University of Castilla-La Mancha

A publication of the

Department of Computer Science

PROTOTIPO DE UNA HERRAMIENTA DE EJECUCION DE ESPECIFICACIONES EN TPAL

by

Juan José Pardo, Valentín Valero

Technical Report #DIAB-99-01-05 Sept. 1999

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete - 02071 - Spain
Phone +34.967.599200, Fax +34.967.599224

Proyecto CICYT TIC-0669-c03-02
DESARROLLO FORMAL DE SISTEMAS DISTRIBUIDOS
SUBPROYECTO 2
PROTOTIPO DE UNA HERRAMIENTA DE EJECUCIÓN DE ESPECIFICACIONES EN TPAL
(Informe Tecnico No. 1 del Subproyecto, Int. report DIAB-99-01-05)

1 Objetivos del proyecto

El objetivo principal de este proyecto es el de contribuir a facilitar la utilización en la práctica de lenguajes formales para la concurrencia. Para ello partiremos de un lenguaje base como es E-LOTOS, el cual será extendido con nuevas características que harán posibles nuevas variantes del lenguaje. Esta extensión del lenguaje consistirá principalmente en la introducción de tiempos y probabilidades dentro del lenguaje, lo que nos permitirá especificar comportamientos de sistemas temporales y probabilísticos.

Como consecuencia directa de este objetivo principal se plantea uno final como es realizar un prototipo de herramienta que permita la ejecución simbólica de las especificaciones, así como realizar el análisis de éstas mediante la traducción a un modelo temporizado y probabilístico de redes de Petri.

Para conseguir este nuestro principal objetivo nos planteamos algunos subobjetivos que pasamos a describir a continuación clasificados de la forma siguiente:

1.1 Semánticas y propiedades de E-LOTOS y sus extensiones

Una vez exista una definición del lenguaje fijada con una semántica "oficial", que será operacional, nuestros objetivos serán:

1. Definición de una semántica de pruebas, tanto en su forma convencional como en su versión amigable introducida en [FLN97], que defina de forma razonable el comportamiento observable deducible a partir de la semántica operacional.

2. Una semántica denotacional equivalente a la de pruebas que ponga de manifiesto la estructura del nuevo lenguaje. En el intentaremos introducir un aspecto novedoso como es la integración armoniosa de los aspectos de procesos y datos.

3. Una semántica axiomática que permitir razonar algebraicamente sobre las propiedades de los procesos y derivar transformaciones correctas que puedan utilizarse para derivar implementaciones a partir de las correspondientes especificaciones.

4. Estudio de extensiones probabilísticas del lenguaje, lo cual nos llevará a desarrollar modelos temporizados probabilísticos .

5. Definición de una semántica de tiempo real mediante la traducción a R edes de Petri.

1.2 Verificación de propiedades de procesos

Una vez definidas las distintas semánticas tendremos definidos el marco teórico sobre el que comprobar propiedades para lo que habrá que realizar el desarrollo e implementación de algoritmos. En concreto desarrollaremos los siguientes:

1. Algoritmos de exploración de estados
2. Algoritmos de transformación de especificaciones
3. Algoritmos eficientes para la decisión de la equivalencia entre procesos.
4. Algoritmos de comprobación de modelos.
5. Definición de una maquina virtual que ejecute especificaciones E-LOTOS.

1.3 Prototipado de un entorno de diseño

Una vez alcanzados los objetivos anteriores tendremos una base sobre la que fundamentar los prototipos de herramientas de soporte al diseño que faciliten el uso de E-LOTOS. Las herramientas desarrolladas permitirán las siguientes tareas:

1. Análisis sintáctico y de la semántica estática del lenguaje.
2. Ejecución simbólica de especificaciones con facilidades de depuracin.
3. Exploración de estados y verificación de propiedades.
4. Paso de test.
5. Traducción a Redes de Petri y Análisis de las redes resultantes de la traducción.

1.4 Tutoriales, Casos de Estudio y Tranferencia Industrial.

Una de las principales tareas para conseguir la facilidad de uso del lenguaje E-LOTOS será la creación de unos buenos tutoriales y casos de ejemplo. Para ello se tomarán casos de la vida real realizando un diseño completo.

2 Orientación

Con el lenguaje que se está desarrollando se prentende realizar más fácil y permitir la modelización de sistemas con restricciones temporales como pueden ser sistemas de tiempo real así como sistemas que por su necesidad de seguridad y sus posibilidades de fallo necesitan una caracterización probabilística.

