

CONTRIBUCIÓN AL DISEÑO DE LA ARQUITECTURA DE RED  
PARA LA INTEGRACIÓN DE SERVICIOS DE PAQUETES DE  
DATOS EN EL SISTEMA UMTS

Tesi Doctoral presentada en la  
Universitat Politècnica de Catalunya

Autora: **Anna Umbert Juliana**  
Directora: **Pilar Díaz Romero**

Julio 2008

## Capítulo 7. Conclusiones

Los sistemas de comunicaciones móviles de segunda generación (2G) están dirigidos básicamente a dar servicio de voz con una calidad aceptable. La evolución de las necesidades de los usuarios hacia aplicaciones multimedia típicamente relacionadas con Internet ha creado la necesidad de incorporar los servicios ofrecidos en Internet en los futuros sistemas de comunicaciones móviles de tercera generación (3G). La mayoría de estos nuevos servicios están orientados a transmisión de paquetes, por lo que es más eficiente diseñar un sistema de conmutación de paquetes que utilizar los sistemas existentes de conmutación de circuitos.

La presente Tesis Doctoral es una **contribución al diseño de la arquitectura de red para la integración de servicios de paquetes de datos en el sistema UMTS**, nombre que recibe el sistema de comunicaciones móviles de 3G en Europa.

### 7.1. Conclusiones de la tesis

En esta tesis se ha pretendido evaluar las prestaciones de diferentes alternativas para la integración de servicios de paquetes de datos en el sistema UMTS. Para llevar a cabo este objetivo, se ha reproducido el acceso radio del UMTS hasta la capa de enlace con el fin de evaluar su impacto en las capas superiores.

En una primera fase de la tesis se estudiaron y analizaron las características básicas del sistema UMTS y de la tecnología IP. El resultado de este trabajo de investigación ha intentado recogerse en el capítulo 3. Fue en esta fase donde se fijó como punto de partida, y dentro de la definición del sistema UMTS, el modo **FDD (Frequency-Division Duplex)**. Este modo se basa en la técnica de acceso **WCDMA (Wideband Code Division Multiple Access)**.

En una segunda fase se construyó un emulador del canal radio y la capa física, para el modo de acceso WCDMA, basado en Modelos Ocultos de Markov (**HMM: Hidden Markov Model**).

---

En el capítulo 4 se presenta el emulador creado, los modelos ocultos de Markov, y el simulador necesario para entrenar al emulador. El emulador intenta imitar el sistema real, pero sin reproducir cada uno de sus procesos. En efecto, el emulador puede verse como una caja negra que tiene el mismo comportamiento entrada-salida que el sistema real. Sin embargo para que el emulador consiga reproducir este comportamiento es necesario entrenarlo, mediante simulaciones, marcando las pautas de respuesta a diferentes señales de entrada. Una vez obtenidas las simulaciones y entrenado el emulador se puede prescindir del simulador. Antes de prescindir del simulador se ha validado extensamente que el emulador construido reproduzca adecuadamente el sistema real. Los resultados de esta validación han sido muy gratificantes por la gran precisión conseguida con el emulador.

El HMM es un modelo formado por un conjunto de estados ( $N$ ), cada uno de los cuales representa un conjunto de símbolos (en nuestro caso los símbolos corresponden a un determinado número de errores por trama), y tres distribuciones de probabilidades: la probabilidad de estar en cada uno de los estados ( $\pi_i$ ), la distribución de probabilidad de los símbolos dentro del estado, y las probabilidades de transición entre estados. Esta última constituye lo que se llama matriz de transición entre estados. Se ha observado que cuanto mayor es  $N$  más resolución se obtiene, pero a partir de cierto tamaño se detecta que quedan estados vacíos, y eso no es eficiente. Existe un compromiso entre la precisión del modelo y la complejidad. Para acotar el número de estados rentables del modelo se han establecido unos criterios para decidir el tamaño óptimo del modelo oculto de Markov, que se presentan y justifican en el capítulo 4.

La posibilidad de cambiar la velocidad instantánea de los datos de una trama a la siguiente, es decir cada 10 ms, es una de las características más importantes e innovadoras del sistema UMTS, ya que ofrece a los usuarios una gran capacidad de adaptación a diferentes condiciones de la celda para manejar en la misma conexión diferentes situaciones de tráfico. Estos cambios implican que para emular correctamente un canal del sistema UMTS debe considerarse la posibilidad de cambiar el Spreading Factor (SF), y/o de que varíe la  $E_b/\eta$ , de una trama a la siguiente. Cuando alguno de los valores que definen el canal emulado por el HMM cambia, los parámetros de este también cambian, y en consecuencia debe cambiarse el HMM. Para dicho cambio de modelo se han estudiado dos opciones: (1) **Cambio de canal sin memoria de errores**: significa que una vez seleccionado el nuevo HMM se busca el estado inicial a partir de las probabilidades iniciales de los estados  $\pi_i$ , (2) **Cambio de canal con memoria de errores**: significa que una vez seleccionado el nuevo HMM se busca el estado inicial partiendo del estado que representa el mismo número de errores que han sucedido en la última trama, y usando la matriz de transición entre estados del nuevo modelo. Tras varias pruebas se ha llegado a la conclusión de que la memoria no es relevante, ya que aunque la secuencia instantánea de errores generados no es igual, en una simulación suficientemente larga los histogramas de error de una y otra alternativas son muy similares.

En una tercera fase se construyó un simulador de la capa de enlace: MAC, RLC y PDCP, para la red terrestre de acceso radio del UTRAN (UTRAN: UMTS Terrestrial Radio Access Network). Para implementar la capa de enlace se buscó la simplicidad y la flexibilidad. Para ello, se decidió trabajar con diferentes módulos, de manera que sea fácil cambiar parámetros y funciones de cada una de las entidades involucradas. Con el objetivo de poder dar el máximo de difusión a estas investigaciones y pensando en otros futuros usuarios de este simulador, se buscó un lenguaje de programación fácilmente entendible por alguien ajeno al entorno de programación. El lenguaje que en nuestra opinión mejor se adapta a dichas necesidades es el SDL (Specification and Description Language). Este lenguaje fue inventado y estandarizado por la CCITT, para la especificación y descripción de sistemas principalmente en el campo de las telecomunicaciones.

En el capítulo 5 se describen las tres subcapas de la capa de enlace, MAC (Medium Access Control), RLC (Radio Link Control) y PDCP (Packet Data Convergence Protocol), se detalla su implementación y se muestra el proceso de validación.

La implementación de las subcapas MAC y RLC se realizó usando dicho lenguaje SDL y también especificándolo en lenguaje estándar C. A primera vista este doble trabajo puede parecer redundante e inútil. Sin embargo diferentes situaciones vividas en el transcurso de esta tesis justifican la existencia de estos dos frentes. En primer lugar la necesidad de colaborar con otras investigaciones dentro del grupo de comunicaciones radio impulsó el desarrollo en lenguaje C clásico. Estas investigaciones estaban desarrollando, en lenguaje C clásico, algoritmos de gestión de recursos radio y requerían un emulador de las capas bajas del interfaz radio de la UTRAN para poder testarlos en tiempo real. En el proceso de ajuste y manipulación del simulador se hizo patente la dificultad de usar el simulador por parte de alguien diferente al propio programador, y por este motivo se buscó un lenguaje que fuera simple y ofreciera gran flexibilidad. Así pues se inició el diseño en SDL con el objetivo de construir un simulador fácilmente entendible y manejable por alguien diferente a su creador.

En la implementación del MAC se ha reproducido únicamente el MAC para un usuario, llamado usuario de referencia, y los canales lógicos dedicados DTCH (Dedicated Traffic CHannel) y DCCH (Dedicated Control CHannel). Estos canales lógicos pueden proyectarse en un canal de transporte dedicado (DCH) o bien en los canales comunes RACH (Random Access CHannel) y FACH (Forward Access CHannel).

En la implementación del RLC se han reproducido los tres modos de operación definidos en la especificación del 3GPP: modo transparente (TM), modo sin confirmación (UM) y modo con confirmación (AM). Para el modo AM se eligió entre los diferentes tipos de esquemas ARQ (Automatic Repeat Request) existentes un mecanismo ARQ de repetición selectiva (selective-repeat ARQ) con un buffer de recepción finito.

---

La implementación de la subcapa PDCP se realizó únicamente en lenguaje C ya que en ella los parámetros a configurar son mínimos. Tan sólo debe considerarse si se aplica compresión de cabeceras o no. En caso de tener la compresión habilitada se aplica el protocolo ROCH (**R**obust **H**eaders **C**ompression) para las aplicaciones que usen los protocolos RTP/UDP/IP, y el protocolo IPHC (**I**P **H**eaders **C**ompression) para las aplicaciones que usen los protocolos TCP/IP.

