

**ANÁLISIS DE LOS SERVICIOS DE TELECOMUNICACIONES BASADOS EN
INTERFACES PARLAY OSA**

**JUAN JOSE ACEVEDO RUEDA
2038241**

**OSMER EDUARDO RUIZ MATEUS
2038264**

Proyecto para optar el título de
Especialistas en Telecomunicaciones

Director

Ph D. HOMERO ORTEGA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERIA ELECTRICA ELECTRÓNICA Y
TELECOMUNICACIONES
ESPECIALIZACION EN TELECOMUNICACIONES
BUCARAMANGA
2005**

AGRADECIMIENTOS

La mayor parte del tiempo en que escribimos este libro pudimos observar el apoyo que recibimos de tantas personas que sin conocernos nos brindaron su mano y que además compartieron sus conocimientos para que esta monografía llegará a su feliz término, por eso, a todos ellos quiero agradecerles todos esos momentos y consejos que nos brindaron.

Agradecemos a nuestros Padres por los principios sembrados gracias a ellos cosecharemos grandes frutos.

Al profesor Homero Ortega por su tiempo y colaboración en el desarrollo de este escrito.

CONTENIDO

	pág.
INTRODUCCIÓN	
1. REDES INTELIGENTES E IMPACTO EN COLOMBIA	10
1.1 EL MODELO DE INTELIGENCIA DE RED	12
1.2 APORTES Y SERVICIOS DE TELECOMUNICACIONES EN COLOMBIA	17
1.3 SERVICIOS DE TELECOMUNICACIONES DE VALOR AÑADIDO	19
2. PARLAY OSA SERVICIOS INALÁMBRICOS	21
2.1 PARLAY OSA	21
2.2 LA ARQUITECTURA DE PARLAY OSA	22
2.3 FRAMEWORK DE PARLAY OSA	23
2.4 PARLAY OSA EN LA RED	24
2.5 EL PAPEL DEL GATEWAY DE PARLAY OSA	24
2.6 CONECTANDO LOS SERVIDORES DE APLICACIÓN EN LA RED DE LA EMPRESA	25
2.7 EL PAPEL DE LOS SERVIDORES DE APLICACIONES EN LA RED	26
3. HERRAMIENTA ERICSSON NRG COMO DESARROLLO DE SERVICIOS PARLAY OSA.	27
3.1 PARLAY.	28
3.2 PARLAY OSA.	28
3.3 PARLAY HOSA. ALTO NIVEL DE ACCESO DE SERVICIO ABIERTO	29
3.4 NETWORK RESOURCE GATEWAY (NRG).	29
3.5 SDK.	30
3.5.1 SOFTWARE LIBRERÍAS DE JAVA.	31
3.6 SIMULADOR NRG.	34
3.6.1 REQUERIMIENTOS DEL SISTEMA.	34
3.6.2 INSTALACIÓN AUTOMÁTICA.	35
3.6.3 INICIAR EL SIMULADOR NRG.	35
3.6.4. ADICIONANDO TELÉFONOS	36

3.6.5 HACER UNA LLAMADA.	37
3.7 EJECUTANDO APLICACIONES PARLAY OSA.	38
3.7.1 EJECUTANDO LA APLICACIÓN WEBDIAL	39
3.7.1.1 DESCRIPCIÓN DE LA APLICACIÓN WEBDIAL	39
3.7.1.2 CORRIENDO LA APLICACIÓN.	39
3.7.1.3 DISEÑO DE DESCRIPCIÓN DE WEBDIAL.	41
3.7.1.4 SECUENCIA MAIN.	41
4. TECNOLOGIAS ACTUALES IMPLEMENTADAS EN COLOMBIA	43
4.1 ELEMENTOS DE REDES MOVILES	43
4.2 SERVICIOS QUE OFRECEN LAS COMPAÑÍAS DE COMUNICACIÓN MÓVILES EN COLOMBIA	47
4.3 ALGUNOS TERMINALES MÓVILES COMUNES, IMPLEMENTADOS POR LAS ACTUALES COMPAÑÍAS DE SERVICIO EN COLOMBIA.	53
5.REVISIÓN DE SERVICIOS PARLAY OSA INTERFAZ DE PROGRAMACIÓN DE APLICACIONES APIs.	57
5.1 CONTROL DE SESIÓN DE DATOS	57
5.2 PRESENCIA Y DIRECCIÓN DE DISPONIBILIDAD	57
5.3SERVICIOS DE INFORMACIÓN DE USUARIO.	58
5.4SERVICIO DE MENSAJERÍA	59
5.5SERVICIOS DE ADMINISTRACIÓN DE INFORMACIÓN PERSONAL (PIM).	60
5.5.1 PIM DE CONTACTO	60
5.5.2 PIM DE CALENDARIO	61
5.6 SERVICIOS DE INTERACCIÓN Y CONTROL DE LLAMADAS DE LOS USUARIOS.	61
5.6.1. PROTOCOLOS SOPORTADOS EN LA RED.	62
5.6.2 WEBDIAL	62
5.6.3 CALLROUTER	63
5.6.4 DIFERENTES CONDUCTAS DE LLAMADAS (DIFFERENT CALL BEHAVIORS)	63
5.6.5 SERVICIO DE COBRADO (CHARGING)	64
5.6.6. DURACIÓN DE LA LLAMADA	65
BIBLIOGRAFÍA	66

LISTA DE FIGURAS

	pág.
Figura 1. Modelo de Inteligencia de Red	14
Figura 2. Modelo funcional de la arquitectura IN CS2	16
Figura 3. Framework Parlay	23
Figura 4. Parlay OSA	24
Figura 5. Elementos de Redes de Telecomunicaciones	27
Figura 6. Interfaz Parlay	28
Figura 7. Ericsson Network Resource Gateway (NRG)	30
Figura 8. NRG Software Development Kit (SDK)	30
Figura 9. JAVA librerías de Software	59
Figura 10. API CORE	32
Figura 11. API Utility	33
Figura 12. API Test	34
Figura 13. SIMULADOR NRG	35
Figura 14. Pantalla Principal NRG	36
Figura 15. NRG anexo de Teléfonos	37
Figura 16. NRG realizando llamadas	38
Figura 17. NRG Llamada establecida	38
Figura 18. Aplicaciones NRG	39
Figura 19. Ejecutando Aplicación Web dial	40
Figura 20. Especificación de usuarios	40
Figura 21. Conferencia de Llamadas	41
Figura 22. Secuencia de eventos cuando el usuario requiere la aplicación	42

LISTA DE ANEXOS

Anexo A: Código Fuente aplicación WebDial

INTRODUCCIÓN

Gracias a los adelantos presentados en las últimas décadas en la ciencia y la tecnología, la rama de las telecomunicaciones ha estado muy bien beneficiada, por ello temas como comunicaciones portátiles integradas han podido acogerse en el desarrollo del nuevo mundo y el crecimiento de nuevas tecnologías aplicadas en Colombia, favorecen este gran paso hacia lo imposible.

A continuación presentaremos esquemas básicos que han motivado al desarrollo de sistemas que logran hacer converger las comunicaciones a un nuevo camino.

Las redes inteligentes e impacto en Colombia nos presentara una visión de cuales son los ideales que se quieren cubrir para poder implementar este gran sueño la convergencia de las telecomunicaciones.

El capítulo 2 da una apreciación global de los API de Parlay OSA, una interface abierta creada por el consorcio de 65 compañías de tecnologías de información (TI) e industrias de Telecom que habilitan la creación de nuevos servicios de las telecomunicaciones. Los APIs de Parlay OSA pueden usarse para el desarrollo de servicios móviles, así como los servicios de la red fija, y servicios de convergencia de la próxima red de generación. Parlay OSA incluye capacidades específicas, diseñado para reforzar los servicios de datos móviles, incluyendo localización, capacidad del Terminal, dirección de conexión entre otros. Operadores de la red y Proveedores de Servicio están buscando nuevas fuentes de crédito, y el nuevo valor agregado, los servicios son una parte importante de esta iniciativa. Parlay OSA habilita el desarrollo de estos nuevos servicios que usan las mismas tecnologías para el rápido desarrollo de aplicaciones en las tecnologías de información: los APIs abiertos, la informática distribuida, Java y Servicios de Web.

El capítulo 3 describirá la arquitectura de Parlay OSA y la adopción de Parlay OSA alrededor del mundo mostrará una breve descripción de como crear aplicaciones de servicios PARLAY OSA, Ericsson provee el Network Resource

Gateway (NRG) y el Software Development Kit (SDK), el SDK contiene los siguientes cuatro paquetes:

Documentación: contiene toda la información que se necesita para programar las aplicaciones hacia NRG, incluyendo la guía de programación, especificaciones PARLAY OSA y especificación de interfaces de Ericsson.

Software Librerías de Java: el Software de las librerías de Java simplifica el desarrollo de las aplicaciones que usan NRG en las cuales se estudia las clases API CORE, API Test y API Utility.

Simulador NRG: el simulador NRG simula las redes de telecomunicaciones y el Gateway de Parlay. En esta sección se describirán los requerimientos para poder instalar el Simulador así como la utilización del mismo para ejecutar las aplicaciones.

Ejemplo de Aplicación. se analiza un Diagrama de Clases que ilustra las clases de modelos y sus principales relaciones de la mayoría de las aplicaciones, un ejemplo junto con el código fuente para usar el software de las librerías de java NRG este código fuente puede ser usado para aplicaciones reales.

El capítulo mostrará una información global de como utilizar las aplicaciones y como utilizar las librerías de Java usando NRG.

El tema Tecnologías Actuales Implementadas en Colombia ubicara al lector en los principales bloques de una red celular para poder entender la arquitectura de ella y con ello dar pie a una breve mención de los operadores nacionales con sus respectivas tecnologías, en este capitulo se presentan algunas de las terminales móviles actuales mas usadas por los operadores del servicio de comunicaciones móviles personales de nuestro país como Bellsouth, Colombia Móvil y Comcel, y se muestran los principales servicios que los operadores prestan.

Por último presentamos una breve revisión de servicios PARLAY OSA interfaz de programación de aplicaciones APIs, que incluye el control de sesión de datos, servicios de información de usuario, servicio de mensajería, y algunos servicios de información personal.

1. REDES INTELIGENTES E IMPACTO EN COLOMBIA

El éxito del desarrollo de las telecomunicaciones estará basado en dos características fundamentales:

- El incremento en el número de servicios de telecomunicaciones, así como de las aplicaciones que hacen uso de estos, además en el tipo de medios utilizados en la provisión de estos servicios (redes de acceso, tipos de terminales, tecnologías de transporte, etc).
- La integración del suministro, operación y gestión de las redes de telecomunicaciones.

Ambas tendencias tienen su origen en los desarrollos tecnológicos pero también, en las demandas de los usuarios, que desean más servicios, sin que esto suponga, no obstante, un aumento de la complejidad de su utilización, ni tener que interaccionar con una gran cantidad de distintos operadores.

La diversidad de servicios e integración de redes no sólo no son incompatibles, sino que, a largo plazo, resultan complementarias. Es decir la gran mayoría de usuarios utilizarán los servicios más necesitados o que consideren básicos dando ello una mayor demanda a los proveedores que permitirá minimizar algunos costos debido a la tendencia de usuarios a solicitar primero los servicios que más valoran. Y con ello se entablara un punto de equilibrio entre el costo de servicios complementarios y un número reducido de usuarios que darán uso a ellos; sin embargo lo ideal sería que a medida que el incremento de usuarios y servicios alcance un punto alto, la tendencia al uso de servicios complementarios se mantuviera con costos similares.

Se considera que en un entorno integrado los usuarios valoran más la provisión de nuevos servicios, al ser más sencilla su utilización.

Las soluciones no integradas son generalmente más fáciles de implementar y poner en funcionamiento (de ahí su menor costo inicial), con los posibles beneficios que esto implica a corto plazo, sobre todo en un entorno competitivo.

Sin embargo, los operadores y proveedores de servicios que prefieran no atarse a un único proveedor, o fijar sus objetivos a más largo plazo, encontrarán más interesantes las posibilidades de un entorno integrado.

Es en este contexto donde hay que entender la convergencia entre las redes móviles y los sistemas de redes inteligentes. La disponibilidad de una plataforma de red inteligente facilitará la creación e implantación de servicios de forma rápida y flexible. Entre los servicios que está previsto que soporte a corto plazo se encuentra la movilidad personal. Las redes móviles soportan la movilidad del terminal.

De su convergencia e integración se derivan, entre otras, dos tipos de ventajas fundamentales:

1. Por parte del usuario: La posibilidad de soportar las comunicaciones personales y otros nuevos servicios avanzados (por ejemplo, aquellos que utilizan la localización del usuario para proveer otros servicios).
2. Para el operador/proveedor de servicios: Una explotación más optimizada de sus redes, incrementando su eficiencia, reduciendo los costos de operación y mantenimiento, reutilizando la infraestructura, etc. Un ejemplo de las ventajas que pueden obtenerse son la minimización de los costos de interconexión utilizando las redes inteligentes (IN) para proporcionar enrutamiento óptimo.

Sin embargo, conviene aclarar que esta integración presenta en el momento actual importantes problemas que la dificultan, mientras no se incrementen las capacidades y prestaciones de las actuales plataformas de redes inteligentes.

Una Red Inteligente (IN) es una arquitectura que se caracteriza por el control de servicio centralizado y lógica de servicio, funciones de conmutación distribuida y recursos especializados. Al emplear este concepto de arquitectura, los proveedores de servicio y operadores de la red podrán introducir nuevos servicios con rapidez y facilidad, poniendo estos servicios a disposición de todos los abonados. Las funciones de la red necesarias para brindar servicios son modulares, reutilizables y portátiles entre las entidades físicas. La arquitectura de la IN prevé la rápida introducción de servicios nuevos así como

flexibilidad para los abonados y usuarios de los servicios a fin de controlar determinados aspectos basados en la IN. Las prestaciones podrán crearse y personalizarse cuando sea necesario teniendo en cuenta la facilidad de uso.

La idea básica de la IN es que muchos servicios pueden implementarse como una sucesión de acciones elementales ejecutadas por la red, independientes de la prestación y ordenadas de acuerdo a una lógica propia de uso, que pueden ser reutilizadas para implementar otros servicios.

Por otro lado, se busca separar las funciones de control avanzado de los servicios de las funciones básicas de conmutación y transporte, de forma que las primeras residen en los equipos que configuran las plataformas de IN avanzadas, separadas de las centrales de conmutación. De esta forma, la IN no modifica los servicios portadores (de transporte de información) de las redes sobre las que actúa, sino que optimiza y proporciona nuevas capacidades de control de los servicios.

1.1 EL MODELO DE INTELIGENCIA DE RED

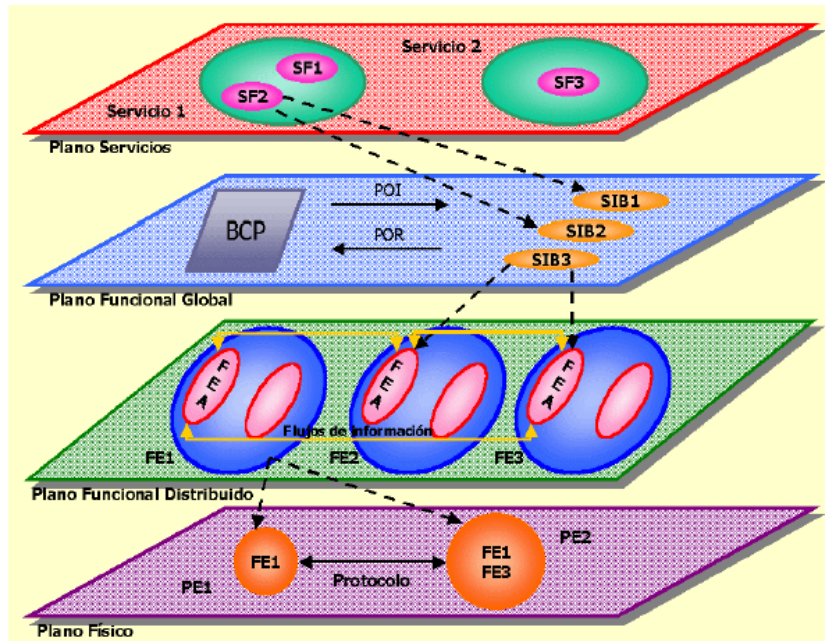
De acuerdo con el modelo conceptual de la IN, una red estructurada como red inteligente se puede describir por medio de cuatro planos diferentes:

1. *El Plano de Servicios (PS)*: describe los servicios desde el punto de vista del usuario; representa una visión exclusivamente orientada a los servicios en la que todo lo que se percibe es el comportamiento de la red en relación con el servicio visto por un usuario del mismo.
2. *El Plano Funcional Global (PFG)*: describe los servicios como una sucesión de bloques elementales, independientes del servicio y reutilizables, que los conforman de acuerdo con la lógica específica de cada uno de ellos. No se tienen en cuenta aspectos de distribución. Modela una red estructurada como Red Inteligente como una sola entidad. Lo componen, un SIB especializado en el proceso de llamada básica y los demás SIB (Bloques de Construcción Independientes del Servicio) requeridos por el sistema, el punto de iniciación POI (Point of Initiation) que define el momento en el cual debe ejecutarse el servicio y

el punto de retorno POR (Point of Return) que es el momento en el cual se define que el servicio ha concluido. SIB se refiere a eventos y funciones invisibles, los cuales, al ponerlos juntos en una secuencia, dictaminan la forma individual del servicio o los servicios que deben operar.

