



**Traductor de Especificaciones XML
de Redes de Petri Coloreadas
a un Lenguaje para la Resolución
de Problemas de Optimización.**

Memoria del proyecto
de Ingeniería en Informática
realizado por
Jaime Ruiz Sánchez
y dirigido por
Mercedes Elizabeth Narciso Farias
del Departamento
de Telecomunicación e Ingeniería
de Sistemas
Bellaterra,.....de.....de 2009

Firmante **Mercedes Elizabeth Narciso Farias**
Profesora de la Escuela Técnica Superior de Ingeniería de la UAB,

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha sido realizado bajo su dirección por
Jaime Rúa Sánchez

Y para que así conste firma la presente.

Firma:

Bellaterra,de.....de 2009

Las grandes obras
son hechas no con
la fuerza, sino con
la perseverancia.

Samuel Johnson

Dedicatoria

A mis padres por apoyarme y ayudarme en cada momento. A mis hermanos por aportarme su experiencia de la vida.

A Lisa que ha llenado mi vida de amor. T'estimo.

A mis amigos que han estado ahí en todo momento.

Agradecimientos

A la profesora Mercedes Elizabeth Narciso por la ayuda recibida para completar este proyecto.

Al profesor Sergi Robles que ha sabido encontrar la mejor forma de transmitir conocimientos a sus alumnos.

Tabla de contenidos

Capítulo 1 Introducción	8
1.1 Motivación.....	8
1.2 Objetivo	10
1.3 Estado del arte.....	10
1.4 Organización de la memoria	11
Capítulo 2 Estudio de viabilidad.....	12
2.1 Requerimientos funcionales.....	12
2.2 Requerimientos de software y hardware	12
2.3 Planificación temporal del proyecto prevista	12
2.4 Costos del proyecto.....	13
2.4.1. Detalle costes del proyecto	13
2.4.2. Detalle costes mantenimiento/ampliaciones del traductor	14
2.5 Conclusiones del estudio de viabilidad	15
Capítulo 3 Fundamentos teóricos.....	16
3.1 Sistema de Eventos Discretos (SED).....	16
3.2 Redes de Petri Coloreadas	16
3.2.2 Definición formal de Red de Petri Coloreada	18
3.3 Redes de Petri Coloreadas Temporales.....	20
3.3.1 Árbol de alcance	21
3.4 CPN Tools	21
3.5 Lenguaje XML.....	21
3.6 Lenguaje para la Especificación de Redes de Petri Coloreadas.....	22
3.6.1 Definición Formal del Lenguaje	22
Capítulo 4 Desarrollo del traductor.....	25
4.1 Análisis de requerimientos.....	25
4.1.1 Evaluación y estudio de la herramienta CPN Tools.....	25
4.1.2 Proceso de traducción	26
4.1.3 Diagramas de clases.....	31
4.2 Diseño del traductor.....	32
4.2.1 Estructura de datos.....	32
4.2.2 Diagramas de actividades	33
4.2.3 Diagrama de secuencia	35
4.2.4 Visualización del fichero con la traducción	37
4.3 Implementación	39
4.4 Pruebas.....	40
4.4.1 Pruebas unitarias	40
4.4.2 Pruebas funcionales	40
4.4.3 Pruebas de integración	40
4.4.4 Pruebas de validación	40
Capítulo 5 Manual de usuario	41
5.1 Instalación del traductor.....	41
5.2 Interfaz de usuario	41
5.2.1 Descripción de los menús	43
5.2.1.1 Menú Archivo	43

5.2.1.2 Menú Edición.....	45
5.2.1.3 Menú Ver.....	46
5.2.1.4 Menú Ayuda.....	46
5.2.1.5 Menú secundario.....	47
5.2.2 Descripción de la barra de herramientas.....	48
5.2.3 Descripción de la barra de estado.....	49
5.2.4 Ventana vista preliminar.....	49
5.2.4.1 Descripción de la barra de herramientas de la vista preliminar.....	51
5.2.5 Introducción de texto.....	52
5.2.6 Ajustes de ventana.....	52
5.2.7 Finalización de la ejecución.....	52
5.3 Obtención de la Red de Petri Coloreada en XML.....	53
5.4 Obtención de la traducción de la Red de Petri Coloreada.....	53
5.5 Cambiar la vista XML a ASCII y de ASCII a XML.....	55
5.6 Aspectos de la traducción.....	56
5.7 Guardar un fichero.....	58
5.8 Configuración de la impresora.....	59
5.9 Resolución de problemas.....	60
Capítulo 6 Ejemplo de diseño y traducción de una Red de Petri Coloreada.....	64
6.1 Especificaciones del problema a modelar.....	64
6.2 Pasos para la realización de una traducción.....	66
6.3 Resultado obtenido de la traducción del sistema Job Shop.....	69
Capítulo 7 Conclusiones.....	72
7.1 Objetivos cumplidos.....	72
7.2 Posibles ampliaciones.....	72
7.3 Conclusiones generales.....	73
Anexo A Descripción de clases.....	74
A.1 Clase nodo_lugar.....	74
A.2 Clase transición.....	77
A.3 Clase arco.....	80
A.4 Clase color.....	84
A.5 Clase función.....	86
A.6 Clase variable.....	89
A.7 Clase red_petri.....	92
Anexo B Sintaxis de la gramática generadora del lenguaje de especificación RPC.....	97
B.1 Sintaxis de Especificación del Número de Sub-árboles.....	97
B.2 Sintaxis de Especificación del Número de Niveles.....	97
B.3 Sintaxis de Especificación de la Banda de Costo.....	97
B.4 Sintaxis de Especificación de la Banda de Tiempo.....	97
B.5 Sintaxis de Definición de Constantes.....	97
B.6 Sintaxis de Declaración de Conjuntos Color.....	99
B.7 Sintaxis de Declaración de Variables.....	99
B.8 Sintaxis de Definición de Funciones.....	99
B.9 Sintaxis de Definición de Nodos Lugar.....	100
B.10 Sintaxis de Definición de Nodos transición.....	101
B.11 Sintaxis de Definición de Arcos.....	101
Bibliografía.....	102
Resum.....	103

Capítulo 1

Introducción

1.1 Motivación

La optimización combinatoria estudia el modelado y solución algorítmica de problemas donde se busca maximizar (o minimizar) una función de varias variables definidas sobre un conjunto discreto, muchos de los problemas de optimización combinatoria han sido considerados problemas NP-Hard. Los problemas NP-hard o NP-difícil (Nondeterministic Polynomial time), son problemas para los cuales todos los algoritmos actualmente disponibles para encontrar soluciones óptimas a ellos requieren un número de pasos computacionales que crece exponencialmente con el tamaño del problema, el número de variables de decisión o el número de combinaciones de valores de las variables de decisión. Este proyecto surge de la idea de intentar resolver problemas de optimización combinatoria a partir de modelos de redes de Petri Coloreadas (RPC). Para ello se intenta facilitar la creación de modelos RPC, usando la herramienta de diseño gráfico CPN Tools [3] (Colored Petri Net Tools), para su posterior utilización como entrada a un simulador/optimizador de sistemas modelados con RPC.

Este proyecto forma parte de un Sistema de Ayuda a la Toma de Decisiones (en adelante SATD) [6,7] que permite evaluar diversas estrategias heurísticas de búsqueda para conseguir la mejor configuración de las variables de decisión de un problema de optimización, que pueda ser concebido como un Sistema de Eventos Discretos (SED). El formalismo de RPC se utiliza para representar el conocimiento que el experto tiene sobre el sistema.

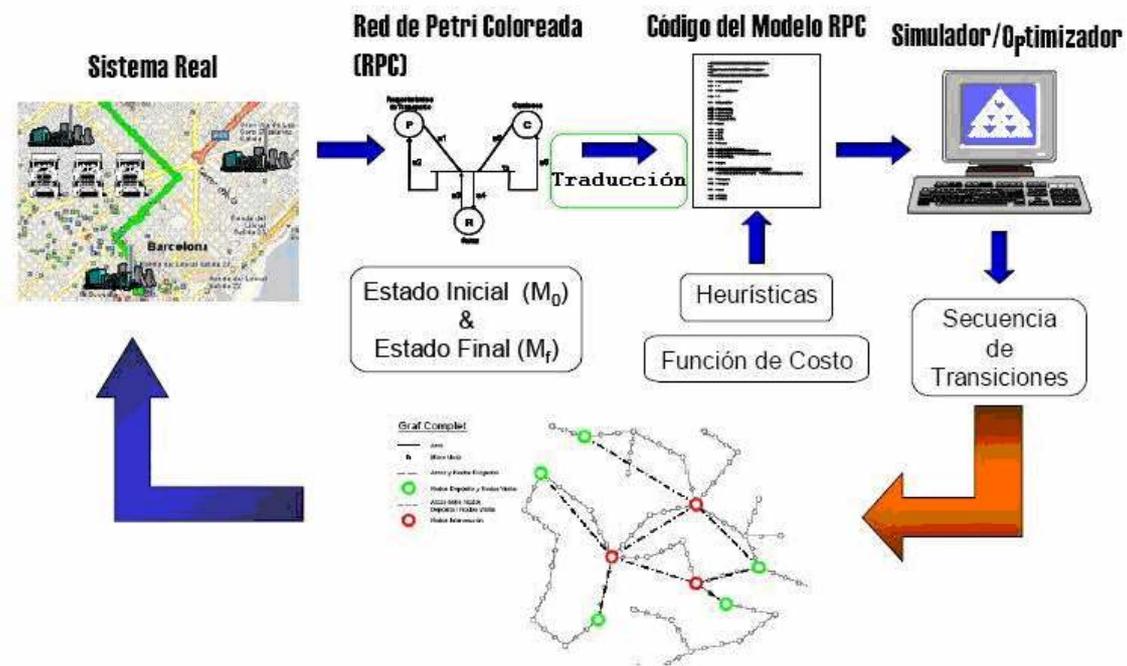


Figura 1. Principales componentes del SATD.

Este SATD consiste en un entorno de simulación basado en el modelado de un SED usando RPC, a fin de explotar las características de representación y análisis de este formalismo de modelado de sistemas para la resolución de problemas de optimización. La Figura 1 muestra los componentes del SATD que permite la evaluación de diferentes algoritmos heurísticos de búsqueda para encontrar soluciones a problemas de planificación, scheduling o routing de sistemas logísticos [7]:

- El modelo RPC: se describe sintácticamente (codifica) en un fichero ASCII. Para ello se ha diseñado tanto un lenguaje [7] (ver anexo B) de especificación de RPC que permite codificar el modelo, como un intérprete de dicho lenguaje, de tal forma que el modelo codificado pueda ser suministrado como entrada al simulador.

La ejecución del SATD da como resultado, si se trata de un sistema pequeño, la mejor secuencia de transiciones (eventos, actividades) para conducir el sistema desde el estado inicial hasta el estado final (óptimo global). En el caso de sistemas complejos, mediante el uso de heurísticas la herramienta ofrece soluciones bastante próximas al óptimo (soluciones factibles u óptimos locales), cuando las heurísticas diseñadas son eficaces.

1.2 Objetivo

El objetivo de este proyecto es desarrollar un traductor para Redes de Petri Coloreadas a partir de un fichero generado mediante la herramienta de modelado y simulación de RPC: CPN Tools[3]. El traductor deberá generar como resultado un fichero de texto con las especificaciones de la RPC con la estructura requerida por un lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC [6] que forma parte del SATD.

Este traductor tendrá una interfaz gráfica para seleccionar el fichero XML [4] que contiene cualquier RPC generada por CPN Tools que se desee traducir. Los ficheros generados por el traductor deberán ser validos para la sintaxis del lenguaje interpretado por el simulador/optimizador de RPC. [7]

El traductor que se desarrollará en este proyecto formará parte del SATD, siendo el elemento del sistema que proporcione los ficheros en ASCII con la codificación de la RPC. Los datos que manejará el traductor serán ficheros XML proporcionados por la herramienta de modelado gráfico CPN Tools.

La interfaz del traductor con la vista XML permitirá abrir, modificar, guardar y traducir los ficheros XML. En cambio con la vista ASCII permitirá abrir, modificar y guardar los ficheros ASCII con la codificación de la RPC, pudiendo pasar rápidamente de la vista XML a la vista ASCII y viceversa.

1.3 Estado del arte

Una buena herramienta para el diseño gráfico de RPC es CPN Tools, sin embargo tal herramienta no puede calcular una solución para un problema modelado mediante RPC, ya que resulta complicado y tedioso seguir paso por paso cada uno de los estados de la RPC. Para seguir los estados se ha de modificar la RPC para que copie en un fichero los diferentes estados y buscar en el fichero los estados que son solución y recorrer en orden inverso los diferentes estados hasta llegar al estado inicial, todo este recorrido se ha de hacer de forma manual.

Con la finalidad de aprovechar la facilidad de modelado gráfico, con la posibilidad de obtener la solución de un problema de optimización se ha desarrollado el Sistema de Ayuda a la Toma de Decisiones (SATD), este traductor consigue unir un fichero creado en CPN Tools al lenguaje de especificaciones del SATD.

1.4 Organización de la memoria

En el segundo capítulo de la memoria se presenta el estudio de viabilidad del proyecto. En el tercer capítulo se explican los fundamentos teóricos para el desarrollo de este proyecto: las Redes de Petri Coloreadas, lenguaje XML, la herramienta CPN Tools, etc. El cuarto capítulo explica el desarrollo seguido para la creación del traductor. En el quinto capítulo está detallado el manual de usuario. En el sexto capítulo muestra un ejemplo de traducción de una RPC en CPN Tools y traducida al lenguaje de especificaciones del fichero de entrada a un simulador/optimizador. En el séptimo capítulo están las conclusiones del proyecto.

Capítulo 2

Estudio de viabilidad

En este capítulo se explican los requisitos funcionales y tecnológicos que se necesitan para desarrollar el traductor así como la planificación de tiempo prevista y los costos directos del proyecto.

2.1 Requerimientos funcionales

El traductor debe tener como entrada y ser capaz de reconocer ficheros de una RPC en formato XML [4] creados por la herramienta CPN Tools [3], que el traductor deberá traducir en forma de fichero de salida en formato ASCII a un lenguaje interpretado por un simulador/optimizador de RPC diferente de CPN Tools. La figura 2 muestra un esquema que describe el marco del proyecto. Además el traductor deberá contar con una interfaz gráfica para interactuar con el usuario.

2.2 Requerimientos de software y hardware

- Sistema operativo: Windows XP (o superior).
- Espacio de en el disco: >2 Megas
- Memoria ram: >= 512 Megas

Este programa ha sido desarrollado en Microsoft Visual Studio C++ [1] por ser el lenguaje de desarrollo del SATD (descrito en el capítulo 1) con miras a una futura integración con el mismo.

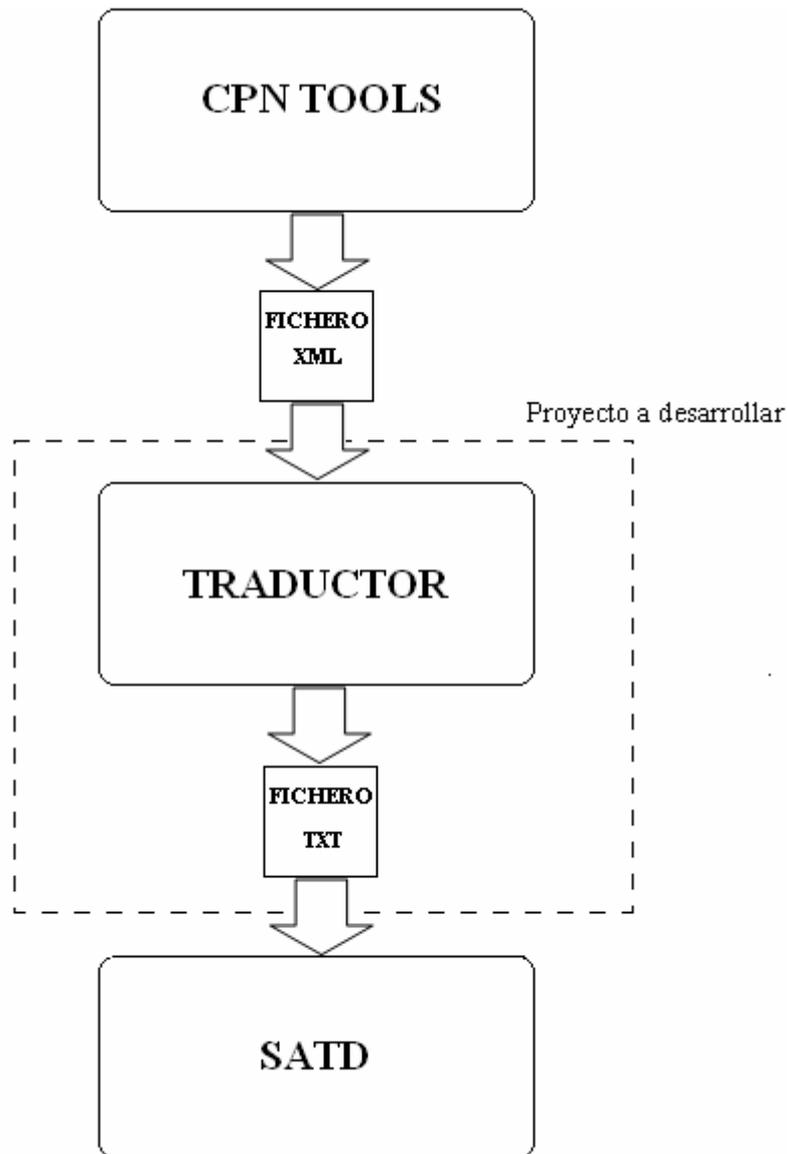


Figura 2. Esquema que resume la función del traductor.

2.3 Planificación temporal del proyecto prevista

Fase de análisis del sistema

16/10/08 a 30/10/08 Evaluación y estudio de la herramienta de CPN Tools.

30/10/08 a 13/11/08 Evaluación y estudio de los ficheros XML generados por CPN Tools.

13/11/08 a 04/12/08 Diseño de varias redes de Petri Coloreadas en CPN Tools.

04/12/08 a 11/12/08 Evaluación y análisis del lenguaje de especificaciones del

fichero de entrada al simulador/optimizador.

Fase de diseño del sistema

11/12/08 a 14/01/08 Diseño del sistema.

Fase de implementación del sistema

14/01/08 a 05/02/09 Implementación del sistema.

Fase de pruebas y mejoras del sistema

05/02/09 a 26/02/09 Pruebas y modificaciones.

26/02/09 a 16/03/09 Verificación del sistema.

Fase de documentación

16/03/09 a 14/04/09 Memoria.

21/04/09 a 28/04/09 Modificaciones de la memoria.

2.4 Costos del proyecto

2.4.1. Detalle costes del proyecto

	nº horas	precio/h	total
Importe por el tiempo dedicado al estudio de la viabilidad del PFC	2	60,00	120,00
Importe por el tiempo dedicado al diseño de software	40	30,00	1.200,00
Importe por el tiempo dedicado a la implementación del software	240	25,00	6.000,00
Importe por el tiempo dedicado a la depuración del	10	30.00	300,00

software			
Importe por el tiempo dedicado a pruebas funcionales	8	15.00	120,00
Importe por el tiempo dedicado a la confección de documentación	36	15.00	540,00
Importe del software utilizado			(a ¹) 0,00
Importe de los gastos generales y amortización del equipo			307,52
Suma costes PFC			8.587,52

Tabla 1. Relación costes/hora

a¹) La herramienta CPN Tools tiene una licencia de uso sin costes. La licencia para usar MS Visual Studio C++ fue proporcionada por Microsoft al ser estudiante de la comunidad de la UAB.

2.4.2. Detalle costes mantenimiento/ampliaciones del traductor

Modificación muy sencilla	15,00 €
Modificación sencilla	30,00 €
Modificación dificultad media	50,00 €
Modificación alta dificultad	80,00 €
Modificación muy alta dificultad	120,00 €
Compatibilidad con otra herramienta de edición RPC que no sea CPN Tools	500,00 €
Mantenimiento por 12 meses del software (a ²)	1.700,00 €

Tabla 2. Relación servicio/precio

a²) En ningún caso el mantenimiento incluirá la compatibilidad con otras herramientas de edición RPC.

2.5 Conclusiones del estudio de viabilidad

Se determina como viable la realización del traductor de especificaciones XML de RPC a un lenguaje para la resolución de problemas de optimización en el periodo de tiempo previsto.

Capítulo 3

Fundamentos teóricos

En este capítulo se describen los fundamentos teóricos necesarios para el desarrollo del traductor.