Con el fin de mostrar la efectividad de la compresión y comprobar si es o no útil introducir compresión de cabeceras en las aplicaciones disponibles se han fijado dos parámetros de análisis: el *Tamaño medio de las cabeceras* y la *Eficiencia media*. Al final del capítulo 5 se presentan los resultados de analizar las aplicaciones de videoconferencia y video streaming, que usan RTP/UDP/IP, y de web, telnet, e-mail y ftp, que usan TCP/IP.

De dicho análisis se ha llegado a las siguientes conclusiones: para la videoconferencia el uso de la compresión ROCH es beneficioso ya que disminuye el tamaño medio de la cabecera y las eficiencias mejoran. Por contra, para el video streaming el uso de la compresión ROCH no es recomendable ya que, para la aplicación usada, el campo `SSCR_ID` que se suponía constante, varía, y obliga al envío de paquetes de inicialización y refresco para comunicar estos cambios al descompresor. Estos paquetes hacen aumentar el tamaño medio del paquete y en consecuencia disminuir la eficiencia.

Para las aplicaciones web, telnet, e-mail y ftp, la compresión consigue reducir el tamaño medio de las cabeceras de forma importante y la eficiencia siempre mejora. La reducción más importante en el tamaño de las cabeceras se ha observado en la aplicación ftp que llega a niveles de reducción de la cabecera del 90%. La aplicación que en media mejora más su eficiencia con la compresión es la telnet que consigue una mejora de aproximadamente 30%. Y las aplicaciones que mejoran menos son la web y la ftp, ya que la eficiencia ya es de por sí muy alta pues los paquetes llevan gran cantidad de datos y la presencia del compresor no es imprescindible. Todo lo contrario, si no se comprime se ahorra tiempo y problemas de recepción.

En una cuarta fase se integraron estos dos bloques (emulador del canal radio y la capa física, y simulador de la capa de enlace) y se construyó así un emulador de la red de acceso radio del UMTS. Este emulador se bautizó con el nombre de URAE, **U**MTS **R**adio **A**ccess **E**mulator. El URAE permite estudiar en tiempo real los servicios de paquetes de datos en el sistema UMTS. Este análisis es muy importante para integrar los servicios ofrecidos en Internet, en los futuros sistemas de comunicaciones móviles de tercera generación.

En el capítulo 6 se describe a nivel conceptual el emulador creado, sus opciones y algunos ejemplos de estudios que se pueden realizar. En concreto se presentan seis análisis realizados con el URAE:

- Impacto del tamaño de los paquetes: permite analizar el tamaño adecuado de los paquetes de las aplicaciones para ser transmitidos por el interfaz radio del UMTS.

- Modo transparente versus modo reconocimiento: permite analizar la inclusión o no de un mecanismo de retransmisiones en diferentes aplicaciones en función de sus requerimientos.
- Efecto de la velocidad del móvil: permite comparar las prestaciones de un usuario desplazándose a 50 km/h con uno desplazándose a 120 km/h. El URAE también permite comparar con usuarios desplazándose a 3 km/h.
- Necesidad de códigos de canal: permite analizar diferentes esquemas de codificación de canal.
- Influencia de la carga del sistema: permite estudiar el comportamiento de aplicaciones en diferentes situaciones de carga del sistema.
- Tráfico interactivo versus tráfico background: permite comparar el comportamiento de diferentes tipos de tráfico con el objetivo de elegir el más adecuado para una determinada aplicación.

La flexibilidad del URAE permite probar técnicas, protocolos y algoritmos de gestión de recursos en distintas situaciones de la red para servicios orientados a paquetes. En particular se destaca su importante papel en el análisis de algoritmos innovadores de gestión de recursos radio para garantizar calidad de servicio.

## 7.2. Líneas futuras de investigación

Existen varios aspectos que podrían ampliar el trabajo realizado en la presente Tesis Doctoral y crear líneas futuras de investigación. A continuación se comentan las que se consideran más destacables:

En primer lugar sería interesante completar el emulador de la red de acceso radio implementando todos los canales definidos en el estándar para el modo FDD. En particular resultaría interesante reproducir el canal de transporte compartido del enlace descendente (DSCH: **D**ownlink **S**hared **C**hannel), y del ascendente (CPCH: **C**ommon **P**acket **C**hannel), junto con la inclusión de más de un usuario de referencia. Es decir, probar aplicaciones de más de un usuario que compartan un mismo canal físico de datos.

También resultaría interesante ampliar el emulador al modo TDD, y obtener así un emulador general del acceso radio del sistema UMTS. En principio este trabajo no debería suponer un gran esfuerzo ya que se puede aprovechar el emulador creado para el modo FDD, y realizar sólo los cambios en aquellos módulos que lo requieran. La metodología utilizada para crear el emulador de la red de acceso radio para el modo FDD permite hacer estos cambios con rapidez y simplicidad.

---

Otra posibilidad consistiría en utilizar el emulador del acceso radio construido para estudiar diferentes protocolos de acceso al medio físico. En particular se ha empezado a estudiar los protocolos con contención DQRAP/CDMA [1] (**D**istributed **Q**ueueing **R**andom **A**ccess **P**rotocol) y ISMA-DS/CDMA [2] (**I**nhibit **S**ense **M**ultiple **A**ccess). El protocolo DQRAP/CDMA permite mantener controlado el número máximo de transmisiones simultáneas sin necesidad de reservar un código de transmisión para cada usuario. Este hecho permite realizar todas las transmisiones en modo paquete usando un número muy reducido de códigos. El protocolo ISMA-DS/CDMA, entre otras particularidades, dota a los terminales del conocimiento de los códigos ocupados, evitando así un buen número de colisiones.

Asimismo se podría estudiar en profundidad los mecanismos de gestión de recursos radio (RRM: **R**adio **R**esources **M**anagment) propuestos en UTRA para mejorar el rendimiento de los canales en el interfaz radio. En este sentido se deberían incorporar con detalle las funciones de gestión de la admisión de nuevos usuarios, control de colisiones, cambio de velocidad, etc. Cabe destacar que los algoritmos de RRM no van a ser objeto de estandarización, por lo que el estudio de nuevos algoritmos usando el emulador creado en la presente tesis puede ser de gran utilidad para verificar la viabilidad real de implantación de los mismos.

También se podría usar el emulador creado para implementar y estudiar la modificación de aplicaciones existentes y la creación de nuevas aplicaciones, para hacerlas más adecuadas al interfaz aire.

Por último, también sería interesante tratar de forma más profunda los aspectos relacionados con el impacto del entorno radio en el funcionamiento de protocolos como son TCP, IP, UDP, etc. Algunos trabajos ya han presentado propuestas de mejora para TCP en entornos móviles [3].

### **7.3. Bibliografía**

[1] Luis Gonzaga Alonso Zárate, “Técnicas de Acceso y gestión de Recursos para garantizar Calidad de Servicio en Sistemas de Comunicaciones Móviles basados en CDMA”, Tesis doctoral, Universitat Politècnica de Catalunya, Mayo 2001.

[2] Jordi Pérez Romero, “Contribución al Estudio de las Técnicas de Acceso Aleatorio y al Análisis de la Calidad de Servicio en Sistemas de Transmisión por Paquetes para Comunicaciones Móviles”, Tesis doctoral, Universitat Politècnica de Catalunya, Abril 2001.

[3] Juan Gabriel Rendón Scheider, “Evaluación de las prestaciones de los protocolos de internet en ambientes móviles celulares”, Tesis doctoral, Universitat Politècnica de Catalunya, Octubre de 2001.

## Apéndice I. Ejemplo de parámetros de un HMM

En este apéndice se presentan a modo de ejemplo los parámetros de uno de los HMMs que forman el banco de parámetros que se ha creado. En concreto se muestra un modelo que emula un canal radio del enlace descendente, sin códigos de canal incluidos, con spreading factor 4, de un móvil con velocidad media (50 km/h), para una relación  $E_b/\eta = 9\text{dB}$  y con control de potencia en lazo cerrado. Su nombre es D4M9s según la nomenclatura descrita en el capítulo 4 página 4-69.

