

EL LENGUAJE DECLARATIVO COMO HERRAMIENTA DE IMPLEMENTACION:

Un Caso de Estudio en Telecomunicaciones

J. Grompone, F. Brum, M. Casamayor, F. Simini

INTERFASE LTDA.

Zabala 1378, Montevideo, Uruguay

Teléfono: 95 31 66, 95 62 65

RESUMEN

El presente trabajo resume una experiencia de Diseño e Implementación de una Central de Conmutación Télex.

Se presenta y discute la alternativa de implementar un Lenguaje Declarativo como solución efectiva, tanto para la Operación del Sistema, como desde el punto de vista de la Ingeniería de la Programación.

Se comparan las diferentes soluciones a través del análisis de las prestaciones, de los tiempos de ejecución y de los tiempos de desarrollo, concluyendo que la construcción de una herramienta tan abstracta y conceptual como un Lenguaje, es también una solución eficiente para problemas concretos de implementación.

1. INTRODUCCION

En este trabajo se presenta uno de los resultados de un proyecto de diseño e implementación de un Conmutador Télex: una manera general de resolver el problema de personalizar la conmutación. Las ideas desarrolladas no están limitadas al tema Télex sino que pueden generalizarse a diversos problemas de telecomunicaciones y de otras áreas.

La personalización de un Conmutador de Datos exige definir todos los elementos potencialmente modificables del equipo:

- cantidad y propiedades de las líneas
- manera como se agrupan las líneas
- enrutamiento de llamadas
- servicios y prestaciones auxiliares

El enrutamiento es el procedimiento por el cual se asocia a un código de selección de abonado una línea para realizar una comunicación.

El problema del enrutamiento fue analizado desde tres puntos de vista:

a) Definir e implementar los algoritmos y las estructuras de datos capaces de enrutar una solicitud de conexión en tiempos aceptables para el correcto funcionamiento

del Conmutador.

b) Lograr que la programación del Conmutador fuera independiente de su personalización.

c) Dotar al personal de operación del equipo de herramientas flexibles para introducir modificaciones y personalizarlo de acuerdo a necesidades cambiantes.

La solución adoptada consistió en desarrollar un Lenguaje Descriptivo que permite definir con precisión las características del Conmutador. Dicho lenguaje fue llamado Lenguaje de Especificación de Personalidades (de ahora en más LEP). Paralelamente al LEP, se desarrollaron Autómatas que interpretan las tablas generadas por el compilador LEP, realiza el enrutamiento y algunas otras funciones y un Simulador utilizado como herramienta previa a la instalación de una Personalidad.

En este trabajo se presentan algunos de los resultados extraídos de la etapa de diseño, implementación y explotación del LEP. El artículo se concentra en el tema de diseño de un Lenguaje como solución de Ingeniería para problemas con las características del que enfrentamos. Este último punto llevó a confrontar el enfoque adoptado con el enfoque convencional (construcción de un conjunto de programas que implementen en concreto cada Plan de Enrutamiento sin proporcionar una visión abstracta y global del mismo).

En este trabajo, además de presentar ambas soluciones y la discusión correspondiente, se incluyen los resultados medidos a lo largo del Proyecto. La utilidad del LEP fue considerada desde dos ópticas:

a) Para las tareas de operación (una vez instalado el Conmutador) es una herramienta poderosa que simplifica ampliamente su gestión y todos los aspectos vinculados al mantenimiento.

b) Desde el punto de vista de la duración del proyecto en su conjunto, los resultados obtenidos muestran que el tiempo invertido en el desarrollo del LEP fue compensado ampliamente por el ahorro obtenido al no tener que implementar varias versiones concretas de Planes de Enrutamiento.

2. EL PROBLEMA A RESOLVER

Se llama Personalidad de un Conmutador al conjunto de tablas y de programas que implementan las características operativas de un equipo dado. En la Personalidad se definen

las modalidades de las líneas (abonado o troncal, etc), las reglas de enrutamiento (numeración, rutas, etc), las reglas de accesibilidad entre líneas y servicios y nombres simbólicos accesibles desde la Consola (que permiten ajustes y monitoreo en tiempo de operación).