3.1 Sistema de Eventos Discretos (SED)

Un sistema de eventos discretos se caracteriza en que las propiedades de interés del sistema cambian únicamente en una secuencia de instantes de tiempo y, se puede considerar que permanecen constantes el resto de tiempo. La secuencia de instantes en los cuales el estado del sistema puede presentar un cambio (eventos), obedece a un patrón aleatorio. Los SED se modelan mediante el formalismo de RPC.

3.2 Redes de Petri Coloreadas

Las Redes de Petri Coloreadas fueron desarrolladas en la década de los 80 por Kurt Jensen, profesor y jefe del Departamento de Ciencias de la Computación en la Universidad de Aarhus, Dinamarca [1]. En el marco de ese departamento nace CPN Tools [3], una herramienta de para el modelado y simulación RPC.

Las Redes de Petri Coloreadas son un formalismo para el modelado analítico de sistemas que presentan concurrencia, sincronización y recursos compartidos. Las Redes de Petri Coloreadas permiten representar gráficamente comportamientos de sistemas simples o complejos que facilitan a los analistas tomar decisiones oportunas y eficientes sobre el sistema modelado. La representación se forma a través de elementos simples, relaciones y reglas. Ejemplos típicos de áreas de aplicación son protocolos de comunicación, teoría de grafos, problemas de transporte, sistemas de producción automatizada y análisis de flujo de trabajo entre otros.

Los elementos de modelado que se encuentran en una Red de Petri Coloreada son:

- **Nodos lugar:** Representados gráficamente por círculos o elipses y se utilizan generalmente para describir colas de espera, así como los recursos y actividades que pertenecen al sistema.
- **Arcos:** Representados gráficamente por flechas y se utilizan para conectar nodos lugar con transiciones o viceversa, pero en ningún caso conectará dos nodos lugar o dos transiciones entre si. Un arco de entrada tiene como origen un nodo lugar y como destino una transición y un arco de salida tiene como origen una transición y como destino un nodo lugar. A los arcos se les asocia un peso que son los tokens.
- **Transiciones:** Representados gráficamente por rectángulos y se utilizan para representar eventos y actividades que se suceden en el sistema. Una transición está habilitada solo cuando en todos los nodos lugar que tienen un arco hacia la transición (nodos lugar de entrada) hay el número suficiente de tokens como indique el arco correspondiente. Una transición habilitada puede ser disparada en cualquier momento. El disparo de una transición consiste en quitar los tokens de los nodos lugar de entrada a la transición como indique el arco que une la transición con el nodo lugar y añadir tokens a los nodos lugar a los que apunte un arco de salida.
- **Marcas o Tokens:** Representados gráficamente por tuplas y representan las entidades que fluyen a través del sistema.
- **Conjunto color:** Permiten modelar el tipo de información asociada a las entidades del sistema. Representados gráficamente con el nombre del conjunto color junto al nodo lugar.
- **Estado inicial:** se determina evaluando las expresiones de inicialización asociadas a cada nodo lugar, las cuales determinarán el número de marcas en cada nodo lugar, así como los valores de los colores de las marcas que representan el estado inicial del sistema.

- **Expresiones de arco:** Permiten especificar una o varios tokens asociados al nodo lugar de donde sale el arco que habilita el evento asociado a la transición o bien especifica como será el o los tokens asociados al nodo lugar a los que apunta el arco después que ocurre el evento asociado a la transición.
- **Guardas:** Son funciones booleanas que están asociadas a las transiciones que se verifican condiciones que se deben cumplir para que ocurra un evento representado por una transición.
- **Marcado:** es la especificación del número de marcas en cada nodo lugar, así como los valores de los colores de cada una de las marcas. Representa la información mínima y necesaria para poder predecir cuales son los posibles eventos y/o actividades que pueden producirse. El marcado inicial (estado inicial del sistema) se identifica como M_0 .

3.2.2 Definición formal de Red de Petri Coloreada

La definición formal de una Red de Petri Coloreada es:

$CPN = \{S, P, T, A, N, C, G, E, I\}$

$S = \{c_1, \dots, c_{nc}\}$, conjunto finito no vacío de conjuntos color.

$P = \{p_1, \dots, p_{np}\}$, conjunto finito no vacío de nodos lugar.

$T = \{t_1, \dots, t_{nt}\}$, conjunto finito no vacío de transiciones.

$A = \{a_1, \dots, a_{na}\}$, conjunto finito no vacío de arcos.

$N =$ función de nodo, $N(a_i)$ asigna al arco a_i sus nodos terminales origen y destino.

$C =$ función de color, $C(p_i)$ asigna al lugar p_i un conjunto color.

$G =$ función booleana asociada a las transiciones. Es necesario que $G(t_i)$ se evalúe a Cierto para que la transición t_i se pueda habilitar.

$E =$ Expresiones asociadas a los arcos.

$I =$ función de inicialización, $I(p_i)$ indica el conjunto de marcas iniciales en p_i .

En la figura 3 se ha representado una RPC con dos nodos lugar ($G1$ y $G2$), una transición ($T1$) que está habilitada por que se cumplen las condiciones de los arcos de

entrada: el arco que sale de G1 con la expresión $1'(a)$ y el arco que sale de G2 con la expresión $2'(b,0)$ y también se cumple la función guarda $[a = b]$, el resultado del disparo de la transición T1 se muestra en la figura 4.

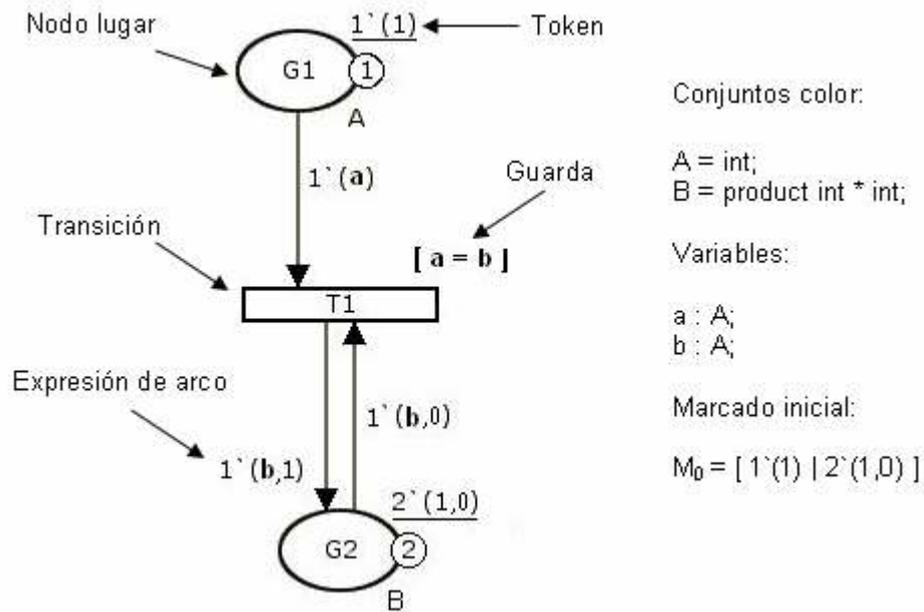


Figura 3. Representación de una RPC.

Cuando es disparada la transición T1, al nodo lugar G2 se le resta una marca del tipo $1'(1,0)$ y al nodo lugar G1 se le resta su única marca $1'(1)$, tras el disparo el nodo lugar G2 gana una marca, por lo que se queda con las marcas $1'(1,0)$ y $1'(1,1)$.

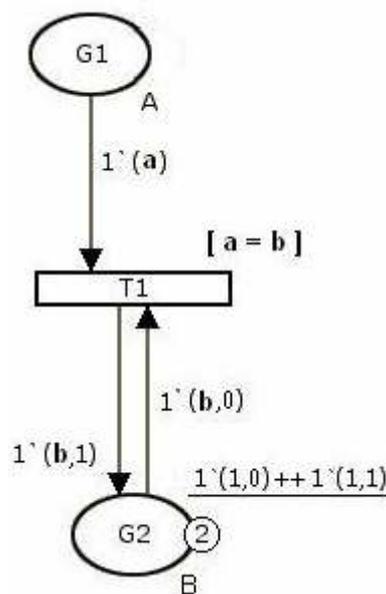


Figura 4. Representación de una RPC después del disparo de una transición.

3.3 Redes de Petri Coloreadas Temporales

Redes de Petri Coloreadas temporales son redes de Petri en las que se introduce el tiempo en forma de retardo asociado a una transición y también asociando a cada token un valor de tiempo (sellos de tiempo).

La introducción del tiempo en los modelos de Redes de Petri Coloreadas permite la descripción del comportamiento dinámico del sistema, teniendo en cuenta tanto los cambios de estado como la duración de cada actividad realizada en el sistema.

En la figura 5 se representa una RPC temporal donde hay tres transiciones (T1, T2 y T3), cada una de estas transiciones equivale al procesado de una pieza en una máquina y el tiempo que requiere el procesado de la pieza viene dado por el valor que acompaña al carácter “@” junto a la transición. La transición T1 tiene un tiempo de retardo de 7, la transición T2 tiene un tiempo de retardo de 2 y la transición T3 tiene un tiempo de retardo de 4. La secuencia que pueden seguir las piezas por el sistema no es única por lo que se representa mediante un árbol de alcance.

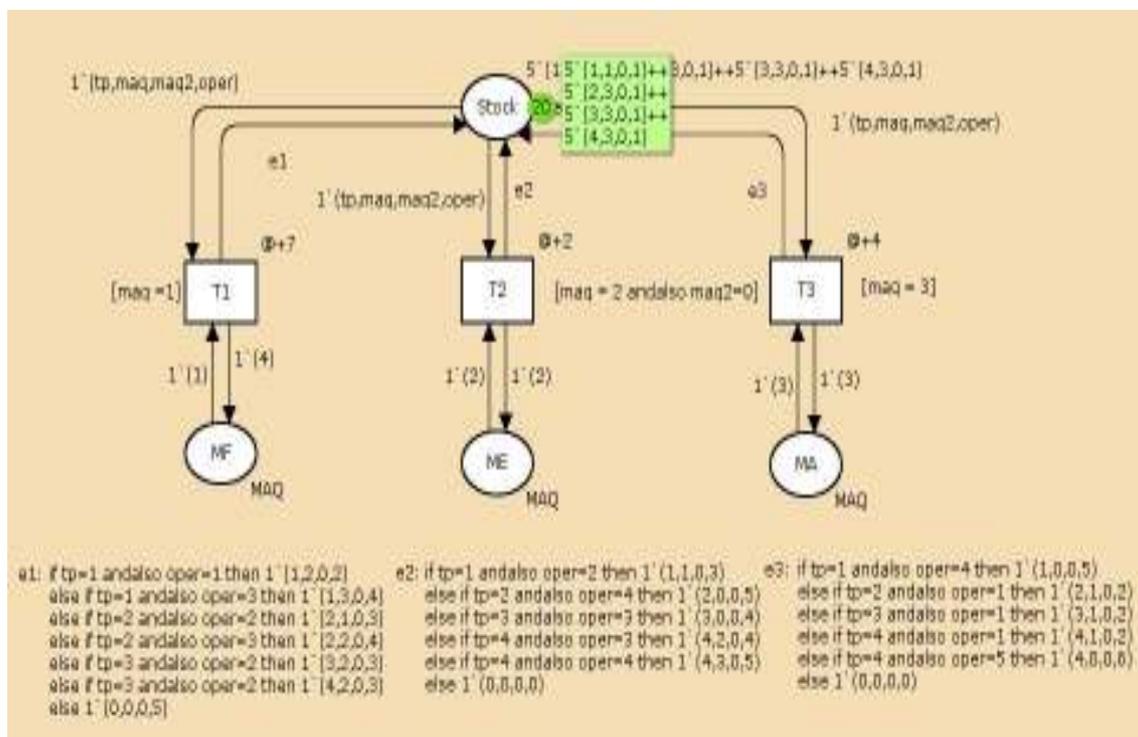


Figura 5. Red de Petri Coloreada Temporal generada con la herramienta CPN Tools [3].

3.3.1 Árbol de alcance

El árbol de alcance de una RPC consiste en un grafo que contiene un nodo por cada estado (marcación) alcanzable en el sistema modelado, y un arco por cada ocurrencia de un evento (disparo de transición), lo cual permite representar las propiedades dinámicas de la RPC. La herramienta CPN Tools genera automáticamente el árbol de alcance, para ello dispone de un icono.

3.4 CPN Tools

CPN Tools es una herramienta gráfica para la edición y validación sintáctica de RPC. CPN Tools dispone de un simulador (tanto interactivo como automático) para poder inspeccionar la evolución de las RPC. La simulación puede servir para eliminar errores iniciales del modelo. Actualmente solo disponible para Windows. Tiene una licencia libre de uso y solo es necesario pedirla vía e-mail. Pueden consultarse manuales en la web [3]. CPN Tools genera un fichero con extensión “cpn” en formato XML.

3.5 Lenguaje XML

El lenguaje XML [4] es un lenguaje que permite al usuario estructurar la información. XML son las siglas en inglés del Lenguaje de Etiquetado Extensible. Con la palabra "Extensible" se alude a la no limitación en el número de etiquetas. Las etiquetas o tags son palabras entre corchetes agudos: ‘<’ y ‘>’ y los atributos de la forma: nombre=”valor”. Las etiquetas se anidan unas dentro de otras para crear un esquema de etiquetas padre e hijo. Todas las etiquetas de un esquema XML deben tener una etiqueta de apertura y otra de cierre.

XML es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio lenguaje, así como la reutilización de partes del mismo. La información estructurada presenta varios contenidos (texto, imágenes, audio, etc.) y formas: hojas de cálculo, tablas de datos, libretas de direcciones, parámetros de configuración, dibujos técnicos, etc. La forma da alguna indicación de qué papel puede jugar el contenido (por ejemplo, el contenido de una sección encabezada con un significado difiere del contenido de una nota a pie de página, lo que significa algo

diferente que el contenido de un pie de foto o el contenido de una tabla de datos). Más o menos todos los documentos tienen la misma estructura. Ejemplo:

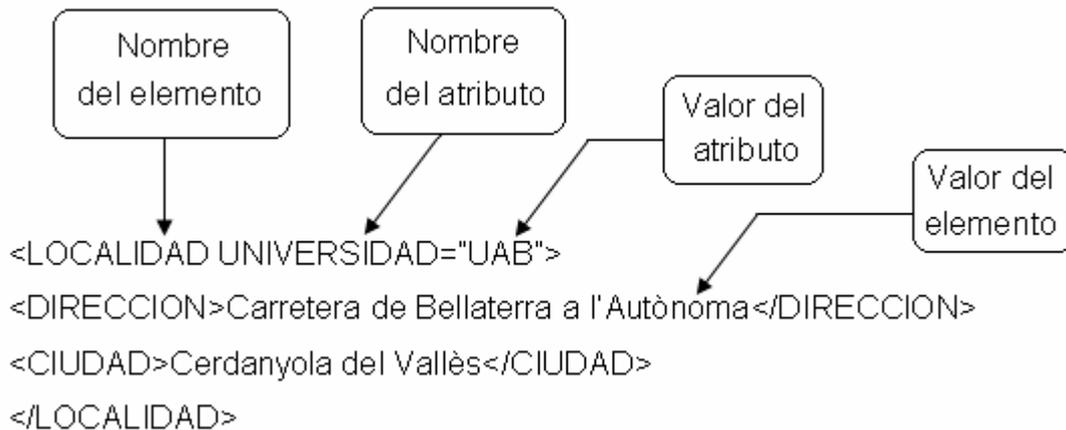


Figura 6. Detalle de una estructura XML.

En el ejemplo, la etiqueta padre `<LOCALIDAD>` contiene dos etiquetas hijas, `<DIRECCION>` y `<CIUDAD>`. No hay restricciones para nombrar y estructurar las etiquetas XML, siempre y cuando se aniden correctamente unas dentro de otras y cada etiqueta de apertura tenga su correspondiente etiqueta de cierre.

3.6 Lenguaje para la Especificación de Redes de Petri Coloreadas

3.6.1 Definición Formal del Lenguaje

El lenguaje para la especificación [6] de RPC's en el SATD que permite resolver un problema de optimización se compone de once partes, cada una identificada con una etiqueta:

SA: Especificación del número de sub-árboles para la construcción del árbol de alcance.

NI: Especificación de la profundidad de cada sub-árbol del árbol de alcance.

BC: Especificación del margen de variación de costo permitida (banda de costo).

- BT: Especificación del margen de variación de tiempo permitida (banda de tiempo).
- CT: Definición de constantes.
- CO: Declaración de conjuntos color asociados a los nodos lugar de la RPC.
- VA: Declaración de variables a utilizar para la manipulación de los colores.
- FU: Declaración de funciones a utilizar en las expresiones de arcos, guardas, funciones de costo y funciones de tiempo.
- LU: Definición de nodos lugar a utilizar en la especificación de la RPC.
- TR: Definición de nodos transición a utilizar en la especificación de la RPC.
- AR: Definición de los arcos que conectan nodos lugar y nodos transición.
- ?: Una línea con comentarios que no forman parte de la especificación de la RPC.

La siguiente regla de producción define la sintaxis de una especificación de un modelo RPC representada en notación BNF extendida.

```

<RPC> ::= <especificación_de_subarboles>
         <especificación_de_niveles>
         <especificación_de_banda_de_costo>
         <especificación_de_banda_de_tiempo>
         <definición_de_constantes>
         <declaración_de_conjuntos_color>
         <declaración_de_variables>
         <declaración_de_funciones>
         <definición_de_lugares>
         <definición_de_transiciones>
         <definición_de_arcos>

```

En el anexo B se presentan las reglas de producción en notación BNF extendida de la gramática generadora del lenguaje de especificación para cada una de las partes que componen el modelo RPC. El traductor a desarrollar deberá respetar la sintaxis del lenguaje de especificación [6] a fin de que el fichero resultante de la traducción pueda ser utilizado como entrada al simulador/optimizador del SATD (ver figura 2).

Capítulo 4

Desarrollo del traductor

En este capítulo se describe el proceso que se ha llevado a cabo para el desarrollo del traductor. Se han utilizado los diagramas de UML [5] para describir las fases de análisis y diseño del traductor.

4.1 Análisis de requerimientos

4.1.1 Evaluación y estudio de la herramienta CPN Tools

La primera fase de análisis de requerimientos fue el aprendizaje de la herramienta de modelado de RPC CPN Tools [3]. Y para aprender a utilizar la herramienta se realizaron tres RPC sencillas:

- Un problema de una cinta transportadora
- Un problema logístico
- Un problema de un sistema de producción job shop

Posteriormente se examinaron los ficheros XML resultantes y se determinó que era posible extraer la información de las RPC para su traducción. El fichero XML que se genera con la herramienta CPN Tools al crear una RPC tiene el aspecto que se muestra en la figura 7.

El traductor ha de ser capaz de leer los tags que contengan información relevante relacionada con la RPC, y en el ejemplo de la figura 7 puede verse resaltado un conjunto color X: `<layout>colset X = int with 1..6;</layout>` y dos variables x: `<layout>var x:X;</layout>` e y: `<layout>var y:Y;</layout>` con sus respectivos atributos de identificación. El traductor almacenará de forma temporal el contenido de dichos tags y los irá traduciendo al lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC.

```

cintaq.cpn - traductor
Archivo Edición Ver Ayuda
[Icons]
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE workspaceElements PUBLIC "-//CPN//DTD CPNXML 1.0//EN"
"http://www.daimi.au.dk/~cpntools/bin/DTD/6/cpn.dtd">

<workspaceElements>
  <generator tool="CPN Tools"
    version="2.2.0"
    format="6"/>
  <cpnet>
    <globbox>
      <color id="ID9269036">
        <id>X</id>
        <int>
          <with>
            <ml>1</ml>
            <ml>4</ml>
          </with>
        </int>
        <layout>colset X = int with 1..6;</layout>
      </color>
      <var id="ID9269271">
        <type>
          <id>Y</id>
        </type>
        <id>y</id>
        <layout>var y: Y;</layout>
      </var>
      <var id="ID9269405">
        <type>
          <id>X</id>
        </type>
        <id>x</id>
        <layout>var x: X;</layout>
      </var>
    </globbox>
  </cpnet>
</workspaceElements>

```

Figura 7. Fichero XML que contiene una RPC generada en CPN Tools.

4.1.2 Proceso de traducción

La traducción ha de contener los mismos elementos y en el mismo orden que en el fichero XML. En la traducción:

- %: Las líneas que tienen este carácter como primer carácter son comentarios, el intérprete del SATD ignora estas líneas.

