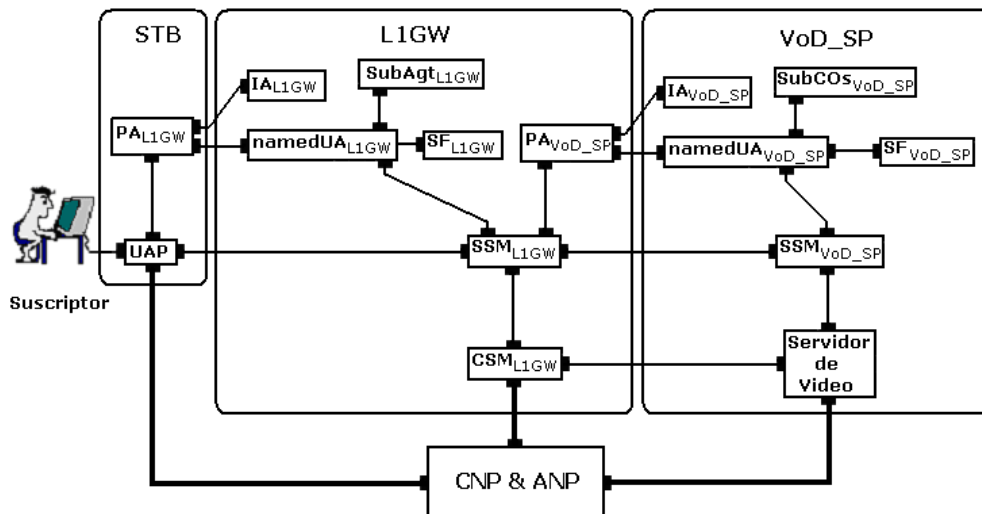


Figura 4.3 Modelo computacional del VoD (vista del usuario)

En el anterior modelo computacional se pueden apreciar los componentes *software* que constituyen el sistema para la prestación del servicio desde el punto de vista del usuario, sin embargo existen otros componentes para la gestión del mismo desde el proveedor del servicio y la interfaz JMX para la gestión desde el SMS, los cuales se pueden apreciar en las siguientes gráficas (figura 4.4 y figura 4.5):

⁶Durante un periodo de tiempo provisional, algunos proveedores de servicio pueden admitir transferencia de 32 octetos de UUI solamente en el servicio 1.

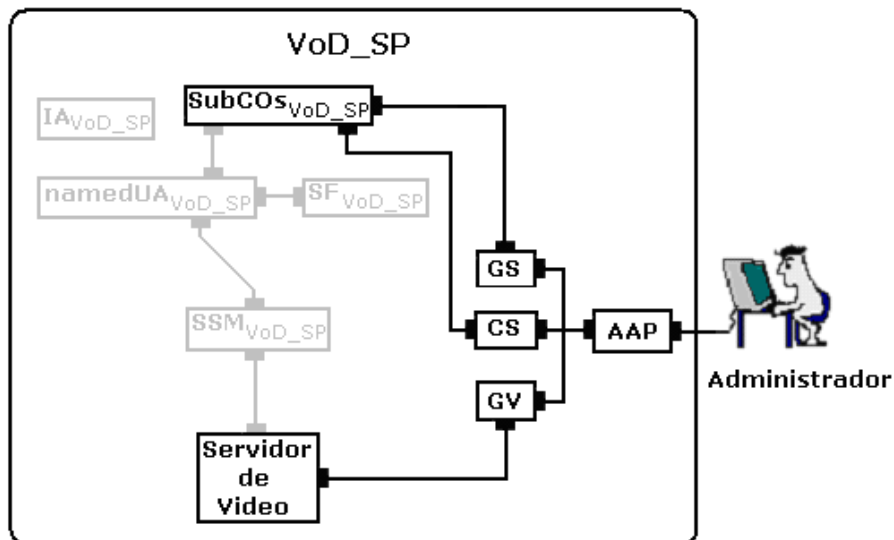


Figura 4.4 Modelo computacional de VoD (vista del administrador)

La interfaz JMX para gestión se centra en el acceso al servicio ya que el control de la reproducción y la QoS para satisfacer las necesidades del usuario se realiza a través de la señalización usuario-usuario nivel 3 (USS3) para garantizar su respuesta en tiempo real, de otra manera se tendría que integrar el SCP y el SSP para garantizar la atención en tiempo real utilizando JMX sobre RTJ.

En la imagen siguiente (figura 4.5) se puede apreciar la interfaz JMX desplegada desde un explorador web, lo cual también es otra de sus ventajas en despliegue para facilitar el acceso a los recursos gestionables con http.

Para finalizar el modelo computacional total del sistema se puede apreciar en la siguiente figura 4.6, donde se puede encontrar todos sus componentes *software*, como se exponen a continuación:

UAP (User Application, Aplicación de usuario): A través de una adecuada interfaz de usuario, permite que los usuarios del servicio utilicen las capacidades del Agente proveedor (PAL1GW), tal como autenticarse para iniciar el servicio y así configurar una sesión de acceso con un Agente de Usuario (namedUAL1GW) y recibir la lista de contenidos a los que tiene acceso de acuerdo a su suscripción, o finalizar la prestación del servicio terminando los procesos inicializados y liberando recursos. Además le permite al usuario que ha inicializado el servicio, utilizar las capacidades del SSML1GW, tales como seleccionar un video de la lista de contenidos, controlar la reproducción del video seleccionado y modificar la calidad de reproducción del mismo.

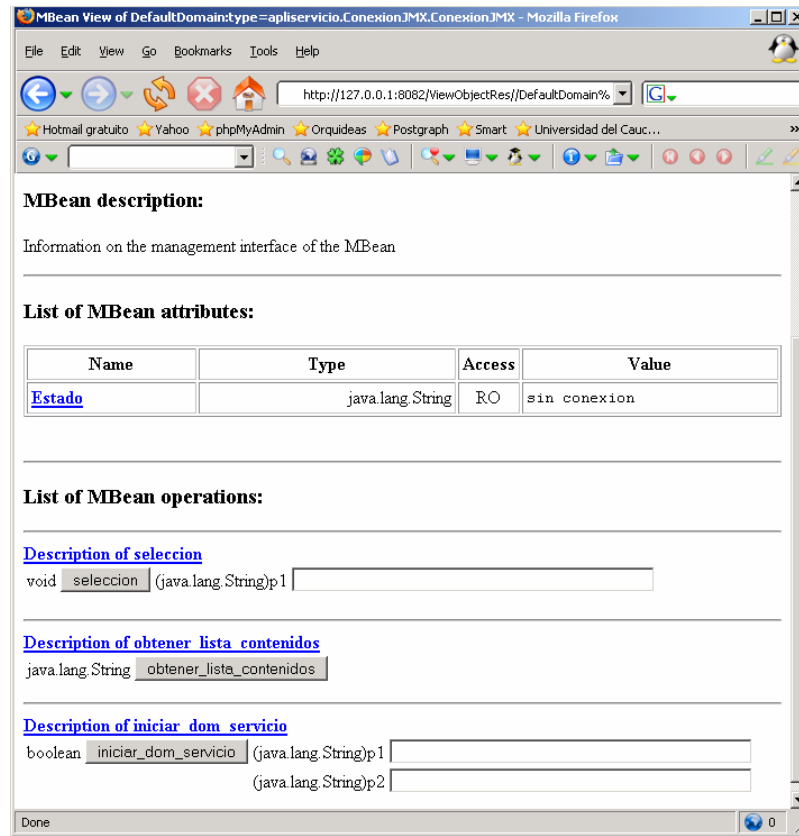


Figura 4.5 Interfaz JMX para acceso al servicio

PAL1GW (Provider Agent, Agente proveedor): Soporta la configuración de una relación confiable entre el usuario y el proveedor del servicio, interactuando con un agente inicial (IAL1GW) y siendo referenciado a un agente de Usuario (namedUAL1GW). Igualmente permite la finalización de la sesión establecida entre el usuario y el proveedor del servicio.

IAL1GW (Initial Agent, Agente Inicial): Permite la autenticación del usuario con el proveedor del servicio interactuando con el PAL1GW, permitiendo así una relación confiable entre el usuario y el proveedor. Igualmente, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

namedUAL1GW (Named User Agent, Agente de usuario de nombrado): Representa a un usuario en el dominio proveedor, asegurando una relación confiable entre el usuario y el proveedor. Permite controlar y manejar (crear/ suspender/reiniciar) una sesión de servicio interactuando con el SF para crear un SSM y permite el acceso a la información de configuración de usuario suministrada por el SubAgtL1GW. De la misma forma, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

SubAgtL1GW: (Subscription Agent, Agente Suscriptor): Permite acceder a la información de suscripción de un usuario particular y cuando el usuario finaliza la sesión de servicio liberar los recursos utilizados.

SFL1GW (Service Factory, Factoría de servicio): Por solicitud del namedUAL1GW instancia una sesión de servicio creando los objetos de sesión SSM necesarios para iniciar la comunicación con el proveedor de servicio de video por demanda, igualmente permite la finalización de esta sesión y liberar los recursos.

SSML1GW (Service Session Manager, Sesión de gestión de servicio): Comprende la especificación del servicio y los segmentos de control de sesión genéricos de la sesión de servicio de usuario. Representa y mantiene el contexto del usuario en una sesión de servicio, almacenando la información y capacidades locales del usuario. Es Instanciado por el SFL1GW cuando el usuario solicita el servicio y es eliminado cuando el usuario abandona la sesión del servicio. Durante la sesión del servicio envía las solicitudes de control, realizadas por el usuario a través del UAP, al SSMVoD SP, permitiendo seleccionar un video de la lista de contenidos y controlar la reproducción del video seleccionado. Si la solicitud del usuario es la de modificar la calidad de reproducción, se comunica con el CSML1GW que a través del proveedor de conectividad se encarga de modificar la configuración del enlace para incrementar o disminuir la calidad de reproducción del video solicitado por el usuario.

CSML1GW (Communications Session Manager, Gestor de sesión de comunicación): Se encarga de gestionar los flujos punto a punto entre el usuario y el servicio de video por demanda, para esto se comunica con un módulo externo (proveedor de conectividad) que se encarga de reservar los canales de comunicación del proveedor de red y establecer el camino. Este provee interfaces que permite al SSM configurar, modificar y remover los flujos de unión.

PAVoD SP: Soporta la configuración de una relación confiable entre el proveedor del servicio y el proveedor del servicio de video por demanda, interactuando con un agente inicial (IAVoD SP) y siendo referenciado a un agente de Usuario (UAVoD SP). Igualmente permite la finalización de la sesión establecida entre el proveedor del servicio y el proveedor del servicio de video por demanda.

IAVoD SP: Permite la autenticación del usuario con el proveedor del servicio de VoD interactuando con el PAVoD SP, permitiendo así una relación confiable entre el proveedor del servicio y el proveedor de servicio de VoD. Igualmente, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

namedUAVoD SP: Representa a un usuario en el dominio del servicio, asegurando una relación confiable entre el usuario y el proveedor del servicio de video por demanda. Permite controlar y manejar (crear/ suspender/reiniciar) una sesión de servicio interactuando con el SFVoD SP para crear un SSMVoD SP y el acceso a la información de configuración de usuario suministrada por el SubAgtVoD SP. Igualmente, cuando el usuario finaliza la sesión de servicio libera los recursos utilizados.

SubCOsVoD SP (Subscription Computational Objects ,Objetos Computacionales de suscripción): Se tienen los siguientes:

- **SubAgtVoD SP** (Subscription Agent ,Agente Suscriptor): Permite acceder a la información de suscripción de un usuario particular y cuando el usuario finaliza la sesión de servicio liberar los recursos utilizados.
- **SubTH VoD SP** (Subscription Template Handler, Manejador de la plantilla de suscripción): Se encarga de almacenar la configuración inicial del sistema que es definida por el administrador a través del GC (gestor de configuración).

SFVoD SP: Por solicitud del namedUAVoD SP instancia una sesión de servicio creando los objetos de sesión SSMVoD SP necesarios para la prestación del servicio de video por demanda, igualmente permite la finalización de esta sesión y liberar los recursos.

SSMVoD SP: Comprende la especificación del servicio y los segmentos de control de sesión genéricos de la sesión de servicio de usuario. Representa y mantiene el contexto del usuario en una sesión de servicio, conteniendo la información y capacidades locales del usuario. Es Instanciado por el SFVoD SP cuando el usuario solicita el servicio y es eliminada cuando el usuario abandona la sesión del servicio. Durante la sesión del servicio envía las solicitudes de control, realizadas por el usuario a través del UAP y recibidas por el SSM LIGW, al Servidor de video, permitiendo seleccionar un video de la lista de contenidos y controlar la reproducción del mismo.

Servidor de video: Almacena los diferentes videos a los que pueden acceder los usuarios y de acuerdo a solicitudes del SSM VoD SP permite seleccionar un video de la lista de contenidos y controlar la reproducción del mismo. Se establece un camino directo entre este y el usuario, por el que se envía el video seleccionado.

GUI_AAP (Administrator APlication, Aplicación de Administrador): Interfaz que le permite al administrador hacer uso de las capacidades que le provee las diferentes clases de gestión como lo es agregar y eliminar un video en la videoteca, suspender, eliminar o suscribir un usuario y establecer la configuración inicial del sistema. Para esto se comunica con los componentes que se encargan de la prestación del servicio, para que actualicen la nueva configuración.

GS (Gestor de Suscripción): Le permite al administrador del sistema a través de la GUI_AAP gestionar el ingreso de los usuarios al servicio, permitiéndole suspender, eliminar o suscribir a los usuarios. Para esto se comunica con los componentes encargados de la prestación del servicio de video por demanda, para que permitan el ingreso o no a los usuarios al sistema con la configuración correspondiente para cada uno de ellos.

GC (Gestor de configuración): Le permite al administrador a través de la GUI_AAP, suministrar las características iniciales de configuración del sistema. Para esto se comunica con el componente SubTHVoD SP para que almacene esta información.

GV (Gestor Videoteca): Le permite al administrador a través de la GUI_AAP, gestionar la lista de los videos que hacen parte de la videoteca virtual, permitiéndole agregar o eliminar videos.

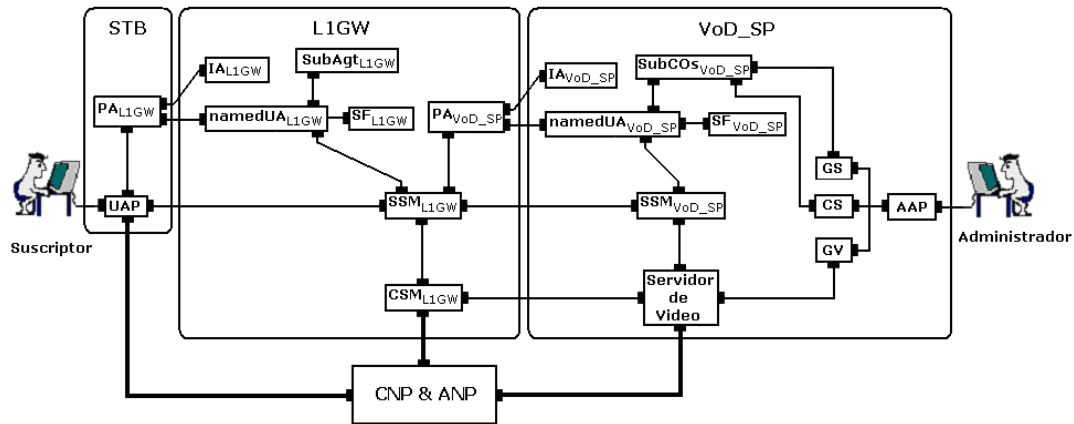


Figura 4.6 Modelo computacional total de VoD

4.2.2 SSP (Service Switching Point)

Punto de Conmutación del Servicio: Equivalente al SSP-SS7 del modelo de despliegue (figura 4.1) se comporta como una “CENTRAL SMART” con la funcionalidad de conmutación del servicio, tiene la capacidad de conectarse a una o varias centrales de telefonía publica. Debe ser capaz además de interactuar con los demás componentes de la RI a través de una interfaz JMX para exponer sus características gestionables y además soporta el manejo de la señalización No. 7, incluyendo la UUS3, para establecer las condiciones de operación de red y de tarificación del servicio.

Este componente es el corazón de la plataforma SMART III ya que sobre él se realiza todo el procesamiento de la señalización hasta el nivel ISUP; este módulo está constituido de dos componentes que soportan toda su funcionalidad, en primer lugar cuenta con una tarjeta CTI donde se procesan los niveles de MTP1 y MTP2, ya que son los que requieren mayor rigurosidad manejo de altas velocidades y tiempo real para el procesamiento de los mensajes de SS7; y también cuenta con un componente en *software* para el procesamiento de los mensajes “lentos” que se utilizan para el MTP3 e ISUP; se expone un poco más en detalle este módulo a continuación.