3 Desglose de tareas

3.1 Tareas del proyecto global

Para llevar a cabo el proyecto y conseguir los objetivos planteados existen tres equipos de trabajo entre los que se distribuyen las diferentes tareas. De este modo la estructuración en tareas que se realiza del proyecto es la siguiente:

Tarea A: Semánticas de más alto nivel.

Subtarea A.1: Semánticas de pruebas.

En primer lugar se realizará un estudio de las semánticas de pruebas tradicionales haciendo hincapié en la semántica *must* para posteriormente centrarse el estudio en la semántica de pruebas amigable prestando en este caso especial atención al orden inducido.

Subtarea A.2: Semánticas denotacionales.

Mediante esta semántica se pretende la caracterización de la semántica de pruebas definiendola de forma composicional para lo que se prestar una especial atención a la técnica de punto fijo. Igualmente se aplicar esta técnica a la semántica amigable.

Subtarea A.3: Semánticas axiomáticas y formas normales.

Se realizará una caracterización algebraica de las semánticas anteriores que nos conducir a una forma normal computable y que nos permitir caracterizar la equivalencia de procesos.

Tarea B: Extensiones probabilísticas y semánticas alternativas.

Subtarea B.1: Extensiones probabilísticas.

Se desarrollará un modelo temporizado probabilístico en el que se conjugarán la simplicidad de la semántica con la capacidad expresiva. Este modelo irá acompañado de una bateria de ejemplos que muestren su capacidad expresiva.

Subtarea B.2: Paralelismo real.

Se realizar el estudio del paralelismo y la causalidad con paralelismo real. Para ello se utilizarán pasos y redes de Petri respectivamente.

Subtarea B.3: Semántica de los tipos de datos.

Se realizará un estudio comparado de los tipos de datos de LOTOS y E-LOTOS indicando la forma de pasar del primer lenguaje al segundo.Este estudio irá acompañado de ejemplos de como utilizar la nueva componente de datos.

Tarea C: Diseño de algoritmos para la evaluación de sistemas.

Subtarea C.1: Exploración de estados y bisimulación.

Se adecuarán los algoritmos de exploración de estados al caso de E-LOTOS haciendo hincapié en el papel de los datos, buscando los correspondientes algoritmos de expansión parametrizada. Estos algoritmos se adecuarán a la semántica de pruebas .

Subtarea C.2: Algoritmos de decisión de propiedades.

Se buscarán o adecuarán a E-LOTOS algoritmos eficientes para el análisis de propiedades clásicas de los sistemas concurrentes, los cuales se aplicarán directamente o mediante la traducción a Redes de Petri.

Subtarea C.3: Evaluación de rendimientos.

A modo de iniciación en el tema se estudiar el tema a partir de la extensión probabilística, como a través de la traducción a redes de Petri utilizando los estudios ya desarrollados sobre el tema.

Tarea D: Herramientas de apoyo al diseño y análisis de sistemas.

Subtarea D.1: Analizador sintáctico y semántico.

Se adecuarán los prototipos disponibles para las distintas versiones del lenguaje que se han ido considerando según se ha desarrollado la nueva norma E-LOTOS a la concreción definitiva de la misma una vez consolidada.

Subtarea D.2: Algoritmos de comprobación de propiedades (model checking)

Se realizará la aplicación de los algoritmos ya conocidos aunque para ello haya que realizar un desarrollo previo que adecue el lenguaje a éstos. Se realizará un estudio directo sobre procesos E-LOTOS y un estudio a partir de la traducción a redes de Petri.

Subtarea D.3: Algoritmos de análisis de Redes de Petri probabilísticas.

Se implementarán los algoritmos de análisis de propiedades de redes de Petri Probabilísticas. Posteriormente se realizará un simulador e interprete de este tipo de redes y sobre el se implementarán los algoritmos de evaluación de rendimientos.

Tarea E: Tutoriales, casos de estudio y transferencia industrial.

Subtarea E.1: Desarrollo de manuales comentados y tutoriales.

Se presentarán y comentarán las distintas características del lenguaje, indicando especialmente las novedades con respecto a LOTOS, incluyendo ejemplos ilustrativos de uso y propiedades.

Subtarea E.2: Casos de estudio.

Se realizarán distintos ejemplos por tres grupos independientes para posteriormente aglutinarlos todos de modo que se encuentren aquellos que ilustren simultáneamente distintas características interesantes del lenguaje.

Subtarea E.3: Transferencia de resultados.

