Especificación de Modelos de Simulación RDEVS: Diseño e Implementación de una Gramática Libre de Contexto

Clarisa Espertino[1], María Julia Blas[1,2], Silvio Gonnet[1,2]
[1]Universidad Tecnológica Nacional-Facultad Regional Santa Fe
[2]Instituto de Desarrollo y Diseño INGAR (UTN-CONICET)
cespertino@frsf.utn.edu.ar, mariajuliablas@santafe-conicet.gov.ar,
sgonnet@santafe-conicet.gov.ar

Resumen

Se presenta una gramática libre de contexto para la definición de procesos de enrutamiento como un caso particular de un modelo de red restringido. La gramática se basa en un metamodelo que define la semántica sobre los elementos sintácticos. Como soporte, se implementó un editor de texto como un plugin para Eclipse, con sugerencias de escritura, una herramienta para la creación de archivos y opciones de validación. Esta herramienta es el paso inicial de un complemento de software para Eclipse que permitirá obtener un modelo de simulación de eventos discretos sin tener conocimientos de programación, utilizando especificaciones escritas en lenguaje natural, que serán procesadas y validadas para garantizar la consistencia de dicho modelo.

1. Introducción

La teoría de redes es una técnica útil para modelar relaciones entre entidades [1]. Dicha representación ha sido ampliamente adoptada para estudios de modelado en diversos campos, como, por ejemplo, el dominio de las redes sociales [2]. En ingeniería de software, estudios han utilizado esta técnica para evaluar problemas de sistemas/software [3, 4, 5].

Cuando en un sistema de componentes que interactúan, la operación de un componente y el enrutamiento de sus salidas depende de qué está sucediendo a lo largo del proceso, la dinámica puede verse como un proceso de enrutamiento. Los procesos de enrutamiento exhiben una fuerte interdependencia dentro de los componentes que permiten modelar su estructura utilizando teoría de redes. Además, cuando los procesos de enrutamiento son estudiados como sistemas de eventos discretos, el formalismo Routed Discrete Event System Specification (RDEVS) [6] provee una base sólida para su modelado y simulación (M&S).

De acuerdo con lo definido en [7], si un proceso de enrutamiento es correctamente definido como un modelo de red restringido, luego los modelos RDEVS pueden ser obtenidos siguiendo un conjunto de reglas de traducción. Es decir, al realizar una interpretación de la definición de un modelo de red restringido, es posible obtener los modelos de simulación de eventos discretos que representan la estructura de dicha situación. En este trabajo proponemos la implementación de una gramática libre de contexto que permite la definición de procesos de enrutamiento a través de la edición en lenguaje natural de sus representaciones. Nuestro objetivo es *i)* ofrecer un entorno de software que facilite la definición de procesos de enrutamiento utilizando lenguaje natural y *ii)* ser capaces de obtener una definición de modelo que admita una generación de código Java para tales situaciones de enrutamiento, de forma que puedan ser ejecutadas en simuladores DEVS como modelos RDEVS. Para cumplir con esta finalidad, se presenta un editor de texto desarrollado como un plugin para Eclipse [8], que incluye una herramienta para la creación de archivos de especificación, sugerencias de escritura y la opción de validación necesaria para obtener una instancia de un modelo de red. El análisis sintáctico se logra a través la gramática definida, y el análisis semántico utiliza un metamodelo (en el que se basa la gramática) para determinar la correctitud de la especificación definida, teniendo en cuenta un conjunto de restricciones invariantes definidas en OCL.

El resto del trabajo se encuentra estructurado de la siguiente manera. La Sección 2 presenta conceptos teóricos relacionados con el uso de modelos de red restringidos para la definición de modelos de simulación RDEVS y las gramáticas libres de contexto como soporte a la definición textual de modelos de red restringidos. La Sección 3 describe la sintaxis implementada para la especificación de procesos de enrutamiento, incluyendo una descripción de su árbol de sintaxis y del editor de texto. La Sección 4 presenta la semántica utilizada en parte del proceso de validación de dichas especificaciones. La Sección 5 describe la herramienta propuesta para definir los procesos de enrutamiento junto con un ejemplo ilustrativo. Por último, la Sección 6 está dedicada a las conclusiones y trabajos futuros.

2. Fundamentos Teóricos

2.1. Procesos de Enrutamiento como Modelos de Redes Restringidos

Un proceso de enrutamiento puede ser definido como "un sistema de componentes que interactúan, donde la operación de un componente y el ruteo de sus salidas depende de lo que sucede a lo largo del proceso". Las interacciones entre componentes dependen de información local y externa. La parte local refiere a información interna de los componentes, mientras que la parte externa refiere a información derivada de la estructura del proceso.