El sistema trabaja sobre una arquitectura *hardware* que realiza las funciones críticas soportada con un microprocesador que realiza las funciones lógicas que pueden no resultar tan críticas. Aunque la estructura *hardware* no pudo ser construida como parte del proyecto, en vez de esto fue adquirido un ambiente integrado que lo reemplazo por razones logísticas y económicas, pero si se pueden analizar las razones que llevaron a los diseñadores a tomar las decisiones de Codiseño apreciables en dicha tarjeta, antes de hacer la partición *hardware/software* del sistema.

Basados en el análisis de sistemas similares y de acuerdo a la experiencia que se ha tenido en el desarrollo de otros sistemas, se llega a la conclusión que las labores críticas de un sistema de telecomunicaciones deben tender a residir en las capas uno y dos que corresponden a las interfaces físicas de transmisión y a los protocolos de enlace y que además siempre resulta conveniente sintetizar en *hardware* aquellas labores que tienen un patrón constante es decir aquellas labores que comúnmente se conocen como “mecánicas y repetitivas”, puesto que resultan procesos con patrones en su mayoría determinísticos que facilitan la labor del *hardware*. Teniendo en cuenta que el nivel de enlace parte de un protocolo HDLC modificado (*High Speed Data Link Protocol*), resultaría muy rígido implementarlo totalmente en *hardware* lo cual lleva a la decisión de implementar sólo las labores comunes a todas las derivaciones del protocolo y que en la misma manera resultan ser aquellas labores repetitivas como son control de errores de entramado e inserción de banderas y ceros para evitar repetición de patrones clave. De allí en adelante se deja la responsabilidad de manejo del protocolo a entidades software que pueden correr tanto en una arquitectura monoprocesador o multiprocesador como es el caso.

A continuación (figura 4.7) se aprecia la división funcional del sistema y su correspondencia *hardware/software* dentro del sistema.

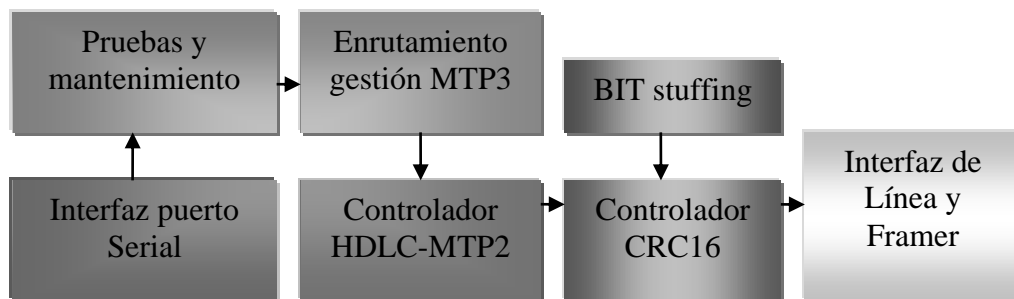


Figura 4.7 División Funcional del Sistema

Interfaz de línea y Framer

Entidad Hardware cuyos mecanismos permiten la generación del código de transmisión/recepción necesario para la comunicación con otra entidad del mismo tipo y el entramado paquetes para su correcta transmisión. El código analógico generado es de tipo HDB3, sirve para transportar tramas digitales PCM de 2048 MBPS. El formato de entramado es Dobletrama o Multitrama digital según sea el caso para el estándar Europeo.

Controlador CRC16

Esta entidad es una entidad hardware, pues aunque es factible su realización por software, reduciría el rendimiento de la máquina. Su función es generar un código de 16 bits calculado de acuerdo a un polinomio de chequeo de redundancia cíclica “ $x^{16} + x^{12} + x^5 + 1$ ” que se aplica a la trama y se transmite al final de esta y que servirá para determinar si la trama transmitida llegó sin errores a su destino [Tutorial-2003].

Controlador HDLC-MTP2

Esta entidad Hardware-Software esta encargada del nivel de enlace de las tramas SS7, el MTP-2 especificado por la ITU-T es un código HDLC modificado para una necesidad especifica y presenta el formato que se muestra en la figura 4.8. Está encargada de funciones *hardware* como inserción de banderas, inserción de ceros, comparación de direcciones, eliminación de tramas repetidas y funciones *software* como numeración de tramas, alineación inicial, retransmisión para corrección de errores, control de estado del enlace [SBEI-2000].



Figura 4.8 Formato de trama HDLC

Enrutamiento gestión MTP3

La entidad de enrutamiento y gestión correspondiente al nivel MTP 3 del SS7, es una entidad software encargada principalmente de lograr que los mensajes lleguen a su destino correspondiente y de llevar a cabo acciones que eviten la perdida permanente de mensajes por la caída enlaces de señalización para lo cual los posee mecanismos de gestión pertinentes. Dentro de sus funciones están, enrutamiento control de enlaces, control de congestión entre otros.

Pruebas y mantenimiento

La entidad de pruebas y mantenimiento es una entidad software encargada de generar y contestar mensajes para la verificación del correcto funcionamiento de cada enlace.

Interfaz puerto serial

La interfaz de puerto serial es una entidad *hardware/software* que trabaja sobre el protocolo RS-232 [DALLAS-2000], a 19200 baudios con corrección de errores por paridad. Esta velocidad es suficiente para el manejo de un solo enlace de señalización, lo cual es suficiente para suplir los requerimientos del proyecto. Se espera evolucionar a interfaces seriales más veloces que permitan manejar mayor volumen de información.

4.2.2.1 Subsistema ISUP

Descripción del ambiente hardware/software de trabajo

El sistema *hardware/software* en el que se desarrollo el subsistema ISUP se refiere a un ambiente PC convencional con un puerto serial COM y protocolo RS-232 modificado, monitor a color preferiblemente, teclado, mouse, y que posea buenas capacidades de procesamiento y la JVM (Maquina virtual de JAVA) montada sobre una plataforma operativa Windows que permita la priorización de procesos como lo

hacen las plataformas Windows derivadas de la arquitectura NT de Microsoft, las cuales permiten establecer prioridad de “Tiempo Real” para procesos críticos como lo es subsistema ISUP, o en el mejor de los casos sobre una plataforma Linux con capacidad de manejo de restricciones para tiempo real, como puede ser TimeSys Linux o Real Time Linux, todo lo anterior se puede expresar de manera visual con la figura 4.9.

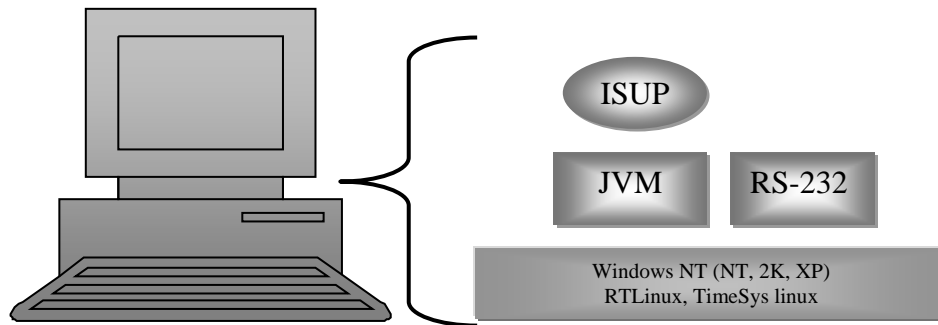


Figura 4.9 Ambiente de trabajo ISUP

Descripción en bloques del subsistema.

El subsistema ISUP se puede describir como un conjunto de entidades que interactúan entre si con el fin de proveer las funcionalidades propuestas por la ITU-T para SS7. Desde el diseño el subsistema se concibió con una arquitectura de despachador de paquetes, que usa el método de comunicación por transferencia de mensajes tal como se muestra en la figura 4.10, a continuación del párrafo, basados en la experiencia que se tienen de sistemas de tiempo real [Rendón-2003] aplicados a sistemas de telecomunicaciones.

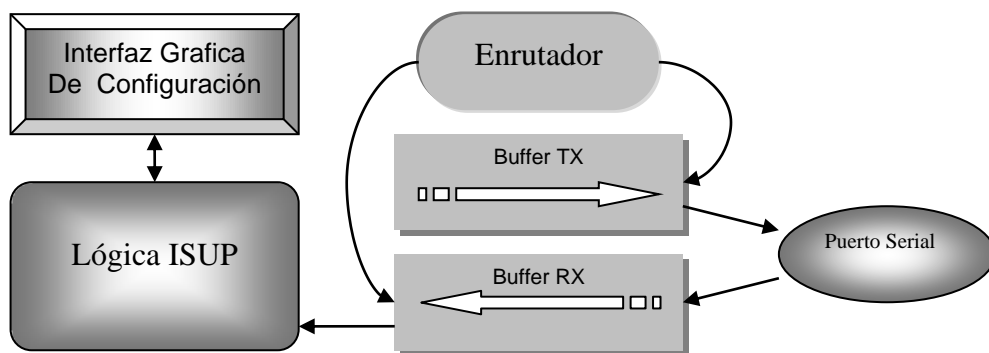


Figura 4.10 Diagrama en bloques funcionales subsistema ISUP

El subsistema desde el punto de vista Arquitectural puede ser concebido como un conjunto de bloques entidad interactuando entre ellos que permiten explicar su comportamiento de una manera relativamente sencilla.

Puerto Serial

Esta entidad funcional esta encargada de proveer los medios necesarios para lograr una comunicación fiable con el subsistema de transferencia de mensajes MTP3 e ISUP. Este proceso permanentemente esta estático a la espera de una interrupción por la llegada de datos desde la MTP o desde el modulo de lógica ISUP a través de los búfer. Dentro de las funciones que cumple el puerto serial están:

- Transrecepción a 115200 baudios.
- Corrección de errores con paridad.
- Numeración de tramas.
- Chequeo de errores.
- Corrección de errores por retransmisión.
- Inserción de ceros.

Enrutador

Basado en la arquitectura de despachador de paquetes, este módulo se encarga de enrutar los paquetes salientes de los buffers de transmisión y recepción a su destino correspondiente. Esta entidad se compone de un proceso dinámico (llamado hilo en java) que permanentemente esta chequeando la existencia de paquetes en la salida de los buffers, cuando obtiene un paquete de alguno de los dos buffers determina hacia donde se dirige el paquete y lo entrega a los sub-módulos correspondientes que se encargan de analizar y discriminar los paquetes y entregarlos a su destino correspondiente. Dentro de las funciones que cumple el modulo enrutador están:

- Enrutamiento.
- Discriminación.
- Transporte.

Buffers de Tx y Rx

Las características de un sistema de comunicación por paso de mensajes generalmente implican la presencia de buffers de transferencia para salvar los problemas de sincronismo debidos a la no coincidencia de velocidad entre los sistemas que se quieren comunicar. Sin embargo también es necesaria la existencia de un mecanismo que sincronice los eventos de acceso a los buffers para evitar lecturas incorrectas o pérdidas de mensajes dado el inconveniente que un proceso de lectura con uno de escritura estén accediendo directamente al recurso compartido, en este caso se usaron las funcionalidades propias de java para acceder a métodos sincronizados que eviten el acceso simultaneo a los buffers, estos fueron implementados como arreglos de vectores configurados como colas FIFO. Dentro de las funciones de los buffers están:

- Transporte.
- Almacenamiento.
- Sincronización.

Lógica ISUP

Este bloque funcional esta encargado de implementar las funcionalidades del protocolo ISUP recomendado por ITU-T adaptadas para dar soporte caso del negocio del proyecto SMART. Antes de entrar a explicar las funcionalidades del subsistema ISUP, se explicará brevemente los componentes que interactúan en la lógica ISUP tal como se ve en la figura 4.11, con el fin de facilitar el entendimiento del esquema funcional.

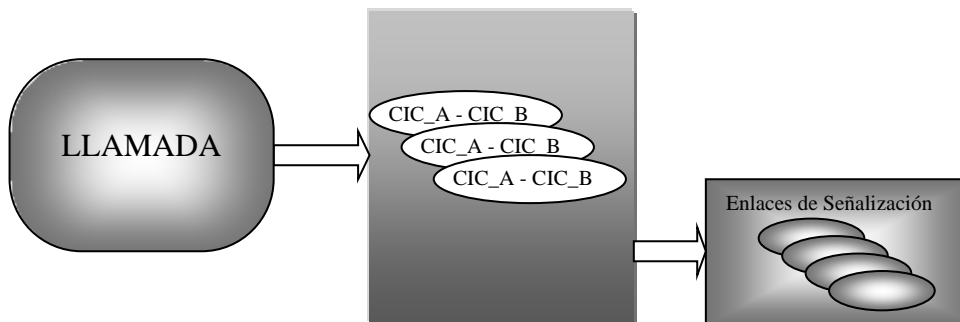


Figura 4.11 Componentes Lógica ISUP

Enlaces de señalización

El entorno *hardware* provee un soporte para cuatro enlaces simultáneos de señalización, para el caso sólo se utilizó uno pero la estructura ISUP se concibió de manera que fuera evolutiva. Por tal razón existen entidades que representan en forma lógica los cuatro enlaces de señalización y además definen sus parámetros y sus estados. Para que sean utilizados en el caso de una llamada ISUP.

- **Canales:** Los canales son entidades configurables que representan en forma lógica los parámetros del procedimiento ISUP-LOOPBACK, como se aprecia en la figura 4.12. Cada canal esta compuesto de dos CICs, los cuales representan dos circuitos correspondientes de dos El distintos que se encuentran en LOOP físico. Estas parejas de CICs deben ser configurados en tiempo de ejecución por el operador del sistema según un acuerdo bilateral con el operador de la central telefónica con la cual se comunica el sistema.
- **Llamada:** Es una entidad software encargada de activar los procedimientos especificados por ISUP, para ello utiliza la información almacenada en los enlaces de señalización, en los canales y otra información proveída en tiempo de ejecución por la entidad que activa los eventos de llamada (Eventos Gráficos, Eventos Interfaz de Gestión). Llamada es un elemento estático que depende tanto de los eventos generados dentro del caso del negocio de ISUP, como de los eventos generados por la llegada de mensajes ISUP provenientes la central telefónica con la cual se esta interactuando.

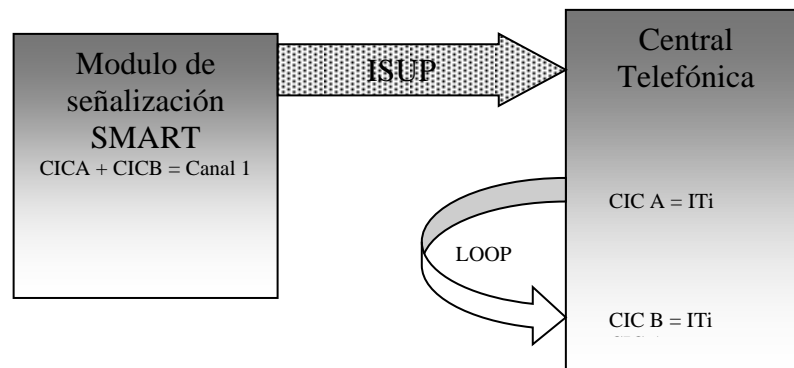


Figura 4.12 Procedimiento ISUP-LOOPBACK

Interfaz grafica de configuración

La interfaz hombre máquina esta hecha para permitirle al operador configurar las condiciones en las que trabajará el sistema que usa el subsistema ISUP y en general el modulo de señalización. En general la interfaz grafica permite crear y configurar canales, generar llamadas de prueba, asignar enlaces de señalización.

Descripción funcional del subsistema

El subsistema ISUP se puede describir como un conjunto de funcionalidades que trabajan soportados sobre la máquina virtual de JAVA, esta particularidad hace que el sistema sea portable entre sistemas operativos, pero por lo pronto sólo se considerará las plataformas Windows NT que nos brindan capacidades de priorización de procesos y nos permiten asignar prioridad máxima a nuestro subsistema ISUP para lograr un funcionamiento eficiente del sistema.

Partiendo del caso del negocio planteado por el proyecto SMART se hizo necesaria y conveniente la implementación del sistema dividiéndolo en las siguientes funcionalidades:

- Configuración de canales: Los CICs son los identificadores de los circuitos del sistema de señalización No 7 en una central telefónica, los canales son entidades compuestas de dos CICs principalmente y de sus estados. Para que el sistema funcione correctamente, deben existir los canales necesarios, correspondientes a los circuitos de los LOOPS puestos en la central telefónica y por tal razón deben existir los mecanismos necesarios para que el operador del sistema pueda crear y configurar canales de acuerdo a las necesidades de configuración del sistema usuario del módulo de señalización. Para proveer esta funcionalidad los canales como entidades software deben ser modificables por una interfaz gráfica de usuario o por una interfaz simple de línea de comando, que para el caso es la primera.