3. *El Plano Funcional Distribuido (PFD)*: Describe las entidades funcionales del PFD. Se implementan en equipos físicos de la red. Modela una perspectiva distribuida de una red estructurada como red inteligente. Aquí se pueden encontrar todas las entidades funcionales que actuarán dentro de la red y son las encargadas del entorno de la creación, gestión y control del servicio. Dentro de las entidades funcionales se encuentran: La entidad de Función Control de Llamada (CCF), la entidad de Función de Conmutación de Servicio (SSF), la entidad de Función de Control de Servicio (SCF), la entidad de Función de Datos de Servicio (SDF) y la entidad de Recursos especializados (SRF). En las recomendaciones de la ITU-T (Q1200 a la Q1219), se describen otras entidades relacionadas con la creación y gestión de servicio, como la entidad de Función de Entorno de creación de Servicio (SCEE) y la entidad de Función de Gestión de Servicio (SMF).
4. *El Plano Físico (PF)*: Describe las entidades funcionales del PFD de red y los protocolos utilizados para la comunicación entre ellas para la implementación de los servicios. Modela los aspectos físicos de redes estructuradas como red inteligente. El modelo identifica las diferentes entidades físicas y protocolos que pueden existir en redes estructuradas como RI. Indica también qué entidades funcionales se realizan y en qué entidades físicas.

Figura 1. Modelo de Inteligencia de Red



<http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol812/inteligencia/inteligencia.html>

El plano funcional distribuido (PFD) modela una visión distribuida de una red estructurada como IN. Consiste en entidades funcionales y relaciones entre ellas. Los bloques elementales de funcionalidad o Service Independent Building Blocks (SIBs), que se describen en el nivel global funcional, se realizan en el PFD como una secuencia de acciones ejecutadas en las entidades funcionales. Algunas de estas acciones generan flujos de información entre entidades.

Una entidad funcional es un grupo único de acciones en una sola localización y un subconjunto de todas las funciones necesarias para proveer un servicio. Las siguientes reglas se aplican a las entidades funcionales:

- Una, o más de una, entidad funcional pueden estar situadas en la misma entidad física.
- Diferentes entidades contienen diferentes funciones, y pueden contener una o más de las mismas funciones

- Una entidad funcional no puede estar dividida en dos entidades físicas diferentes.
- Instancias duplicadas de una misma entidad funcional pueden mapearse en distintas entidades físicas, pero no en la misma.

Cada relación representa el conjunto de interacciones entre la pareja de entidades funcionales que comunican. Cada interacción se especifica mediante un flujo de información. Si las entidades funcionales están situadas en distintas entidades físicas, la relación entre ellas define los requerimientos de transferencia de información para un protocolo de comunicaciones definido entre las entidades físicas. Una relación entre instancias de la misma entidad funcional se representa como un bucle.

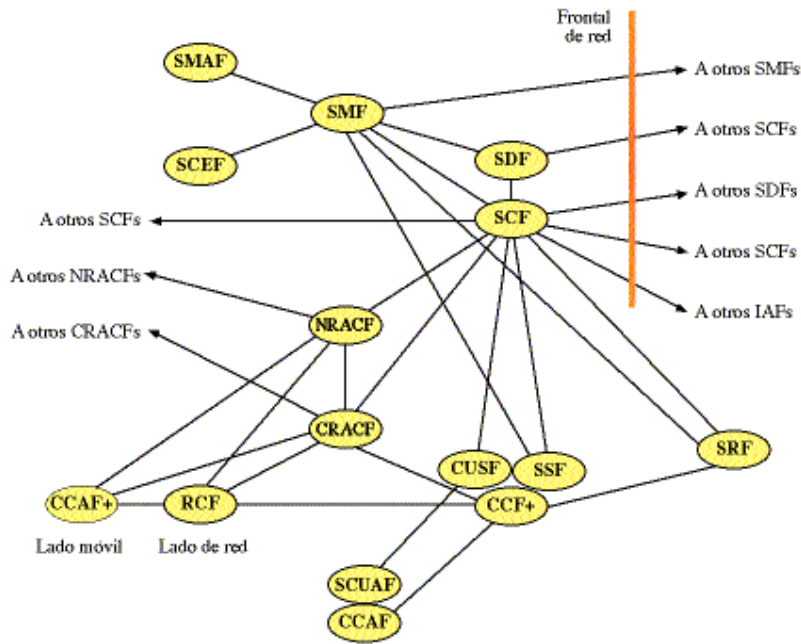
La IN se está especificando en fases, denominadas conjuntos de capacidades (CS). En IN CS1, las principales características son:

- Separación del proceso básico de llamada y el control del servicio.
- Independencia de la red de transporte y conmutación.
- Funciones de creación y gestión de servicios.
- Existencia de bloques elementales reutilizables, independientes de los servicios.
- Terminación y/o punto de control único.

En IN CS2, a punto de concluirse su estandarización, se soportarán funciones que faciliten la provisión de funciones de movilidad como servicios de IN. Entre otras:

- Función de conmutación de servicios no asociada a la llamada.
- Funciones de control asociadas a la red de acceso.
- Control distribuido (punto de control no único).
- Bases de datos distribuidas.

Figura 2. Modelo funcional de la arquitectura IN CS2



CCAF:	Call Control Agent Function
CCAF+:	Call Control Agent Function (Plus)
CCF:	Call Control Function (Plus)
CRACF:	Call / Connection related Radio Acces Control Function
IAF:	Intelligent Access Function
CUSF:	Call-unrelated User Service Function
NRACF:	Non-call / connection related Radio Acces Control Function
RCF:	Radio Control Function
SCEF:	Service Creation Environment Function
SCF:	Service Control Function
SCUAF:	Service Control User AccessFunction
SDF:	Service Data Function
SMAF:	Service Management Access Function
SMF:	Service Management Function
SRF:	Specialized Resource Function
SSF:	Service Switching Function

<http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol812/inteligencia/inteligencia.ht>

ml

1.2 APORTES Y SERVICIOS DE TELECOMUNICACIONES EN COLOMBIA

Algunos de los servicios de telecomunicaciones que los usuarios tienen a su disposición en la actualidad en Colombia son los de la red telefónica, y los servicios basados en difusión de señales o datos, con posibilidad de direccionamiento. Bidireccionales (como la radiotelefonía celular y su evolución hacia los servicios personales de comunicación, y la radiolocalización) y unidireccionales (tales como la radiolocalización de personas).

Existen muchas maneras de clasificar los servicios de telecomunicaciones, con el objeto de establecer un punto de partida, a continuación se resumen las características de las redes empleadas para ofrecer algunos servicios. En particular se utilizan los siguientes criterios de comparación:

a) **Tipo de red.** Se hablará únicamente de servicios ofrecidos al público en general, que utilizan como infraestructura redes públicas de telecomunicaciones, basadas fundamentalmente en transmisiones de radio o en señales guiadas por medio de conductores eléctricos u ópticos.

b) **Cobertura.** La extensión del área geográfica que cubre una red es de particular interés en la comparación, ya que los servicios no pueden ser ofrecidos fuera de dicha área geográfica. La cobertura puede ser caracterizada como local, regional o nacional.

c) **Interconexión.** A pesar de que la cobertura de una red puede ser local o regional, si está interconectada con otras redes de mayor cobertura se amplía de manera automática el área geográfica cubierta por la red. También es importante, el hecho de poder tener acceso a servicios prestados por otras redes interconectadas a la red a la que el usuario tiene acceso.

d) **Direccionalidad.** En una comunicación un usuario puede tener un papel pasivo o uno activo. Se ha incluido esto en el análisis, caracterizándolo por

medio de unidireccional (receptor pasivo) o bidireccional (el receptor tiene un papel activo y también puede transmitir).

e) **Punto-multipunto.** El criterio acerca de los posibles destinos para un servicio se relaciona con varios de los aspectos anteriores. Se han considerado dos opciones: P-P (punto a punto), en la cual existe un solo transmisor y un solo receptor, y P-MP (punto a multipunto), donde hay un solo transmisor pero una cantidad distinta de uno (posiblemente ilimitada) de receptores.

f) **Tipo de información.** Se ha mencionado frecuentemente que la información que se transmite puede ser digital o analógica, lo cual define algunos aspectos del alcance de un servicio; este es otro criterio que se considera digno de mención. Cabe recordar que si se trata de información tipo digital se estaría en posibilidad de tener los beneficios de las comunicaciones digitales, tales como la criptografía digital, la corrección de errores, la compresión del ancho de banda y el procesamiento por medio de microprocesadores de alta velocidad.

g) **Privacía.** Normalmente cuando se hace uso de un servicio de telecomunicaciones se desea tener la certeza de que sólo aquellos usuarios a quienes está destinada la información la reciben, y de que ningún intruso puede tener acceso al servicio sin tener autorización para ello; la privacía que se proporciona a los usuarios en cada servicio es distinta, por lo cual se considera que también es un factor que debe ser considerado.

Evidentemente existen elementos adicionales que podrían ser incluidos por ejemplo, costos del servicio, costos de los equipos terminales, y características del canal de acceso.

Gracias al adelanto de la tecnología, los nuevos sistemas de comunicaciones han sido encargados de darnos día a día los aportes de nuevos servicios, sin embargo el creciente adelanto tecnológico nos permite encontrar en el mercado nuevas tecnologías que desarrollan una forma sencilla y ofrece servicios personales de comunicación tales como comunicación de voz y datos, sin mencionar la aparición de servicios de valor agregado que invaden nuestro mundo.

Algunos ejemplos de servicios de telecomunicaciones básicas:

- (a) Servicios de teléfono
- (b) Servicios de transmisión de datos con conmutación de paquetes
- (c) Servicios de transmisión de datos con conmutación de circuitos
- (d) Servicios de télex
- (e) Servicios de telégrafo
- (f) Servicios de facsímil
- (g) Servicios de circuitos privados arrendados
- (h) Otros servicios
 - Servicios analógicos/digitales de telefonía móvil/celular
 - Servicios móviles de transmisión de datos
 - Servicios de radiobúsqueda
 - Servicios de comunicación personal
 - Servicios móviles por satélite (incluidos, por ejemplo, telefonía, datos, radiobúsqueda y/o servicios de comunicación personal)
 - Servicios fijos por satélite
 - Servicios de terminales de muy pequeña abertura
 - Servicios de estación terrestre de acceso
 - Servicios de teleconferencia
 - Servicios de transmisión de vídeo
 - Servicios de radiotelefonía con concentración de enlaces

1.3 SERVICIOS DE TELECOMUNICACIONES DE VALOR AÑADIDO

Son servicios de telecomunicaciones con valor añadido servicios los cuales los proveedores “añaden valor” a la información de los clientes, mejorando su forma o contenido o mediante su almacenamiento y recuperación. Algunos de ellos son:

- (a) Procesamiento de datos en línea
- (b) Almacenamiento y recuperación de datos en línea

- (c) Intercambio electrónico de datos
- (d) Correo electrónico
- (e) Correo vocal

2. PARLAY OSA SERVICIOS INALÁMBRICOS

2.1 PARLAY OSA

Parlay OSA es una interface para la programación de aplicaciones (API) la cual facilita la rápida creación de servicios de las telecomunicaciones.

Los API de Parlay OSA los define el Grupo Parlay (<http://www.parlay.org>) que es un consorcio superior a 65 compañías que representan las Telecomunicaciones y las Industrias de Tecnologías de la Información (TI). El Grupo Parlay se fundó en 1999, y desde entonces ha producido cuatro versiones de las especificaciones de Parlay, cada versión es construida sobre la última, e incorpora nuevos requerimientos y retoma implementaciones de la industria.

OSA o Arquitectura de Servicios Abierta, se refiere a la arquitectura para servicios móviles desarrollados por el Proyecto de Cooperación en 3G o 3GPP (<http://www.3gpp.org>), también adoptado por 3GPP2.

Los API de Parlay OSA son una tecnología independiente: ellos fueron diseñados para ser usados en las redes móviles, redes fijas y para las redes de la próxima-generación basadas en el protocolo IP. Igualmente, los APIs de Parlay OSA pueden usarse por diseñadores que trabajen en varios lenguajes de programación, como C, C++ y Java.

Parlay OSA es basado en estándares abiertos como CORBA, IDL, Java, UML, y Web, los servicios (SOAP, XML y SDL).

2.2 LA ARQUITECTURA DE PARLAY OSA

El intento del Grupo Parlay cuando fue fundado, era animar la convergencia de la computación y las telecomunicaciones. Una manera de hacer esto es adoptando la tecnología usada por desarrolladores de las aplicaciones de TI para simplificar y acelerar el desarrollo de servicios de comunicaciones.

En las telecomunicaciones y redes de computadoras, los protocolos han jugado un papel importante. Algunos protocolos poseen diferentes tipos de comunicaciones y funciones en las diferentes capas. Las pilas de protocolos individuales definen la forma en la cual se comunican los elementos de la red. Las pilas de protocolos conocidas incluyen TCP/IP el cual posee HTTP, TCP y protocolos IP y muchos otros. En las telecomunicaciones, la señalización No. 7, incluye protocolos como los ISUP, TUP, MTP-3, MTP-2 y protocolos de MTP-1. La definición y estandarización de estos permite garantizar la interoperabilidad entre los elementos de la red, y es una consecuencia para construir las redes con múltiples equipos de diversos vendedores.

En el desarrollo del software, el enfoque ha estado en interfaces entre las capas individuales. Los ejemplos de interfaces de la programación común incluyen POSIX (APIs usados en todos los Servidores Unix y sistemas Linux), y WIN32 (APIs en Microsoft Windows 95/98/2000 etc.).

Uno de los beneficios de usar los APIs comunes es que asegura que las aplicaciones pueden ser portadas por las nuevas plataformas fácilmente. Independientes del Hardware y del vendedor.

Una medida del éxito de este acercamiento se ve cuando consideramos que hay alrededor de 3 millones de diseñadores de software profesionales en todo el mundo lo cual contrasta con que hay menos de 10000 personas especializadas en desarrollo de servicios de Telecomunicaciones.

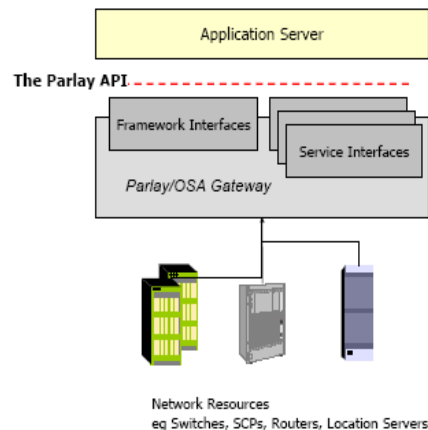
Antes del advenimiento de Parlay OSA, organizaciones de desarrollo de normas en las telecomunicaciones, como TIA, ITU-T y ETSI, habían enfocado hacia las especificaciones de protocolos. Usando el acercamiento de los APIs en los medios de comunicaciones, es posible escribir aplicaciones portátiles que correrán en una variedad de protocolos subyacentes sin cambiar.

2.3 FRAMEWORK DE PARLAY OSA

Uno de los requisitos importantes de los operadores de la red y proveedores de servicio desde el principio de Parlay OSA ha sido asegurar la apertura a la red definiendo un API, sin exponer la infraestructura de comunicaciones al uso desautorizado o a la orden de nuevas amenazas. Esta es la función del Framework de Parlay OSA.