La definición de un proceso de enrutamiento puede contener diferentes tipos de componentes. Cada tipo de componente opera independientemente. Esto significa que la operación interna de los componentes es definida de forma independiente a la estructura del proceso. Ya que el enrutamiento depende igualmente de la descripción operativa del componente y de la estructura del proceso, el componente puede decidir los destinos de las salidas. Luego, los componentes pueden tomar decisiones sobre ruteo, como *i)* alternar el ruteo de sus salidas para evitar congestiones, *ii)* bloquear el ruteo de sus salidas para que no ingresen a un sector de componentes predefinidos, y *iii)* acelerar/desacelerar el procesamiento de entradas (para producir salidas más rápidas/lentas) conociendo que los nodos descendientes están libres/ocupados.

El formalismo RDEVS ha sido definido en [6] como una extensión del formalismo DEVS [9] que facilita la definición de procesos de enrutamiento en modelos de simulación basados en eventos discretos. Este formalismo estructura tres tipos de modelos: *modelo esencial*, *modelo de ruteo* y *modelo de red*. El *modelo esencial* define el funcionamiento de un componente elemental. El *modelo de ruteo* agrupa la definición de un *modelo esencial* junto con una *política de ruteo*. En conjunto, ambos elementos permiten al *modelo de ruteo* ejecutar el comportamiento definido en el componente y, al mismo tiempo, efectuar el ruteo de los eventos entrantes/salientes. Finalmente, el *modelo de red* conecta un conjunto de *modelos de ruteo*.

Una posible forma de conceptualizar procesos de enrutamiento es por medio de modelos de red restringidos. La teoría de redes propone modelar un sistema como un conjunto de nodos que están conectados por enlaces [1]. Ambos elementos, nodos y enlaces, pueden tener diferentes significados. Luego, una red consiste en nodos conectados por un conjunto de enlaces. En base a estos elementos, es posible establecer una correspondencia de los elementos de teoría de redes con los elementos de RDEVS (Tabla 1). De acuerdo con esta correspondencia, es posible establecer que cualquier definición de un modelo de red que satisfaga las restricciones requeridas para modelar un proceso de ruteo puede tomarse como base para la obtención de modelos RDEVS. Es decir, sobre una definición de red en base a nodos y enlaces, las siguientes restricciones garantizan un proceso de enrutamiento: *i)* al menos un nodo debe identificarse como inicial, *ii)* al menos un nodo debe definirse como final, *iii)* los nodos no pueden estar aislados, *iv)* varios enlaces no pueden conectar los mismos nodos, y *v)* los auto-enlaces no están permitidos.

Tabla 1. Mapeo de elementos RDEVS a elementos de la Teoría de Redes (Conceptualización).

Elemento RDEVS	*Elemento Teoría de Redes*
Modelo Esencial	Nodo Comportamiento o funcionalidad.
Modelo de Ruteo	Nodo Estructura o ubicación en la red.
Modelo de Red	Red. Conjunto de nodos conectados todos contra todos*.
Política de ruteo (por nodo)	Enlaces Conjunto de enlaces entrantes/salientes del nodo.

* Los acoplamientos todos contra todos son requeridos por la definición del modelo de red. Esto permite que el ruteo se delegue al conjunto de funciones de ruteo.

2.2. Gramáticas Libre de Contexto

Al modelar un proceso de enrutamiento como una red, los nodos definen componentes, y los enlaces denotan interacciones entre ellos. Sin embargo, el modelo de red que representa un proceso de enrutamiento debe ser restringido para asegurar su correctitud (por ejemplo, componentes no pueden estar aislados, auto interacciones no están permitidas, etc). Este tipo de restricciones limitan el modelo de red original dando lugar a un modelo restringido que representa procesos de ruteo. Desde esta perspectiva, un proceso de enrutamiento puede ser visto como una instancia restringida de un modelo de red.

Una gramática libre de contexto es una forma de describir lenguajes mediante reglas recursivas llamadas producciones. Consta de un conjunto de variables, un conjunto de símbolos terminales y una variable inicial, así como de producciones. Cada producción consta de una variable de cabeza y un cuerpo formado por una cadena de cero o más variables y/o símbolos terminales [10]. En la Sección 3 se introduce la gramática libre de contexto RDEVSNL, basada en un modelo de red restringido, que permite definir la estructura de procesos de enrutamiento.

A su vez, un metamodelo es un modelo que define el lenguaje utilizado para diseñar un modelo [11]. Luego, un metamodelo de un proceso de enrutamiento permite instanciar modelos de procesos de enrutamiento. Los metamodelos son herramientas poderosas de modelado para asegurar la correctitud de la estructura de los modelos. En este contexto, el uso de la sintaxis propuesta en la Sección 3 para la definición de los modelos de red es semánticamente validado con un metamodelo definido para realizar una interpretación basada en las restricciones requeridas para obtener, en una etapa posterior, el conjunto de modelos RDEVS equivalentes. Dicho metamodelo se describe en la Sección 4.