ENTORNO = outdoor50  
 UPDOWN = down  
 EBN0 = 9.00  
 SF = 4  
 BITS = 19200  
 ESTADOS = 16

### #MATRIZ DE TRANSICIÓN

0.0000	0.0000	0.0000	0.1250	0.0000	0.0625	0.1250	0.1875	0.1250	0.1250	0.0000	0.0625	0.0625
	0.0000	0.0625	0.0625									
0.0000	0.1250	0.1875	0.0000	0.0625	0.0625	0.0000	0.1250	0.0000	0.1250	0.0000	0.1250	0.0625
	0.0625	0.0000	0.0625									
0.0625	0.0000	0.0000	0.0625	0.2500	0.0625	0.1250	0.0000	0.0000	0.0000	0.0000	0.0625	0.0625
	0.0625	0.0625	0.1875									
0.0556	0.1111	0.0000	0.0000	0.0000	0.1111	0.0000	0.0000	0.0000	0.2778	0.0556	0.0000	0.1111
	0.1111	0.1111	0.0556									
0.0000	0.0000	0.0667	0.0000	0.0667	0.0667	0.2000	0.0667	0.0000	0.0667	0.0000	0.1333	0.0667
	0.0667	0.1333	0.0667									
0.1875	0.0000	0.0625	0.0625	0.0625	0.0000	0.1875	0.0000	0.1250	0.0625	0.0000	0.0625	0.0000
	0.0625	0.0625	0.0625									
0.1875	0.0625	0.0625	0.1875	0.0000	0.0625	0.0000	0.0625	0.0000	0.0625	0.0000	0.1250	0.0625
	0.1250	0.0000	0.0000									
0.1176	0.1176	0.0000	0.1176	0.0000	0.1765	0.0588	0.1176	0.0000	0.0000	0.0000	0.0000	0.0000
	0.1176	0.0588	0.1176									
0.0000	0.0625	0.0625	0.0625	0.0625	0.0000	0.0000	0.0625	0.1875	0.0000	0.1875	0.0625	0.1875
	0.0625	0.0000	0.0000									
0.0588	0.0000	0.1176	0.0588	0.0588	0.0000	0.1176	0.0588	0.1765	0.0000	0.0000	0.1765	0.0000
	0.1176	0.0000	0.0588									



0.1333	0.1333	0.0667	0.1333	0.0667	0.0000	0.0667	0.2000	0.0667	0.0667	0.0000	0.0000	0.0667
	0.0000	0.0000	0.0000									
0.0588	0.1176	0.0588	0.0588	0.0588	0.0000	0.0000	0.0000	0.0588	0.0000	0.2353	0.0588	0.0000
	0.0000	0.1176	0.1765									
0.0000	0.1875	0.0000	0.0000	0.0625	0.1250	0.0000	0.0000	0.0625	0.0625	0.0625	0.1250	0.1250
	0.1250	0.0000	0.0625									
0.0625	0.0000	0.1875	0.0625	0.0000	0.1250	0.0625	0.1250	0.0000	0.0000	0.0625	0.0625	0.0625
	0.0000	0.1250	0.0625									
0.0667	0.0000	0.1333	0.1333	0.0667	0.1333	0.0667	0.0000	0.0667	0.0000	0.1333	0.0000	0.0000
	0.0667	0.1333	0.0000									
0.0000	0.0625	0.0625	0.0625	0.1250	0.0000	0.0000	0.0625	0.0625	0.1875	0.1875	0.0000	0.1250
	0.0000	0.0625	0.0000									

#	Error mínimo	Error máximo	Media de errores	Probabilidad
#ESTADO 0				
0		91	74.125000	0.061776
#ESTADO 1				
92		108	101.375000	0.061776
#ESTADO 2				
109		120	115.058838	0.065637
#ESTADO 3				
121		130	126.666672	0.069498
#ESTADO 4				
131		141	137.066666	0.057915
#ESTADO 5				
142		155	148.749985	0.061776
#ESTADO 6				
156		164	159.250000	0.061776
#ESTADO 7				
165		173	168.588226	0.065637
#ESTADO 8				
174		181	176.750031	0.061776
#ESTADO 9				
182		191	186.352951	0.065637
#ESTADO 10				
192		202	198.599991	0.057915
#ESTADO 11				
203		216	209.823502	0.065637
#ESTADO 12				
217		235	226.000000	0.061776
#ESTADO 13				
236		260	246.812515	0.061776
#ESTADO 14				
261		292	273.866669	0.057915
#ESTADO 15				
293		19200	349.250000	0.061776

## Apéndice II. El lenguaje de especificación SDL

En este apéndice se describen las principales características y la filosofía del lenguaje estándar de especificación SDL (Specification and Description Language) [1], [2].

La cantidad de productos software para simular sistemas de telecomunicaciones ha aumentado enormemente en las últimas décadas. Cada día más sistemas son distribuidos y se ejecutan en entornos heterogéneos, por lo que se ha generado la necesidad de crear un método internacional de especificación formal capaz de comunicar estos diferentes sistemas, pensando principalmente en el desarrollo de los futuros sistemas de telecomunicaciones caracterizados por una gran flexibilidad. La creación de este método requiere un lenguaje de especificación basado en un conjunto de conceptos bien definidos, incluyendo especificaciones precisas y bases rigurosas para analizar el código generado, herramientas capaces de verificar si la implementación concuerda con la especificación, y un interfaz gráfico para simplificar la generación tradicional del código. El lenguaje SDL ha sido diseñado como solución a dicha situación. En efecto, sus puntos fuertes son la extrema flexibilidad y la capacidad de describir la estructura, el comportamiento, y los datos de los diferentes sistemas.

SDL (en castellano, Lenguaje de Especificación y Descripción) es un lenguaje estándar de programación en alto nivel. Está orientado a objetos, es formal, se representa en modo gráfico, y además proporciona un alto grado de reutilización. SDL está concebido para describir con gran rigurosidad matemática sistemas complejos que se comunican en tiempo real y son dirigidos por eventos. Ha sido desarrollado y estandarizado por ITU-T (The Telecommunications standardization sector of the International Telecommunications Union) en la recomendación Z.100 del Libro Azul.

Las características más importantes del SDL que lo distinguen de otras tecnologías son:

- 
- *Estándar*: SDL es un lenguaje de estandarización internacional, como se ha mencionado anteriormente recomendado por la ITU-T.
  - *Formal*: SDL es un lenguaje formal por lo que asegura precisión, consistencia y claridad en el diseño.
  - *Gráfico y basado en símbolos*: SDL es un lenguaje gráfico que se basa en símbolos lo que proporciona claridad y facilidad de uso.
  - *Orientado a objetos*: SDL es un lenguaje orientado a objetos (OO) por lo que presenta todas las propiedades de los lenguajes OO, y además extiende el concepto tradicional de clase OO personalizándolo para aplicaciones técnicas.
  - *Alta capacidad de testeo*: el formalismo de SDL le proporciona un alto grado de testeo, que a su vez agiliza y proporciona calidad a las mejoras que se deseen realizar.
  - *Portable y Abierto*: las implementaciones SDL son independientes de los compiladores, los sistemas operativos, los procesadores, los mecanismos de comunicación entre procesos y los métodos de distribución. Una misma implementación SDL puede usarse en varias arquitecturas y configuraciones diferentes.
  - *Reutilizable y Escalable*: el grado de reutilización es muy alto y está relacionado con la facilidad de escalar fácilmente los sistemas y con la independencia de las diferentes plataformas.
  - *Eficiente*: el formalismo y el nivel de abstracción proporcionado por SDL lo convierten en un buen candidato para utilizar técnicas de optimización sofisticadas.

Los sistemas descritos mediante SDL consisten en un conjunto de procesos que se ejecutan simultáneamente y se comunican entre ellos por medio de señales. Cada proceso se describe por medio de una máquina de estados finitos. A continuación se presenta brevemente la historia de SDL y su estructura abierta y flexible que lo convierte en un lenguaje eficiente y fácil de usar.

## **AII.1. Historia**

El desarrollo del lenguaje SDL comenzó en el año 1972, gracias al trabajo y esfuerzo de un grupo de estudio formado por quince miembros dentro del CCITT (Comité Consultatif International Télégraphique et Téléphonique) que representaban a varios países y a diferentes compañías del ámbito de las telecomunicaciones, tales como Bellcore, Ericsson, o Motorola. Las primeras investigaciones se centraron en diseñar un lenguaje estándar para la especificación del comportamiento de los sistemas de conmutación para telecomunicaciones.