- **NI:** Son los niveles del árbol. Este valor no existe en el fichero XML, por lo que se ha fijado su valor 1000. (a³)
- **SA:** Son los niveles del subárbol. Este valor no existe en el fichero XML, por lo que se ha fijado su valor a 1000. (a³)
- **BC:** Banda de coste. Este valor no existe en el fichero XML, por lo que se ha fijado su valor a 0. (a³)
- **BT:** Banda de tiempo. Este valor no existe en el fichero XML, por lo que se ha fijado su valor a 0. (a³)
- **CO:** Conjunto color del sistema en formato → Nombre: representación. Un conjunto color puede formar parte de otro conjunto color compuesto.
- **VA:** Variable del sistema en formato → Nombre : Tipo. Pudiendo haber definida más de una variable por línea del mismo tipo.
- **FU:** Función del sistema en formato → Nombre (Argumentos) : tipo retorno = cuerpo de la función.
- **LU:** Nodo lugar del sistema en formato → Nombre : tipo : marcas iniciales : sellos de tiempo iniciales : funciones de costo : variables : estado final.
- **TR:** Transición del sistema en formato → Nombre : función guarda : tiempo.
- **AR:** Arco del sistema en formato → Nodo lugar origen – Transición : función de arco o bien → Transición – Nodo lugar origen : función de arco.

(a³) Datos requeridos por el simulador/optimizador.

Durante la traducción se han seguido los siguientes pasos:

Traducción de colores

Un color en el fichero XML tiene la siguiente estructura:

```
<color id="ID9269036">
  <id>X</id>
  <int><with><ml>1</ml><ml>4</ml></with></int>
  <layout>colset X = int with 1..4;</layout>
</color>
```

El traductor simplemente sustituirá la palabra colset por CO y el carácter “=” por el carácter “:”. La traducción quedará así:

```
CO    X: int with 1..4;
```

Traducción de variables

Una variable en el fichero XML tiene la siguiente estructura:

```
<var id="ID9269271">
  <type> <id>Y</id> </type>
  <id>y</id>
  <layout>var y: Y;</layout>
</var>
```

El traductor sustituirá la palabra var por VA. La traducción quedará así:

```
VA    y:Y;
```

Traducción de un nodo lugar

Un nodo lugar en el fichero XML tiene la siguiente estructura:

```
<place id="ID562263"> < .. >
  <text>Pos_free</text> < .. >
  <text tool="CPN Tools" version="2.2.0">X</text> < .. >
  <text tool="CPN Tools" version="2.2.0">1`(1)++1`(2)++1`(3)++
  1`(4)</text> </initmark>
</place>
```

El traductor obtendrá la siguiente información:

Nombre del nodo lugar: Pos_free

Tipo del nodo lugar: X

Marcado inicial: 1'(1)++1'(2)++1'(3)++1'(4)

Variables asociadas al nodo lugar: Variables de tipo X

La traducción del nodo lugar quedará de esta manera:

```
LU    Pos_free:X:1'(1)+1'(2)+1'(3)+1'(4):1'[0]+1'[0]+1'[0]+1'[0]:x:*'(**);
```

El traductor formará esta línea con el nombre del nodo lugar, el tipo del nodo lugar, el marcado inicial sustituyendo el carácter “'” por el carácter “'”, sustituyendo el doble signo más por uno simple, añadiendo un '[0] (sello de tiempo) por cada token, todas las variables del tipo X y finalmente añadiendo un *'(**) que representa una máscara para el estado final cuya explicación queda fuera del alcance de este proyecto.

Traducción de una transición

Una transición en el fichero XML tiene la siguiente estructura:

```
<trans id="ID564256" explicit="false">
  <text>T2</text> <.. >
  <cond id="ID564257">
    <text tool="CPN Tools" version="2.2.0">[x=y+1]</text>
  </cond>
  <time id="ID564258">
    <text tool="CPN Tools" version="2.2.0">@+t</text>
  </time> <.. >
</trans>
```

Descripción de la transición:

Nombre de la transición: T2

Guarda asociada a la transición: [x = y + 1]

Tiempo asociado a la transición: @+(t+1)

Traducción de la transición, guarda y función de tiempo:

```
TR    T2:G1(PosBussy,Posfree):FT1;    % Transición
FU    G1(y:Y,x:X): bool=(x=(y+1));    % Función guarda
FU    FT1(t:T): real= (t+1);          % Función de tiempo
```

La transición será una línea con el nombre de la transición, el nombre de la función guarda, con el nombre de todos los nodos lugar de entrada a la transición y el nombre de la función de tiempo que describe el tiempo de duración de una actividad representada por la transición.

Para crear la función guarda se sustituye “andalso” por “&” y “orelse” por “|” y todas las variables que aparezcan en la expresión serán parámetros de la función guarda. El retorno de las funciones guarda siempre será “bool”. El nombre de la función será la unión de la letra G con un número que irá incrementado a medida que aumente el número de funciones. (G1, G2, G3, ...)

Para crear la función de tiempo, se hará desaparecer el símbolo “@”, todas las variables que aparezcan en la expresión serán parámetros de la función de tiempo. El retorno de las funciones de tiempo siempre será “real”. El nombre de la función de tiempo será la unión de las letras FT con un número que irá incrementando a medida que aumente el número de funciones de tiempo. (FT1, FT2, FT3, ...)

Traducción de arcos

Un arco en el fichero XML tiene la siguiente estructura:

```
<arc id="ID564924" orientation="PtoT" order="1"> <.. >
  <transend idref="ID564256"/>
  <placeend idref="ID562263"/> <.. >
  <text tool="CPN Tools" version="2.2.0">1'(1)</text>
</arc>
```

Descripción de arco:

Orientación: “PtoT” (nodo lugar origen (P) y transición destino (T))

Identificador de Nodo lugar: "ID562263"

Identificador de Transición: "ID564256"

Expresión de arco: 1'(1)

Traducción del arco y la función expresión de arco:

AR Posfree-T1: F1;

FU F1(x:X): X= 1'(1);

Cada elemento de la RPC creada en CPN Tools tiene un identificador único que el traductor aprovecha puesto que relaciona el arco con los nodos lugar y transiciones que une.

A partir de la expresión del arco se creará una función donde se sustituirá “andalso” por “&”, “orelse” por “|” y el carácter “” por el carácter “””, las variables que aparezcan en la expresión serán los parámetros de la función expresión de arco.

4.1.3 Diagramas de clases

La figura 8 muestra el diagrama de clases que relaciona las clases del traductor. La clase red_petri_xml contiene la descripción de la RPC obtenida de un fichero XML. La clase red_petri_coloreada contiene todas las clases que definen una RPC, que son: la clase arco, la clase variable, la clase nodo_lugar, la clase color, la clase función y la clase transición. Todas estas las clases están detalladas en el anexo A.

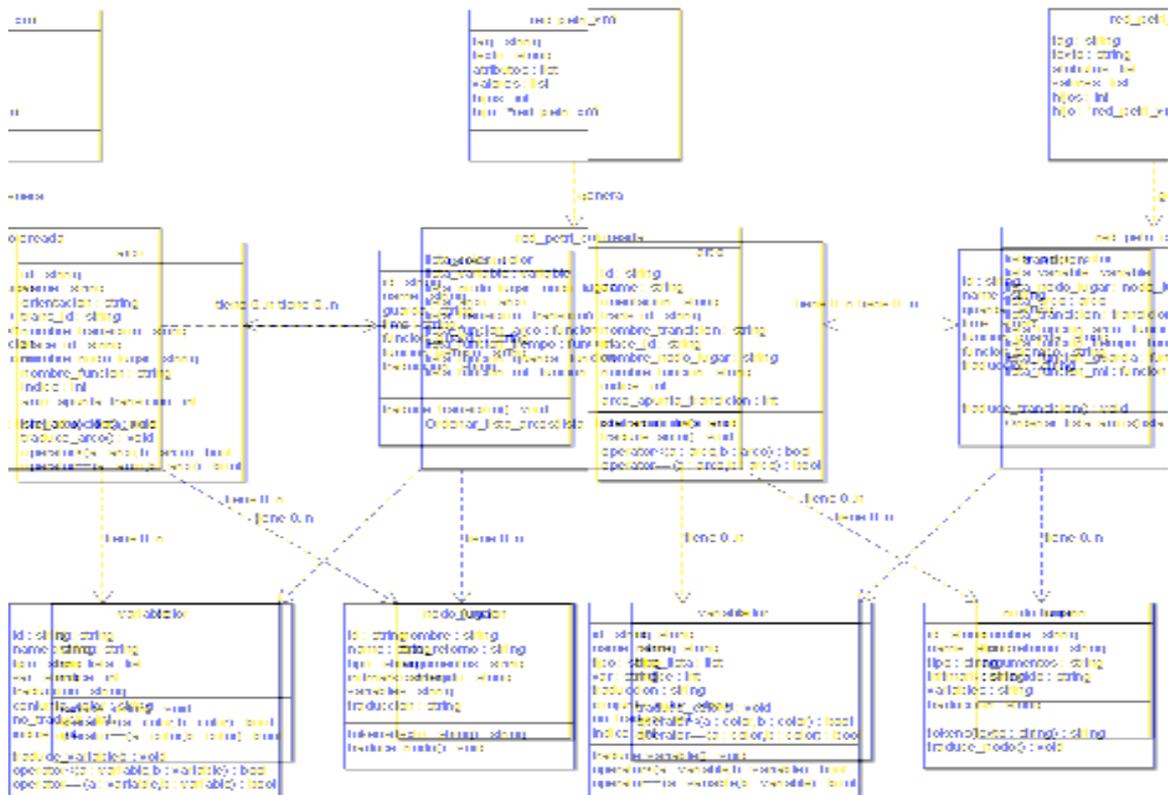


Figura 8. Diagrama de clases que representa el traductor.

4.2 Diseño del traductor

En esta sección se describe el funcionamiento del traductor mediante diagramas de actividades de los módulos de lectura del fichero XML y de la traducción de la RPC. También se describe la interacción de las clases cuando se traduce una RPC.

4.2.1 Estructura de datos

La clase `red_petri_xml` contendrá los datos leídos del fichero XML. Es una lista enlazada (véase figura 9) donde cada elemento contiene el nombre del tag, el contenido, los atributos con sus correspondientes valores, también en listas y un apuntador al siguiente elemento de la lista. El último elemento de la lista apuntará a `NULL`.

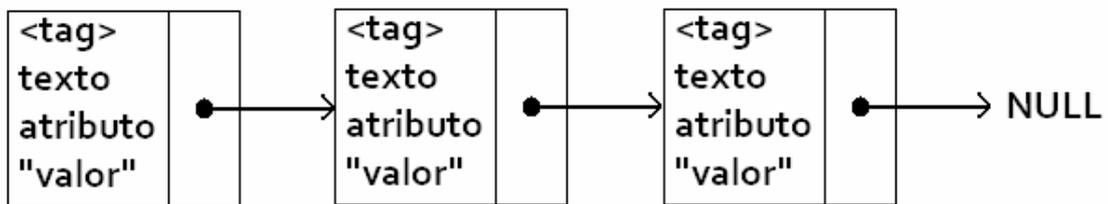


Figura 9. Representación de la lista `red_petri_xml`.

La clase `red_petri_coloreada` contiene listas de los diferentes elementos: arco, transición, nodo lugar, conjunto color y funciones.

4.2.2 Diagramas de actividades

Funcionamiento de la lectura del fichero XML

El traductor debe leer el fichero XML carácter a carácter, busca cada carácter de inicio de etiqueta: '`<`' almacena el nombre de la etiqueta, busca si tiene atributos, los almacena y continúa hasta leer el final de la etiqueta. Si el fichero no ha finalizado vuelve a buscar una nueva etiqueta. Este funcionamiento está reflejado en la figura 10.

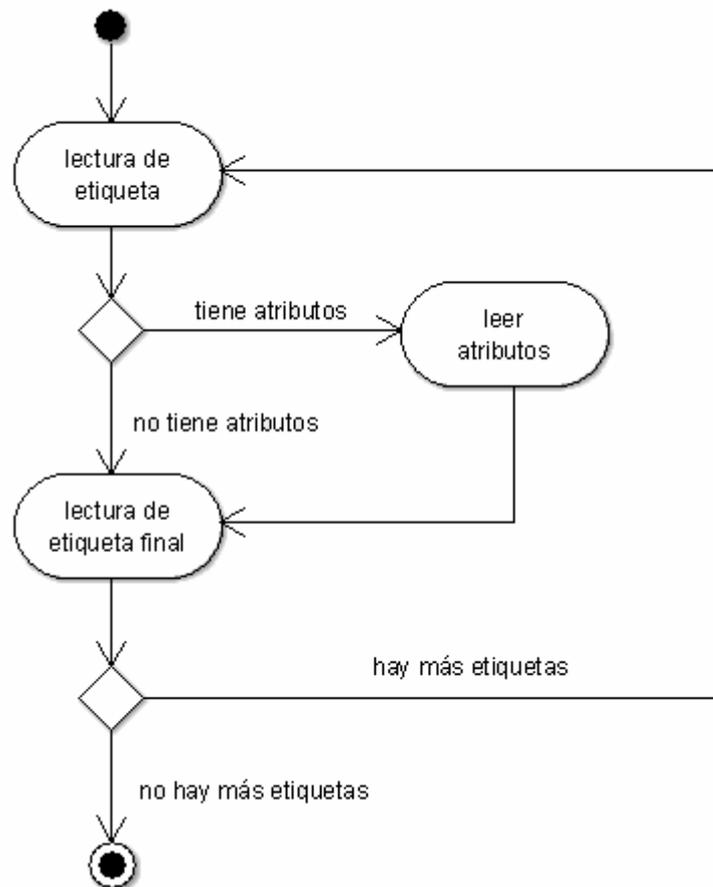


Figura 10. Diagrama de actividades de la lectura del fichero XML.

Funcionamiento de la fase de traducción

Para cada etiqueta (tag) que es leída el traductor comprueba que es una etiqueta de interés: color, variable, nodo lugar, etc. En caso de serlo se almacena en la clase llamada `red_petri_xml`.

Una vez que todos los elementos están guardados en la clase `red_petri_xml`, se procede a su traducción a el lenguaje de especificación en este orden:

- 1 Traducción de los colores.
- 2 Traducción de las variables.
- 3 Se asocia las variables a los nodos lugar para facilitar la traducción de los nodos lugar.
- 4 Traducción de los nodos lugar.

- 5 Traducción de las transiciones.
- 6 Traducción de los arcos.
- 7 Creación de las funciones expresiones de arco.
- 8 Creación de las funciones guarda.
- 9 Creación de las funciones de tiempo.
- 10 Traducción de las constantes.

Este comportamiento queda reflejado en la figura 11.

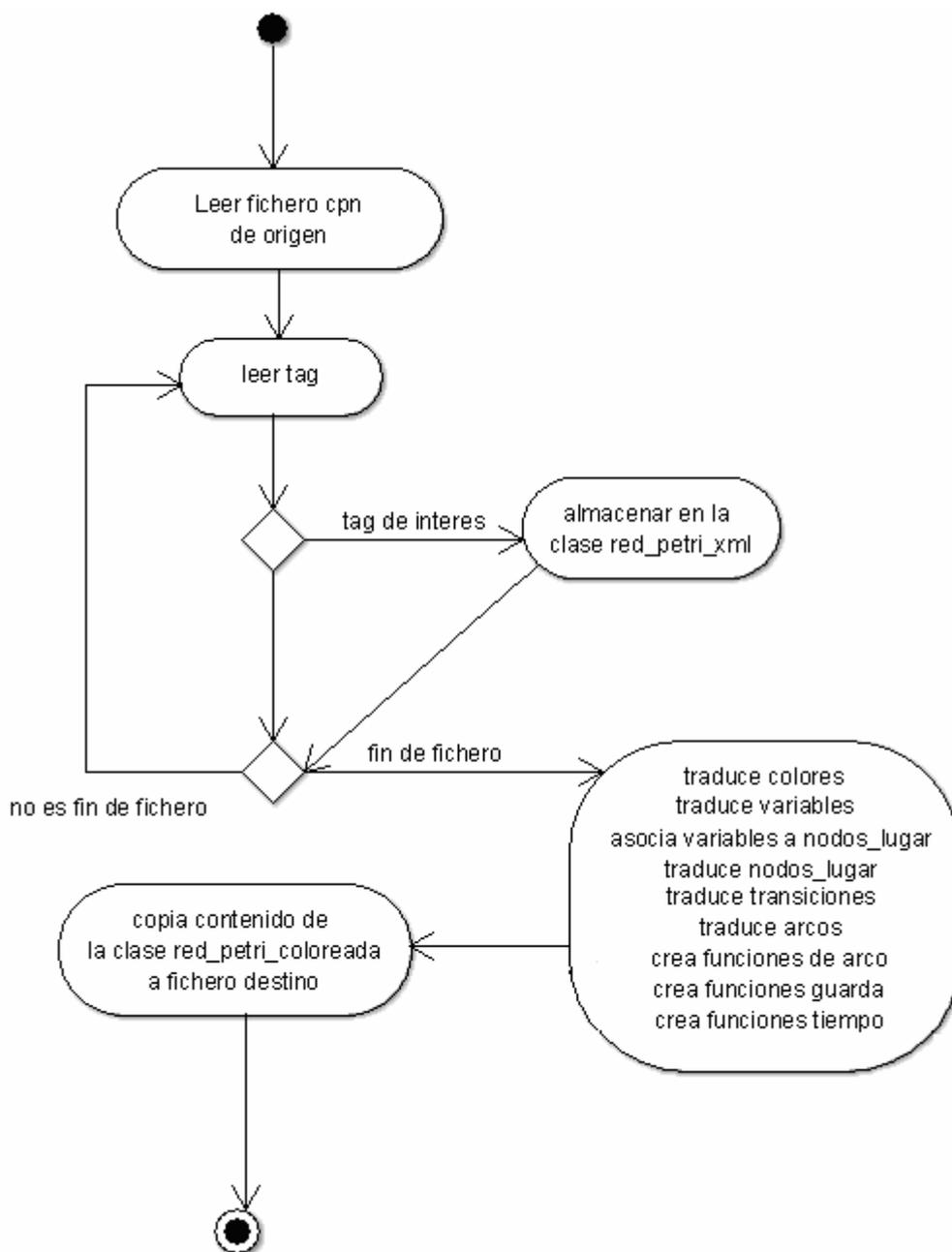


Figura 11. Diagrama de actividades de la fase de traducción de la RPC.

4.2.3 Diagrama de secuencia

Secuencia del traducción

La clase `red_petri_coloreada` contiene listas con los elementos de la RPC sin traducir, el traductor lee cada elemento y lo traduce, entonces el elemento queda traducido en la misma lista de la clase `red_petri_coloreada`.

- Cada elemento de la lista de colores es leído, traducido y almacenado en la clase `red_petri_coloreada`.
- Cada elemento de la lista de variables es leído, traducido y almacenado en la clase `red_petri_coloreada`.
- Cada elemento de la lista `nodo_lugar` es leído, traducido y almacenado en la clase `red_petri_coloreada`.
- Cada elemento de la lista transición es leído, traducido, genera una función y ambas son almacenadas en la clase `red_petri_coloreada`.
- Cada elemento de la lista arco es leído, traducido, genera una función expresión de arco y ambas son almacenadas en la clase `red_petri_coloreada`.

Esta secuencia queda reflejada en la figura 12.

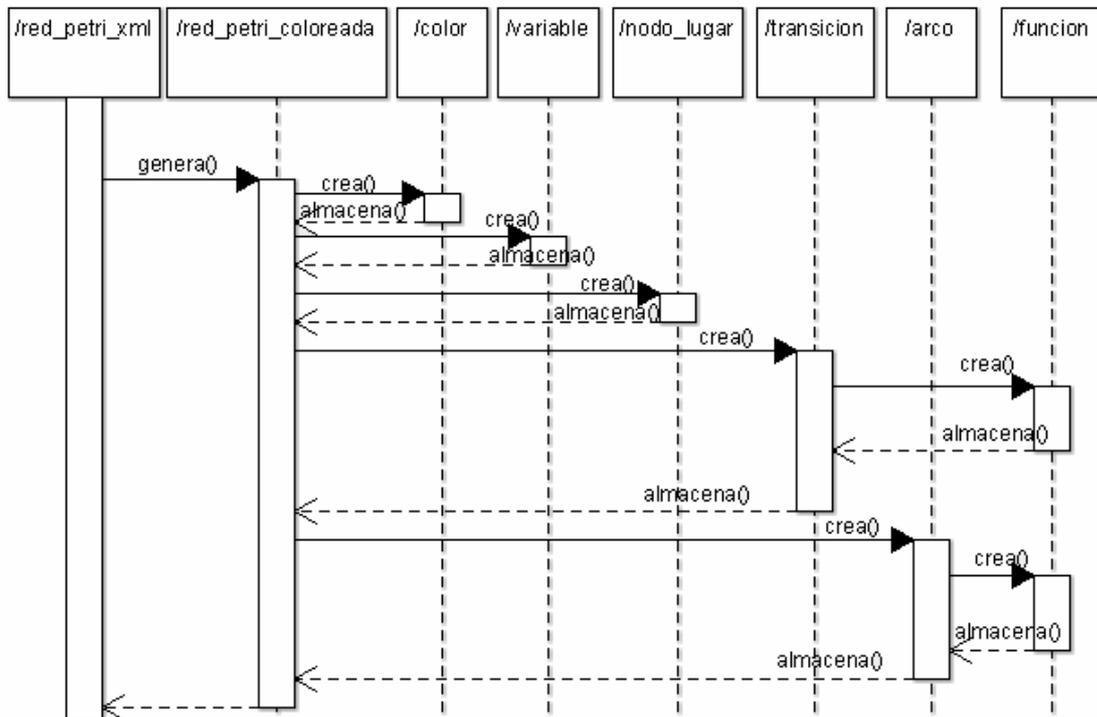


Figura 12. Diagrama de secuencia que sigue el traductor.