- Generación de llamadas: Esta funcionalidad es llevada a cabo por la entidad “LLAMADA” cuando una operación propia es activada por un agente externo con el fin de hacer pruebas. “LLAMADA” utiliza la información almacenada en las entidades ENLACE DE SEÑALIZACION Y CANALES para generar y aceptar los mensajes pertinentes para que sean transferidos hacia la central telefónica de destino y así conseguir una comunicación de voz o datos con un terminal específico de la central telefónica. A través de un circuito específico de una troncal determinada.
- Recepción de llamadas: Esta funcionalidad es atendida por la entidad “LLAMADA” cuando una operación de recepción de llamada es activada por causa de la recepción de un mensaje de inicio de llamada. “LLAMADA” debe responder adecuadamente de acuerdo a la información almacenada en “ENLACE DE SEÑALIZACION Y CANALES” para aceptar y generar los mensajes pertinentes para que sean transferidos hacia la central telefónica de destino y así conseguir una comunicación de voz o datos con un terminal específico de la central telefónica, a través de un circuito específico de una troncal determinada.
- Terminación de llamadas: Cuando la operación de terminado de un llamada es activada por causa de la recepción de un mensaje de desconexión o por causa de un agente externo; La entidad “LLAMADA” debe responder con la generación de los mensajes ISUP necesarios para completar la finalización exitosa de la llamada, de acuerdo a la información almacenada en “CANALES”. Esto con el fin de liberar los circuitos utilizados en la llamada que se esta llevando a cabo.
- Transito de llamadas: Mediante el procedimiento ISUP-LOOPBACK la entidad “LLAMADA” debe proveer los mecanismos necesarios mediante la generación de los mensajes ISUP pertinentes para lograr el transito de una llamada a través de un LOOP físico como se muestra en la figura 4.12. Es decir cuando se recibe una llamada entrante la entidad llamada debe recibirla y generar una nueva llamada por el circuito correspondiente de acuerdo los CICs almacenados en cada canal que previamente han sido configurados por el operador del sistema. Con esto el sistema se habrá comportado como una central tandem virtual. Este procedimiento es realizado con el fin de dar soporte al subsistema del caso del negocio, de modo que se logre conectar un usuario con un proveedor del servicio dentro de la misma central telefónica, pero manteniendo el control de su comunicación desde el sistema SMART a través del modulo de señalización.
- Transferencia de información usuario a usuario tipo 3: La información de usuario a usuario es un servicio suplementario RDSI soportado por el SS7, existen 3 tipos distintos de señalización usuario a usuario. El tipo 3 de este servicio es el de interés para el proyecto SMART y se define como el servicio suplementario que permite el intercambio de información punto a

punto entre dos usuarios, la característica del servicio de tipo 3 es que se efectúa cuando la conexión entre dos usuarios ya está establecida. La entidad “LLAMADA” tiene como función aceptar el requerimiento del servicio y empaquetar y desempaquetar información en mensajes especiales para su transporte dentro de la red SS7 hacia y desde la central de destino.

4.2.2.2 Interfaz de gestión con JMX para SS7

Este subsistema representa en sí el nivel de aplicación específico, para suplir los requerimientos del proyecto SMART III. Aprovechando las funcionalidades prestadas por el “Subsistema ISUP”, el cual se comporta como una centralita de tránsito, se montó sobre dicho subsistema una capa de aplicación que usará las funcionalidades prestadas por ISUP para implementar con ello las funcionalidades necesarias para suplir los requerimientos del proyecto SMART y así publicar estas funcionalidades sobre una interfaz abierta de gestión como JMX, de modo que las funcionalidades prestadas por el sistema puedan ser accedidas fácilmente por cualquier agente externo que use esta interfaz de gestión. Para el caso del proyecto SMART III el agente externo es un sistema de gestión que maneja la red de servicios de SMART III.

Capacidades de gestión de SS7

SS7 incorpora una gran cantidad de mensajes de gestión de señalización que son controlables desde la OMAP (Operation, Maintenance and Administration Part) y muchos más mensajes para realizar control de llamadas sobre RDSI. Estos mensajes que originalmente se concibieron para dar fiabilidad a los enlaces de señalización, y para ofrecer servicios RDSI, respectivamente, pueden ser utilizados para conocer el estado de la red de telecomunicaciones y de la mayoría de sus equipos terminales y de conmutación.

Si se desea conocer el estado de funcionamiento de un enlace E1 en particular se puede realizar la siguiente rutina a nivel de señalización: (Se supone que la señalización se está transmitiendo sólo por el IT16 y no a través de un E1 dedicado sólo a señalización).

**Enviar un CQM (Circuit group Query Message) para conocer el estado general del enlace.
Solicitar la toma de un IT con retorno para una llamada de prueba IAM (Initial Address Message).
Despachar el mensaje COT (Continuity Message) y CCR (Continuity Check request Message) para establecer y probar las condiciones de conectividad sobre la petición de llamada.
Enviar los mensajes de liberación de recursos para finalizar el proceso, REL (Release Message).**

Los mensajes anteriores son un bosquejo general de lo que se podría hacer para verificar el estado de operación de un enlace, podrían incluirse otros mensajes de consulta y verificación, y además entre cada uno de ellos se esperan mensajes de respuesta o confirmación, si alguno no llegara dentro de los tiempos determinados eso ya sería una respuesta del estado del enlace.

Para la realización de las pruebas de gestión sobre los enlaces de comunicación se debe tener un adecuado manejo de los posibles mensajes a utilizar, su lógica de operación y el sentido de las respuestas (o ausencia de estas) de acuerdo a la información solicitada; Aunque los mensajes de los niveles MTP (Message Transfer Part) e ISUP (Integrated Services digital network User Part) se utilizan para gestionar la señalización, se pueden utilizar esos mismos mensajes (en diferentes secuencias) para poder obtener datos de gestión de fallas sobre los equipos de la red, tarificación de servicios utilizados, manejo de tráfico sobre las rutas y estadísticas de desempeño. [ITU-T 1997c]

Los aspectos sobre los que se desarrolla la arquitectura propuesta son los orientados a la gestión básica de llamada RDSI o llamadas no RDSI (analógicas), para determinar sus condiciones de tarificación y estadísticas de desempeño (# de intentos de llamada, # de llamadas exitosas, # de llamadas cursadas, entre otros), y conocer las condiciones de tráfico de los enlaces, a nivel de MTP, donde se pueden obtener los datos concernientes al nivel de congestión de un enlace, uso de las líneas de desborde para tráfico y # de mensajes perdidos por congestión (entre otros) para conocer el aprovisionamiento y operación de los E1 de la red y sus interconexiones.

Por ejemplo se puede realizar una prueba de verificación sobre un terminal en particular, bloqueando un enlace sobre él, y solicitando asignación de un canal sobre su equipo, y aunque no se obtendrá una respuesta explícita de error se obtendrá los desbordes de los “timers de espera de respuesta” que al hacerlo en un ciclo repetitivo (p.e. 3 pruebas en 2 minutos) nos permitirá inferir que dicho terminal está inoperante por fallas de equipo. Al igual que esta simple prueba se pueden analizar datos de desempeño del sistema y aislar problemas potenciales que pueden ser verificados con pruebas como la mencionada y así poder mantener un mayor nivel de control y respuesta a fallas sobre el sistema.

Modelo de la plataforma JMX

Para la implementación de la interfaz de gestión abierta se ha analizado las capacidades que nos ofrecen arquitecturas como TINA, o modelos como CCM, en los cuales se encuentran grandes ventajas para la implementación de sistemas distribuidos abiertos para gestión basada en TMN; sin embargo dentro de estas referencias se vió la dificultad de manejar sistemas en tiempo real, como lo son los equipos de red de una empresa de telecomunicaciones, que normalmente operan sobre plataformas con SOTR (Sistema Operativo de Tiempo Real), pero al tener JMX como interfaz de acceso a la gestión se pueden implementar beans con RTJ (Real Time Java) [RTJ 2000] que nos permiten acceder a las capacidades de gestión de los MBeans de JMX pero actuando con restricciones de tiempo real, de forma que se facilita el acceso a la gestión sobre un SOTR.

Cabe anotar que los modelos basados en CCM nos ofrecen la ventaja de facilitar la migración de componentes *software* y elementos de red, pero para nuestro objetivo es mucho más relevante contar con un sistema que tenga capacidades de tiempo real para no alterar los flujos de datos sobre las redes de telecomunicaciones ya que una

falla sobre la señalización puede causar muchas pérdidas sobre la comunicación y por ende la tarificación de las llamadas y servicios.

MBeans para la gestión con SS7

Existe una arquitectura modelo, basada en un sistema empotrado, que permite el acceso a la información transportada por la red de SS7, y su comunicación se realiza a través de los Mbeans que al registrarse en el servidor de Mbeans ofrecen sus interfaces desde la plataforma JMX para ser accedidas por aplicaciones de gestión, esto se puede ver mejor en el diagrama de dicha arquitectura (figura 4.13).

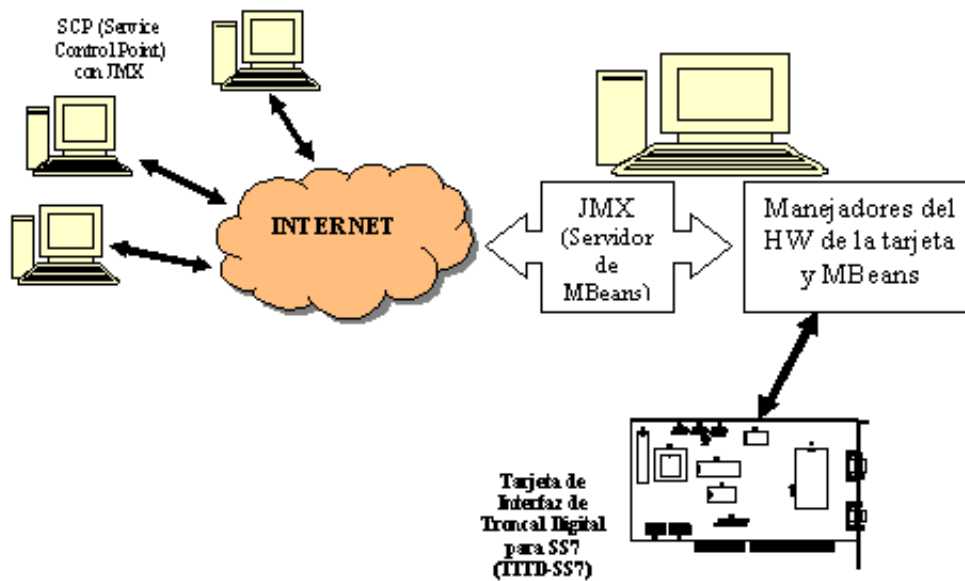


Figura 4.13 Arquitectura del sistema de gestión basado en SS7 a través de JMX

Los mensajes de señalización que se envían a la central para gestionar un servicio, son solicitados directamente desde la interfaz de JMX que cubre al SCP de la RI, desde esta interfaz se accede de manera local, o a través de una red IP, a la plataforma SMART III.

La plataforma SMART III se encarga de enviar los mensajes de SS7 a la central a través de la TITD (Tarjeta de Interfaz de Troncal Digital) que se conecta a la central por medio de E1s, esta tarjeta se considera como el módulo SSP-SS7 del modelo de despliegue de la figura 4.1.

Los MBeans actúan de manera directa con el *hardware* de soporte que permite monitorear, extraer e insertar los mensajes de SS7 que sean necesarios para cualquier función prevista.

Los Mbeans que hemos definido (de manera no exclusiva) para la gestión soportada con SS7 son: Gestión de tarificación, gestión de tráfico y gestión de elemento de red. Aunque entre ellos existen similitudes, y redundancia en los mensajes, cada uno es completo e independiente en su función, de manera que en cualquier momento

cualquiera puede ser removido, modificado o reemplazado sin afectar la operación del sistema completo.

Dentro del componente de gestión de tarificación se pueden invocar métodos para registrar inicio y terminación de llamada para efectos de la tasación, identificación de abonado A y abonado B, ruta utilizada, duración de llamada y modificaciones a la tarificación de acuerdo al acceso que haga sobre el servicio (accesos RDSI con uno o múltiples canales).

En el componente de gestión de tráfico se pueden conocer el # de llamadas cursadas, el # de llamadas exitosas, # de llamadas por determinado enlace (o grupo de enlaces), # rutas congestionadas, fecha y hora de los eventos, entre otras, de forma que se puedan generar gráficas estadísticas para determinar los niveles de congestión y provisionamiento de la red para tomar medidas correctivas cuando sea necesario; lastimosamente no se hacen modificaciones directas sobre las configuraciones del tráfico (aunque se podría hacer) dado que eso está cubierto por condiciones administrativas de cada uno de los operadores de red y no se puede entrar en conflicto.

En los MBean para gestión de elemento de red se puede consultar si el elemento está o no activo y si es un elemento de transporte o señalización, sin embargo esta parte de la gestión no fue implementada en la plataforma actual dado que se centra en la prestación del servicio a través de la RI y no la gestión de la red como un todo; sin embargo se realiza gestión de red en cuanto a la asignación de canales sobre una comunicación (cuando el servicio se soporta sobre RDSI) de forma que se puede ampliar o reducir el ancho de banda de acceso a un servicio de acuerdo al número de canales B asociados a la sesión y así mantener la calidad del servicio (por lo cual esto se consideró como gestión de red para gestión del servicio).

Cabe anotar que todas las invocaciones a los métodos de los MBeans se pueden hacer como llamadas RMI dinámicas o invocaciones dinámicas de CORBA o como llamadas de funciones desde Internet, esto para facilitar la implementación de la aplicación de gestión de acuerdo a las capacidades del desarrollador y los recursos técnicos de que dispongan.

4.2.3 SMS (Service Management System)

Sistema de Gestión del Servicio. Realiza el control del diseño, implementación y administración de un servicio. Entre otras funciones, administra la base de datos de la lógica del servicio, administra el tráfico y la recolección de datos de la red.

Al igual que en cualquier otra arquitectura de RI, en la arquitectura SMART III el SMS no es un elemento indispensable para la implementación y despliegue de un servicio, sin embargo es un componente muy importante cuando se trata de evaluar el desempeño del sistema como un todo, además en esta plataforma el SMS es más un

agente gestor de capacidades para el SCP y el SSP, de manera que puede servir como puente comunicador de datos para solicitudes de gestión a través de JMX.

Este módulo de gestión se soporta sobre un sistema operativo de tiempo real (S.O.T.R.) de Linux, de forma que permita aprovechar las características de la máquina virtual de java para tiempo real (tjvm); con este soporte el sistema puede capturar las peticiones que se realizan desde JMX o a través de la UUS3 para enrutarlas hacia el módulo correspondiente para su atención.

El sistema cuenta con una base de datos en la cual se almacenan los datos estadísticos y de tarificación que son reportados desde el SSP y el SCP para control del servicio y sus recursos de red; aunque el SMS no tiene una gran incidencia dentro de la prestación propiamente dicha del servicio, es un componente fundamental para controlar la parte “lucrativa” de la plataforma cuando se instala comercialmente.

4.2.4 Características generales de operación del sistema implementado

La plataforma SMART III se constituye en una de las mejores alternativas de plataforma de bajo costo para servicios multimediales con RI; esto dado que no conlleva mayores costos en la adquisición de la plataforma en sí misma y la reconfiguración necesaria en los enlaces de señalización en la central telefónica a la que estaría conectada la plataforma no son arduos y mucho menos costosos ya que cada central cuenta con sus técnicos capacitados para este tipo de configuraciones.

También se debe resaltar una vez más las ventajas que ofrece el manejo de la señalización No. 7 sobre un sistema multiprocesador para ofrecer la velocidad de desempeño necesaria para cubrir las necesidades de señalización de una conexión convencional, además de las capacidades de gestión de tarificación y red para cada servicio prestado sobre la plataforma.

En general dentro de la arquitectura diseñada e implementada se cubrieron todos los aspectos planteados como objetivos de esta nueva versión de la plataforma, inclusive algunos resultados excedieron las expectativas originales y crearon nuevas puertas para futuros desarrollos; sin embargo no todo es color de rosa dado que en la implementación actual existen algunos limitantes en el servicio que restringen la cobertura de la plataforma, y aunque pueden ser subsanados dentro de un desarrollo comercial, actualmente esto acarrearía costos adicionales en tiempo y dinero que no mejorarían sustancialmente la relación costo/beneficio de la plataforma; dentro de los limitantes que se deben tener en cuenta para la implementación de un servicio multimedial sobre esta plataforma se debe contar:

- Capacidad de gestión de señalización limitada a 30 canales.
- Necesidad de uso de UUS para comunicaciones directas entre los módulos de la plataforma.

- Desarrollo actual soportado sobre enlaces RDSI para manipulación del ancho de banda de las comunicaciones.

A pesar de que estas limitaciones son parte de la propuesta original del proyecto, después de recorrer el camino del diseño, implementación y pruebas se han podido encontrar nuevas técnicas y tecnologías que permitirán extender las capacidades de la plataforma para mejorar su desempeño y cobertura en los servicios, lo cual será planteado en el capítulo 8 que propone algunos de los posibles trabajos futuros.