Por un lado se seguirá participando en el Comité que desarrolla la norma ISO. Por otro lado se realizar un caso de estudio de magnitud industrial que refleje las ventajas y capacidades de la utilización de E-LOTOS. Para finalizar se contactará con las empresas que han expresado su interés para presentarles los resultados.

3.2 Tareas asociadas al subproyecto 2

Hasta ahora se han presentado las tareas y objetivos del proyecto global. Como ya se comentaba anteriormente existen tres grupos de trabajo a cada uno de los cuales se les ha asignado un subproyecto. A continuación pasamos a describir las tareas asociadas al subproyecto 2, que están siendo llevadas a cabo por el grupo de la Universidad de Castilla-La Mancha en Albacete.

Tarea B: Extensiones probabilísticas y semánticas alternativas.

La contribución de este segundo subproyecto a esta tarea consiste en la definición de un lenguaje temporizado y probabilístico as como de la semántica asociada al mismo. Estas definiciones irán acompaadas de una batería de ejemplos que muestren la capacidad expresiva del lenguaje.

Tarea C: Diseño de algoritmos para la evaluación de sistemas.

Dentro de esta tarea se realizará la definición de algoritmos que permitan la decisión de propiedades de las especificaciones mediante la traducción a redes de Petri. así como el estudio y evaluación de rendimientos.

Tarea D: Herramientas de apoyo al diseño y análisis de sistemas.

El desarrollo de esta tarea será casí exclusivo de este subproyecto por lo que se realizará la creación de la herramienta para lo cual será necesario realizar la implementación de los siguientes elementos:

1. Analizador sintactico y semántico
2. Procedimiento de creación del diagrama de transición de estados
3. Procedimientos de ejecución simbólica de las especificaciones
4. Algoritmos de comprobación de propiedades a partir de la traducción a redes
5. Algoritmos de redes de Petri probabilísticas

Tarea E: Tutoriales, casos de estudio y transferencia industrial.

La contribución realizada será la definición de casos de estudio que ilustren la utilización de la herramienta para posteriormente fusionarlos con otro y conseguir así un ejemplo que muestre conjuntamente la mayoría de las características del lenguaje. Además se realizará la transferencia de los resultados de la investigación a las empresas que mostraron su interes por el proyecto.

Tarea B: Extensiones probabilisticas y semanticas alternativas

ACTIVIDADES	GRUPO	PRIMER AÑO	SEGUNDO AÑO	TERCER AÑO
Subtarea B.1. Extensiones Probabilisticas				
B.1.1. Modelos temporizados y probabilisticos	DIAT-ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
B.1.2 E-LOTOS Probabilistico	ALB-DIAT	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
B.1.3. Metodologia de uso	ALB-DIAT	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
Subtarea B.2. Paralelismo Real				
B.2.2. Traduccion a redes de Petri	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
B.2.4.- Semantica de Redes de petri probabilisticas	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D

Tarea C: Diseño de algoritmos para la evaluacion de sistemas

ACTIVIDADES	GRUPO	PRIMER AÑO	SEGUNDO AÑO	TERCER AÑO
Subtarea C.2.- Algoritmos de decision de propiedades				
C.2.2. Sobre redes de Petri	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
Subtarea C.3. Evaluacion de rendimientos				
C.3.2. Sobre redes de Petri Probabilisticas	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D

Tarea D: Herramientas de apoyo al diseño y analisis de sistemas

ACTIVIDADES	GRUPO	PRIMER AÑO	SEGUNDO AÑO	TERCER AÑO
Subtarea D.2. Comprobacion de propiedades				
D.2.2. Sobre redes de Petri	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
Subtarea D.3. Analisis de redes probabilisticas				
D.3.1. Algoritmos de analisis	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
D.3.2. Simulador e interprete	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
D.3.3. Evaluacion de rendimientos	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D

Tarea E: Tutoriales, casos de estudio y transferencia industrial

ACTIVIDADES	GRUPO	PRIMER AÑO	SEGUNDO AÑO	TERCER AÑO
Subtarea E.2. Casos de estudio				
E.2.2. Paralelismo real	ALB	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
E.2.4. Casos compuestos ilustrativos	DIAT-ALB-VIGO	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D
Subtarea E.3. Transferencia de resultados				
E.3.3. Relacion con empresas interesadas	DIAT-ALB-VIGO	E F M A M J J A S O N D	E F M A M J J A S O N D	E F M A M J J A S O N D

4 Informe del primer año

4.1 Modelos Temporizados

La estrecha relación con el tiempo de los sistemas especificados ha llevado a la extensión de los lenguajes básicos(CCS, CSP, LOTOS) para que incorporasen las características de tiempo y probabilidad.