4.2.4 Visualización del fichero con la traducción

Para ver el resultado de una traducción se muestra la figura 13 con una RPC que representa una cinta transportadora con cuatro posiciones y cuatro piezas, el estado de las posiciones será libre (pos_free) o ocupado (pos_bussy), cada una de las transiciones representa un movimiento de la cinta, por ejemplo, la transición T1 cambiará el estado de la posición 1 de la cinta de libre a ocupado.

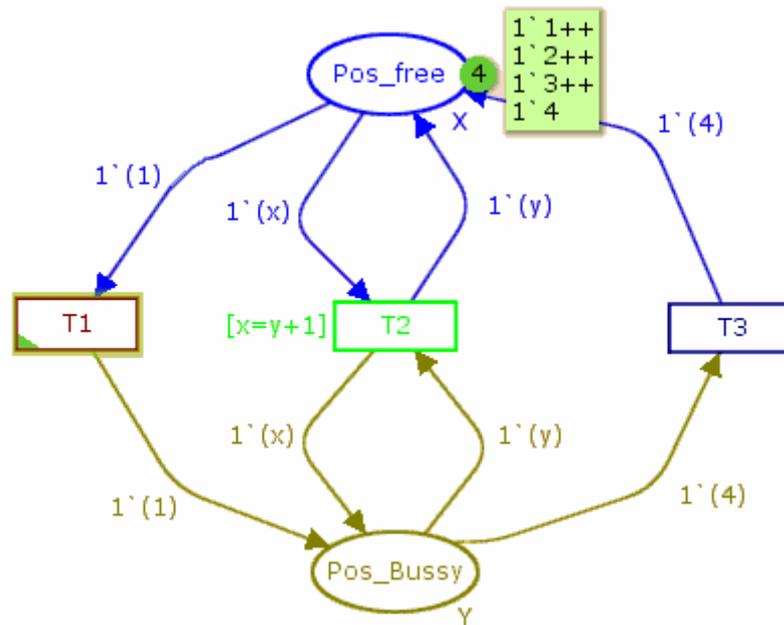


Figura 13. RPC de una cinta con cuatro posiciones.

La figura 14 muestra el diseño de la interfaz del traductor con la traducción de esta RPC en el lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC. Se ha dividido en dos imágenes (14.a y 14.b) por cuestiones estéticas.

```

cinta.txt - traductor
Archivo Edición Ver Ayuda
%
% Nombre_fichero:cinta.txt          FECHA: 10 - 9 - 2009
%
%
% Número de niveles del árbol
NI 1000
% Número de Subárboles
SA 1000
% Constantes
% Bandas de coste y tiempo
BC 0
BT 0
% Conjuntos color
CO INT: int;
CO UNIT: unit;
CO BOOL: bool;
CO STRING: string;
CO X: int with 1..4;
CO Y: int with 1..4;
CO Posfree: X;
CO PosBussy: Y;
% Variables
VA y: Y;
VA x: X;
% Funciones expresiones de arco
FU F1[x:X]: X= 1'[1];
FU F2[y:Y]: Y= 1'[1];
FU F3[y:Y]: Y= 1'[4];
FU F4[x:X]: X= 1'[4];

```

Figura 14.a Fichero ASCII generado al traducir una RPC.

```

FU    F5[x:X]: X= 1'[x];
FU    F6[y:Y]: X= 1'[y];
FU    F7[y:Y]: Y= 1'[y];
FU    F8[x:X]: Y= 1'[x];

%    Guardas

FU    G1[y:Y,x:X]: bool={x={y+1}};

%    Funciones de costo

%    Lugares

LU    PosBussy:Y:::y:*(**);
LU    Posfree:X:1'[1]+1'[2]+1'[3]+1'[4]:1'[0]+1'[0]+1'[0]+1'[0]:x:*(**);

%    Transiciones

TR    T1:true:0;
TR    T2:G1(PosBussy,Posfree):0;
TR    T3:true:0;

%    Arcos

%    Transición T2

AR    Posfree-T2: F5;
AR    PosBussy-T2: F7;
AR    T2-Posfree: F6;
AR    T2-PosBussy: F8;

%    Transición T1

AR    Posfree-T1: F1;
AR    T1-PosBussy: F2;

%    Transición T3

AR    PosBussy-T3: F3;
AR    T3-Posfree: F4;

```

Figura 14.b Fichero ASCII generado al traducir una RPC (segunda parte).

4.3 Implementación

La implementación del traductor se dividió en cuatro módulos:

- **Modulo Lectura de fichero XML**, donde se obtiene toda la descripción de la Red de Petri Coloreada de un fichero estructurado en XML creado con la herramienta de diseño CPN Tools.

- **Modulo Red de Petri Coloreada Temporal**, este modulo está alimentado por el modulo de lectura, almacena los elementos de la RPC sin traducir en las listas de la clase `red_petri_coloreada`.
- **Modulo Red de Petri Coloreada Traducida**, este modulo está alimentado por el modulo de Red de Petri Coloreada Temporal, aquí se interpreta y se traducen los componentes de la Red de Petri Coloreada para posteriormente escribir el resultado estructuradamente en un fichero de texto ASCII.
- **Modulo Interfaz de Usuario**, este modulo, como su propio nombre indica, se encarga de comunicar al programa las órdenes del usuario y muestra al usuario las respuestas a las órdenes recibidas.

La implementación de los diferentes módulos se hizo de forma lazo en cascada, como se muestra en la figura 15. Durante la implementación de cada módulo es posible volver al modulo anterior para corregir o modificar la implementación del módulo anterior, y por tanto una modificación de un módulo puede tener efectos en el módulo que le sigue.

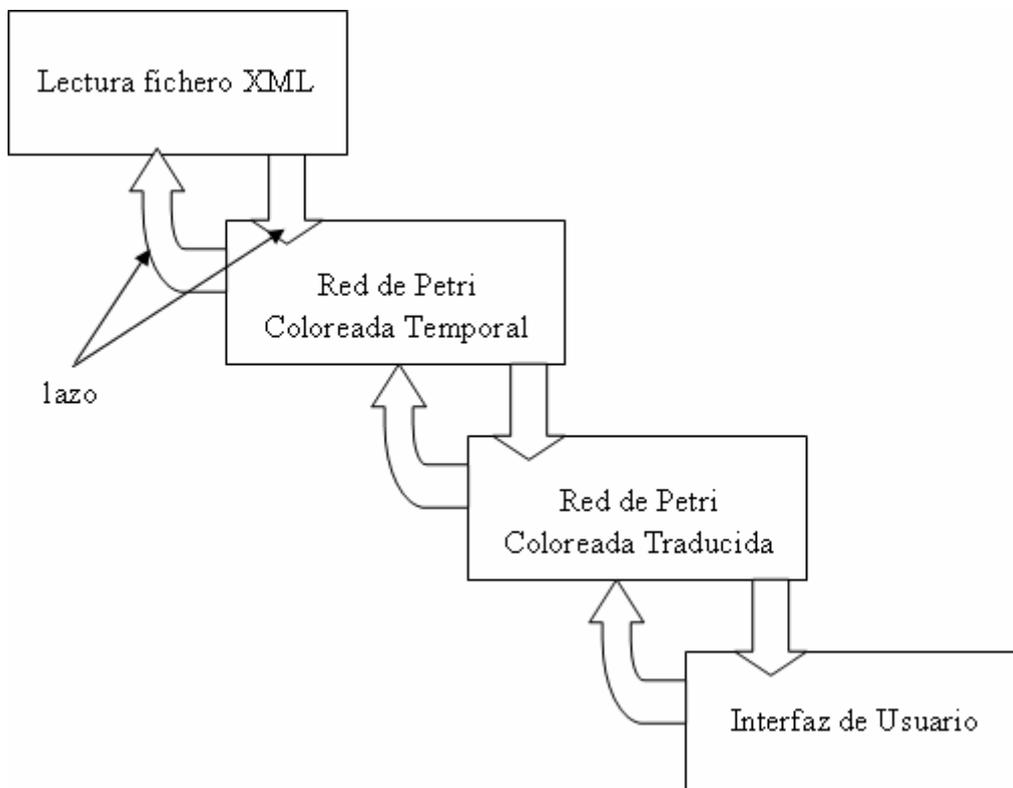


Figura 15. Modelo de lazo en cascada utilizado en la implementación del traductor.

4.4 Pruebas

4.4.1 Pruebas unitarias

Las pruebas unitarias consistieron en:

- Comprobar que el fichero XML era reconocido.
- Obtención de un color de un fichero con un solo color.
- Obtención de un arco de un fichero con un solo arco.
- Obtención de una variable de un fichero con una sola variable.
- Obtención de una transición de un fichero con una sola transición.
- Obtención de un nodos lugar de un fichero con un solo nodo lugar

4.4.2 Pruebas funcionales

Durante el desarrollo del traductor se fueron haciendo pruebas funcionales con RPC simples para obtener y traducir los diferentes elementos de la RPC por separado: conjuntos color, variables, transiciones, arcos y nodos lugar. Pudiendo así concretar el correcto funcionamiento de cada elemento.

4.4.3 Pruebas de integración

A lo largo del desarrollo del traductor se hicieron traducciones completas, donde se pudieron registrar fallos que fueron subsanados. La mayoría de los fallos estaban relacionados con la sintaxis del lenguaje en el fichero resultado.

4.4.4 Pruebas de validación

Las pruebas de validación han consistido en la traducción de varias RPC y comprobando que el resultado era el esperado. Las traducciones han sido utilizadas como entrada al simulador/optimizador y se ha comprobado que los ficheros generados cumplen con la especificación sintáctica del lenguaje (ver anexo B).

Capítulo 5

Manual de usuario

En este capítulo se presenta el manual de usuario para la asistencia a los usuarios del traductor.

5.1 Instalación del traductor

El traductor se presenta en el fichero ejecutable “traductor.exe” y no necesita instalación previa. El traductor funciona en el sistema operativo (S.O.) Microsoft Windows XP. Es posible que el fichero se visualice simplemente por “traductor” en el caso de que el usuario tenga configurada la opción de ocultar las extensiones de archivo para tipos de archivo conocido.

5.2 Interfaz de usuario

La interfaz del traductor se compone de las siguientes partes: (ver figura 16)

- **Menú archivo:** Para abrir, guardar, imprimir, traducir un fichero y salir del traductor.
- **Menú edición:** Para cortar, copiar, pegar y deshacer texto escrito.
- **Menú ver:** Habilitar o deshabilitar la barra de herramientas y la barra de estado.
- **Menú ayuda:** Información sobre el traductor y su funcionamiento.
- **Barra de herramientas:** Contiene los diferentes botones que hacen funcionar el traductor.
- **Botón traducción:** De color blanco y letra T de color rojo, sirve para indicar al programa que realice la traducción de un fichero XML.

- **Botón cambio de vista:** De color azul y verde, sirve para cambiar de la vista XML a la vista ASCII o viceversa.
- **Barra de estado:** (opcional) Mostrará al usuario el estado del traductor, siendo el estado “listo” cuando el programa escuchará las peticiones del usuario y mostrará información sobre las diferentes opciones de menú y botones de los cuales se compone el traductor.
- **Barra de desplazamiento vertical:** Se utilizará para desplazarse por el documento de forma vertical en el caso de que dicho documento ocupe más de una ventana de visualización.
- **Botón ayuda:** Mostrará al usuario una ventana con la información de la versión del traductor y su autor.
- **Zona de visualización/edición:** En esta parte de la interfaz se visualizará el contenido del fichero XML o ASCII dependiendo de la elección del usuario. Los ficheros abiertos podrán ser editados a voluntad, teniendo en cuenta que las modificaciones podrán ser almacenadas en el sistema de ficheros del usuario previo aviso de dicha operación.

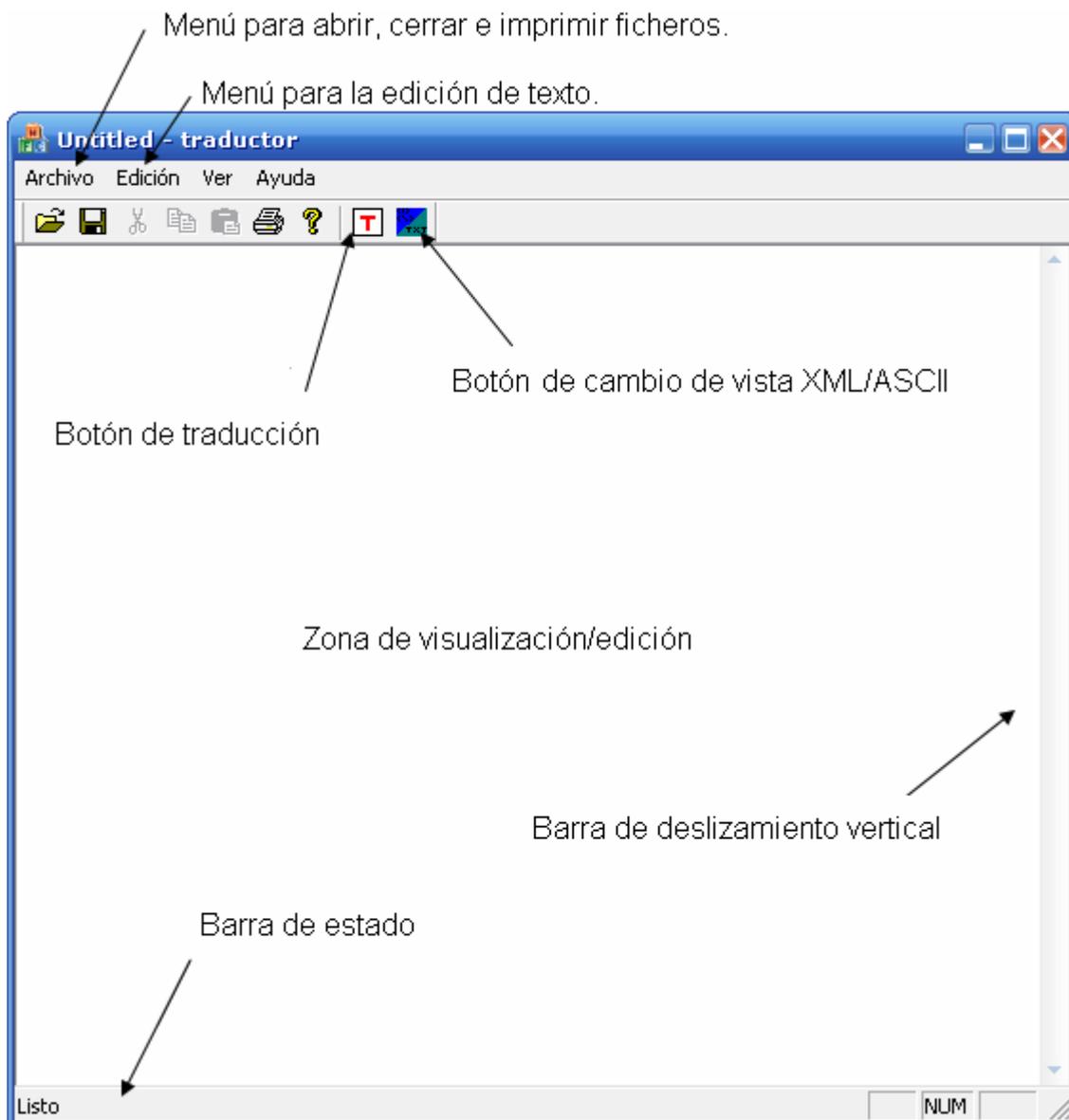


Figura 16. Ventana principal del traductor.

5.2.1 Descripción de los menús

5.2.1.1 Menú Archivo

El menú Archivo contiene las opciones (Figura 17):

- **Abrir...** (con la combinación de teclas Ctrl+O de acceso rápido): El usuario deberá usar esta opción para abrir tanto ficheros XML como ficheros ASCII.

- **Guardar** (con la combinación de teclas Ctrl+S de acceso rápido): El usuario podrá usar esta opción para almacenar en su sistema de ficheros el contenido del documento que esté visualizando.
- **Guardar como...:** Esta opción permite guardar una copia con diferente nombre al fichero ya existente en el sistema de ficheros del usuario.
- **Traducir:** Una vez abierto un fichero con una Red de Petri Coloreada en XML el usuario podrá traducir la RPC para obtener un fichero ASCII con la traducción.
- **Imprimir...:** (con la combinación de teclas Ctrl+P de acceso rápido) El usuario podrá enviar a una impresora el contenido del documento que haya abierto, ya sea XML o ASCII.
- **Vista preliminar:** El usuario podrá utilizar esta opción para previsualizar el contenido del documento que será enviado a una impresora en caso de desear imprimir el documento.
- **Configurar impresión:** Con esta opción aparecerá una ventana donde el usuario podrá seleccionar una impresora, el tamaño del papel, el origen del papel, la orientación de la impresión (vertical u horizontal), entre otras configuraciones antes de proceder a la impresión de un documento.
- **Salir:** Para finalizar la ejecución del traductor.

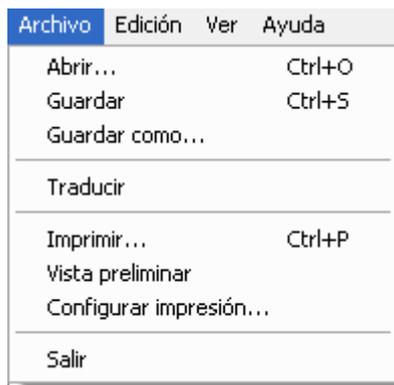


Figura 17. Detalles del menú Archivo.

5.2.1.2 Menú Edición

El menú Edición contiene las siguientes opciones (Figura 18):

- **Deshacer:** (Con la combinación de teclas Ctrl+Z de acceso rápido) Para volver a un estado anterior a un cambio realizado sobre el documento. (**a⁴**)
- **Cortar:** (Con la combinación de teclas Ctrl+X de acceso rápido) Esta opción permite al usuario quitar una selección de caracteres del documento que quedarán almacenadas en la memoria del S.O. (**a⁵**)
- **Copiar:** (Con la combinación de teclas Ctrl+C de acceso rápido) Esta opción permite al usuario copiar una selección de caracteres del documento que quedarán almacenadas en la memoria del S.O. (**a⁵**)
- **Pegar:** (Con la combinación de teclas Ctrl+V de acceso rápido) Esta opción permite al usuario pegar el contenido de una selección previa de caracteres previamente almacenada en la memoria del S.O.

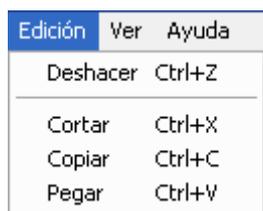


Figura 18. Detalles del menú Edición.

(a⁴) **Aviso:** La opción deshacer quedará neutralizada en el caso de cambiar la vista XML/ASCII.

(a⁵) **Aviso:** El S.O. normalmente solo guarda una selección de caracteres a la vez en memoria, por lo que si se utiliza dos veces esta opción, se perderá la primera de las selecciones.

5.2.1.3 Menú Ver

El menú Ver permite las siguientes opciones como puede verse en la figura 19:

- **Barra de herramientas:** Seleccionando esta opción aparecerá un a su izquierda indicando que la ventana principal de el traductor mostrará la barra de herramientas. (a⁶)
- **Barra de estado:** Seleccionando esta opción aparecerá un a su izquierda indicando que la ventana principal del traductor mostrará la barra de estado.

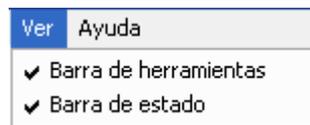


Figura 19. Detalles del menú Ver.