El Framework es implementado en la entrada (Gateway) de Parlay OSA. Todas las aplicaciones y servicios que quieren usar los APIs de Parlay OSA primero tienen que registrarse con el Framework de Parlay OSA. El Framework de Parlay OSA (ver figura 3), es un componente de software que criptográficamente autentica la aplicación, y retorna el objeto referente a la aplicación para esas funciones de Parlay OSA, para que el proveedor de servicio les permita ser usadas. El Framework de Parlay resuelve varios problemas de seguridad y de disponibilidad.

Figura 3. Framework Parlay

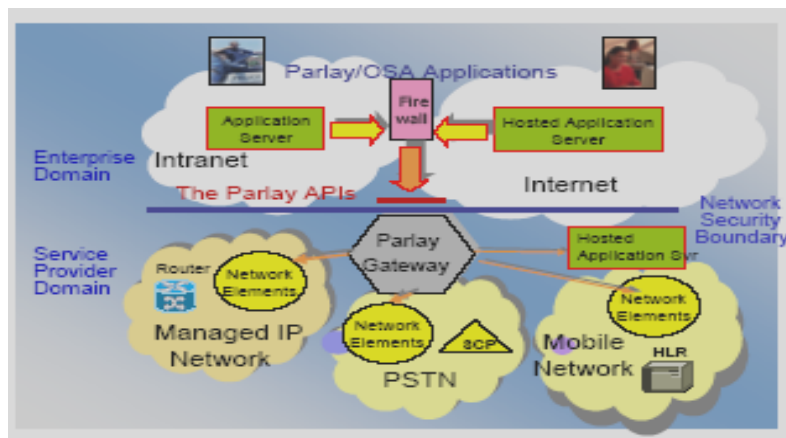


Fuente www.parlay.org

2.4 OSA PARLAY EN LA RED

Parlay OSA trabaja como un perfeccionamiento a la red actual (ver figura 4) o las redes usadas por un proveedor de servicio, busca integrar los diferentes servicios prestados por redes móviles, redes fijas, Internet, Intranet con el fin de permitir la interoperabilidad entre diferentes tecnologías permitiendo a las empresas que no son dueñas de las redes entrar a competir con servicios de valor agregado mediante el alquiler de un espacio en el servidor de cualquier operadora utilizando las APIS de PARLAY OSA y sin conocer a fondo la arquitectura de cada tecnología poder desarrollar aplicaciones que se puedan utilizar bien desde Internet o con un teléfono fijo o móvil.

Figura 4. Parlay OSA



Fuente www.parlay.org

2.5 EL PAPEL DEL GATEWAY DE PARLAY OSA

El modelo de Parlay OSA agrega un nuevo elemento de la red la Entrada (Gateway) de Parlay OSA que es usada para unir aplicaciones que usan los APIs de Parlay OSA con los existentes elementos de la red. El Gateway de Parlay OSA está bajo el control de la red, operador o proveedores de servicio, y es un solo punto a través del cual todas las interacciones de Parlay OSA pasan. Esto significa que las aplicaciones son aisladas desde los protocolos específicos usados dentro de la red por el operador, y las redes pueden ser evolucionadas sin afectar las aplicaciones y servicios existentes. El Gateway

de Parlay/OSA es el elemento que implementa el Framework de Parlay. Este acercamiento todavía les proporciona flexibilidad a vendedores en como ellos implementan el Gateway. Muchos vendedores optan por un Gateway autosuficiente, que une los elementos de la red individuales como SSPs, Servidores de Información de Localización y Políticas de Administración, equipos que usan los protocolos nativos SS7 / INAP, LIF ejemplos de este son el Gateway de Recurso de Red de Ericsson, Lucent MiLife Parlay Gateway y AePONA Parlay Gateway. La flexibilidad de acercamiento que otros vendedores tienen integrando el Gateway de Parlay OSA dentro de una existente plataforma de red inteligente (IN), como un SCP. Los ejemplos de este acercamiento incluyen el Gateway de Marconi's Aplicación Mediation que es un componente del Sistema-X SCP, y los Servicios Abiertos de Tecnología de Telcordia Gateway, que es un componente de la plataforma de ISCP.

2.6 CONECTANDO LOS SERVIDORES DE APLICACIÓN EN LA RED DE LA EMPRESA

La arquitectura de aplicaciones de la Empresa ha convergido en un modelo el cual es usado a través de la industria de TI. Las aplicaciones del cliente (Basadas en Web, corriendo en PCs) interactúan con aplicaciones comerciales localizadas en los servidores de aplicaciones Web que a su vez hace uso de datos de los grandes sistemas informáticos. Uno de los elementos claves de este acercamiento es el servidor de la aplicación que es un sistema del software altamente disponible el cual soporta muchas aplicaciones Web al mismo tiempo. Parlay OSA asume que las aplicaciones correrán en un servidor de aplicación, y conectado al Gateway de Parlay OSA sobre una red TCP/IP. Hay dos escenarios. En el primero, las aplicaciones y los servidores de la aplicación están en un dato corporativo central, soportando aplicaciones comerciales, y usando las facilidades de los servicios que proveen las redes de las comunicaciones, a través de los APIs de Parlay OSA para preguntar la localización de un camión o un ingeniero de campo. En el segundo, las aplicaciones están organizadas por el nombre de la empresa por una tercera parte. Esta podría ser una Aplicación de Proveedor de Servicios (ASP). Las

aplicaciones todavía usan los APIs de Parlay OSA para interactuar con los proveedores de servicios de la red.

2.7 EL PAPEL DE LOS SERVIDORES DE APLICACIONES EN LA RED

Los proveedores de servicio también quieren tomar ventaja de los beneficios del Parlay OSA acercando los servicios que ellos ofrecen a los subscriptores. Varios proveedores de servicio han desplegado los servidores de aplicación dentro de la red, el cual ejecuta los servicios desarrollados por terceras partes y que conecta a un Gateway estándar de Parlay OSA. Esto tiene el beneficio que el proveedor de servicio está usando la misma interface para su propios servicios así como aquéllos ofrecidos por terceras partes.

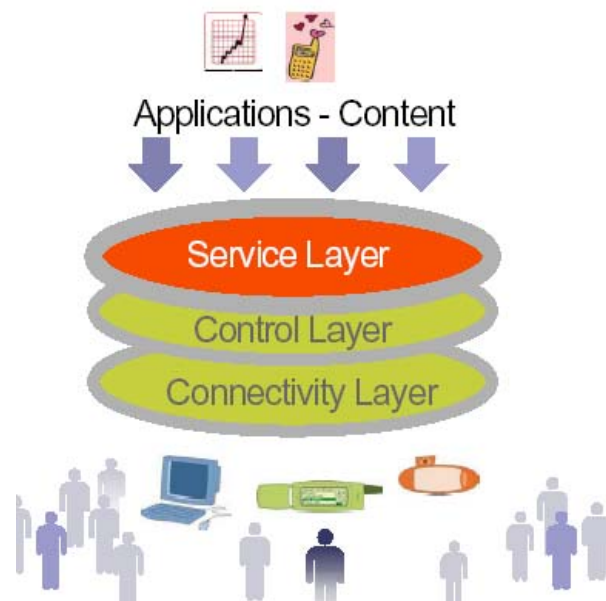
3. HERRAMIENTA ERICSSON NRG COMO DESARROLLO DE SERVICIOS PARLAY OSA.

El concepto de **OSA** consta de tres capas la capa superior es la capa de servicio el Ericsson Network Resource Gateway (NRG) es localizado en esta capa.

Los elementos de redes de telecomunicaciones son localizados en la capa de control y la capa de conectividad, la capa de control manipula la señalización y la capa de conectividad manipula el tráfico (ver figura 5).

El operador usa el NRG para crear un enlace entre las aplicaciones y las redes de telecomunicaciones. El operador hará sólo esto cuando el operador sabe efectivamente que la red de telecomunicaciones no puede emplearse mal. El NRG esconde muchas de las complejidades de las redes de telecomunicaciones proporcionando las interfaces PARLAY OSA.

Figura 5. Elementos de Redes de Telecomunicaciones

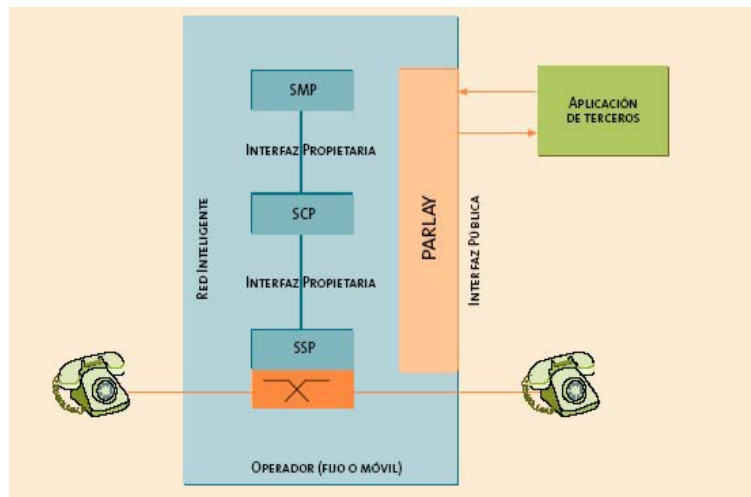


Fuente Understanding Telecommunications Tomo I y II, Ericsson 1997

3.1 PARLAY.

La interfaz Parlay es la definición de una interfaz estandarizada y abierta (ver la Figura 6), para permitir el acceso a los recursos de las redes de los operadores a aplicaciones que no tienen por qué residir en el dominio del operador. Esta interfaz, además de permitir el acceso a los servicios de la red del operador de telecomunicaciones, define unos servicios para asegurar que el acceso a los recursos se realice de forma controlada y sólo a aquellas zonas que permita el operador de la red.

Figura 6. Interfaz Parlay



Fuente Understanding Telecommunications Tomo I y II, Ericsson 1997

3.2 PARLAY OSA.

En pocas palabras PARLAY OSA es una forma de simplificar la vida de los desarrolladores, encapsulando la infraestructura de la red de los operadores, tanto móviles como fijas. El objetivo de PARLAY OSA es hacer más transparente la complejidad de la red desde la óptica del desarrollador, facilitando acceso a funcionalidades de la red a través de la biblioteca de funciones abiertas al dominio publico. Y así eliminando las necesidades de conocer las características de los elementos de la infraestructura y lo que tiene

el operador, resultando en una disminución drástica de tiempo de desarrollo de aplicaciones.

3.3 PARLAY HOSA. ALTO NIVEL DE ACCESO DE SERVICIO ABIERTO

El NRG se ha reforzado por un superset de Ericsson de Parlay OSA, Alto nivel de Acceso de Servicio Abierto (HOSA). Los API de HOSA son las extensiones estándar de Parlay OSA con funcionalidades y servicios adicionales.

Las cosas siguientes son posibles con los servicios de Parlay HOSA del NRG:

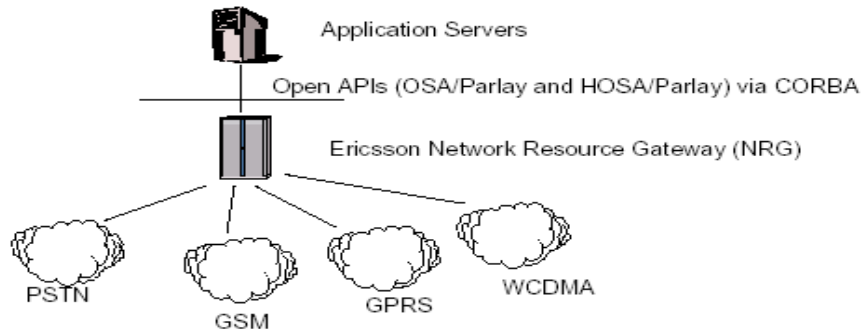
- Enviar mensajes usando el servicio de Mensajería.
- Leer los e-mails usando el servicio de Recuperación de Mensaje.
- Pedir el estado de los teléfonos móviles usando el servicio de Estado del Usuario.
- Pedir la localización de los teléfonos móviles usando el servicio de Localización del Usuario.
- Manipular los contactos suscritos usando el PIM servicio de Contacto.
- Manipular los calendarios del suscriptor usando el PIM Servicio del Calendario.
- Crear las rutas de llamadas usando el Multy Party Call del servicio de contacto

3.4 NETWORK RESOURCE GATEWAY (NRG).

Ericsson Network Resource Gateway es un servidor de capacidad de servicio, que proporciona las capacidades a la aplicación de una manera segura (ver figura 7).

El NRG fue construido con seguridad asegurando el acceso a los elementos de las redes de telecomunicaciones haciendo negocios en términos seguros el NRG apoya una gama amplia de protocolos de la red.

Figura 7. Ericsson Network Resource Gateway

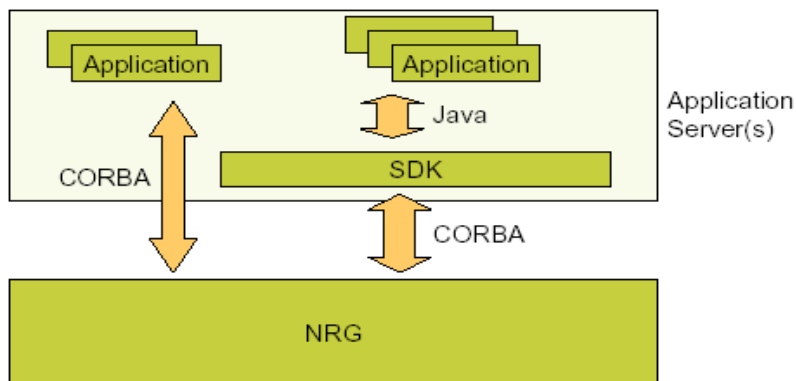


Fuente Understanding Telecommunications Tomo I y II, Ericsson 1997

3.5 SDK.

NRG Software Development Kit (SDK) ofrece a los diseñadores el software de librerías de java, el cual simplifica el desarrollo de aplicaciones que usan las capacidades de NRG (ver figura 8). Uno de los rasgos del SDK es la abstracción de CORBA. El SDK puede ser usado para el desarrollo de aplicaciones hacia el NRG, el SDK es un Kit basado en Java el cual simplifica el desarrollo y disminuye los tiempos para el desarrollo de la aplicación.

Figura 8. NRG Software Development Kit (SDK)

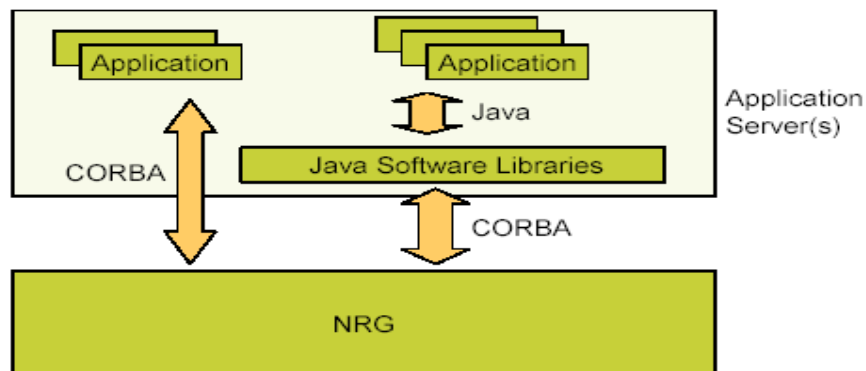


Diseñadores pueden escoger programar directamente hacia NRG usando CORBA, permitiendo mayor flexibilidad y programando con el lenguaje escogido, pero teniendo que tratar con todas las implicaciones de CORBA. Alternativamente el SDK, ofrece una interface abstracta de Java desde CORBA y unos componentes adicionales para acelerar el desarrollo de la aplicación. Además de la abstracción de CORBA, el SDK ofrece también la funcionalidad de apoyar los sistemas de NRG redundantes (2 nodos de NRG en paralelo) y otras clases de utilidades.

3.5.1 SOFTWARE LIBRERÍAS DE JAVA.

Las librerías de Java simplifican el desarrollo de aplicaciones que usan las capacidades de NRG. Uno de los rasgos importantes de estas bibliotecas es la abstracción del protocolo de comunicación (CORBA). Esto les permite a programadores de las aplicaciones usar los APIs de Java (ver figura 9). También significa que es transparente a una aplicación de SDK si está comunicando con un solo nodo de NRG o con un cluster de nodos de NRG.

Figura 9. JAVA librerías de Software



Al lado del beneficio principal por la abstracción de CORBA las librerías del Software proporcionan los medios para supervisar todas las comunicaciones entre NRG y la aplicación.