Para el Conmutador una llamada es una solicitud de conexión. La función básica de la Personalidad es asociar a un destino a una llamada. La noción de destino encierra diferentes contenidos:

- la solicitud no es posible
- es una línea (o un haz) de salida para una comunicación télex
- es un mensaje aclaratorio
- es un servicio interno del Conmutador

También es necesario que el Conmutador suministre información a los efectos de la tarificación, por lo que la Personalidad debe poder identificar al abonado llamante (aquel que corresponde a la línea por la que llega la llamada).

Del análisis de las funciones de la Personalidad surge la identificación de cuatro tareas independientes:

- Enrutamiento: asociar un destino a una llamada.
- Verificación de accesibilidad: determinar si la línea llamante tiene derecho a conectarse con el destino solicitado.
- Identificación del abonado correspondiente a la línea llamante.
- Conversión de códigos: en el caso de que el destino sea un nuevo Conmutador (por ejemplo una llamada internacional).

De acuerdo a esto, el paquete de la Personalidad consta de cuatro procedimientos bien definidos. A los efectos de este trabajo interesa considerar las diferentes alternativas para su implementación y discutir las ventajas y desventajas relativas a cada una de ellas.

3. VARIANTES DE IMPLEMENTACION

Segun la experiencia de años de diseño de conmutadores es posible distinguir diferentes niveles de implementación de una Personalidad [1], [2]:

En el primer nivel se diseña una programación dedicada,

escrita en Assembler, y alojada en ROM. En este caso el mantenimiento de la personalidad no es simple.

En el segundo nivel de implementación, la programación está escrita en "C" pero reside por igual en ROM. Se mejora la posibilidad de mantenimiento, pero al igual que en nivel anterior, el personal de operación del equipo no puede realizar modificaciones que no impliquen la reprogramación de las EPROMs.

En el tercer nivel de implementación, reside en ROM un conjunto de programas interpretadores de tablas y la Personalidad consiste en un conjunto de tablas que constituyen la entrada de los algoritmos residentes. El mantenimiento queda en manos del personal de operación del equipo, y se resuelve el problema de la independencia. Sin embargo, la tarea de modificación de las tablas es tan compleja que resulta inviable.

En el cuarto nivel, las tablas de entrada de los algoritmos residentes son generadas en forma automática por un Compilador a partir de un programa escrito en el lenguaje declarativo LEP. Esta opción deja en manos del personal de operación del equipo la responsabilidad del mantenimiento proporcionándole herramientas que hacen posible su tarea. La descripción de la Personalidad en LEP constituye además un resumen global de un alto nivel de abstracción que facilita el trabajo tanto del personal de operación como del equipo de diseño.

Estos cuatro niveles se resumen en dos opciones:

- Considerar a cada Conmutador concreto como un caso único, escribiendo programas específicos que implementan su Personalidad.

- Construir un Conmutador genérico, donde la Personalidad sea suministrada desde la Consola mediante tablas. Esta opción implica, debido a la complejidad de las tablas, la construcción de un Lenguaje Declarativo. El LEP es imprescindible en esta opción ya que el mantenimiento manual de las tablas es inviable.

El proyecto consistió en la implementación de un modelo de Conmutador de 512 líneas y la instalación de cuatro ejemplares con diferentes personalidades. Se eligió la solución LEP con los siguientes objetivos:

- Independencia entre la programación y la entonación. Los ajustes de la Personalidad son realizados por el personal de operación del equipo, resultado importante si tenemos en cuenta que por ejemplo el Conmutador Internacional

de nuestro País se encuentra en este momento en la versión 36 de su Personalidad.

- Obtención de una herramienta descriptiva general utilizable en otros modelos de Conmutadores. Un programa LEP describe completamente un Conmutador, proporcionando un lenguaje preciso compartido por los usuarios y el equipo de diseño.

El diseño y la implementación del compilador LEP fue realizado en paralelo con la implementación de personalidades dedicadas para los primeros ejemplares del Conmutador que entraron en funcionamiento. Esto es que pese a optar por la solución LEP, por razones de plazo de entrega, también fue necesario utilizar la solución de programación dedicada. El desarrollo de esta experiencia en paralelo permitió confrontar ambas aproximaciones no solamente desde la óptica de las prestaciones, sino también comparando los tiempos invertidos en cada alternativa.