(a⁶) **Nota:** Se recomienda al usuario tener esta opción activada en todo momento para una mayor comodidad en el manejo del traductor.

5.2.1.4 Menú Ayuda

El menú Ayuda contiene la siguiente opción (Figura 20):

- **Acerca de traductor...:** Esta opción abre una ventana secundaria para mostrar al usuario información sobre la versión del traductor que está utilizando y sobre el autor.



Figura 20. Detalles del menú Ayuda.

5.2.1.5 Menú secundario

El menú secundario que se obtiene haciendo clic con el botón derecho contiene las siguientes opciones (Figura 21):

- **Deshacer:** Para volver a un estado anterior a un cambio realizado sobre el documento.
- **Cortar:** Esta opción permite al usuario quitar una selección de caracteres del documento que quedarán almacenadas en la memoria del S.O.
- **Copiar:** Esta opción permite al usuario copiar una selección de caracteres del documento que quedarán almacenadas en la memoria del S.O.
- **Pegar:** Esta opción permite al usuario pegar el contenido de una selección previa de caracteres previamente almacenada en la memoria del S.O.
- **Seleccionar todo:** Esta opción permite al usuario seleccionar todo el contenido del documento que está visualizando.

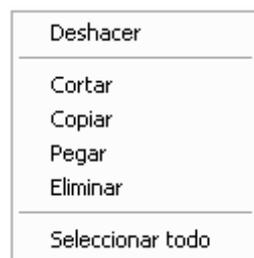


Figura 21. Detalles del menú secundario.

5.2.2 Descripción de la barra de herramientas

La barra de herramientas contiene las siguientes opciones (Figura 22):

- El icono  representa la opción de Abrir fichero.
- El icono  representa la opción de Guardar y Guardar como de un fichero.
- El icono  representa la opción de Cortar una selección de caracteres.
- El icono  representa la opción de Copiar una selección de caracteres.
- El icono  representa la opción de Pegar una selección de caracteres de la memoria del S.O.
- El icono  representa la opción de Imprimir un documento.
- El icono  representa la opción de Ayuda del traductor.
- El icono  representa la opción de Traducir del traductor.
- El icono  representa la opción de Cambio de vista XML/ASCII.



Figura 22. Detalles de la barra de herramientas.

Esta barra no está fija, es decir, puede ser desplazada por cualquier parte de dentro y fuera de la ventana principal. Para realizar esta operación haga un clic sostenido en la parte más a la izquierda de la barra de herramientas y desplace la barra hasta la ubicación que desee. También es posible cambiar el tamaño del recuadro de la barra de herramientas para ello coloque la barra de herramientas en forma de ventana y haga clic

sostenido en cualquiera de los bordes de esta ventana y desplace el cursor hasta obtener la forma (siempre cuadrada) de la ventana deseada. No debe preocuparse de cómo queda la configuración de esta barra pues una vez finalizada la ejecución del traductor esta recuperará la posición inicial para cuando el usuario vuelva a utilizarla.

5.2.3 Descripción de la barra de estado

La barra de estado se compone de las siguientes partes (figura 23):

- **Prompt:** Texto descriptivo del estado del traductor e información sobre los botones y opciones de menú de la ventana principal cuando el cursor pasa por encima de estos.
- **CAP:** Indica que el bloqueo de mayúsculas está activado. Cualquier texto introducido por teclado aparecerá en mayúsculas. Para introducir caracteres en minúscula con este indicador activo bastará con apretar shift+tecla con el carácter deseado en minúscula.
- **NUM:** Indica que el bloqueo numérico está activado, dependiendo de si el bloqueo está activo o no, la funcionalidad del teclado número cambiará. (Es preferible tenerlo activado)
- **SCRL:** Indica que el bloqueo de deslizamiento está activado.
- **Icono ajuste de ventana:** Se utiliza para redimensionar el tamaño de la ventana.

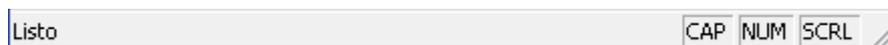


Figura 23 Detalle de la barra de estado.

5.2.4 Ventana vista preliminar

La ventana de vista preliminar se compone de (Figura 24)

- **Barra de herramientas:** Contiene diferentes botones para el manejo de la vista preliminar por parte del usuario.

- **Zona de visualización:** En esta zona el usuario podrá visualizar el contenido del fichero XML o bien el contenido del fichero ASCII previo paso a la impresión en papel.
- **Barra de estado:** Mostrará al usuario la página actual del documento que se está visualizando e información sobre los diferentes botones a medida que desplaza el cursor por encima de ellos.



Figura 24. Ventana vista preliminar del traductor.

5.2.4.1 Descripción de la barra de herramientas de la vista preliminar

La barra de herramientas de la vista preliminar se compone de los botones (Figura 25):

- **Print** (Con la combinación de teclas Alt+P de acceso rápido) abre la ventana de configuración de la impresora.
- **Next Page** (Con la combinación de teclas Alt+N de acceso rápido) muestra la siguiente página del documento.
- **Prev Page** (Con la combinación de teclas Alt+v de acceso rápido) muestra la página anterior del documento.
- **Two Page** (Con la combinación de teclas Alt+T de acceso rápido) muestra el contenido de la página actual en dos páginas).
- **Zoom In** (Con la combinación de teclas Alt+I de acceso rápido) permite ver el contenido del documento en una fuente mayor. Es posible ejecutar esta opción dos veces.
- **Zoom Out** (Con la combinación de teclas Alt+O de acceso rápido) permite ver el contenido del documento en una fuente menor. Solo es posible ejecutar esta opción después de utilizar el Zoom In.
- **Close** (Con la combinación de teclas Alt+C de acceso rápido) cierra la ventana de la vista preliminar.

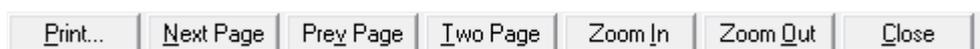


Figura 25. Detalle de la barra de herramientas de la ventana vista preliminar.

5.2.5 Introducción de texto

El usuario puede introducir texto en el documento que presente la vista haciendo clic izquierdo en la zona de visualización/edición en la sección que desee, utilizando el teclado, o bien a través de la memoria del S.O. (portapapeles). Una vez que un documento haya sido modificado y el usuario quiera traducirlo o bien cerrar el programa, el traductor preguntará al usuario si desea que la última modificación del documento pase al sistema de ficheros del S.O. En caso contrario las modificaciones no se reflejarán ni en la traducción ni en el fichero que el usuario guarde en el sistema de ficheros de su S.O.

5.2.6 Ajustes de ventana

El tamaño de la ventana podrá ser ajustado de diferentes maneras:

- En la parte inferior derecha de la ventana se encuentra el icono:  que haciendo clic sostenido sobre él y desplazando el puntero del ratón en cualquier dirección y soltando el clic sostenido la ventana adoptará el tamaño deseado.
- En la parte superior derecha se encuentra el icono:  que al hacer clic sobre él la ventana se minimizará por completo, es decir, solo ocupará un espacio en la barra de tareas del S.O.
- También en la parte superior derecha se encuentra el icono:  que al hacer clic sobre él la ventana se maximizará y ocupará todo el tamaño de ventana posible.
- Cuando la ventana esté maximizada al completo se podrá hacer clic sobre el icono:  y la ventana recuperará el tamaño previo a ser maximizada.

5.2.7 Finalización de la ejecución

El usuario dispone de dos maneras para finalizar la ejecución del traductor:

- En el menú Archivo la opción Salir.
- Haciendo clic sobre el icono:  que se encuentra en la parte superior derecha de la ventana.

5.3 Obtención de la Red de Petri Coloreada en XML

En el menú Archivo se encuentra la opción Abrir y como se muestra en la figura 26, se ha de seleccionar el tipo de fichero *.cpn, a continuación el contenido del fichero XML será mostrado en la zona de visualización/edición.

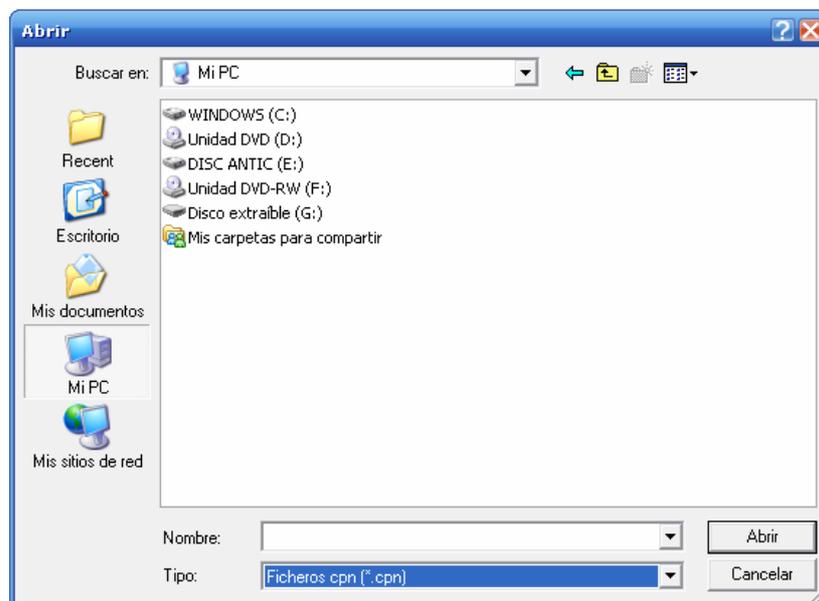


Figura 26. Ventana Abrir con la selección de ficheros cpn.

5.4 Obtención de la traducción de la Red de Petri Coloreada

Existen dos posibilidades para obtener la traducción de una Red de Petri Coloreada:

- Obteniendo en primer lugar una Red de Petri Coloreada creada en CPN Tools [3] y luego pulsando el botón de traducir (o bien en el menú Archivo opción: Traducir) el programa responderá con la traducción de la Red de Petri Coloreada

en ASCII que se visualizará al momento en la ventana y con un fichero con el mismo nombre que el fichero XML pero con la extensión txt. (a⁷) (a⁸)

- Abriendo un fichero ASCII desde el menú Archivo con la opción Abrir, pero esta vez escogiendo un fichero *.txt que ya contenga una traducción de una Red de Petri Colorada calculada previamente. (figura 27)

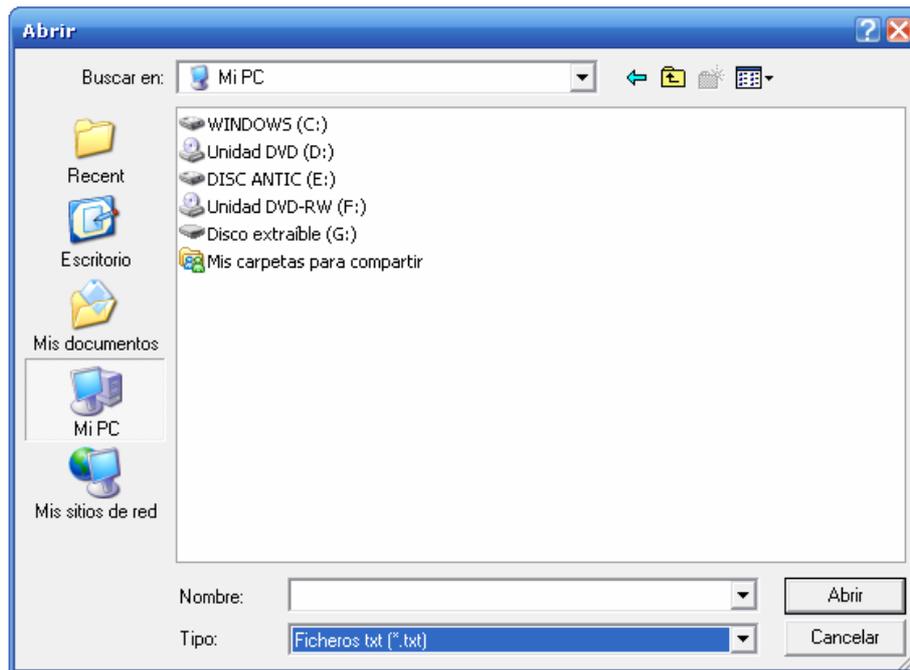


Figura 27. Ventana Abrir con la selección de ficheros txt.

(a⁷) **Aviso:** El lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC, no permite espacios es blanco en el nombre del fichero, por lo que si el nombre del fichero XML contiene espacios, el fichero con la traducción cambiará espacios por guión bajo. (p. ej: Si el fichero lleva por nombre → “Cinta transporte.cpn” su traducción se almacenará en un fichero con nombre → “Cinta_transporte.txt”)

(a⁸) **Aviso:** Se aconseja cambiar el nombre del fichero XML si este contiene espacios en blanco y guión bajo, por un nombre que no contenga ambos caracteres a la vez, ya

que el traductor no podrá encontrar el fichero en el sistema de ficheros por lo explicado en el aviso anterior.

5.5 Cambiar la vista XML a ASCII y de ASCII a XML

Cuando el programa traduce una Red de Petri Coloreada en XML automáticamente cambia la vista a la traducción realizada en ASCII. Una vez traducida una Red de Petri Coloreada se podrá volver a la vista XML pulsando el botón cambio de vista XML/ASCII. Por otra parte si el usuario se encuentra en la vista XML pero de una Red de Petri Coloreada previamente traducida, pulsando el botón de cambio de vista pasará a la vista ASCII, ya que esta vista abre el último fichero traducido de la Red de Petri Coloreada.

5.6 Aspectos de la traducción

Se aconseja hacer la traducción de una Red de Petri Coloreada solo si ha pasado el test ML de la herramienta CPN Tools, es decir que la Red no contenga ningún subrayado en rojo o naranja en su estructura. Esto solo se puede comprobar abriendo la Red de Petri Coloreada con la herramienta de CPN Tools y esperar la verificación ML automática. Para saber si una Red de Petri Coloreada diseñada en CPN Tools ha pasado el test ML bastará ver si no tiene resaltado en naranja o rojo ningún elemento del sistema.

La herramienta de diseño RPC CPN Tools permite nombrar elementos con espacios y guión bajo, pero el lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC, no permite espacios en blanco ni guión bajo en el nombre de sus elementos, por lo que el traductor cambiará los nombres de los elementos eliminando dichos caracteres. (p. ej: Si en el fichero XML existe un nombre de conjunto color → “colset P_1 = int;” será traducido como → “CO P1:int;”)

Las funciones asociadas a los arcos, las funciones guardas y temporales asociadas a las transiciones son representadas sin nombre en la herramienta de diseño CPN Tools, pero no será así en la traducción:

- Las funciones de arco serán nombradas con la letra F y seguidas de un número: F1, F2, F3, etc.
- Los guardas serán nombrados con la letra G y seguidas de un número: G1, G2, G3, etc.
- Las funciones de tiempo asociadas a una transición serán nombradas por las letras FT y seguidas de un número: FT1, FT2, FT3, etc.

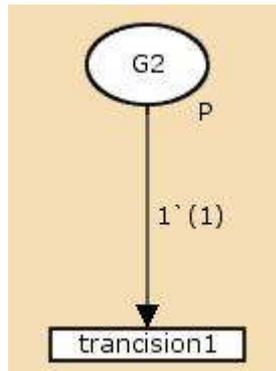
Los elementos número de niveles del árbol y número de subárboles no existen conceptualmente en las RPC diseñadas en CPN Tools por lo que las traducciones contendrán estos elementos con el valor por defecto 1000, es decir a la traducción se añadirá en todos los casos estas líneas:

- NI 1000 → para representar los niveles del árbol de la RPC.
- SA 1000 → para representar el número de subárboles de la RPC.

Los elementos de banda de costes y banda de tiempo no existen conceptualmente en las RPC diseñadas en CPN Tools por lo que las traducciones contendrán estos elementos con el valor por defecto 0, es decir a la traducción se añadirá en todos los casos estas líneas:

- BC 0 → para representar el valor para la banda de costes.
- BT 0 → para representar el valor para la banda de tiempo.

Puesto que la herramienta de diseño de RPC CPN Tools es una herramienta gráfica las conexiones entre transiciones y nodos lugar representadas con arcos son visibles (valga la redundancia) y el lenguaje para la generación de ficheros de entrada a un simulador/optimizador de RPC es estrictamente descriptivo por lo que se han de añadir explícitamente las conexiones entre transiciones y nodos lugar. (p. ej: En la figura 28 se aprecia un Arco que conexiona un nodo lugar y una transición junto con la traducción del arco.)



```
% Transición transicion1
AR G2-transicion1:F1;
```

Figura 28. Representación de un nodo lugar, una transición y un arco que une estos elementos en la herramienta de diseño RPC CPN Tools. A la derecha la traducción del arco.

En CPN Tools no es posible establecer un estado final para un nodo lugar, por lo que en la traducción de un nodo lugar se establecerá como `*`(`**`) el estado final de cada nodo lugar definido en la RPC ya que el lenguaje del optimizador sí lo requiere.

Una vez que el usuario haya abierto un fichero XML y proceda a su traducción, el traductor comprobará si el fichero ha sufrido modificaciones desde el momento de su apertura, por lo que en caso de haber sido modificado por el usuario mostrará una ventana de información y preguntará al usuario si desea sobrescribir el contenido del fichero o si por el contrario el usuario no desea que estos cambios se reflejen en el sistema de ficheros de su S.O. (Figura 29)(a⁹) (a¹⁰)



Figura 29. Ventana requiriendo al usuario para la modificación o no del fichero.

(a⁹) Aviso: Si se modifica un fichero XML a través de este programa, el fichero puede dejar de ser reconocido por la herramienta de diseño de RPC CPN Tools, por lo que se recomienda tener una copia del fichero o no sobrescribirlo.

(a¹⁰) **Aviso:** Cuando el usuario traduzca una RPC diseñada en CPN Tools, el fichero que genera dicha traducción deberá ser único, es decir, si ya existe un fichero en el mismo directorio de igual nombre, éste será irremediablemente sobrescrito sin avisar al usuario.

Todas las traducciones realizadas por el traductor contendrán una cabecera en la que aparecerán:

- Nombre_fichero: Nombre que ha adoptado el fichero con la traducción realizada.
- Fecha: Fecha en la que se realizó la traducción.

Durante el transcurso de la traducción el usuario perderá momentáneamente el control sobre la ventana principal del traductor y el puntero del ratón mostrará el icono de espera: ⌚ (o similar).

5.7 Guardar un fichero

Siempre que lo desee el usuario, podrá guardar el fichero/vista activa en ese momento en el sistema de ficheros de su Sistema Operativo (S.O.). Tanto los ficheros XML como los ficheros ASCII tendrán dos maneras para ser almacenados:

- **Guardar:** Si el fichero había sido previamente creado, se sobrescribirá el contenido que se este visualizando en el momento de usar esta opción (Atención: Al sobrescribir un fichero no será posible recuperar el estado anterior del fichero). En caso de no existir el fichero, el S.O. pedirá que se escriba un nombre con su extensión para su almacenaje. Se aconseja no usar espacios en blanco en el nombre del fichero.
- **Guardar como:** Se podrá usar esta opción para duplicar el fichero que se esté visualizando con otro nombre. La opción Guardar como puede funcionar como

la opción Guardar si se escribe (o se selecciona) el nombre del fichero que se está visualizando. Se aconseja no usar espacios en blanco en el nombre del fichero. (Figura 30)

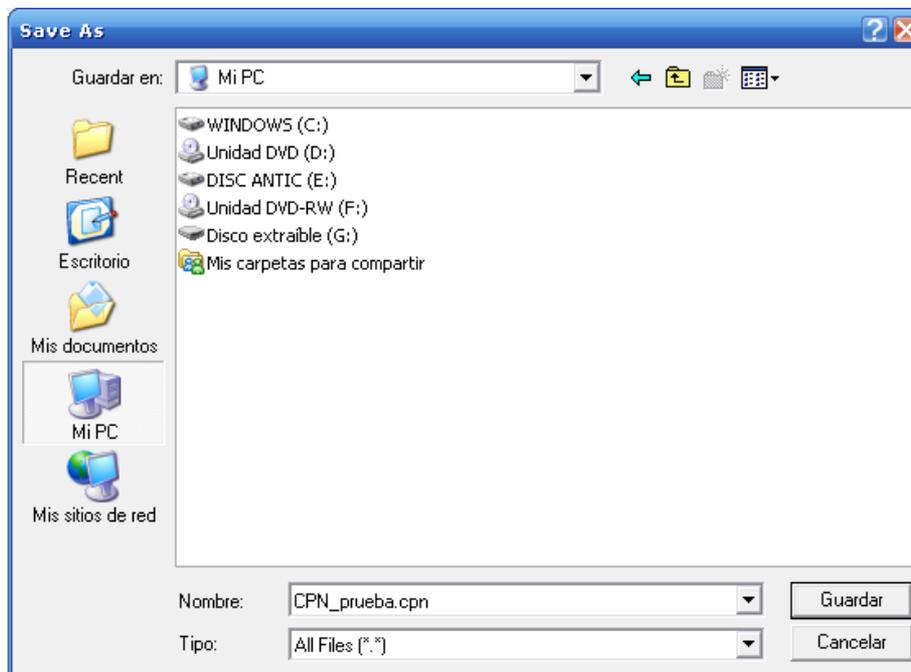


Figura 30. Ventana Guardar como con nombre de fichero para una RPC XML.