La primera versión del lenguaje fue publicada en 1976 en las Recomendaciones Z.101 a Z.103 del Libro Naranja, seguida por nuevas versiones en los años 1980, 1984, 1988. En 1992 se introdujo por primera vez la característica de orientado a objetos (OO), y en 1996 se hicieron algunas actualizaciones al respecto. Finalmente en el año 2000 se realizaron algunos cambios para eliminar características que no eran soportadas por las herramientas, y para fortalecer la programación OO. Las versiones más recientes han ampliado el lenguaje considerablemente y han simplificado notablemente la interconexión, de manera que SDL es a día de hoy un lenguaje maduro y completo en todos los sentidos.

## **AII.2. Áreas de aplicación del lenguaje SDL**

En un principio el lenguaje de programación SDL se desarrolló para y por el ámbito de las telecomunicaciones, aunque últimamente está llegando a ser cada vez más popular en otras áreas [3].

Algunos campos donde se usa el lenguaje SDL fuera del área de las telecomunicaciones son:

- ⇒ Comunicaciones basadas en los satélites
- ⇒ Estandarización aeronáutica
- ⇒ Equipo médico
- ⇒ Sistema de control ferroviario
- ⇒ Protocolos de comunicación en vehículos

## **AII.3. SDL y otros lenguajes**

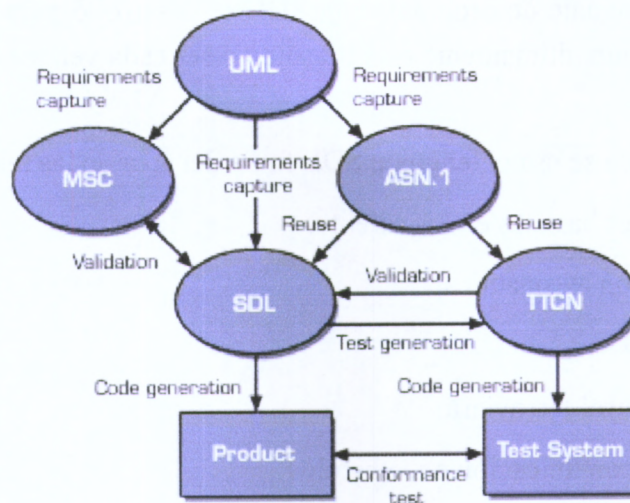
SDL dispone de un número considerable de ventajas frente a los lenguajes de programación de alto nivel y más aún, frente a los tradicionales lenguajes de bajo nivel, tales como C, C++ o Java. SDL tiene una gramática muy extensa que permite definir el comportamiento de un sistema de manera no ambigua. Además, SDL es un lenguaje gráfico, por tanto, sus diagramas son fácilmente comprensibles para personal no técnico. Esto hace que se aumente la comunicación entre el diseñador del sistema y el cliente, asegurando que el proceso se desarrolle perfectamente desde la especificación de los requerimientos hasta la implementación.

La precisión y formalidad de SDL proporcionan la posibilidad de emplear compilación del código desarrollado por medio de herramientas SDL a lenguajes de más bajo nivel, como C ó C++. Esto significa que los sistemas SDL pueden ser traducidos a aplicaciones ejecutables sin programación manual.

SDL se usa en todo el mundo para el desarrollo de todo tipo de sistemas de comunicación complejos. En el campo de las telecomunicaciones, SDL es el lenguaje elegido para el desarrollo

de un amplio rango de aplicaciones hardware y software. Como ejemplos tenemos los teléfonos móviles, switches, sistemas de radio, plataformas de gestión de redes y sistemas de servicios de red. Tenemos otros ejemplos en estándares de telecomunicación, como GSM (Global System for Mobile communications), ISDN (Integrated Services Digital Network), INAP (Intelligent Network Application Protocol), etc.

SDL está desarrollado para ser el centro de proyectos de gran escala, debido a sus facilidades para cooperar con otros lenguajes. En efecto SDL ofrece una buena interfaz con lenguajes tradicionales como C++, MSC (Message Sequence Chart), TTCN (Tree and Tabular Combined Notation), ASN1 (Abstract Syntax Notation One), y UML (Unified Modeling Language). En la Figura AII-1 se muestran las relaciones entre estos lenguajes.



**Figura AII-1. Relaciones entre SDL y otros lenguajes**

#### **AII.4. Modelo teórico SDL**

El modelo teórico básico de un sistema SDL consiste en un conjunto de máquinas de estado finito que se ejecutan en paralelo. Estas máquinas son independientes y se comunican por medio de señales discretas.

Un sistema SDL está formado por una estructura jerárquica formada por sistema, bloque, proceso y procedimiento. La comunicación entre bloques se realiza por medio de señales, opcionalmente con parámetros, que viajan por los canales. Además dispone de tipos abstractos de datos y de módulos, que son paquetes de bloques tipo, procesos tipo, datos tipo y/o señales tipo que han sido desarrollados por separado y se importan al sistema para su uso. A continuación se describen brevemente estos conceptos.

### AII.4.1 Estructura

El lenguaje SDL contiene cuatro niveles jerárquicos principales, que se relacionan como indica la Figura AII-2:

- ⇒ Sistema.
- ⇒ Bloque.
- ⇒ Proceso.
- ⇒ Procedimiento.

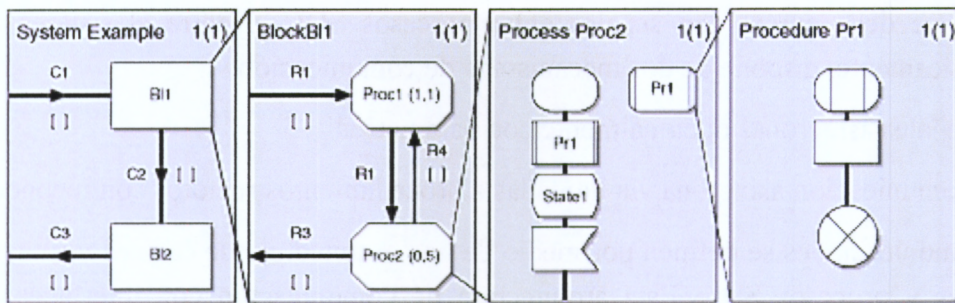


Figura AII-2. Estructura jerárquica en SDL

Los objetivos de dividir el sistema en la jerarquía: sistema, bloque, proceso y procedimiento son los siguientes:

- ✓ Esconder información, es decir, mover los detalles que no son importantes para una idea general del sistema a niveles más bajos.
- ✓ Seguir unas sub-divisiones funcionales naturales.
- ✓ Crear módulos de tamaño comprensibles.
- ✓ Crear una correspondencia con el actual hardware o software.
- ✓ Reutilización de especificaciones de sistemas ya existentes.

Cada tipo de proceso SDL está definido como una máquina de estados finita. Cada estado de la máquina es implementado por un procedimiento. Los procedimientos pueden ser recursivos; locales a sus procesos o globales, dependiendo de su definición. El lenguaje SDL también soporta el paradigma de procedimientos remotos, que permite hacer una llamada a un procedimiento que se ejecuta en el contexto de otro proceso.

Los procesos SDL tienen espacios de memoria separados, es decir, los datos son locales a un proceso o a un procedimiento. Este es un aspecto muy importante que reduce enormemente el número de deficiencias y hace más robusto el sistema.

---

Un conjunto de procesos pueden agruparse en un bloque (es decir, un subsistema). Los bloques pueden ser incluidos unos en otros, para dividir el sistema recursivamente en subsistemas encapsulados más pequeños y fácilmente manejables. Estos mecanismos de división, y las interfaces entre los distintos subsistemas, facilitan los esfuerzos de desarrollo de grandes proyectos.

#### **AII.4.2 Comunicación**

El lenguaje SDL es un lenguaje que no utiliza datos globales. Toda la información a intercambiar se debe enviar con señales entre procesos SDL, o entre el sistema SDL y su entorno. Básicamente, dispone de dos mecanismos de comunicación:

- ✓ Señales asíncronas opcionalmente con parámetros.
- ✓ Comunicación asíncrona vía llamadas a procedimientos remotos con reconocimiento.

Estas comunicaciones se definen por medio de una arquitectura de canales y rutas de señales entre bloques y procesos SDL. Esta arquitectura de comunicación con interfaces de señales simplifica el desarrollo de grandes proyectos y asegura la consistencia entre las diferentes partes de un sistema.

En SDL se definen temporizadores en cada elemento abstracto e inteligente. El tiempo es un aspecto importante en los sistemas en tiempo real, así como en la mayoría de los sistemas distribuidos. Un proceso SDL puede inicializar temporizadores que expiren dentro de ciertos periodos de tiempo para implementar alarmas cuando ocurre una excepción y también para medir y controlar el tiempo de respuesta desde otros procesos y sistemas.