4. EL LENGUAJE DECLARATIVO LEP

El lenguaje LEP se diseñó con dos objetivos a la vista:

- tratar que fuera simple y natural en su expresión
- adherir a las recomendaciones del Comité Consultivo Internacional para la Telefonía y Telegrafía (CCITT) para el lenguaje hombre-máquina [3].

A los efectos de ilustrar el lenguaje declarativo definido, se presenta, a modo de ejemplo, un programa simple:

```
plan DEMO D'16 /* Programa de Demostración */

/* Declaración de reglas de enrutamiento */
rule
    TELEX;
end rule

/* Declaración de servicios de consola */
features
    CONSOLA-AUX      { BAU:L=@ };
    TEXTO-ABC        { TEP:ABC:L=@ };
end features

/* Declaración de mensajes */
messages
    NCH      "<^NCH<^";
end messages
```

```

/* Declaración de formatos */
recode
    TAL-CUAL          *      *;
    SACA-1-CIFRA      ??     *;
end recode

/* Declaración de haces de líneas */
group

    ABONADOS
        ( 0'0..0'11 )
    TELEX multiple trunk Akey WRU ;

    TRONCAL
        ( 0'12..0'17 )
    TELEX single terminal Akey log bill WRU ;

end group

/* Declaración de permisos */
forbidden
    ABONADOS  CONSOLA-AUX;
    TRONCAL   TRONCAL, CONSOLA-AUX;
end forbidden

/* Numeración externa de las líneas */
identification
    ABONADOS + 10[0..9];
    TRONCAL    000;
end identification

/* Plan de enrutamiento */
routing  TELEX
    000          CONSOLA-AUX;
    0[2..9]*     TRONCAL SACA-1-CIFRA;
    10[0..9+]    ABONADOS;
    115          TEXTO-ABC;
    20?          NCH;
end routing

end plan

```

El elemento básico de la Personalidad es el Plan de Enrutamiento. Este Plan se define como una secuencia de declaraciones de la forma:

código de selección	destino
---------------------	---------

Los códigos de selección se describen o bien completamente (P. Ej. 000) o bien utilizando abreviaciones (P. Ej. 0[2..9]* que representa los códigos 02, 03, hasta 09 seguidos de otros dígitos).

El objetivo más frecuente de una llamada es establecer una comunicación con otro abonado a través de una línea. El LEP permite describir completamente una línea indicando sus atributos. Para ello las líneas se agrupan en haces. Un haz es un conjunto ordenado de líneas designado por un identificador. Los haces son particiones disjuntas, ordenadas y completas del conjunto de las líneas.

En la sección group se asignan nombres a los haces.

En la sección recode se asignan nombres a funciones de recodificación de códigos utilizadas cuando el destino es un nuevo Conmutador. La cláusula:

0[2..9]* TRONCAL SACA-1-CIFRA

indica que si por ejemplo se disca 05247, que tiene al haz TRONCAL como destino, el código que se retransmite es 5247.

El LEP permite definir otros destinos mediante las siguientes secciones:

- features: se asignan nombres a los servicios provistos por la Consola (P. Ej. TEXTO-ABC se usa para designar el servicio de texto de prueba accesible a los abonados). De acuerdo con los ejemplos, la cláusula:

115 TEXTO-ABC

indica que al discar el 115 el abonado obtendrá el texto de prueba.

- messages: se asignan nombres a los mensajes provistos por el Conmutador. La sentencia:

20? NCH

indica que al discar cualquiera de los códigos de tres dígitos que comienzan con 20 se obtendrá el mensaje:

<CR> <LF> NCH <CR> <LF>

Como puede observarse, un programa LEP está formado por declaraciones que describen o bien atributos de los elementos primitivos de nuestro problema de enrutamiento (líneas, mensajes, etc.) o bien relaciones entre los elementos primitivos que permiten agregarlos en estructuras más complejas.

El compilador traduce la descripción de alto nivel en una serie de tablas que a su vez son utilizadas por los algoritmos

residentes que conforman el núcleo de la Personalidad.