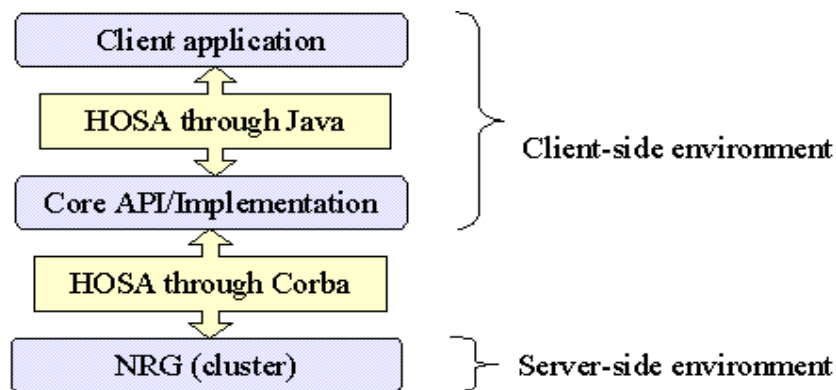
El NRG SDK esta diseñado para programadores que son experimentados en Java principalmente, y que tiene poca experiencia con CORBA o ambientes de J2EE.

Dentro del software de librerías para Java se encuentran las siguientes librerías:

API CORE.

La API Core ofrece una implementación en Java del protocolo HOSA (ver figura 10). Cada aplicación del cliente tiene su propio caso del API de Core (y su implementación) que se une a la aplicación del cliente al tiempo de correrla (para que la aplicación y la implementación del API estén corriendo en la misma máquina virtual Java).

Figura 10. API CORE



La aplicación del cliente usa el API core para traducir todos los tipos de datos de Java y las invocaciones del método a los tipos de datos de CORBA . Esto es transparente para la aplicación del cliente si se está comunicando con un solo nodo de NRG, o un cluster de nodos de NRG (ese actúa como una copia reciproca de los nodos).

API UTILITY.

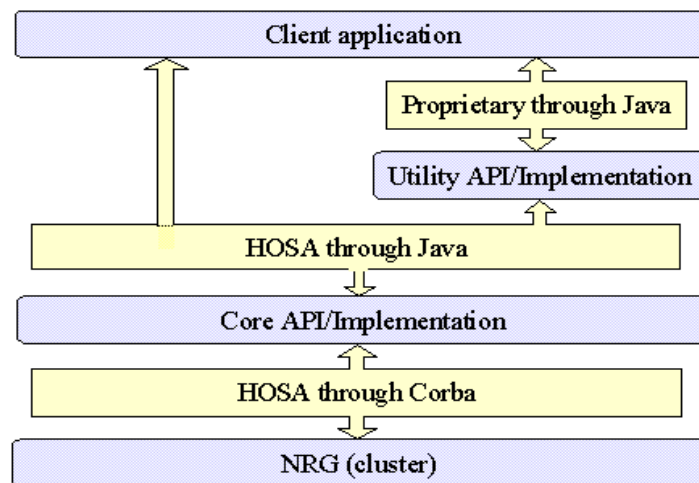
El API Utility va un paso más allá ofrece una abstracción no sólo desde CORBA sino también HOSA. Como opuesto al API Core, que estrechamente sigue el estándar de HOSA, el API Utility es propietario.

Por ejemplo, para ganar el acceso a un servicio, deben seguirse una autenticación específica y un procedimiento de la autorización. Esto produce una cantidad justa de métodos para ser implementados por las aplicaciones. Usando el API Utility, la aplicación del cliente puede establecer el mismo resultado por la vía de un API simplificado que consta en sólo un método. La implementación del API Utility es responsable para invocar los métodos de HOSA apropiados.

Considerando que los objetivos del API Core para la integridad y conformidad de la especificación de HOSA, el API Utility apunta para las simplificaciones de (las partes de) HOSA. Por consiguiente el API Core y el API de Utility no se excluyen, sino se complementan. Ellos pueden usarse simultáneamente por una aplicación cliente, como se muestra en la figura 11.

El API Utility es construido encima del API Core. Por consiguiente, el API Utility y el API Core son totalmente compatibles.

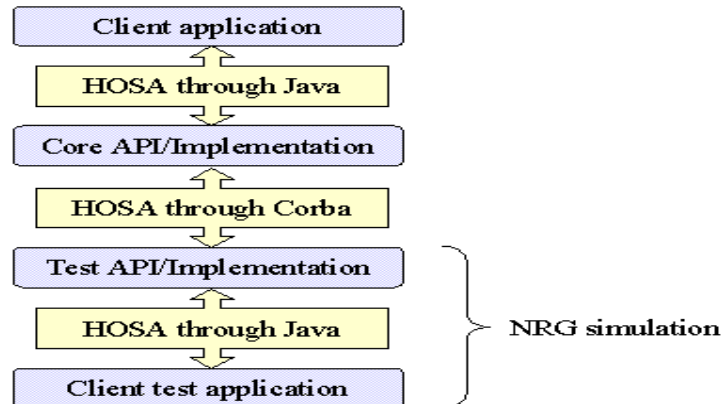
Figura 11. API Utility



API TEST.

Las herramientas están disponibles para ejecutar una aplicación en la ausencia real de un nodo de NRG . Estas herramientas son suficientes para la mayoría de los propósitos de la demostración. El papel del API Test es comparable al API Core: los dos son una abstracción de CORBA y ofrecen una interface de HOSA a través de Java. El API Core es usado para aplicaciones que usan NRG, considerando que el API Test se usa para aplicaciones que emulan o simulan NRG. La figura 12 muestra cómo se posiciona del API Test. El diseñador de la aplicación Test es responsable de asignar la conducta a la interface HOSA (es decir para proporcionar la aplicación Test al cliente). Note que la NRG Herramienta Automatizado Test es un ejemplo de una predefinida aplicación Test cliente. Este usa el API Test para evitar tener que preocuparse con la comunicación detallada, permitiéndole así enfocarse en el comportamiento de HOSA.

Figura 12. API Test



3.6 SIMULADOR NRG.

3.6.1 REQUERIMIENTOS DEL SISTEMA.

El Simulador NRG puede ser usado bajo los siguientes sistemas operativos: Linux, Solaris SPARC/Intel o Microsoft Windows 95, 98 (1era o 2da edición), NT 4.0, ME, 2000 Professional, 2000 Server, 2000 Advanced Server, o XP corriendo sobre hardware Intel.

Es recomendado un Procesador Pentium 400MHz como mínimo o superior y al menos 80 megabytes de memoria RAM .

NRG requiere Java 1.3.1 and Erlang R9C runtime environments. Si el anterior software ya esta instalado, se debe tener 18 megabytes de espacio libre en disco antes de instalar el Simulador NRG. Java y Erlang requieren aproximadamente 70 megabytes.

La disponibilidad de un browser HTML se recomienda Windows Explorer. el Simulador NRG no es compatible con Java 1.4 o superior. Si se tiene instalado Java 1.4 o superior, se necesita instalar Java 1.3 para e.l buen uso del Simulador NRG. Java 1.3 y Java 1.4 pueden coexistir en el mismo computador.

3.6.2 INSTALACIÓN AUTOMÁTICA.

Este procedimiento de instalación automática solo aplica para sistemas operativos Windows se asume que se tiene el instalador con el nombre NRG simulator.exe.

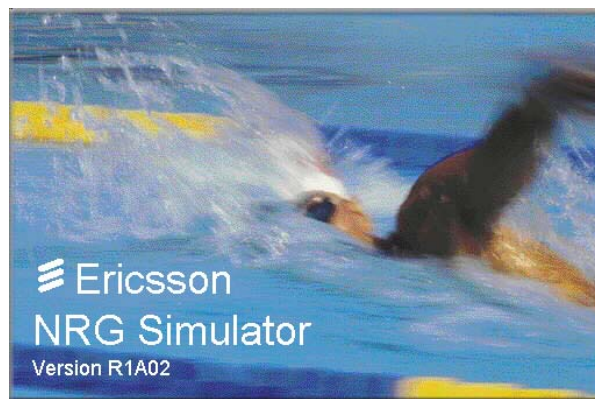
La instalación automática consta con los siguientes pasos:

Comience la instalación y siga las instrucciones de la instalación, el instalador contiene los paquetes de instalación de Java 1.3.1_10, Ericsson Simulador NRG y Erlang R9C-0. Inicialmente se instala el paquete Ericsson Simulador NRG luego se instala Erlang R9C-0 y por último el paquete de Java 1.3.1_10 después de instalados todos los paquetes ya puede empezar a utilizar el Simulador NRG.

3.6.3 INICIAR EL SIMULADOR NRG.

Cuando el simulador NRG ha iniciado se mostrara la siguiente pantalla (ver Figura 13).

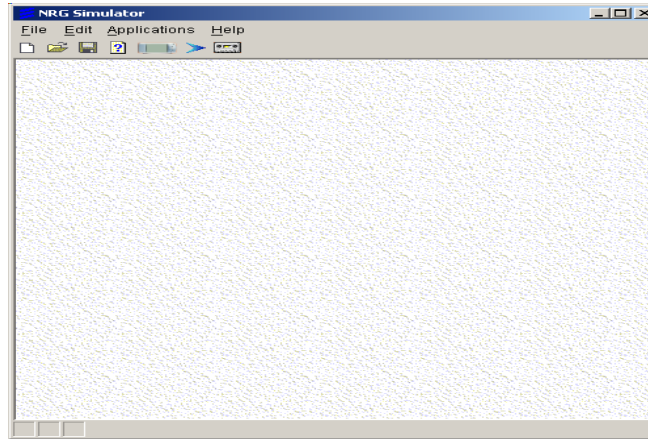
Figura 13. SIMULADOR NRG



Fuente www.ericsson.com

Cuando el simulador ya está listo la pantalla se cierra y aparece la ventana principal (ver la Figura 14).

Figura 14. Pantalla Principal NRG



Fuente tomada del software en ejecución

El área manchada del área principal se usa para organizar las entidades como los teléfonos virtuales. Estos teléfonos pueden usarse para simular la conducta del usuario, como un usuario llamando o enviando un SMS a otro usuario.

3.6.4 ADICIONANDO TELÉFONOS

Para adicionar un teléfono vaya al menú Edit y seleccione Add Phone, alternativamente también puede hacer clic en la barra de herramientas en el botón de teléfono (☎). Se pueden adicionar varios teléfonos con distintos números en la Figura 15 se adicionaron dos teléfonos con números 1 y 2 para realizar la simulación, también se pueden introducir números como los de un celular de alguna operadora conocida, por ejemplo los de Colombia Móvil 3002141202 etc.

Los números de teléfono se ubican en la parte superior, si un teléfono se selecciona este se rodea por un paréntesis en la Figura 14 se encuentra seleccionado el número 1. Cuando un teléfono se selecciona, se muestra información sobre este teléfono en la barra de estado. Por ejemplo, cuando el teléfono seleccionado se arrastra a una nueva situación (usando la ventana del

Mapa), la barra de estado se pone al día automáticamente con la nueva situación geográfica de este teléfono (expresado como la latitud y longitud).

Los iconos en los teléfonos representan las funciones:



Se utiliza para hacer una llamada a otro teléfono.

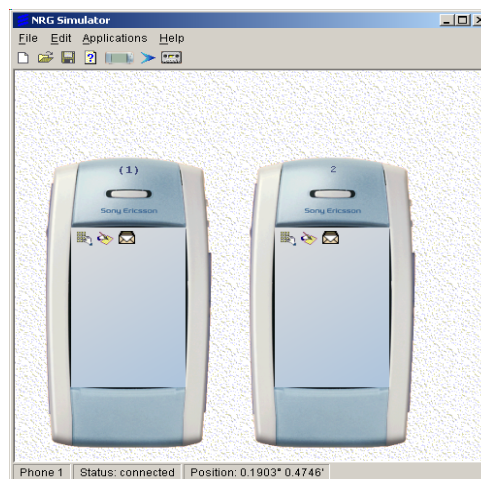


Se utiliza para enviar mensajes SMS a otro teléfono.



Se utiliza para ver todos los mensajes (SMS y MMS) mensajes recibidos por éste teléfono.

Figura 15. NRG anexo de Teléfonos



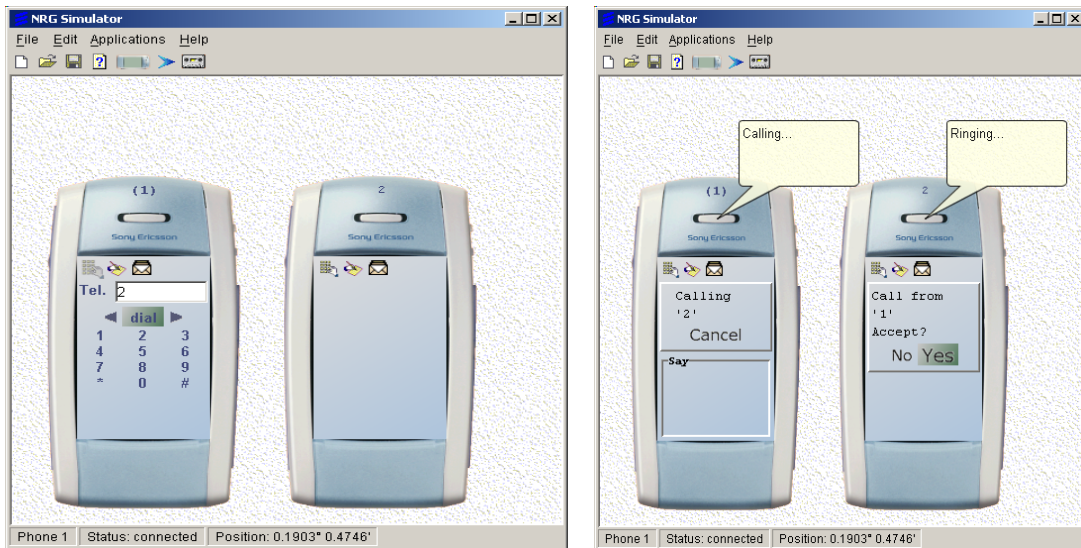
Fuente tomada del software en ejecución

3.6.5 HACER UNA LLAMADA.

Para simular una llamada desde el teléfono 1 al teléfono 2, haz clic sobre el teclado del teléfono 1 un dial aparece en el teléfono, entre el numero 2 correspondiente al número telefónico que desea llamar cuando presione el botón Dial una llamada se produce y el teléfono 2 timbra. (ver figura 16).

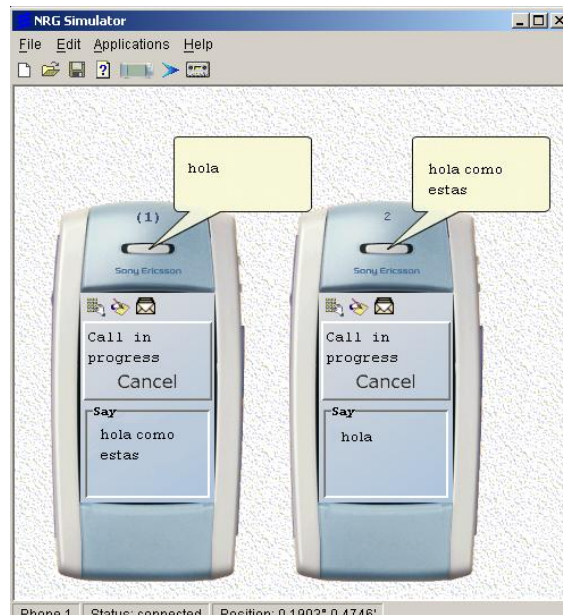
El usuario 1 puede abandonar(Cancel) la llamada, o el usuario 2 puede aceptar(yes) o rechazar(no) la llamada. Si la llamada es aceptada (ver figura 17), se establece una conexión entre los dos teléfonos. Esta conexión puede ser verificada si en el área de texto se puede escribir el mensaje en el ejemplo el usuario1 escribe hola y el usuario 2 visualiza el texto digitado en la etiqueta Say y viceversa.

Figura 16. NRG realizando llamadas



Fuente tomada del software en ejecución

Figura 17. NRG Llamada establecida



Fuente tomada del software en ejecución

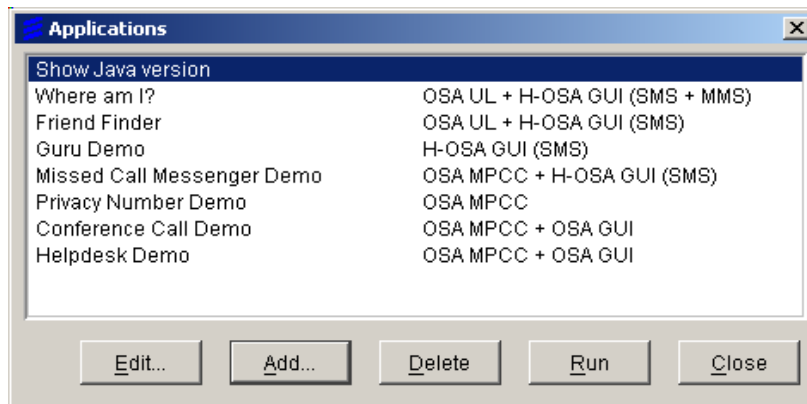
Cualquier usuario puede finalizar la llamada presionando Cancel.