Cuando finaliza un temporizador SDL, el proceso que inicializó ese temporizador recibe la notificación, por medio de una señal, del mismo modo que recibe otras señales. De hecho, un temporizador que finaliza se interpreta del mismo modo que una señal.

Otros aspectos del concepto de señalización en SDL son:

- Las prioridades de señales y procesos no entran dentro del campo de SDL. Estas características se dejan para la fase de implementación, donde el usuario, con directivas especiales, puede asignar prioridades a las señales y a los procesos.
- Una señal SDL no puede enviarse a más de una instancia de un proceso simultáneamente. Para activar el envío a más instancias, el usuario puede incluir un paquete con funciones de propósito general que lo permitan.

En la Figura AII-3 se muestra la señal [*Sig1*] entre dos procesos *Proc1* y *Proc2* viajando a través del canal *R2* dentro del bloque *BL1*.

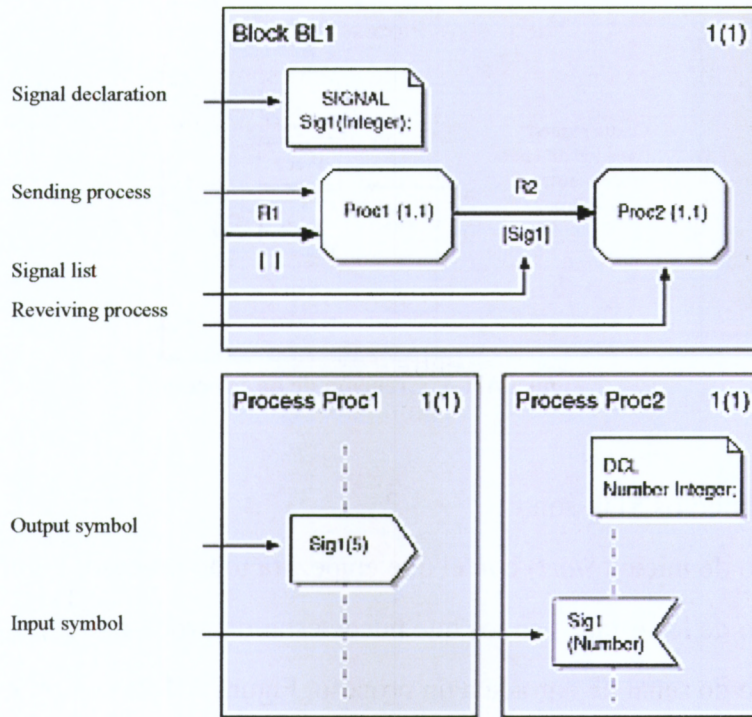


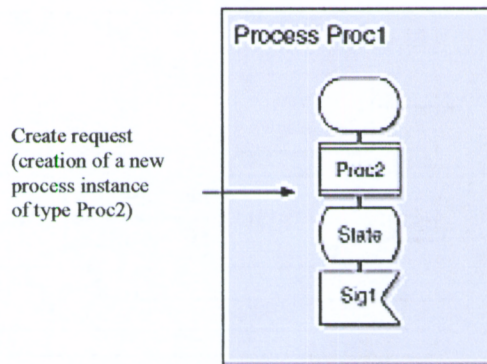
Figura AII-3. Señal entre procesos

### AII.4.3 Comportamiento

El comportamiento dinámico de un sistema SDL viene descrito por medio de sus procesos. La jerarquía sistema/bloque es sólo una descripción estática de la estructura del sistema. Los procesos en SDL pueden ser creados al comienzo del sistema o creados y finalizados en el momento de ejecución. Podrá existir más de una instancia de un proceso. Cada una tendrá un identificador de proceso (Pid) único. Esto permitirá enviar señales a instancias individuales de un proceso. El concepto de proceso e instancias de proceso que se ejecutan de modo autónomo y concurrente convierte a SDL en un verdadero lenguaje en tiempo real.

En la Figura AII-4 se muestra la creación de una nueva instancia de un proceso *Proc2* durante la ejecución.

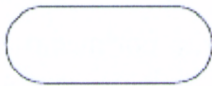




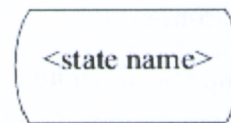
**Figura AII-4. Creación de un proceso**

Los principales símbolos SDL son:

- el símbolo de inicio (*Start*) con el que empezará todo proceso, Figura AII-5;
- el símbolo de los estados en los que puede estar un proceso, Figura AII-6;
- el símbolo de señal de entrada a un proceso, Figura AII-7;
- el símbolo de señal de salida de un proceso, Figura AII-8;



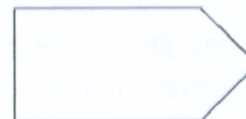
**Figura AII-5. Símbolo de inicio**



**Figura AII-6. Símbolo de estado**

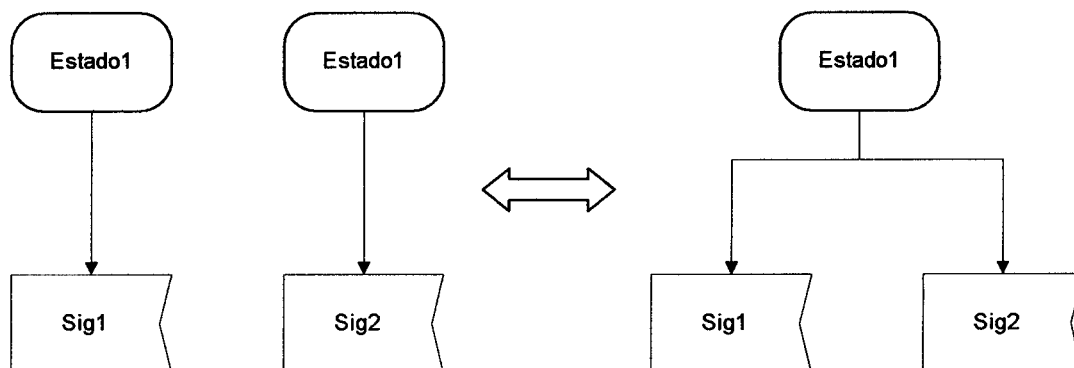


**Figura AII-7. Señal de entrada a un proceso**



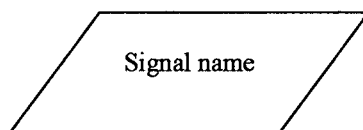
**Figura AII-8. Señal de salida de un proceso**

Un proceso sale de un estado cuando recibe una determinada señal de entrada. Puede haber varias señales que provoquen la salida de un estado. Esta situación puede representarse indistintamente de las dos formas indicadas en la Figura AII-9.



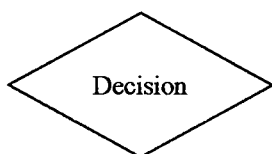
**Figura AII-9. Representaciones de la salida del *Estado1* por la señal *Sig1* o *Sig2***

En algunas ocasiones puede interesar guardar una señal recibida mientras el proceso está en un estado en el que no puede procesar esta señal. Para realizar esta acción existe el símbolo *save*, Figura AII-10. Cuando el proceso pasa a un estado en el que puede procesarse esta señal guardada, ésta se procesa inmediatamente.

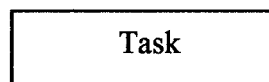


**Figura AII-10. Símbolo de *Save***

En las transiciones entre un estado y el siguiente el proceso puede enviar señales de salida, dirigidas a otros procesos, y también puede tomar decisiones, Figura AII-11, y realizar tareas, Figura AII-12.



**Figura AII-11. Símbolo de decisión**



**Figura AII-12. Símbolo de tarea**

Existen entre otros, símbolos específicos para llamar a procedimientos, para crear procesos, y para insertar comentarios.

Por último en la Figura AII-13 se muestra el símbolo de terminar un proceso.



Figura AII-13. Símbolo de terminar

#### AII.4.4 Datos

SDL acepta dos modos de describir los datos: tipo abstracto de datos (ADT: **A**bstrac**T** **D**ata **T**ype) y notación de sintaxis abstracta uno (ASN.1: **A**bstrac**T** **S**yntax **N**otation **O**ne). ASN.1 permitirá compartir datos entre lenguajes, así como reutilizar estructuras de datos ya existentes.