La operativa es la siguiente: el personal de operación del equipo escribe su programa LEP (o modifica uno existente), lo compila, verifica sus prestaciones utilizando un Simulador y posteriormente lo instala mediante una orden de Consola. Durante la instalación de una nueva Personalidad en un Conmutador se interrumpe el enrutamiento (no se realizan nuevas conexiones) pero las comunicaciones ya establecidas se continúan sin cambio.

Los puntos analizados en este ejemplo constituyen una aproximación ilustrativa a las características del LEP. La descripción detallada del lenguaje aparecerá en publicaciones del área de Telecomunicaciones.

5. IMPLEMENTACION DEL COMPILADOR

El Compilador LEP se implementó en forma clásica [4], mediante cuatro módulos independientes:

- análisis Lexicográfico
- análisis Sintáctico
- generador de Tablas
- verificación de Consistencia Semántica

Los diferentes módulos fueron escritos en "C", a partir de la definición del lenguaje. Inicialmente se escribió un prototipo del analizador sintáctico en Prolog.

6. COMPARACION DE LOS TIEMPOS DE EJECUCION

Para el modelo de Conmutador de 512 líneas existen dos modalidades de implementación de personalidades:

- las implementaciones dedicadas
- las personalidades compiladas utilizando el LEP

Es importante comparar ambas modalidades desde el punto de vista del tiempo de ejecución ya que éste es un aspecto crítico para el funcionamiento de un Conmutador.

Se compararon 15 casos de enrutamiento, comprendiendo tanto casos típicos como casos límite. Los tiempos fueron medidos utilizando un Simulador. Como resultado de esta comparación se tienen siete casos en que la versión LEP demora más que la programación en C dedicada, cuatro casos en que demoran lo mismo, y cuatro casos en que es mejor.

Tomando los tiempos globales se tiene una pequeña

desventaja en tiempos con el LEP. Se debe destacar, sin embargo, que los cuatro interpretadores de tablas poseen todas las protecciones necesarias para asegurar que terminan su ejecución en un tiempo adecuado en todos los casos. Estos resultados permiten afirmar que no hay diferencias apreciables entre el funcionamiento de Conmutadores de uno y otro tipo.

Desde el punto de vista de la operación es claro que el modelo LEP supera ampliamente al modelo de programación dedicada.

7. COMPARACION DE LOS TIEMPOS DE DESARROLLO

Se verá que -contra lo que puede pensarse a priori- el esfuerzo necesario para implementar la solución LEP no es superior al de implementar los Conmutadores con programación dedicada, aún cuando el número de ejemplares a ser instalado sea reducido (en nuestro caso de cuatro). Además, la solución LEP proporcionó, como resultado indirecto, una herramienta de especificación e implementación muy valiosa para proyectos futuros.

Los cálculos de tiempos se realizan sobre la base de la siguiente igualdad:

1 mes-hombre = 170 horas de trabajo realizado

El replanteo de las horas-hombre empleadas en el proyecto en su conjunto conduce al siguiente cuadro de resultados:

ETAPA	HORAS	MESES	%TOTAL
diseño del LEP	410	2.4	5%
preparación programadores	560	3.3	7%
especificación	720	4.2	9%
implementación	4764	28.0	62%
puesta en marcha	1231	7.2	16%
TOTAL	7685	45.2	

El resumen global, dejando de lado el tiempo de formación de programadores, es el siguiente:

ETAPA	MESES-HOMBRE	% TOTAL
Diseño	6.6	15%
Implementación	28.0	67%
Puesta en marcha	7.2	17%
TOTAL	41.8	

El rendimiento de un mes-hombre a lo largo del Proyecto se ajustó notablemente a las estimaciones teóricas (según las cifras del método de estimación COCOMO se obtienen 400 líneas/mes-hombre para este tipo de proyectos) y se presenta en el siguiente cuadro:

LENGUAJE	PAGINAS	LINEAS	TIPO
Assembler (*)	107	5.300	Final
C	150	7.500	Final
Assembler	10	500	Auxiliar
C	54	2.700	Auxiliar
ADA	14	700	Auxiliar
Total Final		12.800	
Total Auxiliar		3.900	23% del total
TOTAL		16.700 líneas	

(*) Se tomó 50% de la programación total (10.600 líneas) como programación nueva ya que se reutilizó gran parte de la programación Assembler correspondiente a modelos anteriores de Conmutadores.