3.7 EJECUTANDO APLICACIONES PARLAY OSA.

Llamadas a otro teléfono o envíos de SMS son los servicios ordinarios de la red. Sin embargo las cosas se ponen más interesantes cuando las aplicaciones de Parlay/OSA se involucran monitoreando o controlando lo que está pasando en la red. Esas aplicaciones pueden ser ejecutadas desde la línea de comandos, pero ellas también pueden ser ejecutadas vía Comando Run por el

menú de aplicaciones, este comando abre la ventana Run (ver figura 18), por defecto esta ventana muestra unas aplicaciones demo que son parte del paquete Simulador; después puede seleccionar una aplicación demo y presionar el botón Run para ver la aplicación.

Figura 18. Aplicaciones NRG



Fuente tomada del software en ejecución

3.7.1 EJECUTANDO LA APLICACIÓN WEBDIAL

3.7.1.1 DESCRIPCIÓN DE LA APLICACIÓN WEBDIAL

La aplicación de WebDial demuestra el uso de OSA MPCC (Multi Party Call Control). La aplicación usa este servicio para implementar la aplicación de inicialización de llamadas; el usuario de la aplicación puede crear una llamada entre un número arbitrario de usuarios. El código fuente de la aplicación puede verlo en el Anexo A.

3.7.1.2 CORRIENDO LA APLICACIÓN.

Al ejecutar la aplicación se muestra la siguiente ventana (ver figura 19)

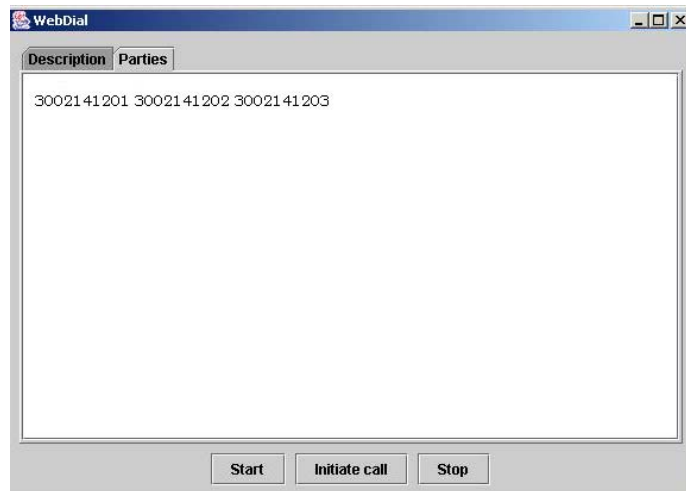
Figura 19. Ejecutando Aplicación Web dial



Fuente tomada del software en ejecución

Para especificar entre cuales usuarios se desea comenzar la llamada,(ver figura 20) se debe , abrir la pagina Parties (pulse el botón en la etiqueta de la página y entre los números de teléfonos con quien desee establecer la comunicación):

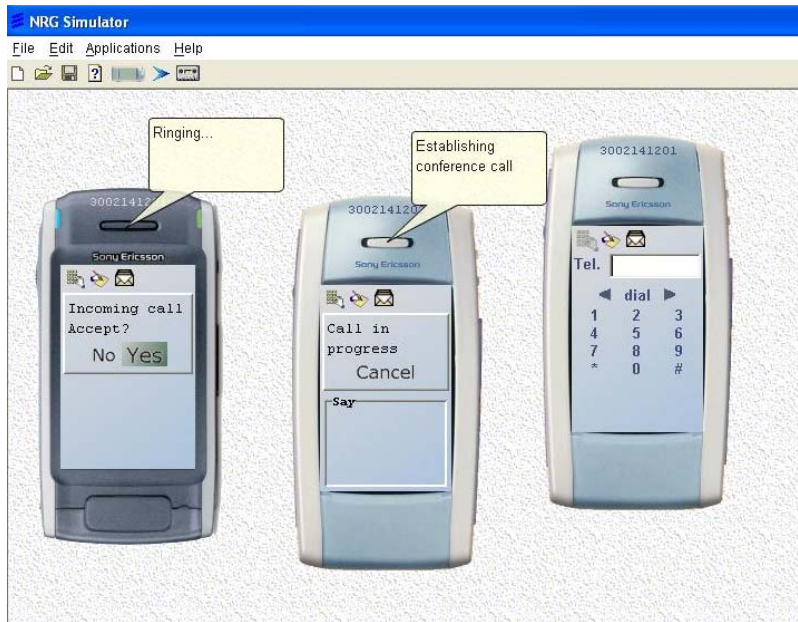
Figura 20. Especificación de usuarios



Fuente tomada del software en ejecución

Después de inicializar las llamadas los usuarios pueden empezar su conferencia de llamadas .(ver Figura 21) y en cualquier momento cualquier usuario puede finalizar su llamada y los demás usuarios continúan conectados hasta que lo deseen.

Figura 21. Conferencia de Llamadas



Fuente tomada del software en ejecución

3.7.1.5 DISEÑO DE DESCRIPCIÓN DE WEBDIAL.

Esta aplicación usa el diseño común explicado en el diseño de descripción común, ésta usa las siguientes clases específicas:

- MPCCProcessor: Usa OSA MPCC (Multi Party Call Control) para crear las llamadas.
- CallProcessor: Usa OSA MPCC (Multi Party Call Control) para crear y rutear legs para cada nueva llamada.
- LegProcessor: usa OSA MPCC (Multi Party Call Control) para recibir los resultados ruteados.

3.7.1.6 SECUENCIA MAIN.

El siguiente diagrama (ver la figura 22) muestra la secuencia de eventos cuando el usuario requiere la aplicación para crear una nueva llamada.

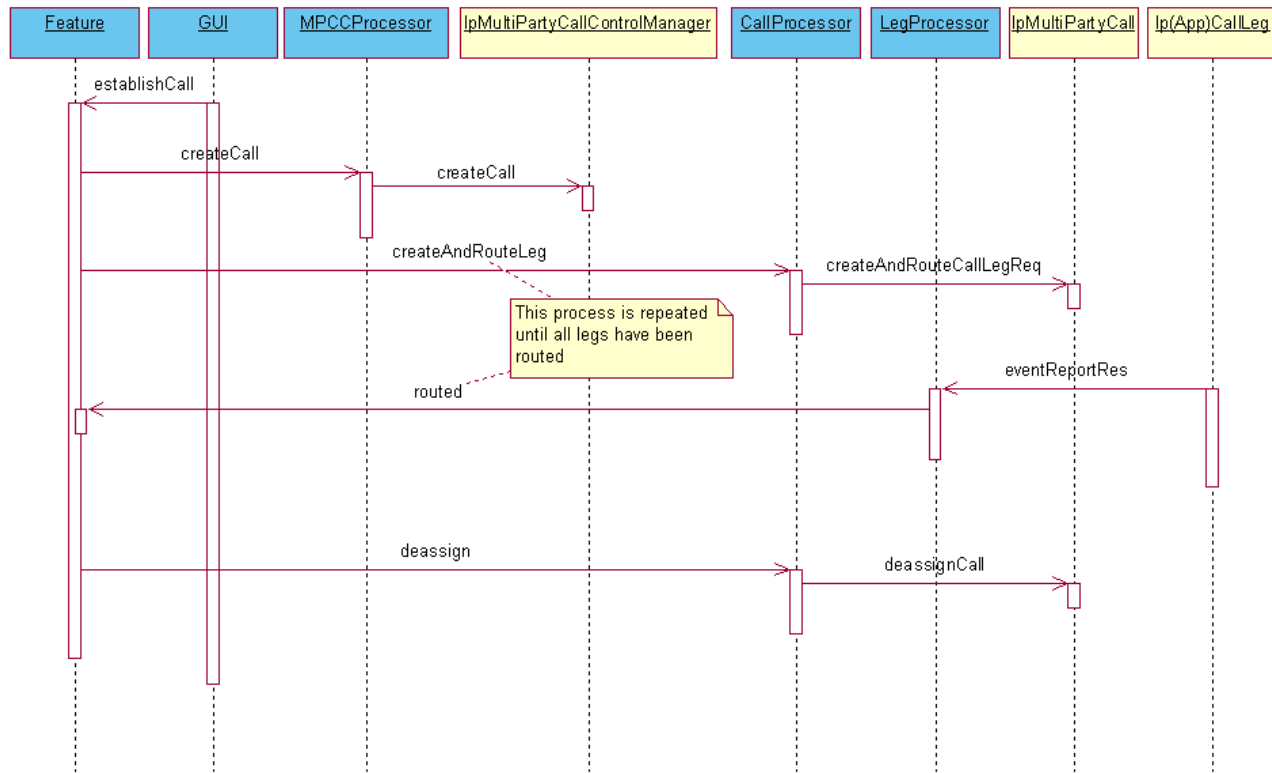


Figura 22. Secuencia de eventos cuando el usuario requiere la aplicación
Fuente www.ericsson.com

4. TECNOLOGIAS ACTUALES IMPLEMENTADAS EN COLOMBIA

Como mencionó en la sección 2.4, PARLAY OSA EN LA RED trabaja como un perfeccionamiento a la red actual, busca integrar los diferentes servicios prestados por redes móviles, redes fijas, Internet, o intranet con el fin de permitir la interoperabilidad entre diferentes tecnologías, a continuación revisaremos algunos conceptos de las redes móviles y mencionaremos algunas de las tecnologías y equipos mas comunes que se han implementado en este tipo de redes en Colombia en operadores como BELLSOUTH, COMCEL y COLOMBIA MOVIL sin embargo es preciso aclarar que la documentación técnica detallada de los equipos implementados es de difícil acceso ya que son datos exclusivos de las empresas y por tanto confidenciales.

4.1 ELEMENTOS DE REDES MOVILES

Una red móvil es constituida por tres elementos: el terminal, la estación base y el subsistema de red o nudo. Adicionalmente existen centros de operación establecidos por las operadoras para monitorizar el estado de la red.

La estación base BS controla la conexión entre el teléfono móvil y la red y es conocida también por célula ya que cubre un área geográfica determinada.

Una BS esta compuesta por dos elementos :

El BTS (Base Transceiver Station) y el BSC (Base Station Controler). Cada BS puede tener uno o mas BTS. Las BTS albergan el equipo de transmisión / recepción (los TRX o transceivers) y gestionan los protocolos de radio con el terminal móvil.

En áreas urbanas existen más BTS que en zonas rurales y en algunos casos con características físicas o geográficas particulares (como por ejemplo, túneles) son colocados retransmisores para garantizar el servicio.

Cada estación utiliza técnicas digitales para permitir que varios usuarios lleguen a la red, así como para permitir que hagan y reciban llamadas simultáneamente.

El BSC administra los recursos de radio de una o más BTS. Entre sus funciones se incluyen el handover (qué ocurre cuando el usuario se mueve de una célula para otra, permitiendo que la llamada se mantenga), y el establecimiento de los canales de radio utilizados y cambios de frecuencias. Finalmente, establece la llamada entre el móvil y el Mobile Service Switching Center (MSC), el cual es el corazón del sistema.

El MSC, como ya fue referido, es el centro de la red, a través del que es hecha la conexión entre una llamada realizada de un móvil hacia las otras redes fijas (las analógicas PSTN o digitales ISDN) o móviles. El nudo en el que se encuentra posee además una serie de equipos destinados a controlar varias funciones, como el cobro del servicio, la seguridad y el envío de mensajes SMS.

El Home Location Register (HLR) contiene toda la información administrativa sobre el cliente del servicio y la localización actual del terminal. Es a través del HLR que la red verifica si un móvil que se intenta conectar posee un contrato de servicio válido. Si la respuesta es afirmativa el MSC envía un mensaje de vuelta al terminal informándole que está autorizado a utilizar la red. El nombre de la operadora aparece entonces en pantalla, informando que se puede efectuar y recibir llamadas.

Cuando el MSC recibe una llamada destinada a un móvil él va al HLR a verificar la localización. Paralelamente, el terminal envía un mensaje para la red, para informarla del sitio donde se encuentra (este proceso es denominado polling).

El Visitor Location Register (VLR) es utilizado para controlar el tipo de conexiones que un terminal puede hacer. Por ejemplo, si un usuario posee restricciones en las llamadas internacionales el VLR impide que estas sean hechas, bloqueándolas y enviando un mensaje de vuelta al teléfono móvil informando al usuario.

El Equipment Identity Register (EIR) y el Authentication Center (AC) son utilizados ambos para garantizar la seguridad del sistema. El EIR posee una

lista de IMEI de terminales que han sido declarados como robados o que no son compatibles con la red GSM. Si el teléfono móvil está en esa lista negra, el EIR no permite que se conecte a la red. Dentro del AC hay una copia del código de seguridad del SIM. Cuando ocurre la autorización el AC genera un número aleatorio que es enviado para el móvil. Los dos aparatos, de seguida, utilizan ese número, junto al código del SIM y un algoritmo de encriptación denominado A3, para crear otro número que es enviado de nuevo para el AC. Si el número enviado por el terminal es igual al calculado por el AC, el usuario es autorizado a usar la red.

Por fin, el Short Message System Center (SMSC) es responsable por generar los mensajes cortos de texto. Otros equipos utilizados en redes GSM pueden adjuntar el recaudo de llamadas, la conexión a Internet, la caja de mensajes de voz, etc.

Las ventajas de CDMA son:

- Contempla un método de control de energía diseñado para el ahorro de la batería y para ayudar a que no hayan interferencias con otro canal. Así se establece una comunicación con el sitio celular receptor y el teléfono para mantener los niveles de potencia constantes y los mas pequeños posibles.
- En CDMA NO se emplean cristales, los cuales al oscilar crean problemas de RFI potenciales para otros equipos electrónicos.
- El handoff (pase entre celdas) es el convencional (fuerte) con uno suave adicional. Cuando el teléfono cruza la frontera de una celda, la celda original continua proporcionando servicio al teléfono. La nueva celda se activa y el teléfono funciona en ambos sitios celulares hasta alcanzar la suficiente intensidad de señal que la nueva celda pueda tomar.
- No hay degradación notable de la calidad de transmisión durante el handoff, lo cual es critico en la transmisión de datos.
- CDMA soporta servicios de datos, conmutación de paquetes y la integración de datos empaquetados digitales celulares(CDPD).

La desventaja de CDMA ante la presencia de equipos AMPS es la gran interferencia originada en los equipos CDMA, problema que ha sido ya rectificado.

El GSM presenta varias características importantes como las llamadas en espera, identificación de la llamada, retención de llamada, llamadas en conferencia y velocidades de datos de 2400 hasta 9600 bps.

Por otro lado, existen tres tipos diferentes de entidades celulares que utilizan GSM. Estas entidades son equipos terminales fijos, portátiles o de mano. La estación móvil fija es aquella que se encuentra instalada permanentemente sobre un automóvil, estas consumen hasta 20 W con lo que se logra obtener una cobertura mayor pero, con un riesgo de interferencia mayor también. Una estación portátil viene a ser un teléfono de maletín, el cual es un poco más grande que los teléfonos de bolsillo y permanecen instalados dentro del maletín funcionando hasta con 8 Watts de consumo. Finalmente, las estaciones móviles son los teléfonos celulares más pequeños y que consumen menos potencia (alrededor de 2W).

Una característica típica de los teléfonos celulares que utilizan GSM es la tarjeta SIM. Esta tarjeta viene a ser un dispositivo pequeño que se inserta en una ranura especial dentro del equipo. Toda la información que el usuario graba en su equipo, tal como directorio, llamadas recibidas, llamadas enviadas, etc. es almacenada en la tarjeta SIM. Así también la línea se encuentra reconocida en la tarjeta, de forma tal que al intercambiar la tarjeta SIM de un equipo a otro la línea o número telefónico de dicho equipo se traslada al otro terminal. Además, las redes GSM proporcionan un sistema de seguridad que permiten desactivar la línea de un equipo celular cuando este ha sido robado.

4.2 SERVICIOS QUE OFRECEN LAS COMPAÑÍAS DE COMUNICACIÓN MÓVILES EN COLOMBIA

El Buzón de voz

El Buzón de voz te permite que tus amigos, familiares o clientes te dejen mensajes de voz cuando no contestes, tengas el móvil apagado o no estés disponible.