5.8 Configuración de la impresora

La ventana de configuración de la impresora se compone de (figura 31):

- **Menú desplegable:** Contendrá cada una de las impresoras de que dispone el usuario. Sirve para escoger la impresora con la que se desea plasmar en papel el documento que se esté visualizando.
- **Botón propiedades:** Este botón abrirá una ventana nueva donde se podrán configurar diferentes aspectos de la impresión. (p. ej: dirección vertical u horizontal de la impresión)

- **Imprimir a un archivo:** Esta opción hará que la impresión se haga en un archivo.
- **Intervalo de impresión:** En esta parte el usuario puede seleccionar que se imprima todo el documento, unas páginas consecutivas en concreto o bien una selección.
- **Copias:** El usuario puede seleccionar el número de copias que desea de la impresión y el orden intercalado o no de estas copias.

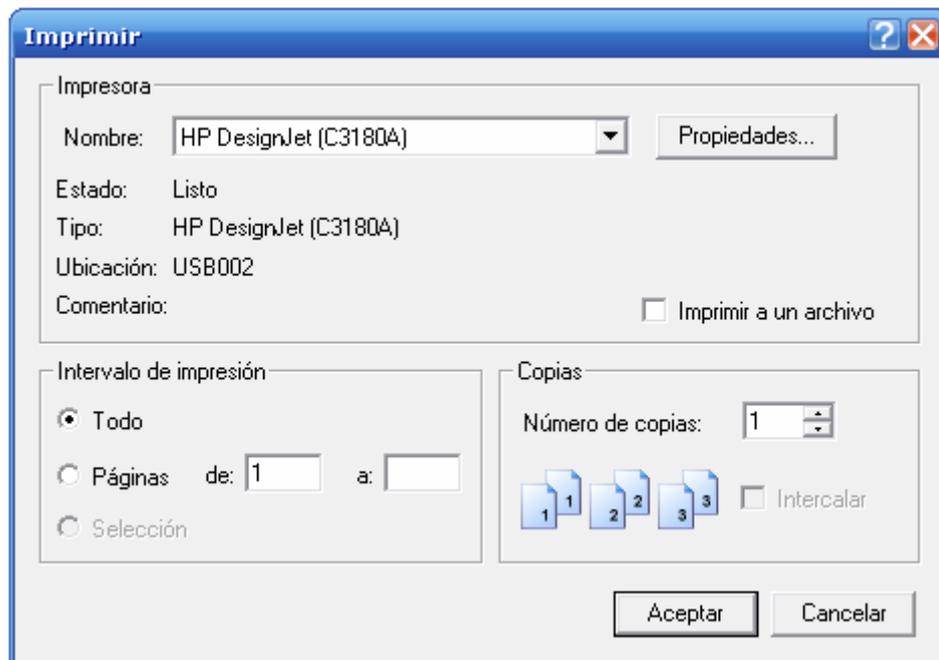


Figura 31. Ventana de configuración de la impresora.

5.9 Resolución de problemas

Problemas frecuentes

El traductor me avisa que he de volver a definir algún conjunto color

Utilice la herramienta de CPN Tools y reescriba el conjunto color, es decir borre y vuelva a escribir el color o los colores que no aparecen en la traducción. Esto suele ocurrir con las Standard declarations que utiliza CPN Tools por defecto.

El traductor me avisa que no se encuentran variables de algún tipo

Utilice la herramienta de CPN Tools y defina alguna variable del tipo que le indica en el aviso. Recuerde que para el buen funcionamiento de una RPC se ha de definir al menos una variable por cada conjunto color definido.

El traductor me avisa que hay una declaración que contiene errores pero no especifica cuales

Utilice la herramienta de CPN Tools, espere a que el evaluador ML recorra todo el sistema de la Red de Petri Coloreada que quiere traducir y CPN Tools le dará los detalles de dicho error. Una vez solucionado el error podrá volver a traducir su RPC sin que vuelva a surgir el error (siempre que dicha RPC no contenga algún otro error).

He modificado un fichero XML desde el traductor, ahora no puedo abrirlo desde CPN Tools y no tengo copia del fichero

- 1) Asegúrese de que no tiene alguna otra copia del fichero. Si no continúe.
- 2) Utilice alguna herramienta de edición de texto avanzada (p. ej: Notepad ++). Seleccione como lenguaje de visualización XML. Verifique que la estructura de tags sea la correcta. Si encontró algún tag mal definido reescríbalo. Si esto no solucionó su problema continúe.
- 3) Abra una nueva red en la herramienta CPN Tools y guárdela seguidamente. Con una herramienta de edición de texto abra esta red y a su vez abra también el fichero con el problema. Ahora copie un conjunto de tags de un fichero a otro, es decir todo el contenido entre tags de algún elemento. (p. ej: <color id="IDI100"> </color>). Una vez hecho una operación de este tipo intente abrir el fichero que originalmente estaba en blanco con la herramienta CPN Tools. Si no se puede abrir el fichero, revise todo el contenido de los tags copiados, en caso de que se pueda abrir vuelva a repetir la operación con el siguiente conjunto de tags hasta detectar el o los errores.

Quiero traducir una RPC que está en ASCII a XML

Este traductor no permite esta operación.

Algunas opciones del menú están en gris

Algunas opciones del menú pueden no ser utilizadas en ciertos momentos, antes se ha de cumplir alguna condición. (p. ej: Cuando seleccione algunos caracteres podrá usar las opciones Cortar o Copiar del menú Edición).

El traductor no almacena ningún archivo con la traducción que ha generado

Asegúrese que tenga acceso de lectura y escritura en el directorio donde se encuentra el fichero XML con la RPC generada por la herramienta de diseño CPN Tools.

He cargado un fichero XML pero el traductor me avisa que he de cargar un fichero XML para traducirlo

Este error puede ser ocasionado porque el fichero XML que intenta traducir no ha sido generado por la herramienta de diseño de RPC CPN Tools.

El traductor me avisa de que no encuentra un fichero, que sí está en mi sistema de ficheros

Asegúrese que el nombre del fichero que busca el traductor no contenga espacio y guión bajo simultáneamente.

El traductor me avisa de que una función contiene variables que no han sido declaradas

La herramienta de diseño CPN Tools permite definir funciones sin declarar las variables previamente, pero el lenguaje de especificaciones del fichero de entrada al simulador/optimizador no permite tener variables en una función que no estén definidas

en el sistema. Así pues abra el fichero con la herramienta de diseño CPN Tools para declarar las variables que use dicha función.

El traductor me avisa de que no se reconoce la estructura de origen

Este error puede ser ocasionado porque el fichero XML que intenta traducir no ha sido generado por la herramienta de diseño de RPC CPN Tools.

Capítulo 6

Ejemplo de diseño y traducción de una Red de Petri Coloreada

En este capítulo se muestra un ejemplo de uso del traductor que ha sido desarrollado. Se describe un sistema modela en CPN Tools [3] y su traducción correspondiente generada por el traductor.

6.1 Especificaciones del problema a modelar

El problema de programación de tareas en sistemas de producción Job Shop, conocido en la literatura como Job Shop Scheduling Problem, engloba una tipología de problemas NP-Hard que tiene la reputación de ser uno de los problemas de optimización combinatoria mas difíciles considerados hasta la fecha. Consiste en programar un conjunto de tareas (jobs) en un conjunto de máquinas, sujetas a la restricción de que cada máquina sólo puede procesar una tarea a la vez y cada tarea tiene un orden de procesamiento especificado en cada una de las máquinas. El objetivo es programar las tareas para reducir al mínimo el tiempo de finalización de todas las tareas (makespan).

Este caso de estudio presentado es un problema de programación de tareas para un sistema de producción Job Shop 3x3 (tres jobs y tres máquinas). El número de tareas que se deben llevar a cabo para cada completar cada job no es igual para todos los jobs, y además, algunas de las tareas de los jobs pueden ser realizadas en una de dos máquinas posibles. La tabla 3 muestra el orden y tiempo de procesamiento de las tareas de cada job en las diferentes máquinas.

Secuencia de Tareas	Job 1				Job 2				Job 3			
	Opción 1		Opción 2		Opción 1		Opción 2		Opción 1		Opción 2	
	Maq	Tpo										
1	1	7	2	8	1	6	2	5	1	8	3	5
2	2	4			2	4	3	2	2	2		
3	1	7	3	4	1	6			2	6	3	4
4					1	3	2	2	1	4	2	2
5									1	2	3	3

Tabla 3. Secuencia de tareas para cada Job

La tabla 4 muestra los diferentes colores que se definen para identificar las tareas, maquinas y tiempos del problema.

Color	Definición	Descripción
O	Int 1..5	Posición de secuencia de la próxima tarea que se debe procesar para un <i>job</i> .
J	Int 1..3	Identificador de <i>job</i>
M	Int 1..3	Máquina en la cual la tarea n° <i>O</i> del <i>job J</i> debe ser procesada.
T	Int 1..10	Unidades de tiempo requeridas para procesar la tarea n° <i>O</i> del <i>job J</i> en la máquina <i>M</i> .
Jo	Product O*J*M*T	Información que describe el estado de cada <i>job</i> .
J1	Int 1..3	Identificador de <i>job</i>
O1	Int 1..5	Posición de secuencia del <i>job J1</i> .
OR	Product O1*J1	Información que describe el número de tareas de cada <i>job</i> .
M1	Int 1..3	Identificador de máquina.

Tabla 4. Descripción de colores del modelo RPC del sistema Job Shop.

La tabla 5 muestra los diferentes nodos lugar que componen la RPC.

Lugar	Color	Descripción
Job	Jo	Las marcas almacenadas en el nodo lugar <i>Job</i> representan la información que describe el estado de cada <i>job</i> en el sistema de producción: identificador del <i>job</i> , máquina en la cual la próxima tarea debe ser ejecutada, tiempo requerido por cada operación y el orden de la tarea en la secuencia de operaciones para finalizar el <i>job</i> .
Maq	M1	Las marcas almacenadas en el nodo lugar <i>Maq</i> representan la identificación de la máquina donde se han de realizar las tareas.
Ord	Or	Las marcas almacenadas en el nodo lugar <i>Ord</i> representa la información asociada con el número de tareas de se deben realizar para finalizar cada <i>job</i> .

Tabla 5. Descripción de nodos lugar del model RPC del sistema Job Shop.

Finalmente la tabla 6 muestra la transición de esta RPC.

Transición	Descripción
T ₁	Ejecución de una tarea de un <i>job</i> en una máquina.

Tabla 6. Descripción de nodos transición del modelo RPC del sistema Job Shop.

La figura 32 muestra el modelo RPC creado con CPN Tools, donde:

Los tokens que tiene el nodo lugar Job indican la secuencia de operaciones, la máquina donde se realizará la operación y los tiempos de cada una de ellas. Los tokens que tiene el nodo lugar OR representa el número de operaciones que se deben realizar para cada job. Finalmente los tokens del nodo lugar Maq indican el número de máquinas de este sistema Job Shop. La función F2 indica las condiciones que se deben cumplir para llevar a cabo una operación en una máquina y el resultado de la operación. El guarda que se observa junto a T1 verifica que la operación, la máquina y la secuencia sea la correcta.

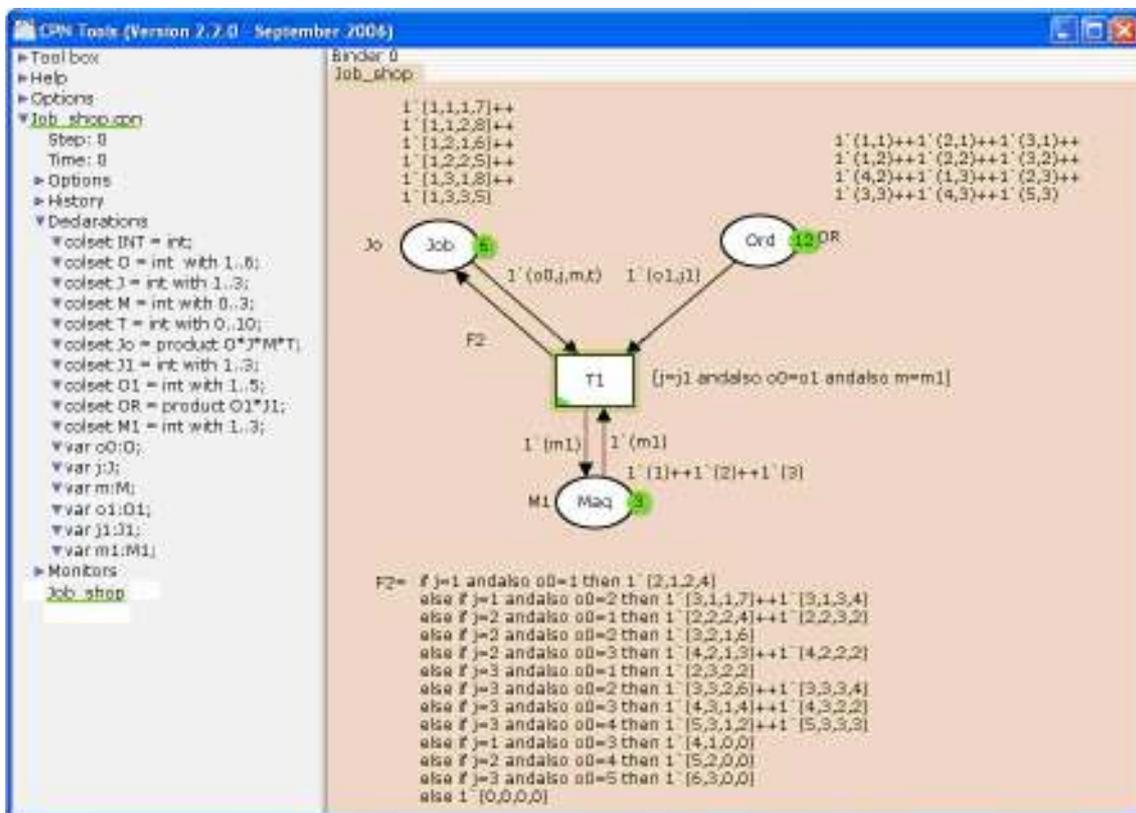


Figura 32. RPC del sistema Job Shop modelado con CPN Tools.

6.2 Pasos para la realización de una traducción

Crear una RPC con la herramienta CPN Tools

La herramienta CPN Tools es bastante intuitiva, los elementos principales de una RPC tienen un icono propio: nodos lugar, transiciones y arcos. Las variables y los conjuntos

color se declararán en la parte izquierda de la ventana. Una vez completada la RPC se hará un clic con el botón secundario del ratón en la parte izquierda y se usará la opción guardar como, se le dará una ubicación y un nombre al fichero cpn. Este fichero podrá ser abierto desde un editor de texto y podrá observar su formato XML.

Abrir el fichero con la RPC en XML

El primer paso para llevar a cabo la traducción de una RPC creada con la herramienta gráfica CPN Tools es abrir el fichero que contiene la RPC. Para ello se deberá hacer clic en el Menú Archivo y usar la opción Abrir... como se muestra en la figura 33.

Se deberá navegar por las carpetas del sistema hasta encontrar el fichero *.cpn, se seleccionará y se pulsará el botón Abrir. Como se muestra en la figura 34.

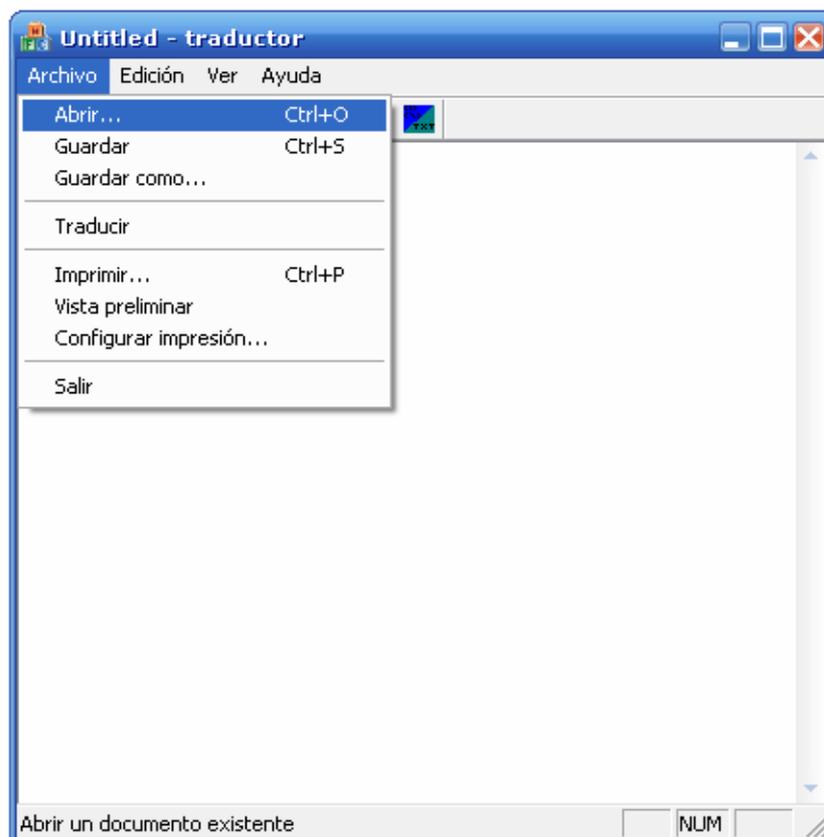


Figura 33. Detalle de la opción de menú Abrir...

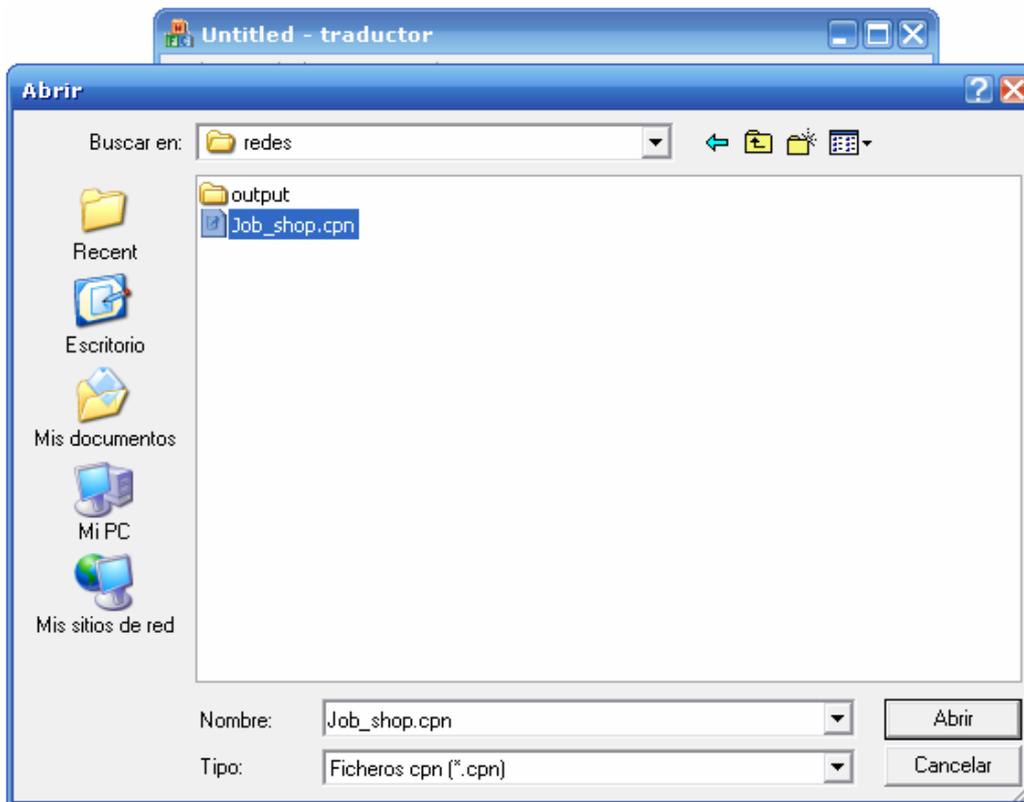


Figura 34. Detalle de la selección de un fichero *.cpn.