$\text{Líneas por mes-hombre} = 16700 / 41.8 = 399.5$

A los efectos de este trabajo, es necesario comparar el tiempo invertido en programación auxiliar de Personalidades con el tiempo de desarrollo del LEP.

El replanteo específico del LEP y de las Personalidades Dedicadas conduce a los siguientes resultados:

Medidas Directas:

	HORAS	MESES-HOMBRE
Diseño	410	2.4
Lexicográfico	90	0.5
Sintáctico	186	1.1
Semántico	153	0.9
Automatas	334	2.0
Manuales y varios	80	0.5
TOTAL	1.253	7.4
Personalidades	690	4.0

Medidas Indirectas:

	PAGINAS	LINEAS	MESES-HOMBRE (estimados a 400 líneas/mes)
LEP	71	3.550	8.9
Autómatas	7	350	0.9
TOTAL		3.900	9.8
Personalidades	44	2.200	5.5

Aún sin entrar en otras consideraciones se puede decir que el tiempo de desarrollo de algunas personalidades dedicadas y el tiempo de desarrollo del LEP son comparables: el primero es el 60% del segundo.

Sin embargo, estos datos son relativos. Si se tiene en cuenta que sólo se programaron personalidades dedicadas para tres Conmutadores de los cuatro instalados (las personalidades del cuarto fueron todas compiladas), surge que hubiera sido necesario un 80% del tiempo de desarrollo del LEP para obtener cuatro personalidades dedicadas.

Por otra parte sólo se programaron personalidades dedicadas en el período julio-diciembre, ya que en diciembre entró en funcionamiento la primer personalidad compilada y a continuación fueron sustituidos los ejemplares dedicados por personalidades obtenidas mediante el LEP.

8. CONCLUSIONES

La experiencia acumulada en proyectos anteriores muestra que el mantenimiento de personalidades dedicadas implica tareas de reprogramación, compilación, linkedición y grabado de EPROMs durante los años de funcionamiento de un Conmutador. En el caso de los Conmutadores de 512 líneas actualmente en funcionamiento, el mantenimiento y ajuste de Planes de Enrutamiento ha pasado a ser tarea exclusiva del personal de operación.

Esto hace que al analizar los tiempos no con el criterio de trabajo inicial, sino con el criterio de desarrollo y seguimiento de una implementación a lo largo de toda su vida útil, se pueda asegurar que la solución LEP implica un enorme ahorro de recursos humanos y técnicos que se suma a las ventajas de Operación mencionadas anteriormente.

La aproximación LEP representa realizar un esfuerzo de

abstracción, cuyo resultado es una definición precisa y de por sí orientadora en cuanto a la solución del problema. Este es el resultado esencial de la experiencia, mas allá de haber obtenido también una herramienta flexible y de gran utilidad en la Operación.

Sin pretender generalizar en exceso, los resultados analizados en el presente trabajo permiten considerar la aproximación LEP al menos como "no descartable a priori", en tanto solución de Ingeniería para problemas de un alto nivel de complejidad. Los resultados muestran, a través de un caso concreto, que la inversión de recursos en conceptualizar y abstraer los elementos principales de un problema es también una importante herramienta en el área de Ingeniería de Software.

8. BIBLIOGRAFIA

[1] "LA VIABILIDAD DE PROYECTOS NACIONALES: UNA CENTRAL TELEX CONSTRUIDA EN URUGUAY", J. Grompone, Boletín de Telecomunicaciones de la UIT, Vol. 49, No. V, páginas 280-284, Ginebra 1982.

[2] "COMMUNICATIONS TECHNOLOGY DEVELOPMENT: A CASE STUDY IN URUGUAY", J. Grompone, F. Simini, Communications Indonesia 1985 (The International Telecommunications and Business Communications Systems Exhibition), Jakarta, Indonesia 1985.

[3] "LENGUAJE HOMBRE-MAQUINA", Recomendación Z.311 a Z.341, Libro Amarillo, CCITT, Ginebra 1981.

[4] "THE THEORY OF PARSING, TRANSLATION AND COMPILING", Aho, Ullman, Prentice Hall, 1972.