Capacidad para 10 mensajes con una duración de un minuto cada uno. Asistente para configuración de saludo personal y contraseña la primera vez que lo use.

Notificación cada vez que recibe un nuevo mensaje.

La contraseña inicial corresponde a los cuatro últimos dígitos de su número móvil y puede cambiarse cuantas veces quiera.

Llamada de respuesta a la persona que le dejó el mensaje.

Capacidad para 20 mensajes con una duración de un minuto cada uno. Envío o reenvío de mensajes de voz, de buzón a buzón, a un usuario o grupo de usuarios mediante listas de distribución.

Mensajes de fax: Recepción en su buzón, reenvío a otros buzones, envío a máquina de fax.

Llamada de respuesta a la persona que le dejó el mensaje.

El servicio de conferencia telefónica

El servicio de conferencia telefónica, permite combinar hasta cinco llamadas telefónicas en una llamada común, permitiendo que te comuniques hasta con cinco personas al tiempo.

Desvío de Llamadas

El Desvío de Llamadas es un servicio que te permite dejar de recibir ciertas llamadas en tu móvil, pero sin perder ninguna de ellas, que serán dirigidas a otros números programados.

Este servicio incluye:

o Desvío automático de llamadas: Las llamadas entrantes a tu móvil son redirigidas inmediatamente al número que hayas programado.

o Desvío de llamadas por no contestación: Cuando no contestes una llamada, ésta se redirige al número que hayas programado.

o Desvío de llamadas por ocupado: Cuando la llamada no puede ser establecida porque tu línea está ocupada, las llamadas entrantes son redirigidas al número que hayas programado.

Llamada en espera

Es un servicio que permite estar comunicado con alguien y aún así poder recibir una nueva llamada y contestarla, o poder consultar tu agenda, en ambos casos sin necesidad de terminar la primera. Este servicio te ofrece:

o Retención de llamadas: permite interrumpir una llamada que ya has contestado, mantenerla en espera, para poder realizar otra llamada o simplemente consultar la agenda de tu móvil.

Servicio de Retención de Llamadas

Este servicio permite interrumpir una llamada en curso para realizar otra, o simplemente consultar la agenda telefónica de tu móvil. El tiempo al aire es facturado al usuario que hace la llamada.

Mensajes de Texto

Es un servicio que permite enviar y recibir mensajes de texto de hasta 160 caracteres en tu móvil. Así puedes comunicarte en forma práctica y fácil con quien quieras.

Mensajes Multimedia

Puede enviar y recibir mensajes multimedia compuestos de texto e imágenes, y puede recibir mensajes con texto, imágenes y sonidos, como los timbrados musicales polifónicos. Puede guardar imágenes y timbres musicales para personalizar su teléfono. Además, puede enviar fácilmente mensajes de texto y multimedia a varios destinatarios.

Pin Virtual

Es un número de recarga para móvil prepago. No se requiere tarjeta física. El número de recarga se entrega en una tirilla en el lugar de compra (inicialmente supermercados). La tirilla tiene las siguientes características:

Valor de la recarga, Clave (Pin) para la recarga, Fecha de vencimiento, Instrucciones para realizar la recarga, Vigencias desde la recarga, Información legal, Número serial.

Bloqueo permitido para las llamadas salientes

Por seguridad y para que nadie llame desde tu móvil sin autorización, puedes bloquear los siguientes tipos de llamadas salientes:

- o Salientes internacionales: Así no se podrá llamar a otro país desde tu móvil.
- o Salientes internacionales excepto país de ubicación y Colombia: Para cuando estés en Roaming, sólo puedan efectuarse llamadas desde el país donde estás y hacia Colombia. Así no se permitirán llamadas desde y hacia otros países.
- o Todas las salientes en Roaming: Así no se podrán efectuar llamadas locales ni internacionales cuando te encuentres fuera del país usando el servicio de Roaming.

o Llamadas Salientes (Voz): Haciendo que desde tu móvil no puedan realizarse llamadas de voz

o Llamadas internacionales: Para llamar de larga distancia y bloquear las llamadas a usuarios de otras redes, cuando te encuentres fuera del país usando Roaming Internacional.

Bloqueo de Llamadas entrantes

Muchas veces no deseas recibir en tu móvil cierto tipo de llamadas. En estos casos, te ofrece las siguientes opciones de bloqueo de llamadas entrantes:

o Entrantes en Roaming: Significa que no recibirás llamadas en tu móvil cuando estés fuera del país (en Roaming).

o Entrantes: Significa que no recibirás llamadas en tu móvil.

o Internacionales cuando esté en Roaming: Significa que no podrás recibir ni realizar llamadas desde tu móvil cuando estés fuera del país (en Roaming).

o No bloquear llamadas entrantes: Significa que puedes recibir llamadas en tu móvil.

Bloqueo permitido para el servicio GPRS

GPRS es un servicio que te permite recibir y transmitir datos inalámbricamente desde tu móvil, con gran ancho de banda y velocidad. Si lo deseas, puedes solicitar bloquear o desbloquear este servicio:

o Bloqueo de GPRS: bloquea comunicaciones de datos GRPS en tu móvil, tanto de envío como de recepción.

o Bloqueo de GPRS en Roaming: Bloquea comunicaciones de datos GPRS en tu móvil cuando te encuentres fuera del país (Roaming).

o No bloqueo de GPRS: Permite que realices comunicaciones de datos GPRS con tu móvil, tanto para recibirlas como para enviarlas.

Bloqueo permitido para mensajes de Texto

Servicio de Mensajes de Texto, para enviar y recibir mensajes cuando quieras y a quien quieras, así como la posibilidad de bloquearlos cuando lo desees:
o Bloqueo de SMS entrante: Significa que no podrás recibir en tu móvil mensajes de tipo SMS.

o Bloqueo de SMS saliente: Significa que no podrás enviar desde tu móvil mensajes de tipo SMS.

o No bloqueo de SMS: Significa que podrás enviar y recibir mensajes de tipo SMS desde tu móvil.

Servicios disponibles:

Emitir y Recibir Llamadas

Enviar y Recibir Mensajes de Texto SMS

Desvío de Llamadas

Consultar Buzón de Voz

Acceso a Servicio al Cliente

Servicios no disponibles en la actualidad:

Roaming GPRS, enviar y Recibir Mensajes Multimedia MMS, Roaming Prepago, Códigos Cortos.

Servicio General de Datos por Paquetes (GPRS)

Con la tecnología GPRS los teléfonos celulares pueden enviar y recibir datos por una red móvil. Las aplicaciones como navegadores, MMS, mensajes de texto y Java™ usan GPRS. Su teléfono respalda hasta tres conexiones GPRS simultáneas.

APLICACIONES JAVA™ MIDP

Si su teléfono respalda Java 2 Micro Edition, J2METM, incluye algunas aplicaciones y juegos Java que han sido diseñados específicamente para teléfonos móviles.

Navegador WEB

Un navegador integrado que puede usar para conectarse a servicios selectos en el Internet. Puede ver informes del tiempo, noticias, horarios de vuelos, información de la Bolsa, y mucho más. El navegador Web de su teléfono puede mostrar contenido WAP.

La tecnología WAP (Wireless Application Protocol - Protocolo para Aplicaciones Móviles) es para los equipos móviles lo que el WWW es para las computadoras. El sector móvil comenzó a desarrollar el WAP hace varios años para brindar a los usuarios de equipos móviles el acceso al Internet.

La mayoría de las páginas WAP consisten en textos e hipernexos. Algunas páginas contienen imágenes de baja resolución, o admiten el ingreso de datos. Puesto que su proveedor de servicio y otros tienen la libertad de diseñar las páginas WAP que elijan, las mismas pueden variar tanto como las de Internet.

4.3 ALGUNOS TERMINALES MÓVILES COMUNES, IMPLEMENTADOS POR LAS ACTUALES COMPAÑÍAS DE SERVICIO EN COLOMBIA

EQUIPOS USADOS EN BELLSOUTH



MOTOROLA JUPITER C210



Variedad de timbres y melodías
Tecnología: CDMA 1X
Tonos descargables, Agenda personal, SMS de doble vía.

BELLSOUTH2100

SMS de doble vía, Timbres polifónicos, Identificador de llamadas.
Marcación rápida, Correo de voz.
CDMA 800 Mhz AMPS 800Mhz

BELLSOUTH 1125C



SMS de doble vía, Identificador de llamadas.
Marcación rápida, Correo de voz.
CDMA 800 Mhz AMPS 800Mhz

MOTOROLA V810



Descarga de aplicaciones BREW 2.0.
Navegador WAP, SMS de doble vía con caracteres latinos, Plantillas para mensajes SMS, Mensajería Multimedia MMS, Marcación rápida
Modo: CDMA 800 MHz, AMPS 800MHz, CDMA 1900 MHz

TELULAR SX-4T



Planta fija CDMA 1X RTT, Fax, Transmisión de datos de alta velocidad 156 Kbps, Compatible con servicios de Caller ID, Llamada en espera, llamada tripartita y desvío de llamadas. Conexión doble para operación de voz y fax. Soporta múltiples teléfonos y PBX. Indicador de señal, Instalación en escritorio o montaje en pared.

Modo: CDMA 800 MHz y CDMA PCS 1900 MHz

Potencia de transmisión: 200 mW (23dBm) 1W (30dBm)

Nokia 3586



Transferencia de datos a velocidad de hasta 153 Kbps

SMS de doble vía.

Acceso a Internet WAP

Modo: CDMA 800 Mhz, AMPS 800 Mhz CDMA PCS 1900 Mhz

NOKIA 2280



Transferencia de datos a velocidad de hasta 153 Kbps

SMS de doble vía.

Acceso a Internet WAP

Modo: CDMA 800 Mhz, AMPS 800 Mhz

NOKIA 6225



Transferencia de datos a velocidad de hasta 153 Kbps

Descarga de aplicaciones BREW 2.0

Navegador WAP 2.0

SMS de doble vía

Sincronización con PC

Modo: CDMA 800 Mhz, AMPS 800 Mhz CDMA PCS 1900 Mhz

EQUIPOS IMPLEMENTADOS EN COLOMBIA MÓVIL

SONY ERICSON P8000



Una verdadera oficina en la palma de las manos. Para navegar, enviar y recibir e-mails con archivos anexos. Tecnología Bluetooth para conectarse sin cables con computadoras, impresoras, cámaras y más. También compatible con tecnología Symbian OS, Java y C++, para instalarle nuevas aplicaciones útiles.

Servicios

GSM para telefonía, mms, sms, cuenta WAP.

NOKIA 7250I



GPRS y WAP para navegar en la red. Este móvil puede usarse como módem para PC o PDA.

El teléfono celular descrito en este manual está aprobado para ser usado en las redes GSM 900, 1800 y 1900.

SAMSUNG V200



Con GPRS, útil para el acceso rápido a Internet vía WAP, para navegar en la red, enviar y recibir e-mails. Este móvil puede usarse como módem para PC o PDA.

EQUIPOS USADOS EN COMCEL 3GSM

TECOM



GSM 850/1900, Puerto RJ-11(conector para PBX) Puerto para PSTN , LCR(enrutamiento del PBX, le permite programar la salida de llamadas)

Base Tecom conectadas al PBX corporativo y al tarificador. Ahorro de dinero para la empresa. Las llamadas salen a través de la red de Comcel y no por la Red fija. Permite la tarificación por llamada.

SONY ERICSSON T226



Bandas GSM 900/1800, MMS , SMS, Chat, GPRS

SONY T616



GSM 850/1800/1900, MMS , SMS, Chat, Email, GPRS, Soporte Java™, Bluetooth™.

MOTOROLA C 332



GSM 850 y 1900 MHz. SMS, Chat

MOTOROLA V300



GSM 850/1800/1900 ,MMS , SMS, Chat, Email, GPRS

NOKIA 1100



GSM 850/1900 , SMS, Chat.

NOKIA 3300



GSM 850/1900 MMS , SMS, Chat, GPRS, Soporte Java™

Nokia 6010



GSM 850/1900 MMS , SMS, Chat, GPRS.

5. REVISIÓN DE SERVICIOS PARLAY OSA INTERFAZ DE PROGRAMACIÓN DE APLICACIONES APIs.

Parlay OSA incluye un juego comprensivo de APIs para las aplicaciones de comunicaciones que incluye: La movilidad, la localización, Presencia y Manejo de Disponibilidad, control de Llamada, Interacción del Usuario, Mensajería, y Políticas de manejo.

La disponibilidad que estos APIs ofrecen son:

Las Definiciones de los Datos comunes. Los objetos y tipos usados a lo largo de las especificaciones de Framework de Parlay, como las aplicaciones se autentican a la red, la manera como las aplicaciones descubren qué medios, están disponibles desde red. Manejo de carga y sobrecarga.

5.1 CONTROL DE SESIÓN DE DATOS

Nos muestra, como las aplicaciones manejan sesiones de los datos inicializadas en los terminales. Esto es típicamente usado para GPRS.

5.2 PRESENCIA Y DIRECCIÓN DE DISPONIBILIDAD

Hace referencia a la manera como las aplicaciones manejan la presencia "yo estoy en mi computadora" / "yo estoy lejos" y la disponibilidad "yo estoy en una reunión" "yo estoy disponible para ser contactado". Típicamente usado, en los tipos de Mensajería corto de aplicaciones y otras extensiones para redes inalámbricas.

5.3 SERVICIOS DE INFORMACIÓN DE USUARIO.

El kit de información de usuario contiene servicios US (Estado del Usuario) y UL (Localización del Usuario). Ambos, el estándar OSA así como las extensiones H-OSA son soportados.

Ambos servicios US Y UL permiten a una aplicación obtener la información sobre los teléfonos móviles. El servicio US obtiene el estado de los teléfonos móviles. El servicio UL obtiene la información relacionada con la localización del usuario.

Las dos aplicaciones usan como ejemplo para discutir los servicios UL y US. La aplicación Tracker usa OSA UL y la aplicación GroupSMS usa H-OSA US.

Los dos servicios UL Y US son comparables en términos de uso, capacidad y complejidad. Tracker es una simple aplicación que usa UL en una forma de reenvío continuamente. GroupSMS es una aplicación algo más compleja cómo el comportamiento asíncrono de Parlay, puede ser convertida dentro de un comportamiento síncrono. Esto muestra también que diferentes servicios pueden ser combinados dentro de una aplicación. El GroupSMS también usa la funcionalidad del kit de mensajería para enviar y recibir mensajes SMS.

La aplicación Tracker usa OSA UL para obtener la Localización de un teléfono móvil para un juego de subscriptores. En la solicitud de la aplicación de localización de un usuario, la localización actual de cada usuario configurado se obtiene y es mostrado en un mapa.

La aplicación de GroupSMS demuestra el uso del H-OSA, el servicio estado del usuario (HUS) y el H-OSA servicio de mensajería (HUI).

La aplicación define grupos de subscriptores. Cada grupo tiene una dirección a qué mensajes SMS pueden ser enviados. Usando HUI, la aplicación intercepta todos los mensajes SMS que son enviados a un grupo.

La aplicación entonces usa HUS para determinar un miembro (al menos uno) de el grupo que es alcanzado.

Usando HUI de nuevo, el mensaje SMS se remite a este miembro del grupo alcanzado. Si todos los miembros del grupo son inalcanzables, HUI es usado para enviar una replica de mensaje SMS, indicando que el mensaje SMS no pudo ser remitido.

5.4 SERVICIO DE MENSAJERÍA

El kit de servicio de mensajería contiene servicios de interacción de usuario (UI) y mensajería genérica (GM). Ambos servicios son servicios PARLAY HOSA.

La interacción con el usuario puede ser usada por ejemplo enviando y recibiendo mensajes, mientras que la mensajería genérica puede usarse para ocuparse de mensajes del e-mail en un buzón.

Los siguientes tipos de mensajes pueden ser enviados con el servicio de interacción de usuario.

- SMS Servicio de mensajería corta
- MMS Sistema de mensajería de multimedia
- WAP-Push¹:
- E-Mail

El siguiente tipo de mensajes pueden ser recibidos con el servicio de interacción de usuario

- SMS

Se usan cuatro aplicaciones como los ejemplos para discutir la Interacción del Usuario y los servicios del Mensajería genéricos:

- Aplicación SMSSender usa HOSA UI para enviar un mensaje SMS a un particular usuario.
- Aplicación SMStoEmail usa HOSA UI para recibir un mensaje SMS y para reenviar el mensaje SMS recibido como un mensaje de e-mail.
- Aplicación SMStoMMS usa HOSA UI para recibir un mensaje SMS y para reenviar el mensaje SMS recibido como un mensaje MMS.