5 EL MODELO DE CONSTRUCCION DE SOLUCIONES PARA SISTEMAS EMPOTRADOS PARA COMUNICACIONES

5.1 METODOLOGÍA

Para el desarrollo adecuado de este trabajo, fue necesario contar con una metodología que facilitara la construcción de un sistema de calidad y que permitiera realizar un adecuado seguimiento al proceso de construcción del sistema. Dentro de las diferentes alternativas, se seleccionó el Modelo de Construcción de Soluciones, desarrollado dentro del grupo de Ambientes de Desarrollo de la facultad, esta selección se basó en la aceptación y buenos resultados que ha tenido al utilizarse en el desarrollo de anteriores proyectos y en el trabajo previo que se ha implementado con esta metodología.

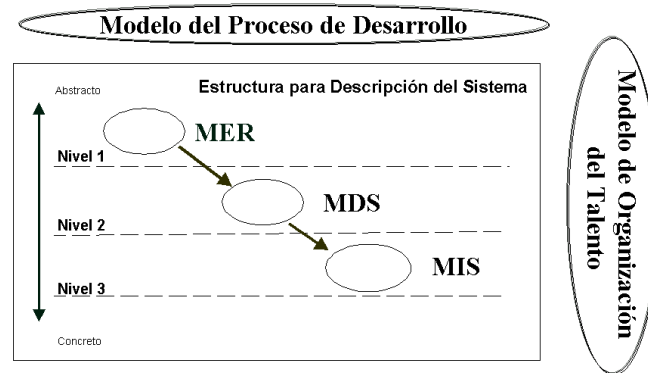
Es importante tener en cuenta que es posible complementar el modelo adicionando estructuras de especificación que permitan describir de manera más clara el sistema que se pretende implementar, sin que esto afecte el desarrollo de esta guía metodológica.

A continuación se muestran las características básicas del Modelo.

5.1.1 MCS (modelo para la construcción de soluciones)

[MCS 1997]

Concebido como una referencia metodológica esencial para cualquier proyecto cuyo propósito sea construir una solución de calidad, oportuna y con costos competitivos que pretenda contribuir a la creación y enriquecimiento de la base de conocimiento y experiencia institucional, se ha constituido en la herramienta metodológica más utilizada y de mayor éxito dentro de la FIET, se puede apreciar la estructura metodológica de este modelo en la figura 5.1.



MER: Modelo para el Establecimiento de Responsabilidades
MDS: Modelo para la Descripción del Sistema
MIS: Modelo de Implementación del Sistema

Figura 5.1 Macro componentes del modelo

5.1.1.1 Macrocomponentes del modelo

En un primer nivel de abstracción puede indicarse que este Modelo para Construcción de Soluciones está constituido por los tres siguientes macrocomponentes:

- Estructura para Descripción del Sistema.
- Modelo del Proceso de Desarrollo.
- Modelo de Organización del Talento.

El primero está constituido por los 3 modelos propuestos por el M.C.S. para describir los 3 niveles de abstracción del sistema/solución. El segundo macrocomponente define las fases de referencia y los criterios de transición entre las mismas. El tercero tiene como propósito servir de referencia para organizar el recurso más valioso con que cuenta cualquier proyecto, el personal, estos macrocomponentes se pueden resumir en la tabla 5.1.

| MACROCOMPONENTES | COMPONENTES PRINCIPALES |
|---|---|
| MODELO PARA ESTABLECIMIENTO DE RESPONSABILIDADES | Modelo del Ambiente del Sistema Modelo de Casos de Uso del Sistema/Solución (ESENCIAL) Arquitectura de Referencia para el Sistema/Solución (ESENCIAL) Registro de Gestión de Riesgos Plan del Proyecto (ESENCIAL) Caso del Negocio |
| MODELO PARA DESCRIPCIÓN DEL SISTEMA | Modelo del Diseño del Sistema/Solución (ESENCIAL) Modelo de Despliegue/Distribución del Sistema/Solución Modelo de Pruebas |
| MODELO DE IMPLEMENTACIÓN DEL SISTEMA | Descripción Detallada del Sistema/Solución Prototipo Operacional de la Solución versión beta (ESENCIAL) Manuales de Usuario e información complementaria |

Tabla 5.1 Estructura de referencia para descripción del Sistema-Solución

5.1.1.2 Fases de referencia

La figura 5.2, al final del párrafo, presenta una visión general del modelo de proceso de desarrollo que el M.C.S. propuesta como referencia para cualquier proyecto que desee construir una solución de calidad.

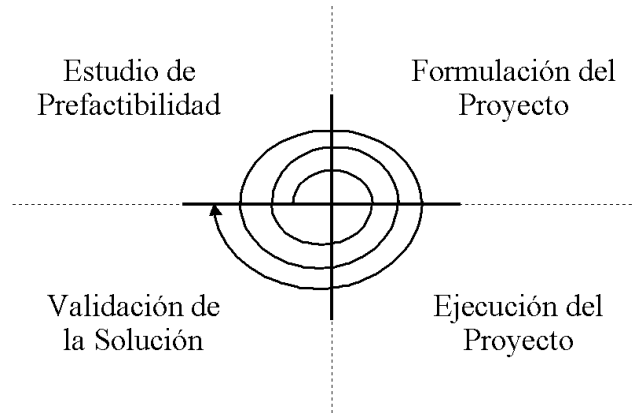


Figura 5.2 Visión general del modelo de proceso de desarrollo

Este modelo tiene como paradigma principal la técnica de desarrollo incremental mediante la cual se busca proveer al cliente - lo más temprano posible - de una capacidad operacional inicial que él pueda ejercitar y a partir de los resultados de la ejercitación (por medio de incrementos de funcionalidad) tratar de obtener una completa satisfacción del cliente y/o de los usuarios del Sistema-Solución. La tabla 5.2 describe las 4 fases establecidas como referencia para cualquier proceso de construcción de una solución. Se indica para cada fase su propósito, el acontecimiento o hito mayor que constituye el criterio esencial de transición a la siguiente fase y se indican los modelos asociados según la denominación establecida en la anterior.

| <i>FASES</i> | <i>PROPÓSITO</i> | <i>HITO MAYOR</i> | <i>MODELOS</i> |
|-----------------------------------|--|---|---|
| ESTUDIO DE PREFACTIBILIDAD | Analizar el dominio del problema y determinar si se puede continuar con el proyecto | Definición del alcance del Sistema-Solución | Modelo para establecimiento de responsabilidades (versión inicial) |
| FORMULACIÓN DEL PROYECTO | Analizar los aspectos esenciales relacionados con la construcción del sistema y asegurar su viabilidad | Establecimiento de una Arquitectura de referencia para todo el ciclo de vida del Sistema-Solución | Modelo para establecimiento de responsabilidades (versión final) |
| EJECUCIÓN DEL PROYECTO | Construir la versión correspondiente del Sistema-Solución | Provisión de la capacidad operacional proyectada | Modelo para descripción del Sistema Modelo de implementación del Sistema |
| VALIDACIÓN DE LA SOLUCIÓN | Ejercitar el Sistema-Solución adecuadamente y corregir los errores detectados | Satisfacción del cliente con el Sistema-Solución | |

Tabla 5.2 Descripción de las fases de referencia

Tomando como base este modelo se desarrolló el proyecto SMART III, pero dadas las condiciones particulares de desarrollo *hardware/software* se han previsto la

inclusión de algunos elementos que permitirían una mejor aplicación del MCS al desarrollo de sistemas que requieran la aplicación de técnicas de codiseño (como es el caso del los sistemas empotrados) y así contar con una fuerte herramienta metodológica para este tipo de proyectos que pueden impulsar rápidamente el desarrollo tecnológico de nuestra región y el país entero.

Para empezar, se expondrán las técnicas de códiseno actuales para el desarrollo de sistemas *hardware/software* para telecomunicaciones.

5.2 TÉCNICAS DE CODISEÑO ELECTRÓNICO APLICADAS A SISTEMAS DE TELECOMUNICACIONES

El gran avance que ha tenido la microelectrónica y las altas escalas de integración que se han logrado alcanzar, han conseguido la integración de sistemas muy complejos en un simple chip. Este hecho ha traído como resultado la introducción de nuevas herramientas enfocadas a la especificación de dichos sistemas que pretenden aumentar la productividad de las empresas y reducir los costos en desarrollo.

Uno de los elementos más versátiles que tiene hoy en día el mundo electrónico es el procesador, el cual trajo la introducción del software como parte fundamental del desarrollo de la gran mayoría de los sistemas, debido a la gran capacidad de abstracción, modelos de desarrollo y pruebas con los que este puede ser descrito sin que pierda sus propiedades, además su reusabilidad, portabilidad, facilidad de adaptación y uso, ha logrado reducir considerablemente los procesos y los costos tanto en desarrollo como en producción. Tanto es así que la tendencia en las herramientas de síntesis *hardware* es asemejarse en la mayor manera a los lenguajes de programación *software* para concurrir en sistemas codiseñados. Sin embargo la gran debilidad de los modelos de procesamiento de instrucciones esta en su capacidad de respuesta, la cual esta ligada por supuesto a la velocidad de procesamiento de la CPU (que solamente puede procesar unas cuantas instrucciones simultaneamente) y a la complejidad del sistema y esto es un punto crítico, sobretodo en los sistemas de comunicaciones, en donde sistemas altamente complejos deben responder a severos y estrictos requerimientos de velocidad.

Es así como se impone la tendencia de los sistemas *hardware/software* que descargan las labores críticas a elementos *hardware* y entregan la responsabilidad de procesamiento lógico más liviano pero no por ello menos complejo, al *software* del procesador, esta tendencia ha logrado aumentar la velocidad de los procesos de desarrollo y disminuir los costos trayendo con ello la introducción de herramientas CAD para síntesis de *hardware* que se asemejan a los ambientes de desarrollo *software* y que interactúan entre ellas para lograr una completa simulación de los sistemas antes de su implementación física, lo que permite detectar fallos y ahorrar esfuerzos antes de plasmar los sistemas sobre el silicio.

5.2.1 Requerimientos de especificación

Uno de los aspectos determinantes en el proceso de desarrollo de sistemas heterogéneos esta en la partición *hardware/software* que se hace antes de empezar el co-diseño pues el resultado de esto determinará la complejidad y rendimiento de los subsistemas resultantes y del sistema total, así como también repercutirá en la velocidad y economía del desarrollo.

Los siguientes requerimientos ayudaran a dar unas pautas para el particionamiento que se haga del sistema antes de determinar que partes se desarrollaran en *hardware* y cuales en *software* al empezar el co-diseño.

Programadores

En la mayoría de los casos, el comportamiento de un sistema puede ser descrito desde el punto de vista de un sistema secuencial. Sin embargo cuando se trata de sistemas de comunicaciones los requerimientos de rendimiento son muy altos lo cual dificulta un poco la especificación de un sistema completo como un algoritmo secuencial; a pesar de ello estos sistemas pueden seguir siendo especificados secuencialmente sin que ello implique que su implementación deba ser hecha de la misma manera, por ello el algoritmo secuencial será una de las técnicas preferidas en la especificación de un sistema heterogéneo.

Descripción de los estados del sistema

La descripción a nivel de estados es una técnica que facilita la abstracción tanto de un sistema *hardware* como de un sistema *software* es por ello que viene a ser una herramienta muy potente a la hora de hacer el paso de un modulo entre su especificación y su implementación.

Particionamiento Jerárquico

La división que se haga de un sistema siempre ayudará a disminuir su complejidad y esto además presenta un mejor panorama a la hora de decidir que funciones serán implementadas en *hardware* y cuales en *software*. Es muy importante una buena división para realizar un buen co-diseño, la partición debe ser hecha como una partición estructural de componentes y como una partición comportamental del sistema.

Comportamiento completo

Una visión completa del comportamiento del sistema ayuda a comprender mejor los subcomportamientos de los subsistemas, lo que facilita la división jerárquica y verificación de la especificación completa del sistema.

Concurrencia de eventos

Cuando se diseñan sistemas de comunicaciones complejos la concurrencia de eventos siempre es uno de los ingredientes fundamentales en el desarrollo y es un punto determinante a la hora de particionar el sistema ya sea arquitectural o

comportamentalmente justo en el momento que se toma la decisión de decidir que elementos van a ser *hardware* y que elementos se implementaran en *software*, pues como ya se sabe las características secuenciales de los procesadores (elementos encargados de ejecutar el *software*) limitan sus capacidades de respuesta y obligan en algunos casos a utilizar sistemas multiprocesador y/o también a sintetizar elementos *hardware* que desarrollen efectivamente las labores de concurrencia de tareas.

Comunicación

Los mecanismos de comunicación entre tareas se pueden clasificar en dos grupos; los de memoria compartida y los de paso de mensajes. Aunque los mecanismos de memoria compartida son económicos y más fáciles de implementar que los de paso de mensajes, se debe tener especial cuidado cuando se usan para intercambio de información entre tareas concurrentes, pues puede dar lugar a complicaciones y a errores muy difíciles de detectar.

Sincronización

Los mecanismos de sincronización entre tareas son muy útiles a la hora de solucionar problemas de comunicación y permitir la integración de múltiples sistemas pues ayudan a que los sistemas se establezcan antes de tomar decisiones críticas que pueden afectar a otros procesos o tareas.

Manejo de excepciones

A menudo la ocurrencia de eventos no determinísticos que pueden afectar el correcto funcionamiento de los procesos y tareas hacen que los sistemas se vuelvan inestables y producen fallas fatales que deben ser capturadas como fallas eventuales y llevar al reestablecimiento del sistema para que este se haga estable de nuevo.

Respuestas no determinísticas

La posibilidad de no poder siempre predecir exactamente los resultados de algunos sistemas ante algún determinado evento debe ser tenida en cuenta siempre que se diseña un sistema complejo de comunicaciones, por varios motivos los sistemas deben ser diseñados bajo estos parámetros y a pesar de ello no deben perder su estabilidad. Existen dos clases de respuestas indeterminísticas, las funcionales y las temporales. Las funcionales son aquellas que para un determinado evento pueden responder de distintas formas de acuerdo al estado del sistema y las temporales son aquellas cuyo tiempo de respuesta para un mismo evento puede variar severamente dependiendo de muchos factores, este comportamiento esta muy ligado con la manera en que se hace el particionamiento del sistema *hardware/software*.

Temporización

Como se dijo anteriormente, la respuesta temporal de algunos sistemas esta ligada con la partición *hardware/software* que se haga en su diseño. Las respuestas temporales de un sistema son a menudo las más críticas y pueden llegar a determinar la imposibilidad de realización de algún diseño, pues son ellas las que son determinadas por el ambiente externo al sistema.

Orientación a objetos

La técnica de orientación a objetos es una de las más poderosas porque además de permitir una aproximación al mundo real para reducir la complejidad del sistema, también presenta características muy útiles como la reusabilidad y el manejo distribuido de elementos. Además de que existen muchas herramientas para describir sistemas a partir de objetos.

5.2.2 Diseño estructurado de sistemas heterogéneos (codiseño)

El diseño estructurado es una herramienta para diseño y pruebas basada en mantener una dependencia mínima de la tecnología en las etapas tempranas del diseño que aumenta a medida que se avanza en el proceso de manera incremental, lo que facilita tomar decisiones a nivel de partición *hardware/software* en etapas tempranas del diseño, este proceso se puede observar en la figura 5.3:

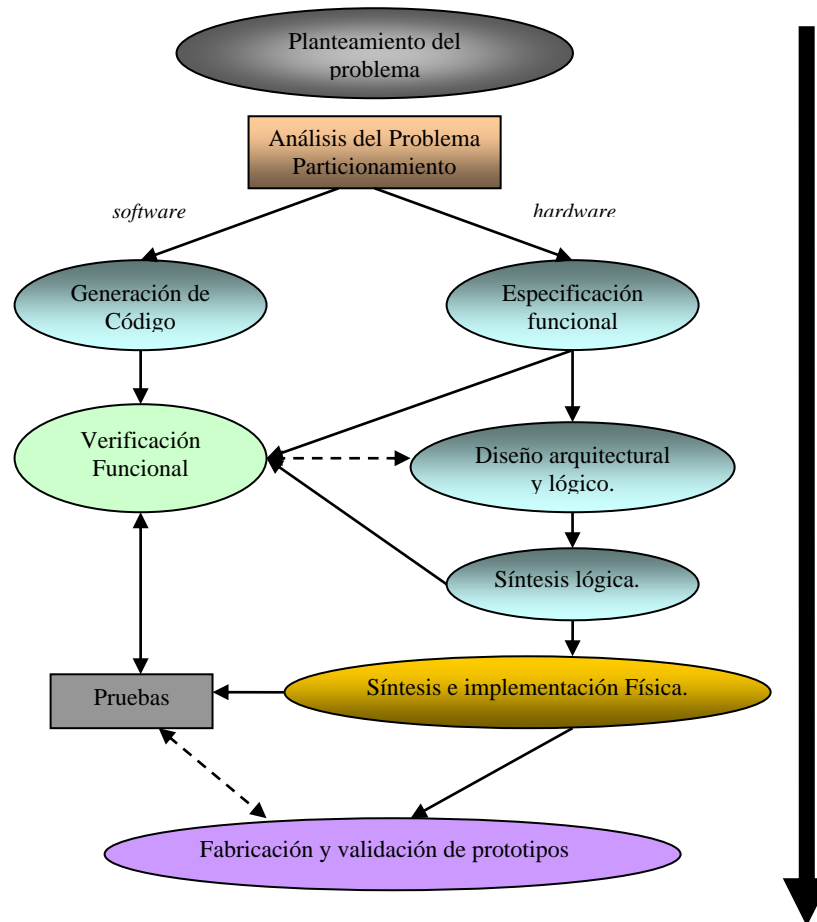


Figura 5.3 Fases del diseño estructurado Top-Down para co-diseño

5.2.2.1 Especificación funcional

En esta etapa se hace un planteamiento detallado de la funcionalidad del sistema; el plan de pruebas previsto en su desarrollo, las herramientas para su diseño y para todo esto se debe tener en cuenta la tecnología elegida. Posteriormente se hace una partición del sistema en bloques funcionales de entrada/salida, se planifica y se modela mediante lenguajes de especificación de alto nivel para descripción de software y hardware, para con esto hacer una primera verificación funcional a alto nivel.