Realizar la traducción

Para efectuar la traducción se pulsará el botón con la “T” de color rojo, como puede verse en la figura 35.

Cuando el traductor termina la traducción de la RPC, se muestra el resultado de la RPC traducida como se observa en la figura 36. Si se desea se puede regresar a la RPC XML pulsando el botón , y pulsándolo de nuevo se volverá a ver la traducción realizada.

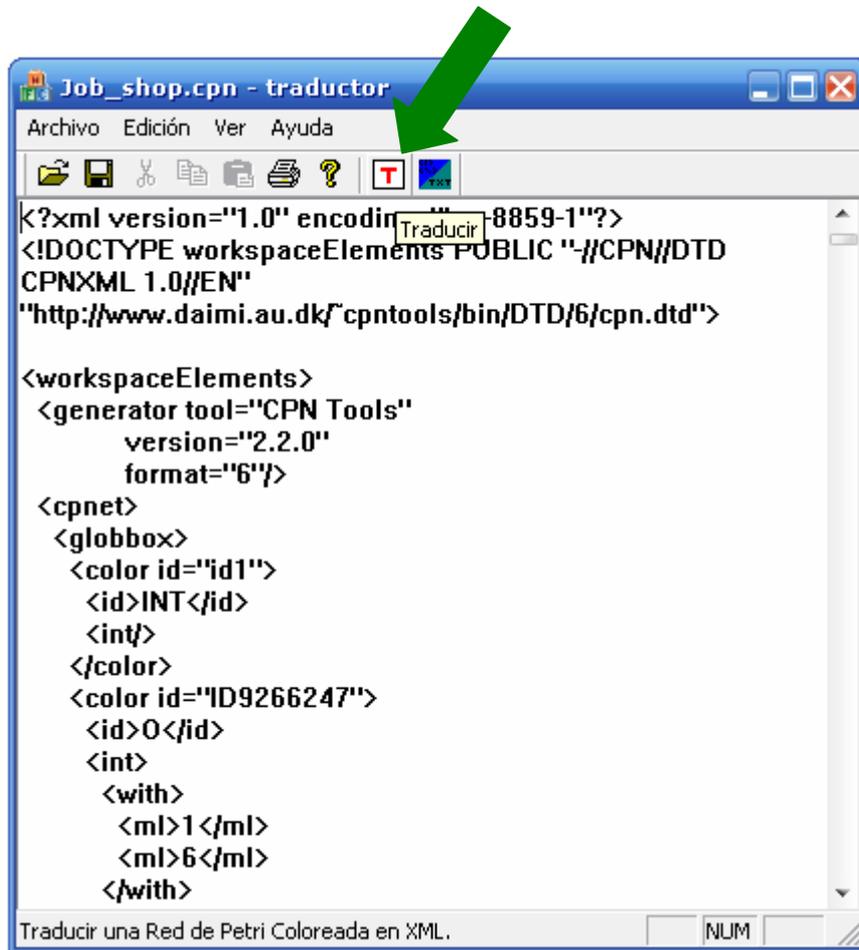


Figura 35. Detalle del botón para traducir.

6.3 Resultado obtenido de la traducción del sistema Job Shop

La figura 36 muestra el contenido del fichero generado por la traducción que está listo para su utilización como entrada del simulador/optimizador SATD.

```

%%%%%%%%%%
%
% Nombre_fichero:Job_shop.txt          FECHA: 20 - 7 - 2009
%
%%%%%%%%%%

%   Número de niveles del árbol

NI   1000

```

Figura 36.a Traducción de un sistema RPC Job Shop.

```

%   Número de Subárboles

SA  1000

%   Constantes

%   Bandas de coste y tiempo

BC  0

BT  0

%   Conjuntos color

CO  O: int with 1..6;
CO  J: int with 1..3;
CO  M: int with 0..3;
CO  T: int with 0..10;
CO  J1: int with 1..3;
CO  O1: int with 1..5;
CO  M1: int with 1..3;
CO  Jo: product O*J*M*T;
CO  OR: product O1*J1;

%   Variables

VA  o0:O;
VA  j:J;
VA  m:M;
VA  t:T;
VA  o1:O1;
VA  j1:J1;
VA  m1:M1;

%   Funciones expresiones de arco

FU  F1(o0:O,j:J): Jo= if j=1 & o0=1 then 1'(2,1,2,4)
    else if j=1 & o0=2 then 1'(3,1,1,7)+1'(3,1,3,4)
    else if j=2 & o0=1 then 1'(2,2,2,4)+1'(2,2,3,2)
    else if j=2 & o0=2 then 1'(3,2,1,6)
    else if j=2 & o0=3 then 1'(4,2,1,3)+1'(4,2,2,2)
    else if j=3 & o0=1 then 1'(2,3,2,2)
    else if j=3 & o0=2 then 1'(3,3,2,6)+1'(3,3,3,4)
    else if j=3 & o0=3 then 1'(4,3,1,4)+1'(4,3,2,2)
    else if j=3 & o0=4 then 1'(5,3,1,2)+1'(5,3,3,3)
    else if j=1 & o0=3 then 1'(4,1,0,0)
    else if j=2 & o0=4 then 1'(5,2,0,0)
    else if j=3 & o0=5 then 1'(6,3,0,0)
    else 1'(0,0,0,0);

FU  F2(o0:O,j:J,m:M,t:T): Jo= 1'(o0,j,m,t);

FU  F3(o1:O1,j1:J1): OR= 1'(o1,j1);

```

Figura 36.b Traducción de un sistema RPC Job Shop.

```

FU  F4(m1:M1): M1= 1'(m1);
FU  F5(m1:M1): M1= 1'(m1);

%  Guardas

FU  G1(o0:O,j:J,m:M,o1:O1,j1:J1,m1:M1): bool=(j=j1 & o0=o1 & m=m1);

%  Funciones de tiempo

FU  FT1(t:T): real= (t+1);

%  Funciones de costo

%  Lugares

LU  Maq:M1:1'(1)+1'(2)+1'(3):1'[0]+1'[0]+1'[0]::m,m1:*(**);

LU  Job:Jo:1'(1,1,1,7)+1'(1,1,2,8)+1'(1,2,1,6)+1'(1,2,2,5)+1'(1,3,1,8)+1'(1,3,3,5):
1'[0]+1'[0]+1'[0]+1'[0]+1'[0]+1'[0]::o0,j,m,t:*(**);

LU  Ord:OR:1'(1,1)+1'(2,1)+1'(3,1)+1'(1,2)+1'(2,2)+1'(3,2)+1'(4,2)+1'(1,3)+
1'(2,3)+1'(3,3)+1'(4,3)+1'(5,3):1'[0]+1'[0]+1'[0]+1'[0]+1'[0]+1'[0]+1'[0]+1'[0]+
1'[0]+1'[0]+1'[0]::o1,j1:*(**);

%  Transiciones

TR  T1:G1(Maq,Ord,Job):FT1;

%  Arcos

%  Transición T1

AR  Job-T1: F2;

AR  Ord-T1: F3;

AR  Maq-T1: F4;

AR  T1-Job: F1;

AR  T1-Maq: F5;

```

Figura 36.c Traducción de un sistema RPC Job Shop.

Capítulo 7

Conclusiones

En este capítulo se describen los objetivos cumplidos, las ampliaciones que podría tener este proyecto y las conclusiones generales.

7.1 Objetivos cumplidos

Todos los objetivos que fueron planteados se han alcanzado:

- **Se ha desarrollado un traductor** para Redes de Petri Coloreadas, de fácil utilización e implementado en c++, que traduce un fichero con especificaciones de una RPC en formato XML [4] generado mediante la herramienta de simulación de RPC: CPN Tools [3].
- **El traductor genera** como resultado de un **fichero** de texto con las especificaciones de la RPC con la estructura y formato requerida por un lenguaje [6] específico para la entrada a un simulador/optimizador de RPC diferente a CPN Tools.
- Para la utilización del traductor se ha desarrollado una **interfaz gráfica** que permite seleccionar el fichero XML que contiene cualquier RPC generada por CPN Tools que se desee traducir, así como para especificar el nombre del fichero resultado de la traducción y mostrará el resultado de la traducción.

7.2 Posibles ampliaciones

- Una posible ampliación a este traductor sería adaptar el algoritmo para que aceptara ficheros generados por otras herramientas de diseño de RPC como: Design/CPN (Design/Colored Petri Nets), INA (Integrated Net Analyzer), etc.

- Integrar el traductor con el simulador/optimizador que consistiría en poder usar el traductor dentro del simulador/optimizador como si fuera una opción del SATD.

7.3 Conclusiones generales

El proyecto estaba planteado para un semestre, a la implementación del traductor se le asignó un mes de duración, pero el módulo de la interfaz gráfica duró mucho más de lo esperado por la falta de experiencia en la programación de interfaces en Windows, por lo que todo el proyecto tuvo una demora considerable. Finalmente la planificación quedó afectada de la siguiente manera:

Fase de análisis del sistema:

16/10/08 a 30/10/08 Evaluación y estudio de la herramienta de CPN Tools

30/10/08 a 13/11/08 Evaluación y estudio de los ficheros XML generados por CPN Tools

13/11/08 a 04/12/08 Diseño de varias redes de Petri Coloreadas en CPN Tools

04/12/08 a 11/12/08 Evaluación y análisis del lenguaje de especificaciones del fichero de entrada al simulador/optimizador.

Fase de diseño del sistema:

11/12/08 a 14/01/08 Diseño del sistema

Fase de implementación del sistema:

14/01/08 a 05/06/09 Implementación del sistema

Fase de pruebas y mejora del sistema:

05/06/09 a 19/06/09 Pruebas y modificaciones

19/06/09 a 26/06/09 Verificación del sistema

Fase de documentación:

26/06/09 a 24/07/09 Memoria

24/07/09 a 14/09/09 Modificaciones de la memoria

Anexo A

Descripción de clases

A.1 Clase nodo_lugar

Clase nodo_lugar
Descripción: La clase nodo_lugar contiene toda la información referente a los nodos lugar de la RPC, así como todos los métodos para su acceso y uso.

Atributo id
Descripción: Identificador alfanumérico único en el entorno de la RPC XML.

Atributo name
Descripción: Contiene el nombre del nodo lugar.

Atributo tipo
Descripción: Contiene el tipo del nodo lugar.

Atributo initmark
Descripción: Contiene el marcado inicial del nodo lugar.

Atributo variables
Descripción: Contiene las variables asociadas al nodo lugar.

Atributo traducción
Descripción: Contiene el texto de la traducción del nodo lugar.

Método SetId
Descripción: Almacena el valor del identificador del nodo lugar. Entrada: El valor en string. Salida: Void

Método SetName
<p>Descripción: Almacena el nombre del nodo lugar.</p> <p>Entrada: El nombre en string.</p> <p>Salida: Void</p>

Método SetTipo
<p>Descripción: Almacena el tipo del nodo lugar.</p> <p>Entrada: El tipo en string.</p> <p>Salida: Void</p>

Método SetInitmark
<p>Descripción: Almacena el marcado inicial del nodo lugar.</p> <p>Entrada: Las marcas en string.</p> <p>Salida: Void</p>

Método SetVariables
<p>Descripción: Almacena las variables asociadas al nodo lugar.</p> <p>Entrada: El nombre de las variables separadas por “,” en string.</p> <p>Salida: Void</p>

Método SetTraduccion
<p>Descripción: Almacena la traducción del nodo lugar.</p> <p>Entrada: La traducción del nodo lugar en string.</p> <p>Salida: Void</p>

Método GetId
<p>Descripción: Recupera el valor del identificador del nodo lugar.</p> <p>Entrada: Void</p> <p>Salida: El identificador del nodo lugar en string.</p>

Método GetName
<p>Descripción: Recupera el nombre del nodo lugar.</p> <p>Entrada: Void</p>

Salida: El nombre del nodo lugar en string.

Método GetTipo

Descripción: Recupera el tipo del nodo lugar.

Entrada: Void

Salida: El tipo del nodo lugar en string.

Método GetInitmark

Descripción: Recupera el marcado inicial del nodo lugar.

Entrada: Void

Salida: El marcado inicial en string.

Método GetVariables

Descripción: Recupera las variables asociadas al nodo lugar.

Entrada: Void

Salida: El nombre de las variables asociadas al nodo lugar separadas por “,” en string.

Método GetTraduccion

Descripción: Recupera la traducción del nodo lugar.

Entrada: Void

Salida: La traducción del nodo lugar en string.

Método tokens

Descripción: Crea un string para los tokens de coste a partir del marcado inicial.

Entrada: Marcado inicial en string.

Salida: Tokens de coste en string.

Método traducenodo

Descripción: Forma la traducción con las diferentes partes del nodo lugar.

Entrada: Void

Salida: Void

A.2 Clase transición

Clase transición

Descripción: La clase transición contiene toda la información referente a las transiciones de la RPC, así como todos los métodos para su acceso y uso.
--

Atributo id

Descripción: Identificador alfanumérico único en el entorno de la RPC XML.
--

Atributo name

Descripción: Contiene el nombre de la transición.

Atributo guarda

Descripción: Contiene el cuerpo de la función guarda de la transición.
--

Atributo time

Descripción: Contiene el cuerpo de la función de tiempo de la transición.

Atributo nombre_funcion_guarda

Descripción: Contiene el nombre de la función guarda asociada a la transición.
--

Atributo nombre_funcion_tiempo

Descripción: Contiene el nombre de la función tiempo asociada a la transición.
--

Atributo traducción

Descripción: Contiene el texto de la traducción de la transición.

Método SetId

Descripción: Almacena el valor del identificador de la transición.
--

Entrada: El valor en string.

Salida: Void

Método SetName

Descripción: Almacena el nombre de la transición.

Entrada: El nombre en string.

Salida: Void

Método SetGuarda

Descripción: Almacena el cuerpo de la función guarda de la transición.

Entrada: El cuerpo de la función guarda en string.

Salida: Void

Método SetTime

Descripción: Almacena el cuerpo de la función tiempo de la transición.

Entrada: El cuerpo de la función tiempo en string.

Salida: Void

Método SetNombre_funcion_guarda

Descripción: Almacena el nombre de la función guarda de la transición.

Entrada: El nombre de la función guarda en string.

Salida: Void

Método SetNombre_funcion_tiempo

Descripción: Almacena el nombre de la función tiempo de la transición.

Entrada: El nombre de la función tiempo en string.

Salida: Void

Método SetTraducción

Descripción: Almacena la traducción de la transición.

Entrada: La traducción del nodo lugar en string.

Salida: Void

Método GetId

Descripción: Recupera el valor del identificador de la transición.

Entrada: Void

Salida: El identificador de la transición en string.

Método GetName

Descripción: Recupera el nombre de la transición.

Entrada: Void

Salida: El nombre de la transición en string.

Método GetGuarda

Descripción: Recupera el cuerpo de la función guarda de la transición.

Entrada: Void

Salida: El cuerpo de la función guarda en string.

Método GetTime

Descripción: Recupera el cuerpo de la función tiempo de la transición.

Entrada: Void

Salida: El cuerpo de la función tiempo en string.

Método GetNombre_funcion_guarda

Descripción: Recupera el nombre de la función guarda asociada a la transición.

Entrada: Void

Salida: El nombre de la función guarda asociada a la transición.

Método GetNombre_funcion_tiempo

Descripción: Recupera el nombre de la función tiempo asociada a la transición.

Entrada: Void

Salida: El nombre de la función tiempo asociada a la transición.

Método GetTraducción

Descripción: Recupera la traducción de la transición.

Entrada: Void

Salida: La traducción de la transición en string.

Método traducetransicion

Descripción: Forma la traducción con las diferentes partes de la transición.

Entrada: Void

Salida: Void

A.3 Clase arco

Clase arco

Descripción: La clase arco contiene toda la información referente a los arcos de la RPC, así como todos los métodos para su acceso y uso.

Atributo id

Descripción: Identificador alfanumérico único en el entorno de la RPC XML.

Atributo name

Descripción: Contiene el cuerpo del arco.

Atributo orientation

Descripción: Contiene la orientación del arco. "TtoP" si va de transición a nodo lugar o bien "PtoT" si va de nodo lugar a transición.

Atributo trans_id

Descripción: Contiene el identificador la transición a la que está asociado el arco.

Atributo place_id

Descripción: Contiene el identificador del nodo lugar al que está asociado el arco.

Atributo nombre_nodo_lugar

Descripción: Contiene el nombre del nodo lugar al que está asociado el arco.

Atributo nombre_funcion

Descripción: Contiene el nombre de la función que representa al arco.

Atributo indice

Descripción: Contiene un identificador numérico de arco.

Atributo arco_apunta_transición

Descripción: Contiene un 1 si es un arco de salida de un nodo lugar.

Método SetId

Descripción: Almacena el valor del identificador del arco.

Entrada: El valor en string.

Salida: Void

Método SetName

Descripción: Almacena el cuerpo del arco.

Entrada: El cuerpo del arco en string.

Salida: Void

Método SetOrientacion

Descripción: Almacena la orientación del arco.

Entrada: La orientación del arco en string.

Salida: Void

Método SetTrans_id

Descripción: Almacena el identificador de la transición asociada al arco.

Entrada: El identificador de la transición asociada al arco en string.

Salida: Void

Método SetPlace_id

Descripción: Almacena el identificador del nodo lugar asociado al arco.

Entrada: El identificador del nodo lugar asociado al arco en string.

Salida: Void

Método SetNombre_funcion

Descripción: Almacena el nombre de la función que representa arco.

Entrada: El nombre de la función que representa arco en string.

Salida: Void

Método SetNombre_transicion

Descripción: Almacena el nombre de la transición asociada al arco.

Entrada: El nombre de la transición asociada al arco en string.

Salida: Void

Método SetArco_apunta_transicion

Descripción: Almacena un 1 si es un arco de salida de un nodo lugar.

Entrada: 1 en caso de si es un arco de salida de un nodo lugar o 0 si es un arco de entrada a un nodo lugar.

Salida: Void

Método SetIndice

Descripción: Almacena el identificador numérico del arco.

Entrada: El identificador numérico del arco en int.

Salida: Void

Método GetId

Descripción: Recupera el valor del identificador del arco.

Entrada: Void

Salida: El identificador de la transición en string.

Método GetName

Descripción: Recupera el cuerpo del arco.

Entrada: Void

Salida: El cuerpo del arco en string.

Método GetOrientacion

Descripción: Recupera la orientación del arco.

Entrada: Void

Salida: La orientación del arco en string.

Método GetTrans_id

Descripción: Recupera el identificador de la transición asociada al arco.

Entrada: Void

Salida: El identificador de la transición asociada al arco en string.

Método GetPlace_id

Descripción: Recupera el identificador del nodo lugar asociado al arco.

Entrada: Void

Salida: El identificador del nodo lugar asociado al arco en string.

Método GetNombre_funcion

Descripción: Recupera el nombre de la función que representa al arco.

Entrada: Void

Salida: El nombre de la función que representa al arco en string.

Método GetNombre_transicion

Descripción: Recupera el nombre de la transición asociada al arco.

Entrada: Void

Salida: El nombre de la transición asociada al arco en string.

Método GetNombre_nodo_lugar

Descripción: Recupera el nombre del nodo lugar asociado al arco.

Entrada: Void

Salida: El nombre del nodo lugar asociado al arco en string.

Método GetIndice

Descripción: Recupera el índice numérico del arco.

Entrada: Void

Salida: El índice numérico del arco en int.

Método GetArco_apunta_transicion
Descripción: Recupera el valor numérico del atributo arco_apunta_transicion del arco. Entrada: Void Salida: El valor numérico del atributo arco_apunta_transicion del arco.

Método pretraduccion
Descripción: Establece el valor del atributo arco_apunta_transicion a partir de la orientación del arco. Entrada: Void Salida: Void

Método traducearco
Descripción: Forma la traducción con las diferentes partes del arco. Entrada: Void Salida: Void

A.4 Clase color

Clase color
Descripción: La clase color contiene toda la información referente a los conjuntos color de la RPC, así como todos los métodos para su acceso y uso.

Atributo id
Descripción: Identificador alfanumérico único en el entorno de la RPC XML.

Atributo name
Descripción: Contiene la definición del conjunto color.

Atributo tipo_lista
Descripción: Contiene los tipos asociados al conjunto color en una lista de strings.

Atributo indice

Descripción: Contiene el identificador numérico del conjunto color.

Método SetId

Descripción: Almacena el valor del identificador del conjunto color.

Entrada: El valor en string.

Salida: Void

Método SetName

Descripción: Almacena la definición del conjunto color.