¹ Un mensaje WAP-Push es un documento de XML estructurado sobre el Protocolo de Aplicación Push (PAP) conteniendo un mensaje WAP

- Aplicación MMSContentRetrieval usa HOSA GM para leer mensajes de e-mail desde un buzón y usa HOSA UI para recibir mensajes SMS y para enviar mensajes MMS.

5.5 SERVICIOS DE ADMINISTRACIÓN DE INFORMACIÓN PERSONAL (PIM).

5.5.1 PIM DE CONTACTO

El Servicio del PIM de Contacto le da la posibilidad de manejar las entradas del contacto en el libro de la dirección personal de un suscriptor, almacenado en un directorio de contacto.

Es posible crear un contacto al mismo tiempo para varios suscriptores. La aplicación puede dar una lista de suscriptores en el método de `hosaCreateContactReq`. En el método del resultado, hay un resultado por el suscriptor. En el método del resultado hay también una identidad del contacto que puede usarse si la aplicación desea modificar o eliminar la información del contacto después.

En la recuperación de la información del contacto, es posible definir qué campos (por ejemplo Nombre del usuario y/o Número telefónico) deben ser incluidos en el criterio de la búsqueda. También es posible usar los comodines (*) en el criterio de la búsqueda. El resultado contendrá todos los contactos que emparejan con el criterio de la búsqueda para el suscriptor especificado.

También es posible crear los contactos en el formato vCard y para recuperar la información del contacto en el formato del vCard. Esto puede hacerse usando los métodos `hosaCreateContactFromVCardReq` y `hosaRetrieveContactsAsVCardsReq`.

5.5.2 PIM DE CALENDARIO

El servicio de administración de información personal de calendario (PIM de Calendario) mantiene una interfase a un Servidor del Calendario externo, la administración de los Calendarios personales.

Este servicio puede usarse para crear, modifique, anular, y recuperar un evento del Calendario o hacer las entradas de uno o más subscriptores a los calendarios. También puede usarse para recuperar la información libre u ocupada de un calendario del subscritor.

5.6 SERVICIOS DE INTERACCIÓN Y CONTROL DE LLAMADAS DE LOS USUARIOS.

El control de llamada hace referencia a la manera de como las aplicaciones prepararon las llamadas en la red, como las aplicaciones preparan (las conferencias) llamadas en la red como se usan las aplicaciones para enrutar las llamadas en la red, para que una aplicación de la empresa pueda manejar la llamada, como las aplicaciones pueden preparar las llamadas de los multi-medios de comunicación.

El kit de servicios de control de llamadas (Call Control) contiene OSA MPCC (Multy Party Call Control) y UI (Servicio genérico de interacción del usuario). Esta también contiene H-OSA una extensión del servicio OSA MPCC.

MPCC es parte del Control de llamada de Parlay/OSA.

El servicio de MPCC permite a una aplicación influenciar llamadas en la red de telefonía. Su funcionalidad consiste en inicializar, detectando, observando, enrutando, y soltando las llamadas. El servicio de UI permite a una aplicación comprometer en un dialogo con los subscriptores. Su funcionalidad consiste en colocar los anuncios y entrar una colección de subscriptores.

UI difiere de los otros servicios porque no puede usarse en aislamiento, pero si puede usarse junto con MPCC.

MPCC también es único. Apoya un grande y variado set de interfaces y operaciones, que cualquiera de los otros servicios. Además, las interfaces se refieren al estado de los objetos: El historial de un objeto puede influir en la conducta y la exactitud de un funcionamiento particular. Como resultado, MPCC es uno de los servicios más complejos que se discuten en este documento.

Se usan dos aplicaciones como ejemplos para discutir los servicios MPCC y UI.

La aplicación WebDial sólo usa MPCC, para ocuparse de la aplicación de las llamadas iniciadas.

La aplicación CallRouter usa ambos servicios MPCC y UI. Esta usa MPCC para manejar la inicialización de las llamadas en la red.

5.6.1. PROTOCOLOS SOPORTADOS EN LA RED.

El NRG soporta los siguientes protocolos de redes de telecom para Multi Party Call Control:

- CS1+
- CAPv2

El operador no necesita soportar ambos protocolos. En caso de que se desee conocer qué protocolos soporta el operador se recomienda contactar con el operador.

5.6.2 WEBDIAL

La aplicación WebDial demuestra el uso de OSA MPCC (Multi Party Call Control). La aplicación usa este servicio para implementar la aplicación de llamadas inicializadas, el usuario de la aplicación puede crear una llamada entre un número arbitrario de suscriptores (a lo máximo 30).

5.6.3 CALLROUTER

La aplicación de CallRouter demuestra el uso del servicio OSA MPCC (Multi Party Call Control) en combinación con el servicio. OSA UI (Servicio genérico de interacción del usuario). La aplicación usa estos servicios para hacer la implementación selectiva de ruta entrante, las llamadas son basadas en un diálogo con el usuario que llama.

5.6.4 DIFERENTES CONDUCTAS DE LLAMADAS (DIFFERENT CALL BEHAVIORS)

El NRG apoya dos conductas de llamada diferentes, éstas son:

- 2-Party Conducta de Llamada
- Multi Party Conducta de Llamada

Qué conducta de llamada se usará depende del acuerdo que se tenga con el operador, esto se configurará en el SLA. (Servicio del nivel acordado).

El método de continueProcessing puede usarse para continuar un leg² interrumpido. Con Multi Party conducta de Llamada se puede continuar siempre un leg interrumpido.

Con 2-Party Conducta de Llamada, se recomienda para poder continuar el primer leg interrumpido enrutar al segundo leg. Cuando se continúa el primer leg antes de enrutar al segundo leg, la llamada se perderá.

Debido a esta conducta es aconsejable seguir la secuencia de la ayuda de la aplicación. Cuando se usa 2-Party Conducta de Llamada la primera parte se oirá un tono resonante cuando la llamada es rutada a la segunda parte.

² Una representación dentro de un modelo de estado de proceso de llamada representando un camino de telecomunicación hacia alguna entidad direccionable (por ejemplo, un usuario o un Periférico Inteligente (IP)).

Cuando se usa Multy Party Conducta de Llamada la primera parte no se oirá un tono resonante durante el enrutamiento.

5.6.5 SERVICIO DE COBRADO (CHARGING)

Se refiere a como las aplicaciones piden el pago para los servicios y la manera como las aplicaciones reservan el pago para un servicio futuro. Esto permite a las aplicaciones trabajar en una red pre-pagada.

La manera en que el NRG maneja cobrando por Multi Party Llamada de control es

descrito en esta sección. NRG soporta cobrado-basado en llamada y basado en leg para Multi Party Call Control (MPCC).

MPCC tiene las siguientes capacidades relacionadas con el cobro de llamada.:

- No cobrado: el NRG no hace ningún cobrado en absoluto. La red es responsable por manejar el cobrado de llamadas.
- La aplicación no puede influir en el cobrado (valor por defecto de cobro): El NRG proporciona la información sobre la aplicación y servicio a ser usado en la Información de Cobrado. Los ítems de los datos extremo-a en el Detalle de la Llamada El registro (CDR). La aplicación no tiene ningún medio para influir en esto. La información cobrado por defecto es fija en el SLA..
- La Influencia de la Aplicación limitada (cobro predefinido).: La aplicación tiene la posibilidad de seleccionar un plan de cobro predefinido y puede especificar la información adicional para un evento de cobro los datos adicionales son incluido en el CDR y puede, por ejemplo, ser usado para especificar el username para la aplicación de inicialización de llamada.
- La Influencia de la aplicación avanzada (el cobro transparente): Al usar el cobro transparente, la aplicación determina la cadena para ser enviado en el mensaje de información de cobro. Este ofrece la máxima flexibilidad para una aplicación. Este tipo de cobro puede, por ejemplo, ser usado por un corto tiempo-en-mercado para los nuevos servicios, si cobra por un servicio

puede ser integrado en la infraestructura de facturación existente sin poner al día, para el caso, un sistema de facturación. El mensaje de cobrado que la aplicación quiere usar, será transparentemente remitida por el NRG a el sistema de facturación en la red.

Cuando hay ya un plan de cargo predefinido definido en el SLA, es posible para la aplicación ser sustituido con otro plan de cobro. Junto con el operador, el diseñador de la aplicación debe decidir qué tipo de cobro se usará con la aplicación.

5.6.6. DURACIÓN DE LA LLAMADA

También es posible pedir o poner la duración de la llamada telefónica. Por medio del uso del método de `getInfoReq` esto es posible para solicitar la duración de la llamada telefónica. La duración se devolverá a la aplicación por medio del método de `getInfoRes`.

El método del `superviseReq` puede usarse para soltar una llamada telefónica después de un cierto periodo de tiempo.

BIBLIOGRAFÍA

ITU-T Recommendation Q.1201/I.312: Principles of intelligent network architecture. Octubre 1992.

Shepard Steven, Convergencia las Telecomunicaciones, Ed Mc Graw Hill.

Muñoz David, Sistemas Inalámbricos de Comunicación Personal, Ed Alfaomega.

Understanding Telecommunications Tomo I y II, Ericsson 1997.

Resumen TV-Cable articulo, Seminario Convergencia en las Telecomunicaciones,

FOROUZAN Behrouz A. Transmisión de datos y redes de comunicación.
Segunda Edición

STALLINGS William. Comunicaciones y redes de computadores. Sexta Edición
Año 2000

Motorola RD-21 BROADBAND REFERENCE GUIDE. August, Año 2003

Biblioteca de consulta Encarta 2005 – Redes Inteligentes

Internet.

www.bellsouth.com.co

www.ola.com.co

www.comcel.com.co

www.noticias3d.com/articulos/200104/adsl/1.asp

<http://broadband.motorola.com>

www.domotica.net

<http://www.ieee.com>

<http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol812/inteligencia/inteligencia.html>

www.parlay.org

www.3GPP.ORG

www.ericsson.com

Anexo A

Código Fuente aplicación WebDial

A continuación se puede ver el código fuente de todos los archivos empleados en la aplicación WebDial:

CallProcessor

```
1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import com.ericsson.hosask.api.TpAddress;
7 import com.ericsson.hosask.api.TpAddressPresentation;
8 import com.ericsson.hosask.api.TpAddressPlan;
9 import com.ericsson.hosask.api.TpAddressScreening;
10
11 import com.ericsson.hosask.api.cc.TpAdditionalCallEventCriteria;
12 import com.ericsson.hosask.api.cc.TpCallAppInfo;
13 import com.ericsson.hosask.api.cc.TpCallEndedReport;
14 import com.ericsson.hosask.api.cc.TpCallError;
15 import com.ericsson.hosask.api.cc.TpCallEventRequest;
16 import com.ericsson.hosask.api.cc.TpCallEventType;
17 import com.ericsson.hosask.api.cc.TpCallMonitorMode;
18 import com.ericsson.hosask.api.cc.TpReleaseCause;
19
20 import com.ericsson.hosask.api.cc.mpccs.IpAppMultiPartyCall;
21 import com.ericsson.hosask.api.cc.mpccs.IpAppMultiPartyCallAdapter;
22 import com.ericsson.hosask.api.cc.mpccs.TpCallLegIdentifier;
23 import com.ericsson.hosask.api.cc.mpccs.TpMultiPartyCallIdentifier;
24
25 /**
26  * This class is responsible for all needed HOSA implementation
27  * regarding
28  * Multi Party Call Control for this application
29  *
30  * @author Ericsson AB
31  */
32 public class CallProcessor extends IpAppMultiPartyCallAdapter
33 {
34     private Feature itsParent;
35     private LegProcessor itsLegProcessor;
36
37     /**
38      * The constructor of this class
39      *
40      * @param aParent the Parent to which this class can callback to
```

```

40     * @param aLegProcessor the legProcessor which the CallProcessor
can use
41     */
42     public CallProcessor(Feature aParent, LegProcessor
aLegProcessor)
43     {
44         itsParent = aParent;
45         itsLegProcessor = aLegProcessor;
46     }
47
48     /**
49     * This method can be used to create a new MPCC leg. Because
this leg will be used in an
50     * application initiated call the originating address will be
specified as undefined.
51     *
52     * @param aCallId The MPCC call identifier
53     * @param aPhoneNumber the phone number to which the leg needs
to be routed
54     *
55     * @return the MPCC leg identifier
56     */
57     public TpCallLegIdentifier
createAndRouteLeg(TpMultiPartyCallIdentifier aCallId, String
aPhoneNumber)
58     {
59         TpAdditionalCallEventCriteria answerCriteria = new
TpAdditionalCallEventCriteria();
60         answerCriteria.Dummy((short) 0);
61         TpCallEventRequest answerRequest = new
TpCallEventRequest(TpCallEventType.P_CALL_EVENT_ANSWER,
62             answerCriteria,
63             TpCallMonitorMode.P_CALL_MONITOR_MODE_INTERRUPT);
64
65         TpReleaseCause[] RELEASE_CAUSES = {TpReleaseCause.P_BUSY,
66             TpReleaseCause.P_CALL_RESTRICTED,
67             TpReleaseCause.P_DISCONNECTED,
68             TpReleaseCause.P_GENERAL_FAILURE,
69             TpReleaseCause.P_NO_ANSWER,
TpReleaseCause.P_NOT_REACHABLE,
70             TpReleaseCause.P_PREMATURE_DISCONNECT,
71             TpReleaseCause.P_ROUTING_FAILURE,
72             TpReleaseCause.P_TIMER_EXPIRY,
73             TpReleaseCause.P_UNAVAILABLE_RESOURCE,
74             TpReleaseCause.P_USER_NOT_AVAILABLE};
75         TpAdditionalCallEventCriteria releaseCriteria = new
TpAdditionalCallEventCriteria();
76         releaseCriteria.TerminatingReleaseCauseSet(RELEASE_CAUSES);
77         TpCallEventRequest releaseRequest = new
TpCallEventRequest(TpCallEventType.P_CALL_EVENT_TERMINATING_RELEASE,
78             releaseCriteria,
79             TpCallMonitorMode.P_CALL_MONITOR_MODE_INTERRUPT);
80
81         TpCallEventRequest[] requestedEvents = {answerRequest,
82             releaseRequest};
83

```

```

84          TpAddress  originatingAddress = new
 TpAddress ( TpAddressPlan.P_ADDRESS_PLAN_NOT_PRESENT,
85             "", "",
86              TpAddressPresentation.P_ADDRESS_PRESENTATION_UNDEFINED,
87              TpAddressScreening.P_ADDRESS_SCREENING_UNDEFINED, "");
88
89         return
aCallId.CallReference.createAndRouteCallLegReq(aCallId.CallSessionID,
90             requestedEvents, createE164Address(aPhoneNumber),
91             originatingAddress, new  TpCallAppInfo [0],
itsLegProcessor);
92     }
93
94     /**
95     * Deassigns the call.
96     *
97     * @param aCallId The MPCC call identifier
98     */
99     public void deassign( TpMultiPartyCallIdentifier  aCallId)
100    {
101        aCallId.CallReference.deassignCall(aCallId.CallSessionID);
102    }
103
104    // unused methods:
105    // public void getInfoRes(int callSessionID,  TpCallInfoReport 
callInfoReport)
106    // public void getInfoErr(int callSessionID,  TpCallError 
errorIndication)
107    // public void superviseRes(int callSessionID, int report, int
usedTime)
108    // public void superviseErr(int callSessionID,  TpCallError 
errorIndication)
109    // public void callEnded(int callSessionID,  TpCallEndedReport 
report)
110    // public void createAndRouteCallLegErr(int callSessionID,
 TpCallLegIdentifier  callLegReference,  TpCallError  errorIndication)
111
112    private static  TpAddress  createE164Address( String  aNumber)
113    {
114        return new  TpAddress ( TpAddressPlan.P_ADDRESS_PLAN_E164,
115            aNumber, "",
116             TpAddressPresentation.P_ADDRESS_PRESENTATION_UNDEFINED,
117             TpAddressScreening.P_ADDRESS_SCREENING_UNDEFINED, "");
118    }
119
120}
121

```

Configuration

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package  se.ericsson.webdialdemo ;

```

```

5
6 import java.util.*;
7 import java.io.*;
8 import com.ericsson.hosasdk.utility.configuration.NestedProperties;
9 import com.ericsson.hosasdk.utility.configuration.NotFoundException;
10 import
com.ericsson.hosasdk.utility.configuration.BadSyntaxException;
11
12 /**
13  * This class is responsible for parsing the configuration file and
14  * presenting the configuration data using convenient data types.
15  */
16
17 public class Configuration extends NestedProperties
18 {
19
20     /** The singleton instance that can be accessed by all users. */
21     public static final Configuration INSTANCE = new
Configuration();
22
23     /**
24      * Private constructor to prevent multiple instances.
25      */
26     private Configuration()
27     {}
28 }
29

```

Feature

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import java.util.HashMap;
7
8 import com.ericsson.hosasdk.api.TpAddress;
9 import com.ericsson.hosasdk.api.cc.TpCallEventInfo;
10
11 import com.ericsson.hosasdk.api.cc.mpccs.IpAppCallLeg;
12 import
com.ericsson.hosasdk.api.cc.mpccs.IpMultiPartyCallControlManager;
13 import com.ericsson.hosasdk.api.cc.mpccs.TpCallLegIdentifier;
14 import com.ericsson.hosasdk.api.cc.mpccs.TpMultiPartyCallIdentifier;
15
16 import com.ericsson.hosasdk.utility.framework.FWproxy;
17
18 import java.util.*;
19 import java.util.Timer;
20 import java.awt.event.*;
21 import javax.swing.*;
22 import javax.swing.border.*;
23
24 /**
25  * This class implements the logic of the application.