En los años 80 los trabajos se centraron principalmente en la extensión temporal, para después en los 90 centrarse en las extensiones probabilísticas. En estos momentos los trabajos se centran fundamentalmente en las extensiones que combinen tanto aspectos temporales como probabilísticos.

La realización de estas extensiones ha dado lugar a la creación de una gran diversidad de modelos de los cuales se da una visión general en el trabajo [Par97].

4.2 Sintaxis del Lenguaje

El estudio desarrollado para la realización del trabajo [Par97] junto con otro realizado sobre redes de Petri temporizadas y probabilísticas ha servido como base para definir la sintaxis de nuestro lenguaje la cual será la que presentamos a continuación.

Consideremos $Act = vis \cup \{\tau\}$,$A \subseteq Act$ $a \in Act$ $e \in Act$ $t \in \mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$

Con esta notación la sintaxis del lenguaje viene dada por la siguiente expresión BNF:

$$T \ ::= \ P \mid N$$

$$N \ ::= \ stop \mid a < t_1, t_2 >; T \mid \tau; T \mid wait(t); T \mid N \Box N \mid N \|_A N \mid N \backslash_a \mid X \mid \mu X.N$$

$$P \ ::= \ \sum_{i \in I} [q_i].N \mid P \|_A N \mid N \|_A P \mid P \|_A P \mid N \backslash_a \mid X \mid \mu X.P$$

Como puede comprobarse esta definición impone una alternancia entre ambos operadores de elección.

Como puede verse en las expresiones anteriores se mantienen los operadores de un álgebra de procesos básica y además se han incluido algunos operadores nuevos, y modificados otros de modo que nos permitan describir comportamientos probabilísticos y características temporales de los sistemas.

Describiremos a continuación estos nuevos operadores.

$a < t_1, t_2 >; T$ .Se establece un intervalo de tiempo durante el cual puede ser ejecutada la acción "a" después del cual la acción ya no podrá ser ejecutada.

$\sum_{i \in I} [q_i].N$.-Representa una elección probabilística donde el sistema decide que es lo que se ejecutará atendiendo a las probabilidades asociadas a cada proceso.

$\tau; T$.- Representa una acción interna que además es urgente por lo que se ejecutará nada más que esté disponible.

$Wait(t)$.- Este operador que representa el paso de t unidades de tiempo fue introducido con la idea de poder expresar retrasos en la ejecución de procesos.

4.3 Prototipo de una Herramienta para la Ejecución de Especificaciones

Como ya se ha comentado anteriormente al especificar las tareas de este subproyecto 2 el objetivo ultimo es el de construir una herramienta para la ejecucion de especificaciones.

Esta herramienta esta pensada que funciones sobre una maquina Sun trabajando sobre ella en entorno gráfico X-windows. Para que esto sea posible en su implemtación serán utilizadas las librerias X-View proporcionadas por el lenguaje de programación C para realizar este tipo de aplicaciones.

A continuación en los siguientes apartados realizaremos una descripción más exhaustiva de los distintos componentes de la herramienta describiendo cada un de ellos.

4.4 Analizador

Es el primer y uno de los más importantes componentes de la herramienta y su principal objetivo es el de comprobar que las especificaciones que sean introducidas se ajustan a la sintaxis y semántica del lenguaje. Como resultado se obtendrá un árbol sintáctico de la expresión introducida, el cual servirá como entrada para las posteriores componentes. Este analizador está compuesto de tres partes interrelacionadas y complementarias las cuales pasamos a describir a continuación.

4.4.1 Scanner o Analizador Léxico

Su principal objetivo es el de ir leyendo el fichero de entrada obteniendo los distintos símbolos que tengan pleno significado dentro del lenguaje.

Para ello la lectura la realizará carácter a carácter construyendo posteriormente los distintos tokens que describen cada uno de los elementos.

Este componente debe ser el encargado del acceso a ficheros realizando la apertura y cierre de estos.

Las funciones proporcionadas por este componente son las siguientes:

```
int scanner(struct token *t)
   Obtiene el siguiente token desde el fichero
int abre_fichero(char * nombre)
   Prepara el fichero fuente para ser leido.
int cierra_fichero()
   Cierra el fichero fuente
void Muestralinea()
   Muestra la situaci'on actual del procesamiento.
```

4.4.2 Parser o Analizador Sintáctico

Este componente es el que controla todo el proceso de análisis de una especificación. El procesamiento del fichero fuente se realiza en dos iteraciones. En una primera construye el árbol sintáctico correspondiente al proceso especificado dejando sin resolver algunas determinaciones de tipos. Para resolver este tema se realiza una segunda pasada en la que se establecen los tipos de los nodos del árbol que no pudieron ser resueltos en la primera iteración.