ADT dará a SDL grandes posibilidades. Un tipo de datos abstracto es un tipo de datos sin una estructura específica. En vez de ello, se especifica un conjunto de valores, un conjunto de operaciones permitidas y un conjunto de ecuaciones que las operaciones deben cumplir, ver Figura AII-14. Esto permite transformar tipos de datos SDL a tipos usados en lenguajes de alto nivel de una manera sencilla.

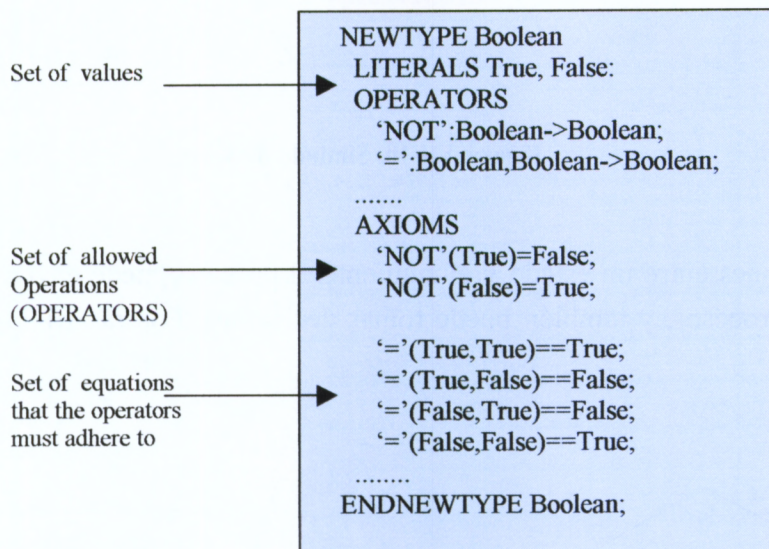


Figura AII-14. Tipo abstracto de datos

El conjunto de tipos predefinidos de SDL hace posible trabajar con datos en SDL de la manera tradicional. Las variables de los tipos estándar pueden ser declaradas como (se indica en inglés, por ser la manera de especificarlo):

- Integer
- Real
- Natural
- Boolean
- Character
- Duration
- Time
- Charstring
- Pid
- Tipos complejos de datos (pueden crearse con Array, String y Struct).

#### **AII.4.5 Herencia**

Los conceptos orientados a objetos de SDL dan al usuario herramientas para re-estructurar y re-utilizar. El concepto está basado en la declaración de tipos. Las declaraciones de tipos pueden situarse en cualquier sitio, bien dentro del sistema cerca de su contexto, o en el nivel del sistema.

Una de las mayores ventajas de usar un lenguaje orientado a objetos es que pueden crearse nuevos objetos de una manera muy simple e intuitiva, añadiendo nuevas propiedades a objetos que ya existen o redefiniendo propiedades de objetos existentes. Esto es lo que se conoce comúnmente como especialización.

En SDL, la especialización de tipos debe cumplir con dos requerimientos:

- Un subtipo debe incluir propiedades no incluidas en el tipo del que se deriva.
- Un subtipo puede redefinir los tipos virtuales y las transiciones virtuales definidas en el tipo del que se deriva.

---

## **AII.5. Bibliografia**

[1] Ferenc Belina and Dieter Hogrefe "The CCITT-Specification and Description. Language SDL", North-Holland, Computer Networks and ISDN Systems 16, pp. 311-341, 1989.

[2] Ermanno Berruto, Tomas Brännlund, Jan Gustafsson and Wolfgang Schott, "An SDL methodology used for specifying the radio protocols in a CDMA system", Vehicular Technology Conference, Stockholm, 8-10 June 1994, pp. 554-558.

[3] [www.sdl-rt.org](http://www.sdl-rt.org)

## Apéndice III. Especificación de los TF para voz y web

En este apéndice se reproducen las tablas de la especificación 34.108 versión 3.2.0 del 3GPP que se han usado para seleccionar los TF del servicio de voz y web.

### AIII.1. Transport channel parameters for Conversational / speech / UL:12.2 kbps / CS RAB

Para el servicio de voz los valores usados se encuentran en el apartado 6.10.2.4.1.4.1.1.1 de la página 71.

Higher layer	RAB/Signalling RB	RAB subflow #1	RAB subflow #2	RAB subflow #3	
RLC	Logical channel type	DTCH			
	RLC mode	TM	TM	TM	
	Payload sizes, bit	39, 81 (alt. 0, 39, 81)	103	60	
	Max data rate, bps	12200			
	RLC header, bit	0			
MAC	MAC header, bit	0			
	MAC multiplexing	N/A			
Layer 1	TrCH type	DCH	DCH	DCH	
	TB sizes, bit	39, 81 (alt. 0, 39, 81)	103	60	
	TFS* <sup>1</sup>	TF0, bits	0x81(alt. 1x0* <sup>2</sup> )	0x103	0x60
		TF1, bits	1x39	1x103	1x60
		TF2, bits	1x81	N/A	N/A
	TTI, ms	20	20	20	
	Coding type	CC 1/3	CC 1/3	CC 1/2	
CRC, bit	12	N/A	N/A		

Max number of bits/TTI after channel coding	303	333	136
Uplink: Max number of bits/radio frame before rate matching	152	167	68
RM attribute	180-220	170-210	215-256

\*1: The TrCH corresponding to RAB subflow #1 should be used as the guiding TrCH, (see section 4.3 in TS25.212).

\*2: CRC parity bits are to be attached to RAB subflow#1 any time since number of TrBlks are 1 even if there is no data on RAB subflow#1 (see clause 4.2.1.1 in TS25.212.).

Y la combinaciones permitidas para estos tres subflujos son: (RAB subflow#1, RAB subflow#2, RAB subflow#3) = (TF0, TF0, TF0), (TF1, TF0, TF0), (TF2, TF1, TF1). Por lo que o bien se puede transmitir  $81+103+60=244$  bits, o no se puede transmitir (0 bits).

## AIII.2. Transport channel parameters for Interactive or background / UL:64 kbps / PS RAB

Para el servicio de consulta de páginas web, que es un servicio interactivo, los valores usados se encuentran en el apartado 6.10.2.4.1.24.1.1.1 de la página 108.

Higher layer	RAB/Signalling RB	RAB	
RLC	Logical channel type	DTCH	
	RLC mode	AM	
	Payload sizes, bit	320	
	Max data rate, bps	64000	
	RLC header, bit	16	
MAC	MAC header, bit	0	
	MAC multiplexing	N/A	
Layer 1	TrCH type	DCH	
	TB sizes, bit	336	
	TFS	TF0, bits	0x336
		TF1, bits	1x336
		TF2, bits	2x336
		TF3, bits	3x336
		TF4, bits	4x336
	TTI, ms	20	
	Coding type	TC	
	CRC, bit	16	
	Max number of bits/TTI after channel coding	4236	
	Uplink: Max number of bits/radio frame before rate matching	2118	
	RM attribute	130-170	

## **Apéndice IV. El lenguaje de descripción MSC**

En este apéndice se describe brevemente la técnica MSC.

### **AIV.1. Definición**

Message Sequence Chart (MSC) es una técnica estandarizada de descripción formal que se utiliza en el campo de las telecomunicaciones para mostrar la comunicación entre diversas partes del sistema. Está estandarizada por el ITU en su recomendación Z.120 [1]. Sus orígenes se encuentran en la herramienta gráfica de flujo de mensajes (Message Flow Graph tool) desarrollada por Peter Ladkin y Barbara Simons, a principios de los noventa. La primera versión del lenguaje apareció en 1992 y se conoce como MSC-1992, mientras que en 1996 se publicó una segunda versión extendida conocida como MSC-1996.

Esta técnica consiste básicamente en dibujar los mensajes que se envían de una instancia a otra; en términos de descripción SDL, estos mensajes corresponden a las señales enviadas por un proceso y consumidas por otro. La estructura básica de un diagrama MSC puede verse en la Figura AIV-1.