5.2.2.2 Verificación funcional

De acuerdo al esquema planteado en la figura 5.4, la especificación funcional planteada de acuerdo a una o varias alternativas, es enfrentada a través de un simulador a un banco de pruebas realizadas por el diseñador, estas pruebas son llevadas a cabo sobre un conjunto de librerías que reúnen las características reales de los componentes a nivel de simulación (dependencia de la tecnología) con el fin de validar los resultados y realizar los ajustes pertinentes a este nivel.

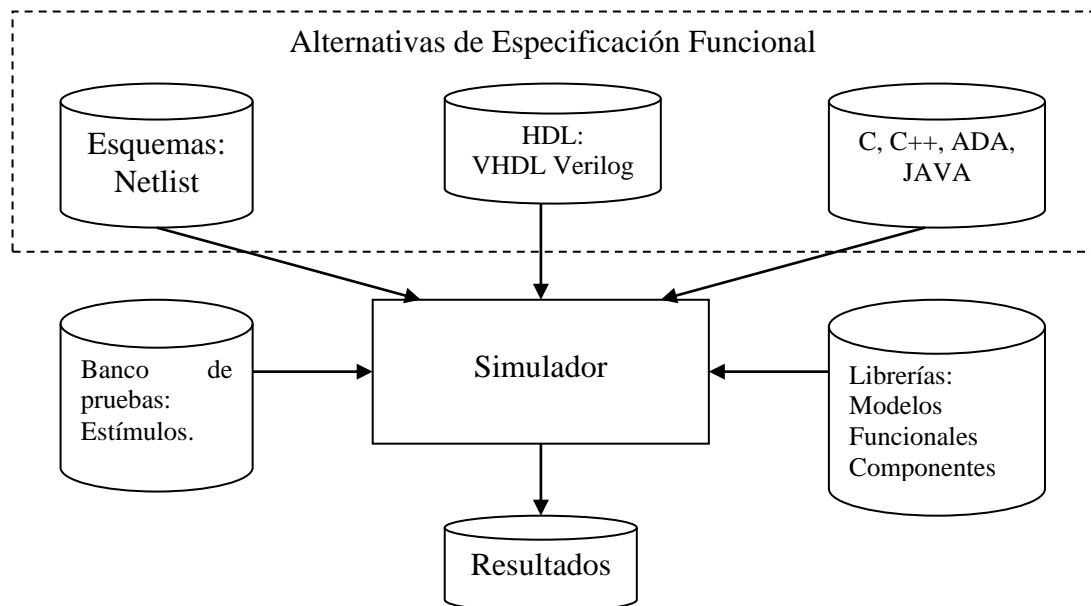


Figura 5.4 Esquema de verificación funcional

Para la realización de las pruebas automáticas también se pueden programar camas de pruebas, estas están definidas como sistemas cerrados con respecto al DUV (Design Under Verification – diseño bajo verificación) ya que ellas generan los vectores de estímulos del DUV y también registran sus respuestas para poder hacer la evaluación de desempeño, este tipo de diseño requiere un esfuerzo adicional por parte del equipo

de trabajo que debe generar un diseño adicional al objetivo y que siempre es más complejo que el sistema-solución en sí mismo, pero es sólo a través de pruebas como estas donde se puede dar garantía de la operación del producto diseñado como solución. Por supuesto existen pruebas ad-hoc que se desarrollan en base a la experiencia de los desarrolladores, pero este tipo de pruebas son muy subjetivas y suelen dejar pasar muchas fallas, por eso se debe mantener en mente el lema de la verificación: “Si no encuentras errores en tu diseño no es porque sea perfecto, es porque no estás haciendo *buenas* pruebas”.

5.2.2.3 Diseño detallado

Arquitectural

El diseño arquitectural busca dividir el sistema total en distintos bloques funcionales con el fin de reducir la complejidad, como se aprecia en la figura 5.5. Para cada bloque funcional se especifica un banco de pruebas propio con la condición que las interfaces entre bloques funcionales deben quedar definidas en forma muy precisa y clara.

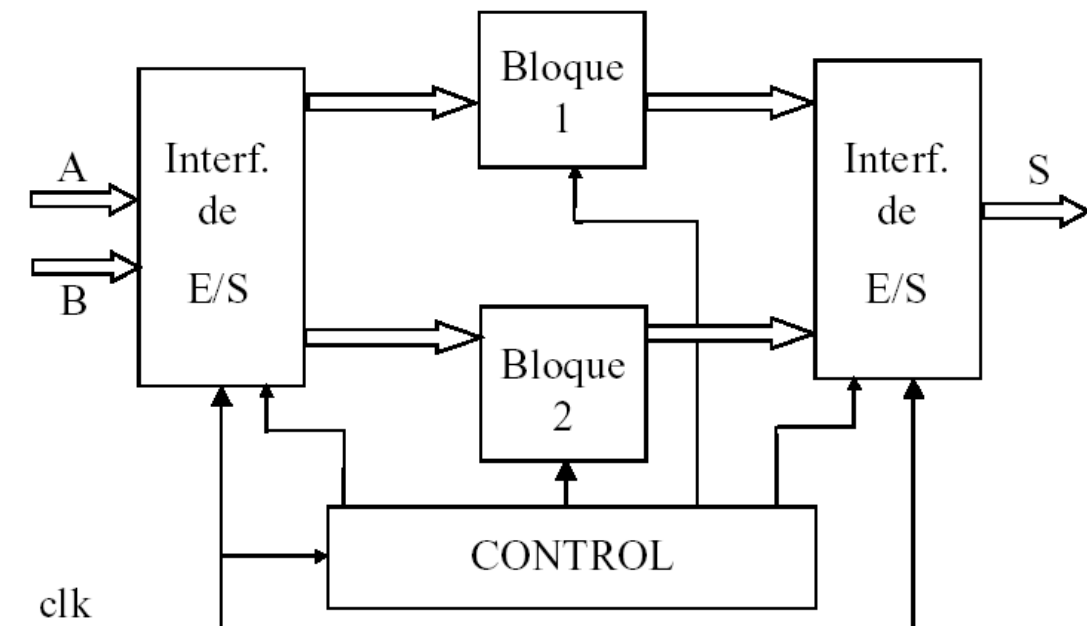


Figura 5.5 División Funcional En Bloques

Lógico

Busca obtener un diseño basado en componentes lógicos (puertas lógicas, flip-flops, memorias, multiplexores, etc.) para cada bloque arquitectural. Existen herramientas automáticas que transforman una especificación funcional en un diseño lógico, estas

pueden estar orientadas a algún tipo de algoritmo o pueden ser dependientes de algún fabricante o determinada tecnología.

Esta fase consume mucho tiempo, sobretodo en el refinamiento del diseño para obtener resultados óptimos que coincidan con los obtenidos en la verificación funcional. Es muy importante no desligarse de la tecnología de implementación, pues decisiones de diseño adecuadas para algunas tecnologías pueden no resultar muy convenientes para otras.

5.2.2.4 Pruebas

Las pruebas de un circuito no consisten sólo en comprobar la funcionalidad del mismo, sino que trata de detectar eventuales fallos en el proceso de fabricación, en teoría se busca comprobar que el circuito este libre 100% de fallos.

Existe la posibilidad de añadir un hardware específico que permita realizar una cobertura de fallos, este procedimiento tiene una efectividad estimada del 93% y no busca una comprobación funcional sino una comprobación de accesibilidad y controlabilidad de los nodos. Su uso tiene sentido para ASICs, PCBs y GateArrays en donde existe un proceso tecnológico de fabricación y no en los FPGAs donde el diseñador, sólo configura la unión a implementar sobre bloques configurables y a programar las interconexiones de pistas que ya existen.

Los fallos de fabricación se producen por fallos al realizar el “placement” de las células o componentes y su interconexión o “routing”, además de su separación, ya que generalmente esto se hace sobre un sustrato de silicio que puede tener impurezas

Objetivos de las pruebas

Las pruebas buscan detectar piezas defectuosas en el proceso de fabricación, por lo tanto el diseño del sistema *hardware/software* debe ir ligado a un diseño de las pruebas, es decir hay que ir diseñando, y pensando como facilitar las pruebas.

Hay que conseguir que el diseño sea controlable y observable ofreciendo:

- Accesibilidad a los nodos desde el exterior.
- Particionamiento y paralelización de los diseños.
- Scan-Path (permitir caminos de rastreo).
- BIST (Built-In Self Test), construcción paralela de un *hardware* específico de pruebas.

5.2.2.5 Alternativas tecnológicas de fabricación

Placa de Circuito Impreso (PCB – Printed Circuit Board)

Esta técnica es muy usada en nuestro país, pues existen multitud de herramientas para su diseño y da la posibilidad que el diseñador tenga control de las dimensiones

físicas, colocación de los componentes y su interconexión (placement/routing), su proceso de desarrollo no es muy costoso y permite adaptarse cambios tecnológicos en forma simple y también permite la utilización de distintas técnicas de pruebas.

Entre sus inconvenientes están que el tamaño físico de los prototipos puede llegar a ser exagerado y a disipar demasiada potencia entre conexiones muy largas, que también pueden dar lugar a efectos de ruido capacitivo e inductivo. En producción a grandes escalas puede llegar a ser muy costoso.

Se pueden clasificar según distintas características, tales como:

- Bicapa/Multicapa.
- Con componentes estándar y/o montaje superficial.
- Taladros metalizados/sin metalizar.
- Para diseño analógico, digital o mixto.

El “routing” y el “placement” se pueden hacer en forma manual, automática o semiautomática, la experiencia aconseja una colocación manual de las señales más críticas (reloj, reset, señales de control, buses de comunicación, etc), minimizando su longitud y evitando ángulos y recodos pronunciados, el resto de las señales se puede *rutear* en forma automática, delimitando áreas críticas no *ruteables* para proteger señales críticas. También es aconsejable, en diseños mixtos (Analógico/digital) separar bien las zonas en la placa y tratar con cuidado las referencias a tierra para evitar el filtrado de ruidos entre zonas. Para diseños digitales no se debe perder de vista la tecnología utilizada (TTL, CMOS, etc), pues unas son más ruidosas que otras y consumen distinta potencia necesitando más grosor y separación entre las pistas, con el fin de evitar ruidos por acoplamiento.

Circuitos de Aplicación Específica (ASIC)

Diseño Full-Custom:

No es una alternativa muy recomendable, pues son difíciles y costosos de verificar, su proceso de desarrollo es poco productivo, difíciles de adaptar a cambios tecnológicos y las herramientas de diseño no son muy conocidas, además que se necesita mucha experiencia del diseñador en el tema y precisa un contacto muy cercano con el fabricante. Sin embargo, cuando las características de especificación del circuito a diseñar (velocidad, consumo, disponibilidad de células estándar, etc) lo aconsejan entonces no hay más remedio.

Diseño Semi-Custom

Es cuando el diseño está ligado a células estándar prediseñadas. Es un diseño más tratable basado en multitud de herramientas, librerías de células y macros (*hardware* y *software*) garantizadas por los fabricantes, es más económico y fácil de probar, mediante el uso de herramientas específicas generadas para esto.

Con la tecnología actual, CMOS por ejemplo, se pueden conseguir chips de varios millones de puertas y una frecuencia de funcionamiento por encima de los 500MHz. Las dimensiones del chip dependen de la tecnología. Un circuito de 100mm² en CMOS de 0,5 micras de canal es GIGANTESCO y de una complejidad enorme. Entre las clases de células y macros que se pueden encontrar están:

- Macro células (macros): RAM, ROM, multiplicadores.
- Bloques funcionales: CPU, RISC, microcontroladores, UARTS, etc.
- Células analógicas: AOs, convertidores A/D y D/A.

Gate Array

En este tipo de fabricación el diseñador sólo se preocupa por especificar las funciones que ha de realizar el diseño, las herramientas de diseño físico únicamente proporcionan la interconexión entre células y al igual que los ASICs es imprescindible las técnicas y la circuitería para pruebas.

Lógica Programable

Son muy versátiles y existen gran variedad de dispositivos y herramientas de lógica programable, el diseñador sólo los configura, por lo tanto no hay un proceso tecnológico y no tiene sentido el test de fabricación. Se pueden clasificar como muestra la figura 5.6.

5.2.2.6 Análisis del sistema y partición

En el codiseño, uno de los aspectos claves está en el particionamiento, en el cual no sólo se determinan cuáles bloques van a ser implementadas en software y cuáles en hardware, sino también se determina que tipos de componentes se usarán (ASICs, ASIPs, Memorias, PLDs, etc.), y los escenarios temporales de asignación de tareas a los bloques funcionales. Es por ello que el particionamiento afectara directamente al resto del desarrollo en cuanto al desempeño y costo del sistema. Lo que convierte al particionamiento en un asunto clave del codiseño.

Mediante un análisis del problema el diseñador deberá decidir que implementar en *software* y que implementar en *hardware*, para ello el diseñador puede utilizar algunas herramientas metodológicas que le permitirán tomar las decisiones, sin embargo, se piensa que las mejores herramientas son:

- La experiencia.
- El conocimiento de los entornos de diseño.
- El conocimiento de tecnologías disponibles.
- El Gran esfuerzo de investigación en Codiseño.

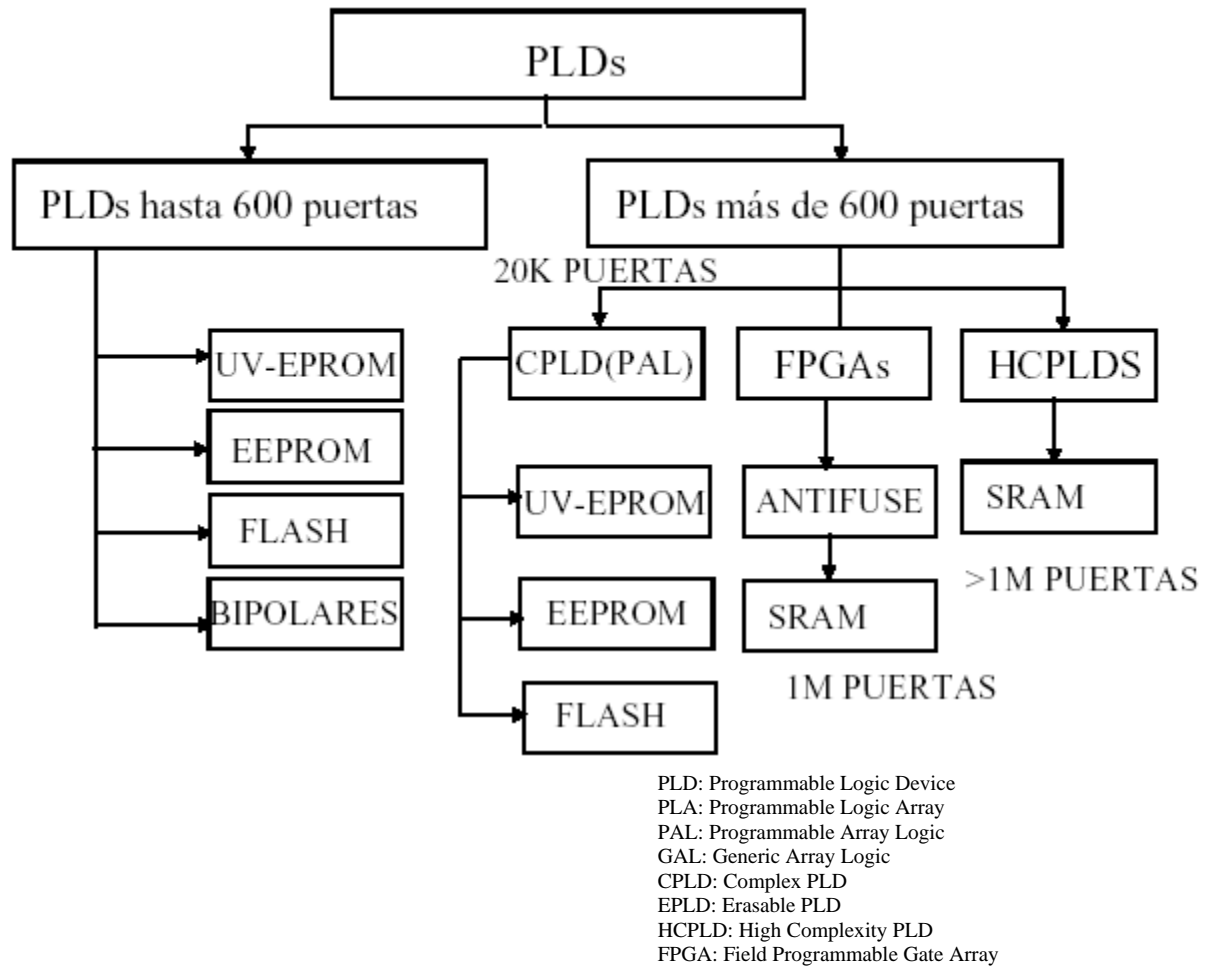


Figura 5.6 Clasificación de los PLDs

Sin embargo, inclusive en este punto es necesario no olvidar las posibles tecnologías de implementación, para hacer el particionamiento del sistema, el diseñador puede guiarse por distintos clases de algoritmos planteados para ello, que conservan algunos puntos en común como:

- Todos ellos son procedimientos de optimización guiados por una función global de costo.
- Hay siempre un soporte estructural basado en graficas.
- Existen siempre unas métricas definidas en cuanto al tiempo de ejecución del software y capacidad (Memoria de datos y programa).