Entrada: La definición del conjunto color en string.

Salida: Void

Método SetTipo_lista

Descripción: Almacena la lista de tipos asociados al conjunto color en el atributo tipo_lista.

Entrada: Los tipos asociado al conjunto color en una lista de string.

Salida: Void

Método SetIndice

Descripción: Almacena el identificador numérico del conjunto color.

Entrada: El identificador numérico del conjunto color en int.

Salida: Void

Método GetId

Descripción: Recupera el valor del identificador del conjunto color.

Entrada: Void

Salida: El identificador del conjunto color en int.

Método GetName

Descripción: Recupera la definición del conjunto color.

Entrada: Void

Salida: La definición del conjunto color en string.

Método GetTipo_lista_size
<p>Descripción: Recupera el número de tipos asociados al conjunto color.</p> <p>Entrada: Void</p> <p>Salida: El número de tipos asociados al conjunto color en int.</p>

Método GetPop_tipo
<p>Descripción: Recupera el primer tipo de la Tipo_lista del conjunto color y lo mueve al final de la lista.</p> <p>Entrada: Void</p> <p>Salida: Un tipo asociado al conjunto color en string.</p>

Método GetTraducción
<p>Descripción: Recupera la traducción del conjunto color.</p> <p>Entrada: Void</p> <p>Salida: La traducción del conjunto color en string.</p>

Método traducecolor
<p>Descripción: Forma la traducción con las diferentes partes del conjunto color.</p> <p>Entrada: Void</p> <p>Salida: Void</p>

A.5 Clase función

Clase función
<p>Descripción: La clase función contiene toda la información referente a las funciones expresiones de arco, funciones guarda de transiciones, funciones de tiempo de transiciones y constantes de la RPC, así como todos los métodos para su acceso y uso.</p>

Atributo nombre
<p>Descripción: Contiene el nombre de la función o declaración de constante.</p>

Atributo tipo_retorno
Descripción: Contiene el tipo de retorno de la función.

Atributo argumentos
Descripción: Contiene los argumentos asociados a la función.

Atributo contenido
Descripción: Contiene el cuerpo que define de la función.

Atributo es_constante
Descripción: Contiene el valor 1 si es una constante, 0 en cualquier otro caso.

Atributo memoria
Descripción: Contiene un apuntador a la propia clase.

Método SetNombre
Descripción: Almacena el nombre de la función o constante. Entrada: El nombre de la función o constante en string. Salida: Void

Método SetTipo_retorno
Descripción: Almacena el tipo de retorno de la función. Entrada: El tipo de retorno de la función en string. Salida: Void

Método SetArgumentos
Descripción: Almacena los argumentos asociados a la función. Entrada: Los argumentos de la función en la forma variable: tipo separadas por “;” en string. Salida: Void

Método SetContenido
Descripción: Almacena el cuerpo de la función. Entrada: El cuerpo de la función guarda en string.

Salida: Void

Método SetConstante

Descripción: Almacena el valor 1 o 0 en el atributo es_constante.

Entrada: 1 si es constante 0 en cualquier otro caso.

Salida: Void

Método GetNombre

Descripción: Recupera el nombre de la función.

Entrada: Void

Salida: El nombre de la función en string.

Método GetTipo_retorno

Descripción: Recupera el tipo del retorno de la función.

Entrada: Void

Salida: El tipo de la función en string.

Método GetArgumentos

Descripción: Recupera los argumentos de la función.

Entrada: Void

Salida: Los argumentos de la función en string.

Método GetContenido

Descripción: Recupera el contenido de la función.

Entrada: Void

Salida: El cuerpo de la función en string.

Método GetEsConstante

Descripción: Recupera el valor del atributo es_constante.

Entrada: Void

Salida: El valor del atributo es_constante en int.

Método Eliminafuncion

Descripción: Elimina todos los punteros asociados a la clase función.

Entrada: Void

Salida: Void

A.6 Clase variable

Clase variable

Descripción: La clase variable contiene toda la información referente a las variables de la RPC, así como todos los métodos para su acceso y uso.

Atributo id

Descripción: Identificador alfanumérico único en el entorno de la RPC XML.

Atributo name

Descripción: Contiene el nombre de la variable.

Atributo tipo

Descripción: Contiene el tipo de la variable.

Atributo var

Descripción: Contiene la declaración de la variable.

Atributo traducción

Descripción: Contiene la traducción de la variable.

Atributo conjunto_color

Descripción: Contiene el nombre del conjunto color al que está asociada.

Atributo indice

Descripción: Contiene el índice numérico de la variable.

Atributo no_traducir

Descripción: Contiene el valor 1 si es una variable declarada en una línea de múltiple declaración de variables.

Método SetId

Descripción: Almacena el valor del identificador de la variable.

Entrada: El valor en string.

Salida: Void

Método SetName

Descripción: Almacena el nombre de la variable.

Entrada: El nombre en string.

Salida: Void

Método SetTipo

Descripción: Almacena el tipo de la variable.

Entrada: El tipo de la variable en string.

Salida: Void

Método SetVar

Descripción: Almacena la declaración de la variable.

Entrada: La declaración de la variable en string.

Salida: Void

Método SetTraducción

Descripción: Almacena la traducción de la variable.

Entrada: La traducción de la variable en string.

Salida: Void

Método SetConjunto_color

Descripción: Almacena el nombre del conjunto color al que está asociado la variable.

Entrada: El nombre del conjunto color en string.

Salida: Void

Método SetIndice
<p>Descripción: Almacena el índice numérico de la variable.</p> <p>Entrada: El índice de la variable en int.</p> <p>Salida: Void</p>
Método SetNo_traducir
<p>Descripción: Almacena el valor del atributo no_traducir.</p> <p>Entrada: 1 si se trata de una variable declarada en una línea de múltiple declaración de variables.</p> <p>Salida: Void.</p>
Método GetId
<p>Descripción: Recupera el identificador de la variable.</p> <p>Entrada: Void</p> <p>Salida: El identificador de la variable en string.</p>
Método GetName
<p>Descripción: Recupera el nombre de la variable.</p> <p>Entrada: Void</p> <p>Salida: El nombre de la variable en string.</p>
Método GetTipo
<p>Descripción: Recupera el tipo de la variable.</p> <p>Entrada: Void</p> <p>Salida: El tipo de la variable en string.</p>
Método GetVar
<p>Descripción: Recupera la declaración de la variable.</p> <p>Entrada: Void</p> <p>Salida: La declaración de la variable en string.</p>
Método GetTraduccion
<p>Descripción: Recupera traducción de la variable.</p>

Entrada: Void

Salida: La traducción de la variable en string.

Método GetConjunto_color

Descripción: Recupera el nombre del conjunto color asociado a la variable.

Entrada: Void

Salida: El nombre del conjunto color asociado a la variable en string.

Método traducevariable

Descripción: Forma la traducción con las diferentes partes de la variable.

Entrada: Void

Salida: Void

A.7 Clase red_petri

Clase red_petri

Descripción: La clase red_petri contiene toda la información de una RPC, así como todos los métodos para su acceso y uso.

Atributo lista_color

Descripción: Lista con todos los conjuntos color de la RPC.

Atributo lista_variable

Descripción: Lista con todas las variables de la RPC.

Atributo lista_nodo_lugar

Descripción: Lista con todos los nodos lugar de la RPC.

Atributo lista_arco

Descripción: Lista con todos los arcos de la RPC.

Atributo lista_transicion

Descripción: Lista con todas las transiciones de la RPC.

Atributo lista_funcion

Descripción: Lista con todas las funciones expresiones de arco de la RPC.

Atributo lista_funcion_tiempo

Descripción: Lista con todas las funciones de tiempo de la RPC.

Atributo lista_funcion_guarda

Descripción: Lista con todas las funciones guarda de la RPC.

Atributo lista_funcion_ml

Descripción: Lista con todas las funciones ML y constantes de la RPC.

Atributo errores

Descripción: Contiene los errores encontrados durante la traducción de la RPC.

Método SetLista_color

Descripción: Almacena la lista de los conjuntos color de la RPC.

Entrada: Los conjuntos color en una lista de strings.

Salida: Void

Método SetLista_variable

Descripción: Almacena la lista de las variables de la RPC.

Entrada: Las variables en una lista de strings.

Salida: Void

Método SetLista_nodo_lugar

Descripción: Almacena la lista de los nodos lugar de la RPC.

Entrada: Los nodos lugar en una lista de strings.

Salida: Void

Método SetLista_arco

Descripción: Almacena la lista de los arcos de la RPC.

Entrada: Los arcos en una lista de strings.

Salida: Void

Método SetLista_transicion

Descripción: Almacena la lista de las transiciones de la RPC.

Entrada: Las transiciones en una lista de strings.

Salida: Void

Método SetLista_funcion

Descripción: Almacena la lista de las funciones de la RPC.

Entrada: Las funciones en una lista de strings.

Salida: Void

Método SetLista_funcion_tiempo

Descripción: Almacena la lista de las funciones de tiempo de la RPC.

Entrada: Las funciones de tiempo en una lista de strings.

Salida: Void

Método SetLista_funcion_guarda

Descripción: Almacena la lista de las funciones guarda de la RPC.

Entrada: Las funciones guarda en una lista de strings.

Salida: Void

Método SetLista_funcion_ml

Descripción: Almacena la lista de las funciones ML y constantes de la RPC.

Entrada: Las funciones ML y constantes en una lista de strings.

Salida: Void

Método AddError

Descripción: Almacena los errores encontrados durante la traducción de la RPC.

Entrada: Descripción del error en string.

Salida: Void

Método GetLista_color

Descripción: Recupera la lista de los conjuntos color.

Entrada: Void

Salida: Los conjuntos color en una lista de strings.

Método GetLista_variable

Descripción: Recupera la lista de las variables.

Entrada: Void

Salida: Las variables en una lista de strings.

Método GetLista_nodo_lugar

Descripción: Recupera la lista de los nodos lugar.

Entrada: Void

Salida: Los nodos lugar en una lista de strings.

Método GetLista_arco

Descripción: Recupera la lista de los arcos.

Entrada: Void

Salida: Los arcos en una lista de strings.

Método GetLista_transicion

Descripción: Recupera la lista de las transiciones.

Entrada: Void

Salida: Las transiciones en una lista de strings.

Método GetLista_funcion

Descripción: Recupera la lista del las funciones expresiones de arco.

Entrada: Void

Salida: Las funciones expresiones de arco en una lista de strings.

Método GetLista_funcion_guarda

Descripción: Recupera la lista del las funciones guarda.

Entrada: Void

Salida: Las funciones guarda en una lista de strings.

Método GetLista_funcion_tiempo

Descripción: Recupera la lista del las funciones de tiempo.

Entrada: Void

Salida: Las funciones de tiempo en una lista de strings.

Método GetLista_funcion_ml

Descripción: Recupera la lista del las funciones ML y constantes.

Entrada: Void

Salida: Las funciones ML y constantes en una lista de strings.

Método GetErrores

Descripción: Recupera los errores encontrados durante la traducción.

Entrada: Void

Salida: La descripción de los errores separados por “:” en un string.

Anexo B

Sintaxis de la gramática generadora del lenguaje de especificación RPC.

B.1 Sintaxis de Especificación del Número de Sub-árboles

<especificación_de_subárboles> ::= SA <número_de_soluciones>

<número_de_soluciones> ::= <natural>

B.2 Sintaxis de Especificación del Número de Niveles

<especificación_de_niveles> ::= NI <número_de_niveles>

<número_de_niveles> ::= <natural>

<natural> ::= <dígito> { <dígito> | 0 }

<dígito> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

B.3 Sintaxis de Especificación de la Banda de Costo

<especificación_de_banda_de_costo> ::= BC <banda_de_costo>

<banda_de_costo> ::= <natural>

B.4 Sintaxis de Especificación de la Banda de Tiempo

<especificación_de_banda_de_tiempo> ::= BC <banda_de_tiempo>

<banda_de_tiempo> ::= <natural>

B.5 Sintaxis de Definición de Constantes

<definición_de_constantes> ::= CT <nombre_constante> = <valor_constante>;

<nombre_constante> ::= <identificador>

<identificador> ::= <letra> { <letra> | <dígito> | 0 | _ }

<letra> ::= a | b | c | ... | z | A | B | C | ... | Z
 <valor_constante> ::= <cadena_de_caracteres> | <número>
 <cadena_de_caracteres> ::= <caracter>{<caracter>}
 <caracter> ::= <letra> | <dígito> | 0 | <carácter_especial>
 <carácter_especial> ::= ¡ | ¨ | ª | ¿ | \$ | # | , | . | ; | : | * | _
 <número> ::= <entero> | <real>
 <entero> ::= [-]<entero_positivo>
 <entero_positivo> ::= <natural> | 0
 <real> ::= <entero>.<entero_positivo>

B.6 Sintaxis de Declaración de Conjuntos Color

<declaración_de_conjuntos_color> ::=
 CO <nombre_conjunto_color>:<descripción_conjunto_color>;

<nombre_conjunto_color> ::= <identificador>
 <descripción_conjunto_color> ::=

- int [with <entero>..<entero>] |
- real [with <real>..<real>] |
- string [with <caracter>..<caracter>] and <entero_positivo>..<entero_positivo> |
- bool [with (<valor_bool>,<valor_bool>)] |
- unit [with <valor_color>] |
- with <valor_color> { | <valor_color> } |
- index <identificador> with <entero_positivo>..<entero_positivo> |
- product <nombre_conj_color>*<nombre_conj_color>{*<nombre_conj_color>} |
- |
- record <nombre_registro>:<nombre_conj_color>
 {*<nombre_registro>:<nombre_conj_color>} |
- union <selector>:<nombre_conj_color>+<selector>:<nombre_conj_color>
 {+<selector>:<nombre_conj_color>} |
- list <nombre_conj_color> [with <número_elementos>] |
- subset <nombre_conj_color> <sub_conjunto> |

<valor_color> ::= <entero> | <real> | <cadena_de_caracteres> | <valor_bool> |

<valor_producto> | <valor_registro> | <valor_union> | <valor_lista> |
<valor_subconjunto>

<valor_bool> ::= true | false | si | no | cierto | falso | yes | cert | fals | 1 | 0

<valor_producto> ::= (<valor_color>, <valor_color> {, <valor_color>})

<valor_registro> ::= {<selector>=<valor_color> {, <selector>=<valor_color>}}

<valor_union> ::= <selector>(<valor_color>) {, <selector>(<valor_color>)} |
 <selector> <valor_color> {, <selector> <valor_color>}

<valor_lista> ::= [<valor_color> {, <valor_color>}]

B.7 Sintaxis de Declaración de Variables

<declaración_de_variables> ::= VA

 <nombre_variable> {, <nombre_variable>}:<tipo_variable>;

<nombre_variable> ::= <identificador>

<tipo_variable> ::= <nombre_conj_color>

B.8 Sintaxis de Definición de Funciones

<definición_de_funciones> ::= FU nombre_función ([<argumento> {, <argumento>}]):

<tipo_resultado> = <descripción_función>;

<nombre_función> ::= <identificador>

<argumento> ::= <nombre_variable>:<tipo_variable> | <NULO>:<NULO>

<tipo_resultado> ::= <nombre_conj_color>

<descripción_función> ::= if [(<expresión_lógica>)] then <resultado> {else
 <resultado>} | case <nombre_variable> of <valor_color> => <resultado>
 { | <valor_color> => <resultado>} | <guarda> | <función_de_costo> |
 <función_de_tiempo>

<expresión_lógica> ::= <expresión_relacional> <operador_lógico>

 <expresión_relacional> { <operador_lógico> <expresión_relacional> }

<operador_lógico> ::= & | |

<expresión_relacional> ::= [(] <operando> <operador_relacional> <operando> [)]

<operando> ::= <nombre_variable> | <entero> | <real> | cadena_de_caracteres |
<valor_bool>

<operador_relacional> ::= < > | = | <= | >= | <>

<resultado> ::= <coeficiente>'(<expresión_color> {,<expresión_color>})

<coeficiente> ::= <natural>

<expresión_color> ::= <operando_color> [<operador_color> <operando_color>]

<operando_color> ::= <nombre_variable> | <valor_color> | <nombre_constante> |
<función> | <expresión_lógica>

<operador_color> ::= + | - | * | /

<guarda> ::= (<expresión_lógica>)

<funcion_de_costo> ::= <nombre_constante>*<coef_función>'(<marca_patrón>)
{+<nombre_constante>*<coef_función>'(<marca_patrón>)}

<coef_función> ::= <constante> | <nombre_variable>

<marca_patrón> ::= <patrón> {,<patrón>}

<patrón> ::= <valor_color> | *

<funcion_de_tiempo> ::= (<expresión_matemática>) |

if <expresión_lógica> then (<expresión_matemática>) {else
(<expresión_matemática>)}

<expresión_matemática> ::=

[(] <operando_numérico> <operador_numérico> <operando_numérico> [)]

{ <operador_numérico> <operando_numérico> }

<operando_numérico> ::= <nombre_variable> | <entero> | <real>

<operador_numérico> ::= + | - | * | / | s

B.9 Sintaxis de Definición de Nodos Lugar

<definición_de_lugares> ::=

LU <nombre_lugar> : <nombre_conjunto_color> : <marcado_inicial> :

<tiempo_inicial>:[<costo>]:[<agrupación>]:<marcado_final>;

<nombre_lugar>::= <identificador>

<marcado_inicial>::= <coeficiente>'(<marca>){+<coeficiente>'(<marca>}}

<tiempo_inicial>::= <coeficiente>[<tiempo>]{+<coeficiente>[<tiempo>]}

<tiempo>::= <entero> | <real>

<costo>::= <nombre_función>

<agrupación>::= <nombre_variable>{,<nombre_variable>}

<marcado_final>::= <coeficiente_patrón>'(<marca_patrón>)

{+<coeficiente_patrón>'(<marca_patrón>}}

<coeficiente_patrón>::= <natural> | *

B.10 Sintaxis de Definición de Nodos transición

<definición_de_transiciones>::= TR <nombre_transición>{,<nombre_transición>}:

<nombre_guarda>:<tiempo_transición>;

<nombre_transición>::=<identificador>

<nombre_guarda>::= <nombre_función> | true

<tiempo_transición>::= <entero> | <real> | <nombre_función>

B.11 Sintaxis de Definición de Arcos

<definición_de_arcos>::= AR <arco>{,<arco>}:<expresión_de_arco>;

<arco>::= <arco_entrada> | <arco_salida>

<arco_entrada>::= <nombre_lugar>-<nombre_transición>

<arco_salida>::= <nombre_transición>-<nombre_lugar>

<expresión_arco>::= <nombre_función>

Bibliografía

- [1] Kurt Jensen (1997): “Coloured Petri Nets: Basic concepts, Analysis Methods and Practical Use” Springer Verlag.
- [2] Manuales C++ en www.cplusplus.com
- [3] Manuales de uso CPN Tools en wiki.daimi.au.dk/cpntools
- [4] Manuales XML en www.w3.org/XML
- [5] Martin fowler, kendall scott, "UML Gota a Gota", 1999.
- [6] Narciso Farias, M.E. (2007). Metodología para la Resolución de Problemas de Optimización Mediante la Exploración del Espacio de Estados Generado a Partir de Modelos de Redes de Petri Coloreadas. PhD diss., Universitat Autònoma de Barcelona, Barcelona, España.
- [7] Narciso, M.E. (2001). Entorno de Planificación de la Producción Basado en el Recorrido del Árbol de Cobertura de una Red de Petri Coloreada. Trabajo de Investigación de Tercer Ciclo. UAB, Barcelona, España.

Resumen

Este proyecto presenta el desarrollo de una aplicación que permite traducir Redes de Petri Coloreadas diseñadas en CPN Tools a un lenguaje para la generación de ficheros de entrada a un simulador/optimizador de Redes de Petri Coloreadas. De esta manera se podrán optimizar modelos creados en CPN Tools ya que esta herramienta no facilita la optimización. Todo el proyecto se ha realizado en C++.

Resum

Aquest projecte presenta el desenvolupament d'una aplicació que permet traduir Xarxes de Petri Acolorides dissenyades en CPN Tools a un llenguatge per a la generació de fitxers d'entrada a un simulador/optimizador de Xarxes de Petri Acolorides. D'aquesta manera es podran optimitzar models creats en CPN Tools ja que aquesta eina no facilita l'optimització. Tot el projecte s'ha realitzat en C++.

Abstract

This project presents the development of a software that allows translate Colored Petri Nets designed in CPN Tools. The output is a file for a language for the generation of input files to a simulator / optimizer Colored Petri Nets. This will allow optimize models created in CPN Tools. CPN Tools does not facilitate the optimization. The whole project has been realized in C++.