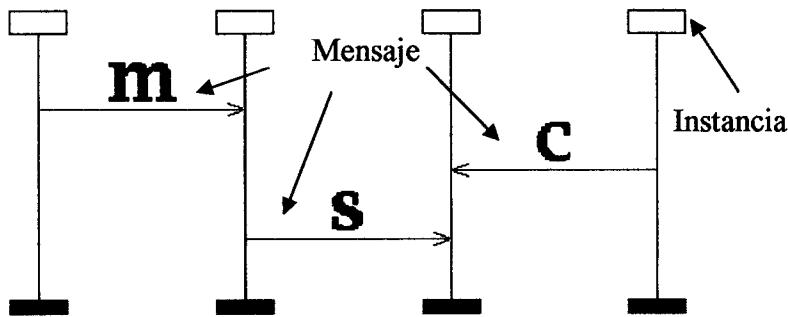


Figura AIV-1. Instancias y mensajes en diagramas MSC

## AIV.2. Áreas de aplicación

El área de aplicación de la técnica MSC es la especificación del comportamiento de la comunicación en sistemas en tiempo real, en particular:

- Presentar la ejecución de una simulación en forma gráfica fácil de comprender;
- Estudiar casos de prueba;
- Producir documentos que definan los requisitos del sistema;
- Contribuir a la fase de diseño y análisis orientado a objetos.
- Especificar interfaces y protocolos de alto nivel;
- Chequear la consistencia de MSC/SDL

## AIV.3. Entorno de trabajo

La herramienta SDT (SDL Design Tool) desarrollada por Telelogic utiliza los diagramas MSC como principal herramienta para simular el trabajo diseñado y creado con SDL. Para ello se ha incluido un editor MSC que soporta la notación gráfica definida en la recomendación Z.120.

## AIV.4. Bibliografía

[1] International Telecommunication Union (ITU), "CCITT Recommendation Z.120: Message Sequence Chart MSC" December 2001.

## Glosario de Siglas

3GPP	<b>3<sup>rd</sup> Generation Partnership Project</b>
3GPP2	<b>3<sup>rd</sup> Generation Partnership Project 2</b>
ACE	<b>Adaptative header ComprEssion</b>
ACTS	<b>Advanced Communications Technologies and Services</b>
ADSL	<b>Asymmetric Digital Subscriber Line</b>
ADT	<b>Abstract Data Type</b>
AICH	<b>Acquisition Indication CHannel</b>
AM	<b>Acknowledged Mode</b>
AMD	<b>Acknowledged Mode Data</b>
AMPS	<b>Advanced Mobile Phone System</b>
AMR	<b>Adaptive Multi Rate</b>
AP-AICH	<b>Access Preamble Acquisition Indication CHannel</b>
API	<b>Application Programming Interface</b>
ARIB	<b>Association of Radio Industry and Business</b>
ARQ	<b>Automatic Repeat Request</b>
ASN1	<b>Abstract Syntax Notation One</b>
ATDMA	<b>Adaptative TDMA</b>
AuC	<b>Authentication Centre</b>
BCCH	<b>Broadcast Control CHannel</b>
BCH	<b>Broadcast CHannel</b>
BER	<b>Bit Error Rate</b>
BG	<b>Border Gateway</b>
BLER	<b>BLock Error Rate</b>
BMC	<b>Broadcast/Multicast Control</b>
BPSK	<b>Binary Phase Shift Keying</b>
CC	<b>Call Control</b>
CCCH	<b>Common Control CHannel</b>
CCITT	<b>Comité Consultatif International Télégraphique et Téléphonique</b>
CCSA	<b>China Communications Standards Association</b>
CCTrCH	<b>Coded Composite Transport CHannel</b>
CD/CA-ICH	<b>CPCH Collision Detection/Channel Assigment Indicator CHannel</b>
CDMA	<b>Code Division Multiple Access</b>
CEPT	<b>Conférence Européenne des Postes et Télécommunications</b>
CI	<b>Connection Identifier</b>
CID	<b>Context IDentifier</b>
CN	<b>Core Network</b>
CoDiT	<b>Code Division Testbed</b>
CPCH	<b>Common Packet CHannel</b>

---

<b>CPICH</b>	<b>Common PIlot CHannel</b>
<b>CRC</b>	<b>Cyclic Redundancy Check</b>
<b>CS</b>	<b>Circuit Switched</b>
<b>CSICH</b>	<b>CPCH Status Indicator CHannel</b>
<b>CTO</b>	<b>Cordless Telephone</b>
<b>CTCH</b>	<b>Common Traffic CHannel</b>
<b>CTCP</b>	<b>Compress TCP</b>
<b>CTDMA</b>	<b>Code Time Division Multiple Access</b>
<b>CWTS</b>	<b>China Wireless Telecommunication Standard</b>
<b>DAMPS</b>	<b>Digital AMPS</b>
<b>DC</b>	<b>Dedicated Control</b>
<b>DCA</b>	<b>Dynamic Channel Allocation</b>
<b>DCCH</b>	<b>Dedicated Control CHannel</b>
<b>DCH</b>	<b>Dedicated CHannel</b>
<b>DCS1800</b>	<b>Digital Communication System en la banda de 1800Mhz</b>
<b>DECT</b>	<b>Digital European Cordless Telephone</b>
<b>DPCCH</b>	<b>Dedicated Physical Control CHannel</b>
<b>DPCH</b>	<b>Dedicated Physical CHannel</b>
<b>DPDCH</b>	<b>Dedicated Physical Data CHannel</b>
<b>DQRAP</b>	<b>Distributed Queueing Random Access Protocol</b>
<b>DS</b>	<b>Direct Sequence</b>
<b>DS-CDMA</b>	<b>Direct Sequence Code Division Multiple Access</b>
<b>DSCH</b>	<b>Downlink Shared CHannel</b>
<b>DSP</b>	<b>Doppler Spread Profile</b>
<b>DSP</b>	<b>Digital Signal Processor</b>
<b>DTCH</b>	<b>Dedicated Traffic CHannel</b>
<b>ECSD</b>	<b>Enhanced Circuit Switched Data</b>
<b>EDGE</b>	<b>Enhanced Data for GSM Evolution</b>
<b>EGPRS</b>	<b>Enhanced GPRS</b>
<b>EIR</b>	<b>Equipment Identity Register</b>
<b>ERMES</b>	<b>European Radio MESSage System</b>
<b>ETSI</b>	<b>European Telecommunication Standards Institute</b>
<b>FACH</b>	<b>Forward Access CHannel</b>
<b>FBI</b>	<b>Feedback Information</b>
<b>FCC</b>	<b>Federal Communications Commission</b>
<b>FDD</b>	<b>Frequency Division Duplex</b>
<b>FDMA</b>	<b>Frequency Division Multiple Access</b>
<b>FEC</b>	<b>Forward Error Correction</b>
<b>FER</b>	<b>Frame Error Rate</b>
<b>FH</b>	<b>Frequency Hopping</b>

---

<b>FMA</b>	<b>FRAMES Multiple Access</b>
<b>FOMA</b>	<b>Freedom Of Mobile multimedia Access</b>
<b>FPLMTS</b>	<b>Future Public Land Mobile Telecommunications System</b>
<b>FRAMES</b>	<b>Future Radio Wideband Multiple Access System</b>
<b>FTTH</b>	<b>Fiber To The Home</b>
<b>G3G</b>	<b>Harmonised Global 3G</b>
<b>GC</b>	<b>General Control</b>
<b>GERAN</b>	<b>GSM/EDGE Radio Access Network</b>
<b>GGSN</b>	<b>Gateway GPRS Support Node</b>
<b>GPRS</b>	<b>General Packet Radio Service</b>
<b>GSM</b>	<b>Global System for Mobile Communications</b>
<b>GTD</b>	<b>Geometrical Theory of Diffraction</b>
<b>HLR</b>	<b>Home Location Register</b>
<b>HMM</b>	<b>Hidden Markov Model</b>
<b>HSCSD</b>	<b>High Speed Circuit Switched Data</b>
<b>HS-DPCCH</b>	<b>High Speed - Dedicated Physical Control Channel</b>
<b>HS-DSCH</b>	<b>High Speed Downlink Shared Channel</b>
<b>HS-PDSCH</b>	<b>High Speed Physical Downlink Shared Channel</b>
<b>HS-SCCH</b>	<b>HS-DSCH related Shared Control Channel</b>
<b>HS-SICH</b>	<b>HS-DSCH related Shared Information Channel</b>
<b>ICG</b>	<b>International Coordination Group</b>
<b>iDEN</b>	<b>integrated Digital Enhanced Network</b>
<b>IETF</b>	<b>Internet Engineering Task Force</b>
<b>IMT-2000</b>	<b>International Mobile Telecommunications for the year 2000</b>
<b>INAP</b>	<b>Intelligent Network Application Protocol</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>IPHC</b>	<b>IP Header Compression</b>
<b>ISDN</b>	<b>Integrated Services Digital Network</b>
<b>ISMA</b>	<b>Inhibit Sense Multiple Access</b>
<b>ISO</b>	<b>Internacional Standards Organization</b>
<b>ITU</b>	<b>International Telecommunications Union</b>
<b>IWF</b>	<b>InterWorking Function</b>
<b>IWU</b>	<b>Inter Working Unit</b>
<b>LAN</b>	<b>Local Area Network</b>
<b>LCD</b>	<b>Long Constrained Delay</b>
<b>LDD</b>	<b>Low Delay Data</b>
<b>LOS</b>	<b>Line Of Sight</b>
<b>MAC</b>	<b>Medium Access Control</b>
<b>MBS</b>	<b>Mobile Broadband Systems</b>
<b>MC</b>	<b>MultiCarrier</b>