A continuación se darán unos parámetros que podrán caracterizar el proceso de particionamiento:

Modelo del sistema para particionamiento

Antes de empezar el proceso de particionamiento, la descripción inicial del sistema es trasladada en una representación intermedia, es decir, un diseño detallado pero sin el nivel de detalle de la implementación física, más conveniente para su manipulación. Esta representación es un mapeo del modelo del sistema y mantiene

una estructura de datos, una estructura comportamental y la información requerida para llevar a cabo el proceso de particionamiento. Este modelo no corresponde necesariamente al modelo de especificación, dado que este debe ser más adecuado para los algoritmos de particionamiento, entre los modelos más conocidos se tienen:

- Control/Flujo de datos.
- Representaciones basadas en FSM.
- Redes de Petri.
- Graficas estructurales.

El modelo interno debe definir el tipo de granularidad con la que será particionado el sistema, la cual describe como puede ser particionado el sistema en un grupo de objetos distribuidos en varios sistemas de componentes; Existen dos tipos de Granularidad; Tosca y Fina, cada una de las cuales se caracteriza por su capacidad de síntesis de componentes lógicos.

Costo por función

La evaluación de la conveniencia de una partición en objetos hardware y software, no es directa, existen muchos factores que afectan estas decisiones, el costo por función esta ligado a factores como: los costos de fabricación, el desempeño del sistema, violación de restricciones. En general se considera que la implementación hardware eleva los costos pero mejora el desempeño mientras que la implementación software es más simple, más barata y más rápida pero su desempeño es más bajo. De acuerdo a esto se asumen algoritmos en los cuales se busca cumplir con las restricciones más severas usando hardware y para el resto se busca la solución más rápida y económicamente más conveniente.

Arquitectura objetivo

La arquitectura objetivo es la descripción de los componentes que interactúan entre si por medio de distintas clases de conectores. Las arquitecturas codiseñadas se caracterizan por tener al menos un componente programable (Procesadores o microcontroladores estándar) y un circuito dedicado con funcionalidad específica (Generalmente un ASIC) usando un mecanismo de comunicación de memoria compartida entre ellos. Es muy difícil clasificar los tipos de tendencias en cuanto a arquitecturas se refiere pero se pueden nombrar cuatro principales:

- La clásica, basada en un microprocesador +ASIC + Memoria + *interfaz* de bus.
- La orientada a datos, es igual a la clásica pero basada en el uso de DSPs.
- La se sistemas on-chip que usa un procesador integrado en un chip con ASICs.
- La de sistemas de tiempo real distribuidos.

Dominio de la aplicación

El dominio de la aplicación es indispensable, pues es ilógico pensar en un ambiente general de codiseño, pues cada dominio tiene distintas tendencias, la mejor división en cuanto a esto esta en clasificarlo en dos grandes grupo, sistemas de control y sistemas de procesamiento de datos (Telecomunicaciones). Mientras los de control se basan en una granularidad tosca, pues sus restricciones básicamente están en sus entradas/salidas y no tienen requerimientos severos de tiempo, los de procesamiento

de datos se adecuan a la granularidad fina pues tienen muchos caminos de procesamiento y grandes restricciones temporales.

Existen gran número de algoritmos en el área del codiseño para particionamiento en distintos marcos de referencia, de los cuales es muy difícil hacer un estudio comparativo, los más conocidos se muestran en la tabla 5.3.

| Procedimiento | Granularidad | Algoritmo | Grupo |
|---------------|--------------|--------------------------------|------------------------------|
| Automático | Tosca | Basado en lista de cronológica | Ptolemy (Berkley) |
| | | Particionamiento Hardware | Fvahid (Riverside) |
| | | Heurística (SA, TS) | P. Eles (Linköping) |
| | Fina | Enfriamiento simulado | Cosyma (Braunshweig) |
| | | Grupo migración | Vulcan (Standfor University) |
| | | Programación dinámico | Lycos (Tech.Denamarc) |
| Manual | Tosca | | Polis, Chinook, CoWare |

Tabla 5.3 Algoritmos de particionamiento en Co-diseño

5.2.3 Java como lenguaje de especificación para sistemas CTI

Como un lenguaje derivado de C, JAVA se presenta una alternativa muy atractiva para especificación de sistemas de telecomunicaciones heterogéneos ya que gracias a su concepción para sistemas distribuidos, permitiría dar flexibilidad a los procesos de especificación para granularidad fina. Además JAVA cumple con todas las requerimientos de especificación presentados, exceptuando los requerimientos para especificación temporal, pues sus características de manejo de procesos en tiempo real son restringidas, sin embargo, la introducción del RT-JAVA como lenguaje de programación para sistemas operativos de tiempo real puede facilitar el manejo de las especificaciones temporales. Entre las características más representativas de JAVA para el caso están:

- Lenguaje de programación orientado a objetos.
- Manejo de herencia simple.
- Manejo de hilos o tareas concurrentes.
- Manejo de excepciones.
- Manejo de interrupciones y eventos.
- Capacidad de sincronización para acceso a procedimientos.
- Comunicación distribuida entre procesos.
- Acceso a puertos.

Para algunos sistemas CTI en donde el objetivo es llevar la computación hasta un hardware específico (de telefonía por ejemplo), resulta conveniente pensar en una arquitectura objetivo basada en un sistema coprocesador comunicado por un puerto estándar de comunicación serial, teniendo en cuenta las altas velocidades que estos

pueden alcanzar hoy en día. El objetivo es manejar eficientemente un hardware de comunicaciones orientado a telefonía por medio de un procesador con periféricos de aplicación específica y controlar este desde una plataforma de cómputo convencional con un sistema operativo de alto nivel.

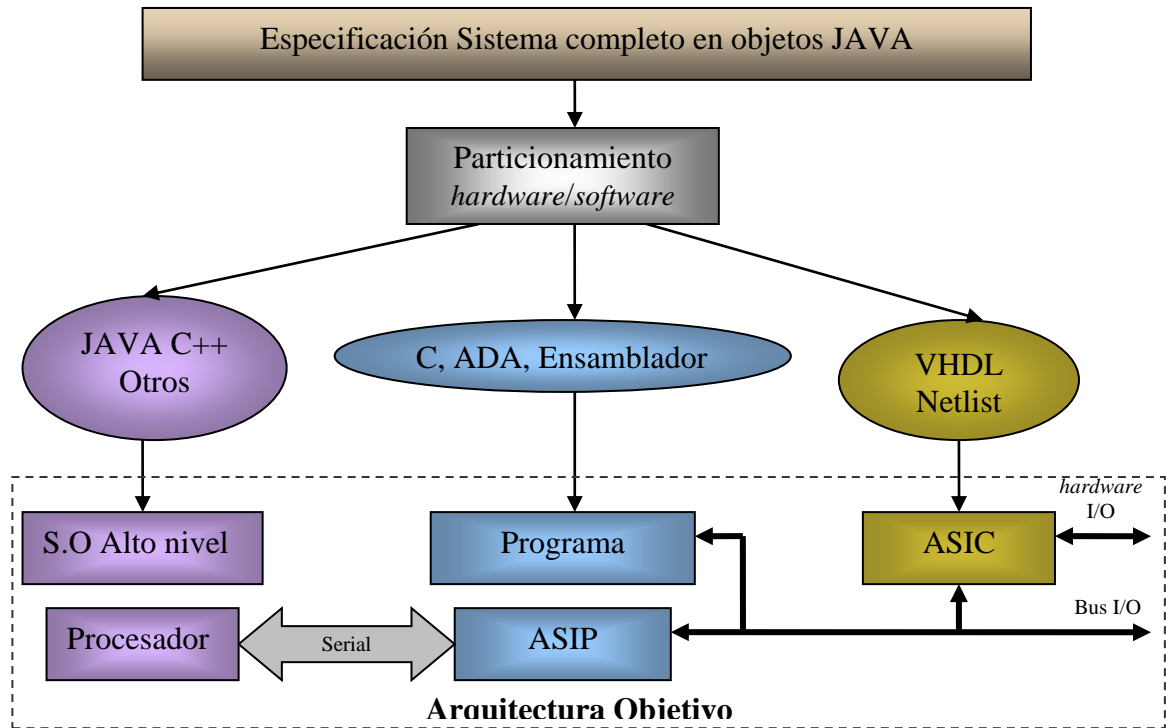


Figura 5.7 Metodología de codiseño propuesta para sistemas CTI

Se propone hacer la especificación del sistema completo en RT-JAVA (como se muestra en la figura 5.7), que permita descomponerlo en un gran número de funciones (granularidad fina) de procesamiento de información. Posteriormente se las funciones se debe agrupar en tres grupos:

- Funciones de alto nivel.
- Funciones especializadas de alto rendimiento.
- Funciones críticas en el tiempo y/o con interfaces no digitales.

De acuerdo a esto las funciones de alto nivel serán implementadas sobre un sistema operativo de alto nivel en general estas funciones se pueden identificar por sus grandes necesidades de interacción hombre máquina, manejo de protocolos de Internet, grandes necesidades de almacenamiento de información o por tener restricciones temporales poco exigentes. Esas funciones pueden ser dejadas en java o trasladadas a otro lenguaje de alto nivel, adaptando sus mecanismos de comunicación a los puertos seriales disponibles.

Las funciones especializadas de alto rendimiento (Tiempo real), de manejo de información a alta velocidad con bajos requerimientos de almacenamiento,

concebidas observando las funcionalidades especiales del ASIP para conservar correspondencia en la granularidad funcional. Estas funciones pueden ser trasladadas a C, ADA o ensamblador según lo requiera, para su compilación y ejecución desde el núcleo de un procesador especializado que puede por si mismo suplir algunas funcionalidades planteadas en JAVA, mejorando así el desempeño del sistema.

Las funciones críticas en el tiempo corresponden a funciones de alto desempeño en tiempo real o con interfaces de línea analógicas. Estas pueden ser trasladadas a un lenguaje de especificación *hardware* como VHDL o a esquemas netlist para la fabricación o configuración del ASIC.

5.3 MODELO DE CONSTRUCCIÓN DE SOLUCIONES CON CODISEÑO (MCS-Co)

Siendo una de las mejores referencias metodológicas para el diseño, desarrollo e implementación de sistemas para solución de problemas ingenieriles, el MCS cuenta con los componentes necesarios para enfrentar cualquier proyecto en general, sin embargo cuando se enfrentan problemas cuya solución está dentro del dominio de los sistemas empotrados, se pueden utilizar algunas especializaciones insertas dentro de este modelo para así adecuarlo más aún a este tipo de desarrollos en particular, facilitando las fases de co-diseño y co-pruebas; como ya se conoce en un primer nivel de abstracción puede indicarse que este Modelo para Construcción de Soluciones está constituido por los tres siguientes macrocomponentes:

- Estructura para Descripción del Sistema.
- Modelo del Proceso de Desarrollo.
- Modelo de Organización del Talento.

El primero está constituido por los 3 modelos propuestos por el M.C.S. para describir los 3 niveles de abstracción del sistema/solución. El segundo macrocomponente define las fases de referencia y los criterios de transición entre las mismas. El tercero tiene como propósito servir de referencia para organizar el recurso más valioso con que cuenta cualquier proyecto, el personal, estos macrocomponentes pueden ser ampliados con nuevos componentes que le permitan una mejor adaptación al desarrollo de soluciones basadas en sistemas empotrados, esta propuesta de ampliación se puede resumir en la tabla 5.4, a continuación:

| MACROCOMPONENTES | COMPONENTES PRINCIPALES |
|---|--|
| MODELO PARA ESTABLECIMIENTO DE RESPONSABILIDADES | Modelo del Ambiente del Sistema Modelo de Casos de Uso del Sistema-Solución incluyendo las restricciones temporales de cada caso de uso (ESENCIAL) Arquitectura de Referencia para el Sistema-Solución donde se definan las restricciones de dimensión física y consumo de potencia (ESENCIAL) Registro de Gestión de Riesgos Plan del Proyecto (ESENCIAL) Caso del Negocio |

Tabla 5.4 Estructura de referencia para descripción del Sistema-Solución

| MACROCOMPONENTES | COMPONENTES PRINCIPALES |
|--------------------------------------|--|
| MODELO PARA DESCRIPCIÓN DEL SISTEMA | Modelo del Diseño del Sistema-Solución especificando los bloques de operación microprocesada y los bloques operación paralela (ESENCIAL) Modelo de Despliegue-Distribución del Sistema-Solución Modelo de Pruebas incluyendo diseño de la cama de pruebas del sistema y métricas Ad-hoc y formales a utilizar (ESENCIAL) |
| MODELO DE IMPLEMENTACIÓN DEL SISTEMA | Descripción Detallada del Sistema-Solución donde se especifiquen las tecnologías propias a utilizar Prototipo Operacional de la Solución versión beta (ESENCIAL) Manuales de Usuario e información complementaria |

Tabla 5.4 Estructura de referencia para descripción del Sistema-Solución (continuación)

Las ampliaciones en cada uno de los componentes principales de los macrocomponentes se pueden ver en detalle dentro del tema “técnicas de codiseño electrónico aplicadas a sistemas de telecomunicaciones” cubierto anteriormente.

5.3.1.1 Fases de referencia

La figura 5.8, al final de párrafo, presenta una visión general del modelo de proceso de desarrollo que el M.C.S. propuesta como referencia para cualquier proyecto que desee construir una solución de calidad.

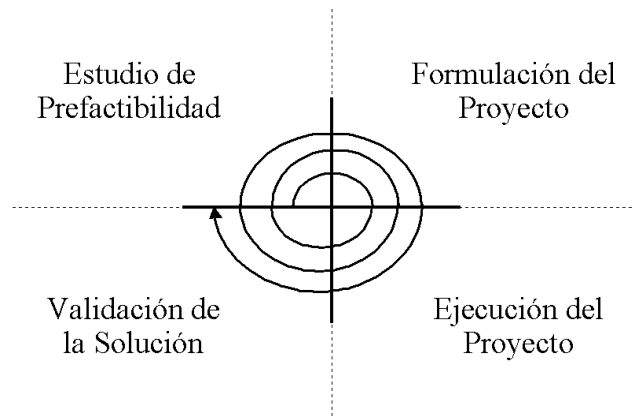


Figura 5.8 Visión general del modelo de proceso de desarrollo

Siendo fundamental para este modelo ampliado mantener la visión de entregar lo más pronto posible una versión beta del sistema-solución y sobre él desarrollar los prototipados incrementales, se debe hacer un fuerte hincapié en la necesidad de tener un buen diseño de verificación, del cual se puedan obtener las métricas necesarias para soportar la culminación de una espiral en el proceso de desarrollo. Sabiendo que la mayoría de los sistemas empotrados interactúan con otros componentes electrónicos y no directamente con humanos (en la mayoría de los casos no hay ningún tipo de interfaz H-M) se debe contar con un personal dedicado única y exclusivamente al diseño de las pruebas del sistema para estresarlo lo suficiente en cada una de sus funciones a través de una cama de pruebas u otro tipo de verificación funcional que no sea basada en la experiencia del ingeniero de verificación.