```

```

26 */
27
28 public class Feature
29 {
30     // the FWProxy used to obtain the service managers
31     private FWproxy itsFramework;
32
33     // the service managers
34     private IpMultiPartyCallControlManager itsMPCCManager;
35
36     // the HOSA callbacks
37     private MPCCProcessor itsMPCCProcessor;
38     private CallProcessor itsCallProcessor;
39     private LegProcessor itsLegProcessor;
40
41     Map itsSessions = new HashMap();
42     Timer itsGuardManager = new Timer();
43
44     /**
45      * Initializes a new instance, without starting interaction with
NRG
46      * (see start)
47      *
48      * @param aGUI the GUI of the application
49      */
50     public Feature(GUI aGUI)
51     {
52         aGUI.setTitle("WebDial");
53         aGUI.addTab("Description", getDescription());
54
55         final JTextArea numbersField = new JTextArea();
56         numbersField.setBorder(new EmptyBorder(4, 8, 4, 8));
57         aGUI.addTab("Parties", numbersField);
58         aGUI.addButton(new AbstractAction("Initiate call")
59         {
60             public void actionPerformed(ActionEvent e)
61             {
62                 StringTokenizer s = new
StringTokenizer(numbersField.getText(),
NRG
63                 "\t\n\r\f,+");
64                 String[] numbers = new String[s.countTokens()];
65                 for (int i = 0; i != numbers.length; i++)
66                 {
67                     numbers[i] = s.nextToken();
68                 }
69                 establishCall(numbers);
70             }
71         });
72     }
73
74     /**
75      * Connect to the Framework and obtain the Service Capability
Features (SCF)
76      * needed.
77      */
78     protected void start()

```

```

79     {
80         itsFramework = new FWproxy(Configuration.INSTANCE);
81         itsMPCCManager = (IpMultiPartyCallControlManager)
itsFramework.obtainSCF("P_MULTI_PARTY_CALL_CONTROL");
82
83         itsLegProcessor = new LegProcessor(this);
84         itsCallProcessor = new CallProcessor(this, itsLegProcessor);
85         itsMPCCProcessor = new MPCCProcessor(this, itsMPCCManager,
86             itsCallProcessor);
87     }
88
89     /**
90      * Releases the Service Capability Features(SCF) which was
obtained by
91      * start() and the disconnect from the Framework.
92      */
93     protected void stop()
94     {
95         itsFramework.releaseSCF(itsMPCCManager);
96         itsFramework.endAccess(null); // Close framework connection
97         itsFramework.dispose();
98     }
99
100    /**
101     * Creates an application initiated call.
102     * @param aPhoneNumbers Specifies the phone numbers for the call
103     */
104
105     protected void establishCall(String[] aPhoneNumbers)
106     {
107         TpMultiPartyCallIdentifier callId =
itsMPCCProcessor.createCall();
108         Session session = new Session(callId, aPhoneNumbers);
109         // route the first leg
110         createAndRouteNextLeg(session);
111     }
112
113     /**
114      * Invoked when a leg has been routed.
115      * @param aLegSessionID identifies the routed leg
116      */
117
118     public void routed(int aLegSessionID)
119     {
120         Session session = removeSession(aLegSessionID);
121         createAndRouteNextLeg(session);
122     }
123
124     /**
125      * Creates and routes the next leg in the call,
126      * or deassigns the call if all legs have been routed.
127      * @param aSession the session that represents the call state
128      */
129
130     private void createAndRouteNextLeg(Session aSession)
131     {

```

```

32     String nextParty = aSession.getNextParty();
33     if (nextParty != null)
34     {
35         TpCallLegIdentifier leg =
itsCallProcessor.createAndRouteLeg(aSession.getCallId(),
36             nextParty);
37         addSession(leg.CallLegSessionID, aSession);
38     }
39     else
40     {
41         itsCallProcessor.deassign(aSession.getCallId());
42     }
43 }
44
45 /**
46  * Adds a new leg session to the session database.
47  * @param aLegID Specifies the leg
48  * @param aSession Contains the leg session
49  */
50
51 synchronized private void addSession(final int aLegID, Session
aSession)
52 {
53     itsSessions.put(new Integer(aLegID), aSession);
54     aSession.itsGuard = new TimerTask()
55     {
56         public void run()
57         {
58             removeSession(aLegID);
59         }
60     };
61     itsGuardManager.schedule(aSession.itsGuard, 2 * 60 * 1000);
62 }
63
64 /**
65  * Removes a leg session from the session database.
66  * @param aLegID Specifies the leg
67  * @return Returns the new hashmap
68  */
69
70 synchronized private Session removeSession(int aLegID)
71 {
72     Session session = (Session) itsSessions.remove(new
Integer(aLegID));
73     if (session != null)
74     {
75         session.itsGuard.cancel();
76     }
77     return session;
78 }
79
80 /**
81  * Returns a descriptive text that explains the application and
its configuration.
82  */
83

```

```

84     private String getDescription()
85     {
86         String s = "Press START to connect to the Framework"
87             + " and request the OSA Multi Party Call Control (MPCC)
service from the Framework.\n";
88         s += "\n";
89         s += "Press INITIATE CALL to setup a call between all
specified parties.\n";
90         s += "\n";
91         s += "Press STOP to release resources in the NRG and the
application.\n";
92         return s;
93     }
94 }
95

```

GUI

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import java.awt.*;
7 import java.awt.event.*;
8 import javax.swing.*;
9 import javax.swing.border.*;
10
11 /**
12  * This class defines a simple graphical user interface.
13  * Users can add (tab) pages and buttons.
14  */
15
16 public class GUI extends JFrame
17 {
18     private JTabbedPane theTabs = new JTabbedPane();
19     private JPanel theButtons = new JPanel();
20
21     /**
22      * Creates a frame containing placeholders for (tab) pages and
buttons.
23      */
24
25     public GUI()
26     {
27         // construct the contents
28         theButtons.setLayout(new FlowLayout(FlowLayout.CENTER, 8,
8));
29
30         JPanel contents = new JPanel();
31         contents.setLayout(new BorderLayout());
32         contents.add(theTabs, BorderLayout.CENTER);
33         contents.add(theButtons, BorderLayout.SOUTH);
34         contents.setBorder(new EmptyBorder(8, 8, 0, 8));
35
36         // add the contents to the frame
37         getContentPane().setLayout(new BorderLayout());

```

```

37         getContentPane().add(contents, BorderLayout.CENTER);
38
39         // the class that opens the window should close it as well
40
41     setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
42 }
43 /**
44  * Centers and shows the window
45  */
46
47     public void showCentered()
48     {
49         pack();
50         Dimension guiSize = getSize();
51         Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
52         setLocation((screenSize.width - guiSize.width) / 2,
53                     (screenSize.height - guiSize.height) / 2);
54         setVisible(true);
55     }
56
57     /**
58     * Adds a tab page with a given title and contents.
59     *
60     * @param aTitle the name of the tab page
61     * @param aComp the contents of the tab page (will be embedded
in a
62     *         JScrollPane)
63     */
64
65     public void addTab(String aTitle, Component aComp)
66     {
67         theTabs.add(aTitle, new JScrollPane(aComp));
68     }
69
70     /**
71     * Adds a tab page with a given title and contents.
72     *
73     * @param aTitle the name of the tab page
74     * @param aText the contents of the tab page
75     */
76
77     public void addTab(String aTitle, String aText)
78     {
79         JTextArea text = new JTextArea(18, 50);
80
81         text.setText(aText);
82         text.setLineWrap(true);
83         text.setWrapStyleWord(true);
84         text.setEditable(false);
85         text.setBorder(new EmptyBorder(4, 8, 4, 8));
86         addTab(aTitle, text);
87     }
88
89     /**

```

```

90     * Adds a button.
91     *
92     * @param anAction the action to perform when the button is
clicked
93     */
94
95     public void addButton(Action anAction)
96     {
97         theButtons.add(new JButton(anAction));
98     }
99 }
00
01

```

LegProcessor

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import com.ericsson.hosask.api.cc.TpCallError;
7 import com.ericsson.hosask.api.cc.TpCallEventInfo;
8 import com.ericsson.hosask.api.cc.TpReleaseCause;
9
10 import com.ericsson.hosask.api.cc.mpccs.IpAppCallLeg;
11 import com.ericsson.hosask.api.cc.mpccs.IpAppCallLegAdapter;
12
13 /**
14  * This class is responsible for all needed HOSA implementation
regarding
15  * Multi Party Call Control for this application.
16  *
17  * @author Ericsson AB
18  */
19 public class LegProcessor extends IpAppCallLegAdapter
20 {
21     private Feature itsParent;
22
23     /**
24      * The constructor
25      *
26      * @param aParent the Parent to which this class can callback to
27      */
28     public LegProcessor(Feature aParent)
29     {
30         itsParent = aParent;
31     }
32
33     /**
34      * This asynchronous method reports that an event has occurred
that was
35      * requested to be reported. The routed method from the feature
will be
36      * used to inform the feature that the leg is routed.
37      *

```

```

38     * @param callLegSessionID specifies the call leg session ID of
the call leg
39     * @param eventInfo specifies data associated with this event
40     */
41     public void eventReportRes(int callLegSessionID, TpCallEventInfo
eventInfo)
42     {
43         itsParent.routed(callLegSessionID);
44     }
45
46     // unused methods:
47     // public void eventReportErr(int callLegSessionID, TpCallError
errorIndication)
48     // public void attachMediaRes(int callLegSessionID)
49     // public void attachMediaErr(int callLegSessionID, TpCallError
errorIndication)
50     // public void detachMediaRes(int callLegSessionID)
51     // public void detachMediaErr(int callLegSessionID, TpCallError
errorIndication)
52     // public void getInfoRes(int callLegSessionID,
TpCallLegInfoReport callLegInfoReport)
53     // public void getInfoErr(int callLegSessionID, TpCallError
errorIndication)
54     // public void routeErr(int callLegSessionID, TpCallError
errorIndication)
55     // public void superviseRes(int callLegSessionID, int report,
int usedTime)
56     // public void superviseErr(int callLegSessionID, TpCallError
errorIndication)
57     // public void callLegEnded(int callLegSessionID, TpReleaseCause
cause)
58 }
59

```

MPCCProcessor

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import com.ericsson.hosask.api.cc.mpccs.IpAppMultiPartyCall;
7 import
com.ericsson.hosask.api.cc.mpccs.IpAppMultiPartyCallControlManagerAda
pter;
8 import
com.ericsson.hosask.api.cc.mpccs.IpMultiPartyCallControlManager;
9 import com.ericsson.hosask.api.cc.mpccs.TpAppMultiPartyCallBack;
10 import com.ericsson.hosask.api.cc.mpccs.TpMultiPartyCallIdentifier;
11
12 /**
13 * This class is responsible for all needed HOSA implementation
regarding
14 * Multi Party Call Control for this application
15 *
16 * @author Ericsson AB

```

```

17 */
18 public class MPCCProcessor extends
IpAppMultiPartyCallControlManagerAdapter
19 {
20     private IpMultiPartyCallControlManager itsMPCCManager;
21     private TpAppMultiPartyCallBack itsMPCCallback;
22     private Feature itsParent;
23
24     /**
25      * The constructor of this class
26      *
27      * @param aParent the Parent to which this class can callback to
28      * @param aMPCCManager manager used to talk to the NRG
29      * @param aMPCCallCallback Specifies the callback for MPCC
30      */
31     public MPCCProcessor(Feature aParent,
32                          IpMultiPartyCallControlManager aMPCCManager,
33                          IpAppMultiPartyCall aMPCCallCallback)
34     {
35         itsParent = aParent;
36
37         itsMPCCallback = new TpAppMultiPartyCallBack();
38         itsMPCCallback.AppMultiPartyCall(aMPCCallCallback);
39         itsMPCCManager = aMPCCManager;
40     }
41
42     /**
43      * Creates a new Multi Party Call Control call.
44      *
45      * @return the call
46      */
47     public TpMultiPartyCallIdentifier createCall()
48     {
49         return
itsMPCCManager.createCall(itsMPCCallback.AppMultiPartyCall());
50     }
51 }
52

```

Main

```

2 1 // Copyright (C) 2003-2004 Ericsson AB.
3
4 package se.ericsson.webdialdemo;
5
6 import java.awt.event.*;
7 import javax.swing.*;
8
9 /**
10 * This class is responsible for initializing, starting, stopping
and
11 * terminating the application.
12 */
13
14 public class Main

```

```

15{
16    private Feature itsFeature;
17
18    /**
19     * Method called by the JVM to launch the application.
20     */
21
22    public static void main(String[] args)
23        throws Exception
24    {
25        new Main();
26    }
27
28    /**
29     * Initializes the application.
30     */
31
32    public Main()
33        throws Exception
34    {
35        Configuration.INSTANCE.load("config/config.ini");
36
37        GUI gui = new GUI();
38
39        gui.addButton(new AbstractAction("Start")
40            {
41                public void actionPerformed(ActionEvent e)
42                {
43                    start();
44                }
45            });
46
47        itsFeature = new Feature(gui);
48
49        gui.addButton(new AbstractAction("Stop")
50            {
51                public void actionPerformed(ActionEvent e)
52                {
53                    stop();
54                }
55            });
56
57        gui.addWindowListener(new WindowAdapter()
58            {
59                public void windowClosing(WindowEvent e)
60                {
61                    terminate();
62                }
63            });
64
65        gui.showCentered();
66    }
67
68    /**
69     * Starts interaction with NRG.
70     */

```

```

71
72     public void start()
73     {
74         itsFeature.start();
75     }
76
77     /**
78      * Stops interaction with NRG.
79      */
80
81     public void stop()
82     {
83         itsFeature.stop();
84     }
85
86     /**
87      * Terminates the application.
88      */
89
90     public void terminate()
91     {
92         System.exit(0);
93     }
94 }
95

```

Session

```

1 // Copyright (C) 2003-2004 Ericsson AB.
2
3
4 package se.ericsson.webdialdemo;
5
6 import com.ericsson.hosasdk.api.TpAddress;
7
8 import com.ericsson.hosasdk.api.cc.mpccs.IpAppCallLeg;
9 import com.ericsson.hosasdk.api.cc.mpccs.TpCallLegIdentifier;
10 import com.ericsson.hosasdk.api.cc.mpccs.TpMultiPartyCallIdentifier;
11 import java.util.*;
12
13 /**
14  * Session contains all the information regarding the call set up
15  * process
16  * @author Ericsson AB
17  */
18
19 public class Session
20 {
21     private  TpMultiPartyCallIdentifier  itsCallId;
22     private  String[]  itsParties;
23     private  int  itsNextPartyToRoute;
24      TimerTask  itsGuard;
25
26     /**
27      * The constructor

```

```

28     */
29     public Session( TpMultiPartyCallIdentifier aCallId,  String[] 
aParties)
30     {
31         itsCallId = aCallId;
32         itsParties = aParties;
33     }
34
35     /**
36     * Returns the call Id
37     */
38     public  TpMultiPartyCallIdentifier  getCallId()
39     {
40         return itsCallId;
41     }
42
43     /**
44     * Returns a party that has not been routed yet,
45     * or null if all parties have been routed
46     */
47     public  String  getNextParty()
48     {
49         return itsNextPartyToRoute < itsParties.length
50             ? itsParties[itsNextPartyToRoute++]
51             : null;
52     }
53}

```