4.4.3 Análisis Semántico

El análisis semántico es realizado a la par que el sintáctico y de hecho está dirigido por este. Durante el análisis sintáctico se construye el árbol sintáctico del proceso analizado, el cual es un árbol,por lo general binario aunque para operaciones como la elección probabilista este sea n-ario, donde el nodo raíz contiene la información del operador principal y cada una de las ramas son los subarboles correspondientes a sus dos argumentos. Para los operadores con un solo argumento como puede ser la acción prefijada se tendrá que el árbol asociado solamente tendr la rama izquierda. La definición en C de las estructuras que implementan estos árboles sintácticos será la siguiente.

```
   /* Argumentos operadores */
   struct Argumentos
   {
    /* stop, div, interna urgente, or  : No tienen */
    /* Prefijo: */
        char Nombre[TAMANOMBRES];  /* Nombre de la accion */
        unsigned int t1, t2;       /* Tiempos del prefijo temporizado */
    /* Wait: */
        unsigned int t1;           /* Tiempo del wait */
    /* Paralelo: */
        struct LISTAR *c_sinc;    /* Conjunto de acciones de sincronizacion */
    /* Ocultamiento: */
        char Nombre[TAMANOMBRES];   /*  Accion ocultada */
    /* Identificador: */
        char Nombre[TAMANOMBRES];   /*  Nombre del identificador */
    /* Recursion: */
       char Nombre[TAMANOMBRES];    /*Nombre del identificador */
    /* Eleccion Probabilistica: */
       float p[MAXPALT];          /* Probabilidades */
   };


struct NODO {
  char Tipo;                    /* tipo de nodo */
  char Desc;                    /* Descripcion de operador de mas alto nivel */
  struct Argumentos Arg;        /* Argumentos */
  struct NODO *Hijos[MAXPALT];  /* Nodos hijos */
  int numhijos;                 /* Numero de hijos */
};
```

Como parte del análisis se realiza una detección de posibles recursiones indirectas sin ningun sentido e irresolubles como la que muestra el presente ejemplo:

```
prog = proc1
proc1= prog or STOP
```

Debido a la irresolubilidad de estas recursiones deberemos evitarlas. Para realizar la identificación de estas definiciones se construye un grafo de relaciones directas entre procesos sobre el que posteriormente se comprueba que éste no posee ningun bucle. La existencia de algun bucle supone la existencia de esas relaciones recursivas irresolubles.

4.5 Grafo de Alcance

Tras realizar el análisis de un expresión, y comprobada la corrección de ésta, a partir del árbol sintáctico se construye el grafo de alcance o de transición de estados definido por la semántica operacional, teniendo en cuenta para esta construcción las restricciones correspondientes. Cada uno de los nodos del grafo posee tres parámetros como son:

1. Tiempo máximo que se puede esperar en ese estado antes de ser abandonado. Este será infinito mientras no exista ninguna restricción temporal.

2. Instante más temprano en el que se puede llegar a este estado.

3. Instante máximo o más tarde en el que se puede llegar al estado.

En cuanto a las transiciones están etiquetadas por el nombre de la acción que la provoca con las siguientes características asociada:

1. Intervalo de tiempo relativo al instante de llegada al nodo de partida durante el cual puede ser ejecutada la acción y por tanto la transición.

2. Digito de control "c" que representa el tiempo que puede transcurrir desde el instante más temprano de llegada al nodo de origen para que sea posible la ejecución de la acción

Durante la construcción del grafo se va teniendo en cuenta la inhabilitacíon que pueden sufrir las distintas acciones por el paso del tiempo para de ese modo poder reducir el grafo.

5 Trabajo actual y futuro

En la actualidad nos encontramos demostrando la equivalencia entre las transiciones inducidas por la semántica operacional y las realizadas en el grafo de alcance. Tras esta demostración se procederá a la implementación de esta construcción para posteriormente sobre ellos aplicar algoritmos de decisión de propiedades.

Referencias

[FLN97] D. de Frutos, L. Llana, and M. Nuñez. *Introducing friendly testing.* 1996.

[Par97] J.J. Pardo. *Modelos Temporizados y probabilísticos. Situación actual.* Int. report, Dpto. Informática. 1997.