6 TRABAJOS DE GRADO VINCULADOS AL PROYECTO Y VALIDACIÓN DE LA ARQUITECTURA

Durante todo el proceso de desarrollo del presente proyecto se ha contado con el apoyo del Grupo de Ingeniería Telemática, pero además de ellos debo resaltar el papel que han desempeñado los diferentes estudiantes de trabajo final de pregrado para los cuales he actuado como su director de tesis, y lo planteo de esa forma porque gracias a ellos es que este proyecto pudo ser culminado en corto tiempo y con unos resultados más que satisfactorios para los investigadores a él vinculados.

Los TGs (Trabajos de Grado) que han estado y están vinculados al proyecto SMART III han permitido la puesta a punto de cada uno de los componentes de la arquitectura y a través de sus resultados se ha podido realizar la validación de la plataforma total; esos trabajos, sus objetivos y resultados se relacionan a continuación.

6.1 VALIDACIÓN DEL MODELO DE GESTIÓN SOPORTADA EN MENSAJES DE SEÑALIZACIÓN NO. 7

Este componente fue validado a través de la implementación del SSP que sirve como soporte y corazón de la plataforma SMART III, esto fue desarrollado dentro del TG titulado: “Módulo empotrado del sistema SMART para la gestión de servicios de redes inteligentes a través de ISUP del protocolo SS7” [Vallecilla 2004], dentro del cual se planteo como objetivo principal el Diseñar y construir un sistema empotrado que implemente el nivel ISUP como parte del modulo SS7 concebido por el proyecto SMART III, de modo que a través de una interfaz abierta de gestión permita procesar y generar mensajes ISUP para controlar los recursos de la red.

Como parte de la aplicación final del proyecto se desarrollo una interfaz *software* en Java que permite configurar y monitorear los mensajes de señalización sobre el SSP (incluyendo UUS) y también gestionar las características propias del SSP a través de una interfaz JMX sobre el Web, a continuación se pueden apreciar las interfaces de esta aplicación.

En la primera imagen (figura 6.1) se puede apreciar el panel de llamadas del sistema; a través de esta interfaz un usuario puede realizar, terminar y monitorear cualquier llamada cursada a través de los canales del E1 cuya señalización se maneja desde el SSP. Teniendo como herramienta esta aplicación se puede utilizar el SSP como un monitor de tráfico activo en línea, todo lo cual certifica la correcta recepción, procesamiento e inserción de mensajes hasta el nivel ISUP dentro del enlace de señalización que se utiliza como vínculo de control con la PSTN.



Figura 6.1 Panel de control de llamadas del SSP

Además del panel de llamadas también se creó una interfaz para configurar las condiciones que pueden variar dentro de un enlace de SS7 para la norma nacional, esto con el fin de garantizar una adecuada sincronización de línea y señalización, esta interfaz se encuentra a continuación en la figura 6.2.

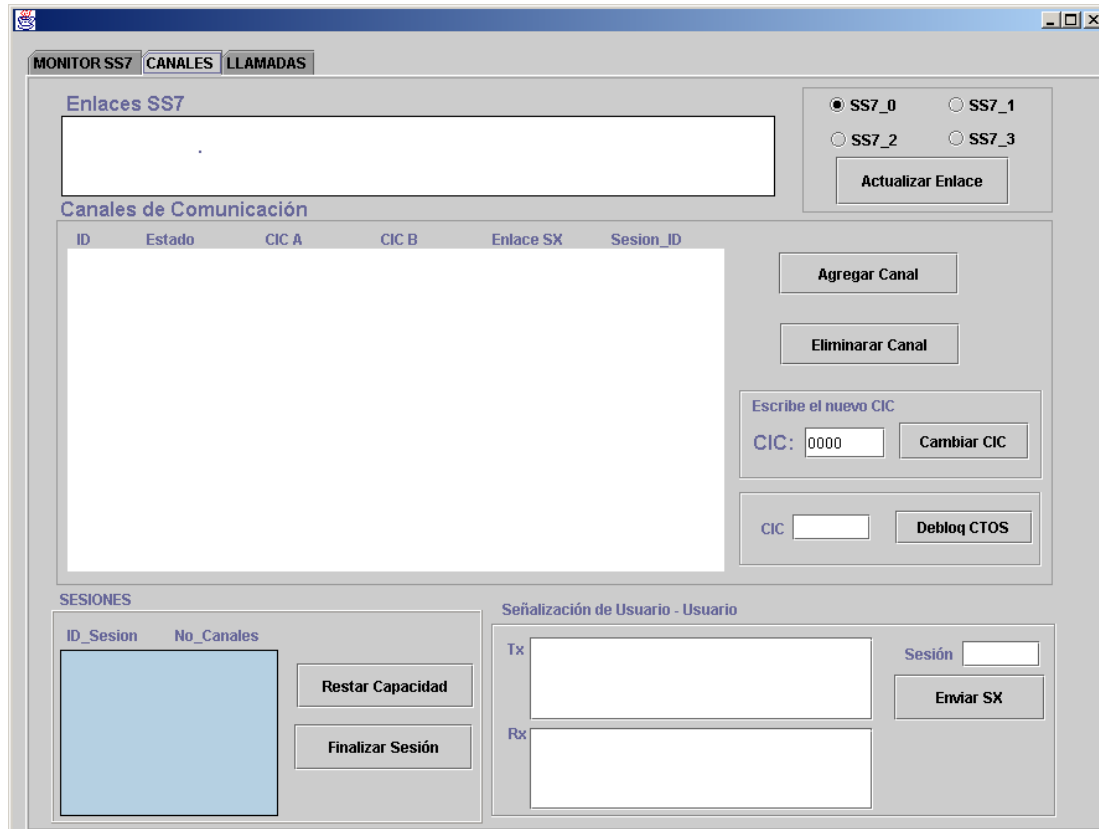


Figura 6.2 Interfaz de configuración del SSP

Como última herramienta de configuración y verificación de este componente de la plataforma se desplegó su interfaz de gestión JMX en el Web para su gestión remota desde un gestor JMX, esa interfaz se puede apreciar en la siguiente imagen (figura 6.3).

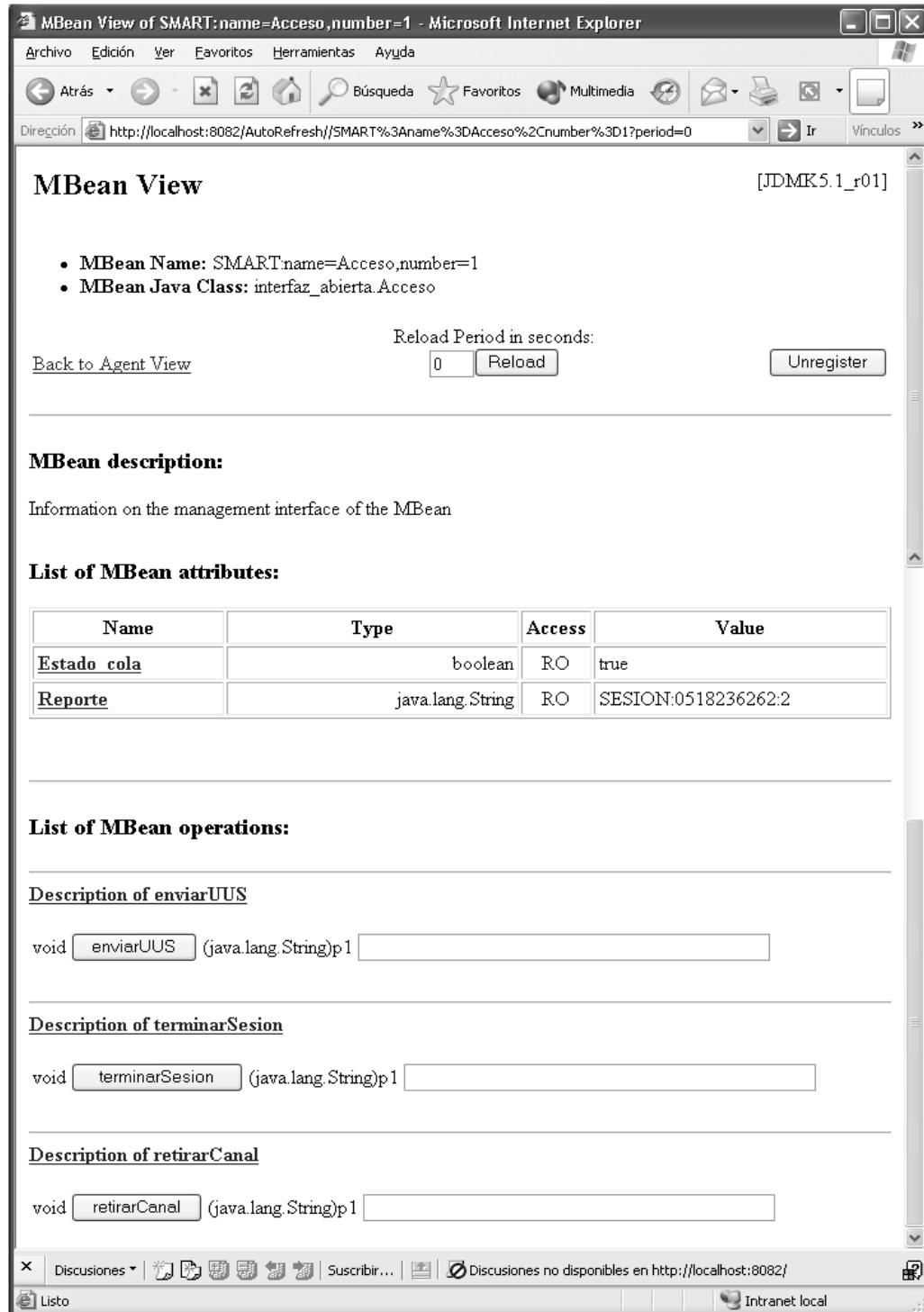


Figura 6.3 Interfaz JMX en el web para SSP

Con las interfaces incluidas y todo el soporte empotrado que se desarrolló para el SSP de SMART III se pudo dar validación a este componente, y para su afirmación se incluyen los documentos de certificación expedidos por Nortel Networks (Ver Anexo 1) donde se realizaron las pruebas contra la central real; también se puede acudir a la monografía aquí reseñada y sus anexos donde se especifican el total de pruebas realizadas y el código de la implementación *software* (empotrada y en PC) del sistema.

6.2 VALIDACIÓN DE LA PLATAFORMA PARA PRESTACIÓN DE UN SERVICIO MULTIMEDIAL GESTIONABLE CON SS7

A través del TG titulado: “Servicio de video por demanda gestionable con SS7” [Moya 2004] se realizaron las pruebas de validación para gestión de tarificación, gestión de red y gestión de servicio para un servicio multimedial soportado en la plataforma SMART III, este servicio se prestaba a través de líneas RDSI básicas.

Este servicio demostró la capacidad de modificar la asignación del número de canales de una comunicación a través de RDSI utilizando mensajes de SS7, además permitió comprobar las capacidades de gestión en tiempo real que puede aprovechar el usuario del servicio utilizando los canales dedicados de la comunicación (UUS3); en las pruebas de este servicio se diseñaron las siguientes interfaces para acceso.



Figura 6.4 Presentación de inicio del servicio

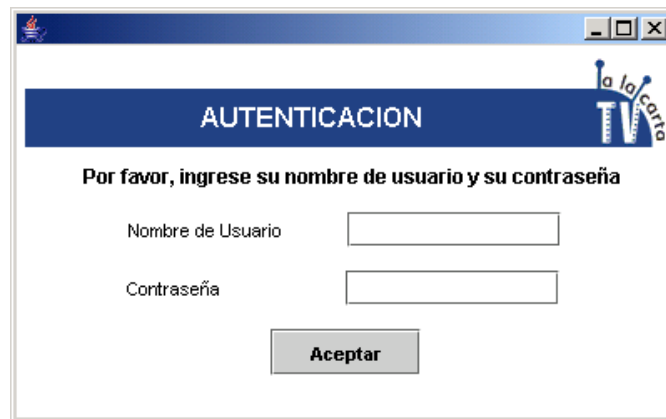


Figura 6.5 Interfaz de verificación de usuario

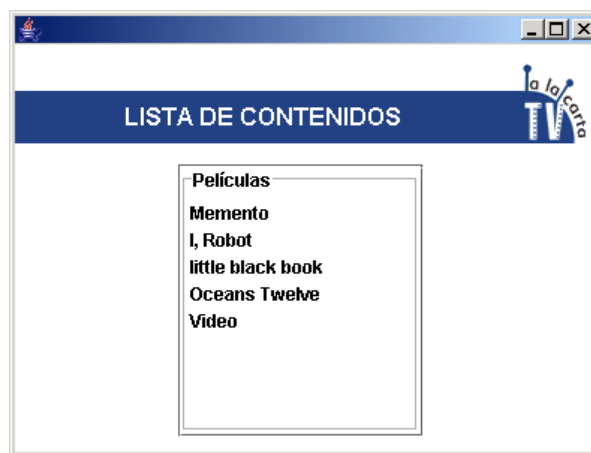


Figura 6.6 Cuadro de despliegue de las películas

Lo más importante era la interfaz de control de reproducción del servicio, desde la cual el usuario tenía todas las capacidades de una reproductora de video virtual, además de que en cualquier momento puede seleccionar si desea reproducir el video con ancho de banda simple (1B) o ancho de banda doble (2B), en este punto la interfaz puede adaptarse fácilmente a aplicaciones sobre conexiones primarias, pero no se implementó dadas las características de público y penetración del servicio planteadas, a continuación se presenta la gráfica de control en la figura 6.7.

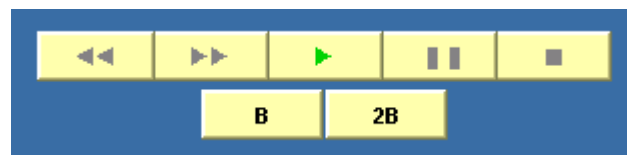


Figura 6.7 Interfaz de control de reproducción

Incluyendo las capacidades ofrecidas hacia el usuario, el suscriptor cuenta con un sistema de gestión que puede ser accedido a través del Web, con un cliente JMX o con un explorador convencional; las interfaces de gestión de usuarios y del sistema en general hacen parte del SCP en si mismo y aunque son parte esencial del sistema se consideran un valor agregado sobre la validación. Todas las pruebas realizadas sobre este módulo, como parte de la validación se realizaron en el centro de redes

inteligentes de Emcali en Cali y como soporte de ello se incluyen los certificados de los resultados expedidos por los ingenieros asistentes a la demostración (ver Anexo 1), para más detalles acerca de las pruebas realizadas se puede acudir a la monografía y anexos de trabajo de grado aquí reseñado. A continuación se exponen las interfaces de gestión desde el Web en las figuras 6.8 hasta la figura 6.12.