---

ME	<b>Mobile Equipment</b>
MIC	<b>Modulación por Impulsos Codificados</b>
MM	<b>Mobility Management</b>
MoU	<b>Memorando of Understanding</b>
MRC	<b>Maximal Ratio Combining</b>
MSC	<b>Mobile-services Switching Center</b>
MSC	<b>Message Sequence Chart</b>
NA	<b>North America</b>
NLOS	<b>Non Line Of Sight</b>
NMT	<b>Nordic Mobile Telephone</b>
Nt	<b>Notification</b>
ODMA	<b>Opportunity Driven Multiple Access</b>
OFDM	<b>Orthogonal Frequency Division Multiplexing</b>
OHG	<b>Operators Harmonization Group</b>
OSA	<b>Open Service Access</b>
OSI	<b>Open Systems Interconnection</b>
OVSF	<b>Orthogonal Variable Spreading Factor</b>
PACS	<b>Personal Access Communications Services</b>
PCCH	<b>Paging Control CHannel</b>
P-CCPCH	<b>Primary Common Control Physical CHannel</b>
PCH	<b>Paging CHannel</b>
PCPCH	<b>Physical Commom Packet CHannel</b>
PDA	<b>Personal Digital Assitant</b>
PDC	<b>Personal Digital Cellular</b>
PDCP	<b>Packet Data Convergence Protocol</b>
PDP	<b>Power Delay Profile</b>
PDSCH	<b>Physical Downlink Shared CHannel</b>
PDU	<b>Protocol Data Unit</b>
PER	<b>Packet Error Rate</b>
PHS	<b>Personal Handyphone System</b>
PICH	<b>Page Indicator CHannel</b>
PLMN	<b>Public Land Mobile Network</b>
PNBSCH	<b>Physical Node B Synchronisation CHannel</b>
PRACH	<b>Physical Random Access CHannel</b>
PS	<b>Packet Switched</b>
PSE	<b>Personal Service Environment</b>
PU	<b>Payload Unit</b>
PUSCH	<b>Physical Uplink Shared CHannel</b>
QoS	<b>Quality of Service</b>
QPSK	<b>Quaternary Phase-Shift Keying</b>

---

<b>RAB</b>	<b>Radio Access Bearer</b>
<b>RACE</b>	<b>Research of Advanced Communication Technologies in Europe</b>
<b>RACH</b>	<b>Random Access CHannel</b>
<b>RAINBOW</b>	<b>Radio Access Independent Broadband On Wireless</b>
<b>RB</b>	<b>Radio Bearer</b>
<b>RB</b>	<b>Radio Bearer</b>
<b>RLC</b>	<b>Radio Link Control</b>
<b>RNC</b>	<b>Radio Network Controller</b>
<b>RNS</b>	<b>Radio Network Subsystem</b>
<b>ROCCO</b>	<b>RObust Checksum based header COmpression</b>
<b>ROCH</b>	<b>Robust Header Compresión</b>
<b>RRC</b>	<b>Radio Resources Control</b>
<b>RRM</b>	<b>Radio Resources Management</b>
<b>RTE</b>	<b>Real Time Emulator</b>
<b>RTP</b>	<b>Real-Time Protocol</b>
<b>SAP</b>	<b>Service Access Point</b>
<b>SC</b>	<b>Site Controller</b>
<b>S-CCPCH</b>	<b>Secondary Common Control Physical CHannel</b>
<b>SCH</b>	<b>Synchronization CHannel</b>
<b>SDL</b>	<b>Specification and Description Language</b>
<b>SDT</b>	<b>SDL Design Tool</b>
<b>SDT</b>	<b>SDL Design Tool</b>
<b>SDU</b>	<b>Service Data Unit</b>
<b>SF</b>	<b>Spreading Factor</b>
<b>SFH</b>	<b>Slow Frequency Hopping</b>
<b>SGSN</b>	<b>Serving GPRS Support Node</b>
<b>SHCCH</b>	<b>Shared cHannel Control CHannel</b>
<b>SIR</b>	<b>Signal to Interference Ratio</b>
<b>SMG</b>	<b>Special Mobile Group</b>
<b>SMS</b>	<b>Short Message Service</b>
<b>SNL</b>	<b>Signalling Network Layer</b>
<b>SRNS</b>	<b>Serving Radio Network Subsystem</b>
<b>SS</b>	<b>Spread Spectrum</b>
<b>SSDT</b>	<b>Site Selection Diversity Transmit Power Control</b>
<b>STTD</b>	<b>Space Time Transmit Diversity</b>
<b>TACS</b>	<b>Total Access Communication System</b>
<b>TB</b>	<b>Transport Block</b>
<b>TB_Set</b>	<b>Transport Block Set</b>
<b>TB_Size</b>	<b>Transport Block Size</b>
<b>TB_SS</b>	<b>Transport Block Set Size</b>

---

<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>TDD</b>	<b>Time Division Duplex</b>
<b>TDMA</b>	<b>Time Division Multiple Access</b>
<b>TE</b>	<b>Terminal Equipment</b>
<b>TETRA</b>	<b>Trans-European Trunked Radio</b>
<b>TF</b>	<b>Transport Format</b>
<b>TFC</b>	<b>Transport Format Combination</b>
<b>TFC_Set</b>	<b>Transport Format Combination Set</b>
<b>TFCI</b>	<b>Transport Format Combination Indicator</b>
<b>TFI</b>	<b>Transport Format Indicator</b>
<b>TFS</b>	<b>Transport Format Set</b>
<b>TH</b>	<b>Time Hopping</b>
<b>TIA</b>	<b>Telecommunications Industry Association</b>
<b>TM</b>	<b>Transparent Mode</b>
<b>TMD</b>	<b>Transparent Mode Data</b>
<b>TPC</b>	<b>Transmit Power Control</b>
<b>TS-CDMA</b>	<b>Time Division Synchronous CDMA</b>
<b>TSTD</b>	<b>Time Switched Transmit Diversity</b>
<b>TTA</b>	<b>Telecommunications Technology Association</b>
<b>TTC</b>	<b>Telecommunication Technology Committee</b>
<b>TTCN</b>	<b>Tree and Tabular Combined Notation</b>
<b>TTI</b>	<b>Transmission Time Interval</b>
<b>UDD</b>	<b>Unconstrained Delay Data</b>
<b>UDP</b>	<b>User Datagram Protocol</b>
<b>UE</b>	<b>User Equipment</b>
<b>U-GGSN</b>	<b>UMTS Gateway GPRS Support Node</b>
<b>U-GMSC</b>	<b>UMTS Gateway MSC</b>
<b>UM</b>	<b>Unacknowledged Mode</b>
<b>UMD</b>	<b>Unacknowledged Mode Data</b>
<b>UML</b>	<b>Unified Modeling Language</b>
<b>U-MSC</b>	<b>UMTS Mobile-services Switching Centre</b>
<b>UMTS</b>	<b>Universal Mobile Telecommunications System</b>
<b>URAE</b>	<b>UMTS Radio Access Emulator</b>
<b>USCH</b>	<b>Uplink Shared Channel</b>
<b>U-SGSN</b>	<b>UMTS Serving GPRS Support Node</b>
<b>USIM</b>	<b>Universal Subscriber Identity Module</b>
<b>UTRA</b>	<b>UMTS Terrestrial Radio Access</b>
<b>UTRAN</b>	<b>UMTS Terrestrial Radio Access Network</b>
<b>UWC</b>	<b>Universal Wireless Communications</b>
<b>UWCC</b>	<b>Universal Wireless Communications Consortium</b>

VHE	<b>Virtual Home Environment</b>
VLR	<b>Visitor Location Register</b>
WAP	<b>Wireless Application Protocol</b>
WARC92	<b>World Administrative Radio Conference in the year 1992</b>
WCDMA	<b>Wideband Code Division Multiple Access</b>
WIMS	<b>Wireless Multimedia and Messaging Services Wideband CDMA</b>
WLAN	<b>Wireless Local Area Network</b>
WLL	<b>Wireless Local Loop</b>
WRC 2000	<b>World Radiocommunication Conference in the year 2000</b>