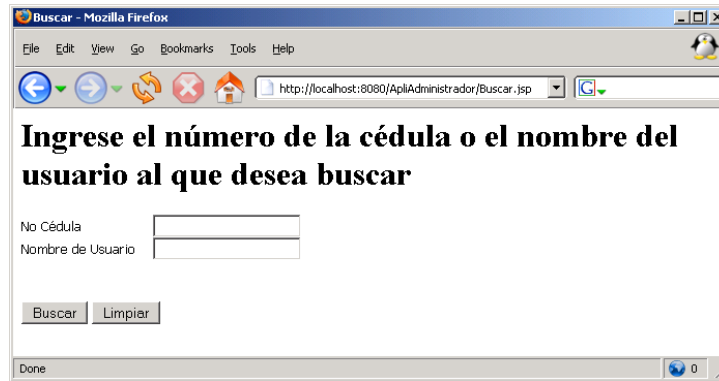


Figura 6.8 Interfaz para búsqueda de usuarios del servicio

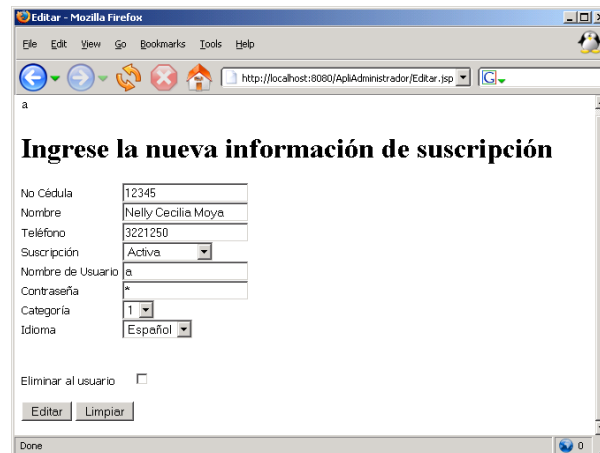


Figura 6.9 Interfaz para edición de suscripción de usuarios del servicio

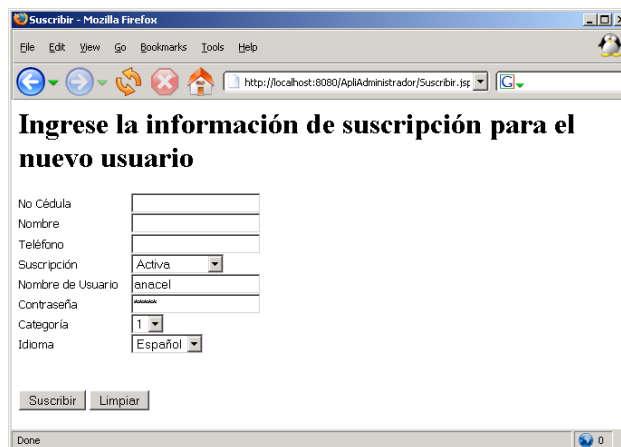


Figura 6.10 Interfaz gráfica para Suscribir un nuevo usuario Buscar/Editar

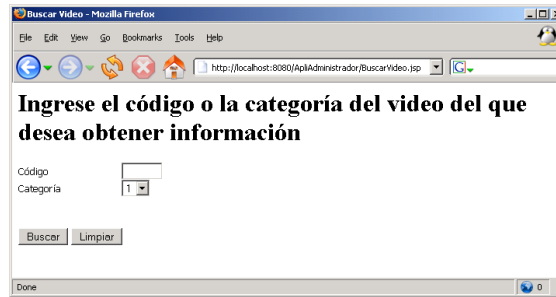


Figura 6.11 Interfaz gráfica para Buscar la lista de videos existentes

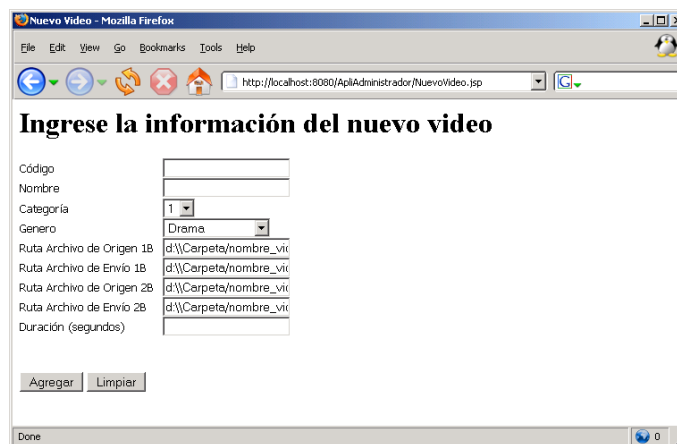


Figura 6.12 Interfaz gráfica para Agregar un nuevo video

6.3 SISTEMA DE GESTIÓN INTEGRADA PARA LOS SERVICIOS DE SMART III

El sistema gestor de la plataforma SMART III se soporta en un S.O.T.R. sobre el que opera RTJ con JMX y se están incorporando las capacidades de comunicación de CORBA-RT para brindar mayor fiabilidad al sistema de gestión y poder desplegarlo de manera distribuida entre varios PC en distintos puntos de la red, pero conservando un monitoreo centralizado.

Esta etapa de la validación se enmarca dentro del TG titulado: “Sistema de gestión en tiempo real de servicios avanzados de telecomunicaciones para la plataforma SMART III” [Miranda 2004], dentro del cual se lleva el control tarifario de todos los servicios que pueden estar montados sobre la plataforma.

Este módulo cuenta con un gestor JMX que hace descubrimiento dinámico de las interfaces JMX publicadas por los otros componentes de la plataforma y permite realizar gestión integrada sobre ellos a través de dichas interfaces, adicionalmente este módulo recibe la señalización usuario-usuario que intercambian el SSP con el

usuario en el establecimiento de la conexión y el SCP con el usuario durante la prestación del servicio, de esta manera se tiene una gestión sobre la red en tiempo real para modificar sus parámetros de operación.

Desde el módulo gestor se puede hacer control de los servicios prestados (crear, modificar y eliminar servicios) y también se puede realizar monitoreo en línea de los servicios activos, además que sobre cada servicio se pueden solicitar las estadísticas de desempeño y tarificación, seguidamente se pueden encontrar las interfaces que facilitan ese procesamiento (figura 6.13 y figura 6.14).

The screenshot shows a software window titled "Servicios Provistos" with a standard Windows-style title bar. Inside the window, there are two tabs: "Control" (selected) and "Monitoreo". Under the "Control" tab, there are three buttons: "Crear Servicio", "Modificar Servicio", and "Eliminar Servicio". Below these buttons is a section titled "Servicios por Proveedor" with a label "Seleccione un proveedor" and a dropdown menu currently showing "CABLEUNION". The next section is "Parámetros del servicio", which contains two input fields: "Id_Servicio" with the value "00005" and "Nombre" with the value "Audio por demanda". Below this is a section titled "Parámetros específicos" with two dropdown menus: "Nombre" set to "Codificación" and "Valor" set to ".mp3". At the bottom of this section are two buttons: "Nuevo" and "Eliminar". At the very bottom of the window are three buttons: "Aplicar", "Aceptar", and "Cancelar".

Figura 6.13 Interfaz para registro de nuevos servicios



Figura 6.14 Interfaz de monitoreo de conexiones y solicitud de estadísticas

Dentro de este TG aún se está trabajando para completar todos los objetivos propuestos para él inicialmente, por lo tanto los resultados incorporados al proyecto SMART III han sido parciales, aunque no por eso poco significativos, ya que gracias a este trabajo se crearon, desplegaron y gestionaron las interfaces JMX de los otros módulos.

7 CONCLUSIONES

El desarrollo de sistemas empotrados para telecomunicaciones es un medio eficaz de enfrentar problemas de orden tecnológico realizando una inversión económica relativamente baja en relación a los beneficios que se pueden obtener, esto es especialmente cierto para el área de las telecomunicaciones donde las restricciones de tiempo y desempeño obligan a los ingenieros a crear sistemas dedicados para satisfacer las necesidades del mercado, alejándose un poco de la concepción de que todas las soluciones deben ser procesadas sobre un PC.

La toma de estándares de señalización como base para realizar gestión de redes y de servicios abre un nuevo espectro de posibilidades de interacción total con cualquier tipo de equipo de la PSTN, independientemente del proveedor del equipo, ya que los sistemas de señalización siempre mantienen un margen de compatibilidad total para garantizar la operación de la red completa.

La gestión en tiempo real de los servicios implementados sobre la plataforma SMART III, facilita los procesos de control de calidad del servicio e interactividad con el usuario dado que se puede contar con canales dedicados dentro de la comunicación para atender los diferentes aspectos que puede requerir la gestión, y en especial los relacionados con gestión de red y gestión de servicios de manera integral, para optimizar al máximo el uso de recursos y la satisfacción de los usuarios.

La capacidad de ofrecer servicios multimediales sobre la plataforma SMART III permitirá a las PyMEs del sector de las telecomunicaciones, e incluso a los grandes operadores, obtener una plataforma a muy bajo costo con una amplia versatilidad para soporte de múltiples servicios, tales como video conferencias, transmisión de radio por cable telefónico, servicios de localización geográfica y muchos más.

El uso de lenguajes como JAVA para la implementación de sistemas empotrados de comunicaciones y de su contraparte a nivel de servicios y gestión ofrecen una vía natural para la integración e interacción de los diferentes componentes del sistema lo cual facilita su mantenimiento y escalabilidad a través del tiempo.

Aunque las metodologías de desarrollo de proyectos en *software* pueden ser aplicadas al desarrollo de proyecto *hardware/software*, es importante contar con herramientas metodológicas propias del co-diseño para obtener resultados igualmente satisfactorios que los obtenidos con el *software* y con tiempos de desarrollo competitivos.

La vinculación del sector empresarial dentro de los proyectos de investigación y desarrollo de la Universidad, ofrecen un verdadero camino para el encuentro de soluciones atemperadas a nuestra realidad regional y nacional, además que a su vez entregan muchos de los nichos tecnológicos donde se requiere investigación y se pueden obtener productos comercializables que sean de beneficio de todos.

En las universidades se debe buscar el desarrollo de una tecnología propia que adapte toda la tecnología extranjera, a través de una verdadera apropiación de las transferencias tecnológicas que se pueden obtener de las empresas, y que facilitaran la creación de soluciones más adecuadas a nuestro entorno social y cultural.

8 TRABAJOS FUTUROS

La plataforma SMART III se puede convertir en una “punta de lanza” para la implementación de toda una gama de servicios multimediales montados en Internet pero con acceso telefónico para que los usuarios pueden disfrutar las ventajas de arquitecturas como SPIRITS que pueden ser complementarias a las capacidades de nuestra plataforma.

Uno de los aspecto más interesantes de la gestión realizada con SS7, consiste en la posibilidad de monitorear y controlar los mensajes para establecimiento y liberación de llamadas, de forma que el sistema además de ser una plataforma de servicios puede actuar como un módulo de monitoreo y control de tráfico telefónico, e incluso podría generar archivos de tarificación integrada por empresa y no por equipo de proveedor como sucede actualmente en todas las empresas de telecomunicaciones nacionales.

La integración de la telefonía IP con la PSTN cobra una nueva vía de solución a través del manejo de mensajes que puede hacer el SSP de SMART III lo cual lo habilita a trabajar como una pasarela SIP/SS7 con sólo algunas modificaciones sobre su diseño original.

Como extensión de la actual plataforma SMART III se podrían estudiar otras tecnologías de interconexión y transporte para ofrecer servicios soportados en redes IP, tales como lo son OSA/PARLAY, SPIRITS, xDLS/RDSI, PINT entre otras que pueden ayudar a convertir la plataforma en un DPE-TINA que facilite la prestación de servicios independientes del tipo de acceso y de terminal de usuario.

Se debe propender por el desarrollo de sistemas empotrados que puedan suplir las necesidades de equipo terminal que puedan tener los usuarios sin acceso a un PC, esto para ampliar la cobertura de la RI y abrir el abanico de servicios que se podría prestar de acuerdo a los terminales desarrollados.

9 REFERENCIAS Y BIBLIOGRAFIA

- [CCIT 2003] “Información sectorial – Estadísticas”. Cámara Colombiana de Informática y Telecomunicaciones.
<http://www.ccit.org.co/pages/html/transaccional/estadisticas.asp>
2003.
- [comp-corba 2002] CORBA Components An Adopted Specification of the Object Management Group, Inc., Version 3.0 formal/02-06-65, Junio 2002.
- [comp-def 2002] “What are components”, Componentsource,
<http://www.componentsource.com/AboutComponents/ComponentEssentials/WhatAreComponents.asp>, Enero 2002.
- [comp-micro 2003] “COM”, MicrosoftCOM,
<http://www.microsoft.com/com/tech/com.asp>, 2003.
- [comp-sun 2003] “Component-Based Software with Java Beans and ActiveX”Sun Microsystems,
http://sunsite.tut.fi/javastation/whitepapers/javabeans/javabean_ch2.html, 2003.
- [CRT 2003] “Indicadores de telefonía local”. Comisión de Regulación de Telecomunicaciones, República de Colombia.
http://www.crt.gov.co/paginas/internas/infsector/ind_tel_local.htm#
2003.
- [DALLAS-2000] “Selecting and Using RS-232”, DALLAS Semiconductor, December 2000.
- [DNP 2003] “Indicadores de Telecomunicaciones”. Departamento Nacional de Planeación.
http://www.dnp.gov.co/01_CONT/INDICADO/Infraest.htm#6
2003.

- [Easy25554-2001] Easy25554 toolbrief description 2001.
- [gest-corba 1999] “Integrated Management of Networked Systems”, Heinz Gerd Hegering, Sebastián Abeck, Bernhard Neumair, Ed Morgan Kaufmann Publishers, 1999.
- [IEC 2000] “Intelligent Network (IN)”, The Internacional Engineering Consortium, <http://www.iec.org>. 2003.
- [Infineon 2002] - C166 UTAH, datasheet 2002.
- [ITU-T 1997a] “Recomendaciones serie Q.1200 – Q.1238” ITU-T. 1997.
- [ITU-T 1997b] “Recomendaciones Serie Q.750-Q.759”. ITU-T. 1997.
- [ITU-T 1997c] International Telecommunication Union, Recomendación Q.762, Septiembre de 1997.
- [jdmk-sun 2003] “JAVA[tm] DYNAMIC MANAGEMENT KIT”,Sun Microsystems, <http://www.sun.com/products-n-solutions/nep/software/java-dynamic/index.html>, 2003.
- [LOPEZ 1998] “Advanced Techniques for Embedded Systems Design and Test”. Juan C. López, Román Hermida, Walter Geisselhardt. Kluwer Academia Publishers. Netherlands. 1998.
- [MCS 1997] “Construcción de Soluciones – Modelo de Referencia para el Desarrollo de Proyectos v.1.1”, Serrano, C. E. Universidad del Cauca. 1997.
- [Miranda 2004] Miranda Diego, Henao Mario, “Sistema de gestión en tiempo real de servicios avanzados de telecomunicaciones para la plataforma SMART III”. Trabajo de grado, Universidad del Cauca. 2004
- [Moya 2004] Moya Ana Cristina, “Servicio de video por demanda gestionable con SS7”. Trabajo de grado, Universidad del Cauca. 2004
- [OMG 2002] “CORBA Components, Versión 3.0”, OMG, Adopted specification of the object management group, Inc. Junio de 2002.
- [Rendón-2003] Rendón Álvaro, Departamento de Telemática Universidad del Cauca, documentación Electiva Sistemas de Tiempo Real, 2003.
- [RTJ 2000] “The Real-Time Specification for Java”, Greg Bollella, Ben Brosgol. Addison-Wesley. 2000. www.rtj.org
- [SBEI-2000] HDLC Derived Protocols, <http://www.sbei.com>, 2000.

- [soft-comp 2000] “TradPerspective_White_Paper_0102”, Len Bass, Charles Buhman, Santiago Comella-Dorda Volume I: Market Assessment of Component-Based Software Engineering, Mayo 2000.
- [TEKNO 2003] Tekno Telecom. Net(s) CCS.
http://www.teknotelecom.com/Products_NetCCS.htm, 2003.
- [TINA-C 1996a] “Service Architecture”, TINA-C, Version 4.0, 28 October 1996.
- [TINA-C 1996b] “Computational Modelling Concept”, TINA-C, Version 3.2, 17 May 1996.
- [Tutorial-2003] Tutorial de redes, <http://www.automatas.org/redes>, marzo de 2003.
- [UUS-rec] Recomendación UIT-T Red digital de servicios integrados; Servicio suplementario de señalización de usuario a usuario; capacidades del servicio. Rec. I.257.1.
- [Vallecilla 2004] Vallecilla Ricardo, Safa Mahdi. “Módulo empotrado del sistema SMART para la gestión de servicios de redes inteligentes a través de ISUP del protocolo SS7”. Trabajo de grado, Universidad del Cauca. 2004

<http://www.tmcnet.com/enews>

<http://www.artesyncp.com/index.html>

<http://www.openss7.org>

<http://ctilabs.de/en/prd>

<http://www.emprix.com>

<http://www.rad.com>

Documentation SMART II GIT FIET UNICAUCA.

Presentación SMART III GIT FIET UNICAUCA.

GIT-SMART-III-V0.5, Octubre de 2001.

RENDON y OSPINA. Sistemas de conmutación digital. Conferencias. FIET Unicauca. 1992.

McGraw Hill Series on Telecommunications. Telecommunications Protocols. 2 edition McGraw Hill

AGUSTIN Froufe, JAVA 2, Alfa Omega, Ra-Ma 2003.

GOTTFRIED Byron, Programación en C, Mc Graw Hill 1997.

CHAPMAN Martin, MONTESI Stefano “Overall Concepts and Principles of TINA”, editor Martin Chapman, 1995

HILL-PETERSON Sistemas Digitales, Organización y Diseño de hardware, Limusa 1994.

DE LA PUENTE Juan A.. "Real-Time Object-Oriented Design and Formal Methods." Real-Time Systems, vol. 18, no. 1, January 2000; pp.79-83

JOYANES AGUILAR Luis, Programación en C++, Mc Graw Hill 1996.

RIPOLL José Ismael. "Tutorial de Real Time Linux". Universidad Politécnica de Valencia. 2001. <http://bernia.disca.upv.es/~iripoll/rt-linux/rtlinux-tutorial/>

C166 Users Guide, Tasking software, 2000.

CISCO, “E1 R2 signalling theory”. 2003.

DMS-100 ETSI-ISUP & CEPT Features, 1995.

Getting Started Guide, ORCAD-Layout, 2002.

Performance Technologies, SS7 Tutorial 2000-2001

Recomendaciones ITU-T Libro Blanco, marzo de 2000.

Sun Microsystems, “Java[Tm] Dynamic Management Kit”, <http://www.sun.com/products-nsolutions/nep/software/java-dynamic/index.html>, 200

ANEXO 1: Certificaciones de pruebas contra centrales reales de la plataforma SMART III

A partir de aquí se pueden encontrar los documentos correspondientes a las certificaciones que fueron entregadas por parte de Nortel Networks en Bogotá y Emcali en Cali, donde se realizaron las pruebas y demostraciones de los diferentes módulos para la validación del presente trabajo de maestría.