Con el objetivo de clarificar el proceso que involucra a la herramienta en desarrollo y los artefactos que le brindan soporte, la Figura 1 presenta un esquema representativo. El modelador, haciendo uso del Editor de texto, realiza una especificación textual del modelo de red restringido que da soporte a su escenario de simulación. Esta descripción, es analizada y posteriormente traducida a código Java correspondiente a su representación RDEVS. Tal como se ha enunciado con anterioridad, en este trabajo se presentan los componentes resaltados en color como parte del proceso de definición de los modelos de simulación.

3. Sintaxis "RDEVSNL"

La gramática RDEVSNL permite abstraer (por medio de su sintaxis), la definición del proceso de enrutamiento en una representación textual basada en nodos y enlaces para describir su estructura. Para brindar mayor versatilidad, se definieron dos versiones de la sintaxis: inglés y español. Ambas fueron especificadas e implementadas utilizando ANTLR4 [12]. De esta forma, el modelador puede optar por el idioma de su preferencia para la definición del modelo.

Además, se implementó un editor de texto para que el modelador tenga la posibilidad de escribir su especificación dentro de Eclipse. Este editor se complementa con una herramienta que le permite crear archivos con una extensión determinada, le brinda ayuda durante la edición (como resaltado de sintaxis y sugerencias de escritura) teniendo en cuenta el idioma seleccionado y, por último, facilita la validación de la especificación final.

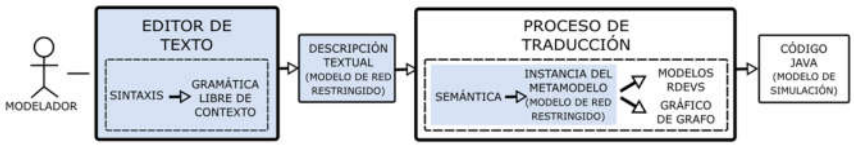
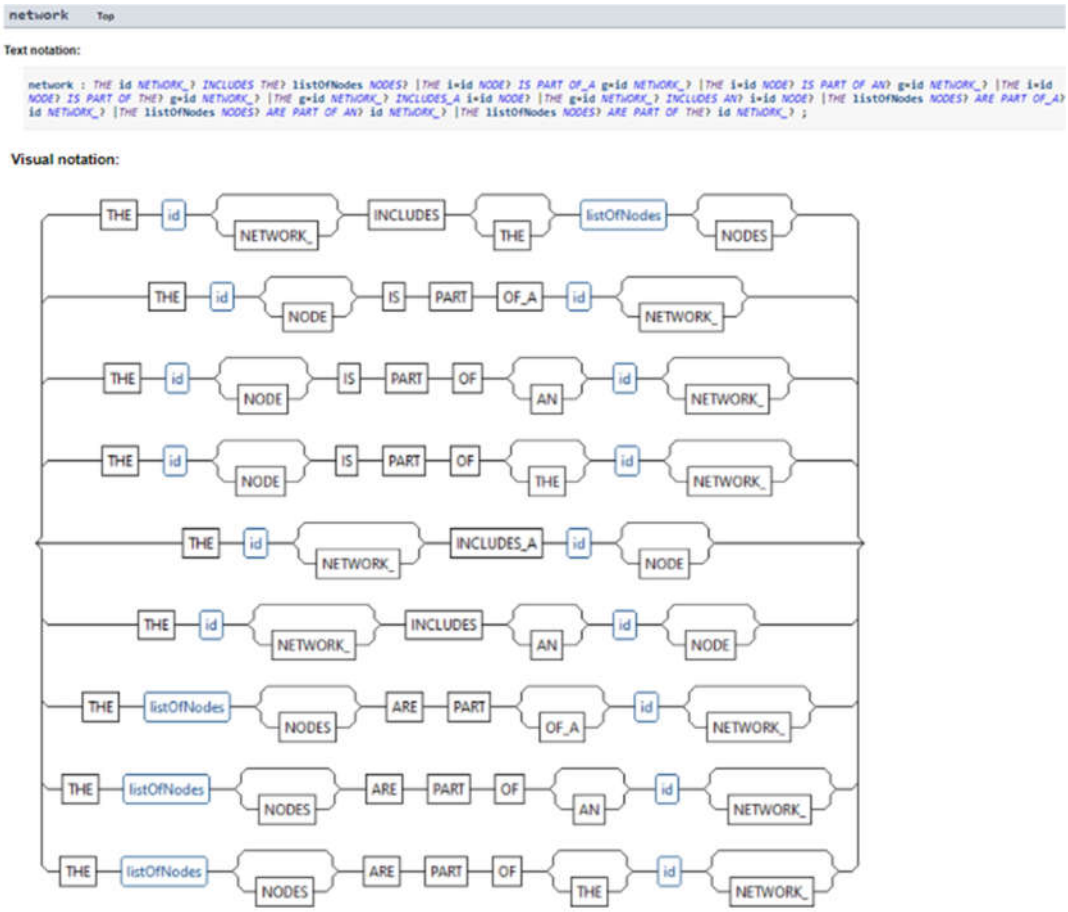
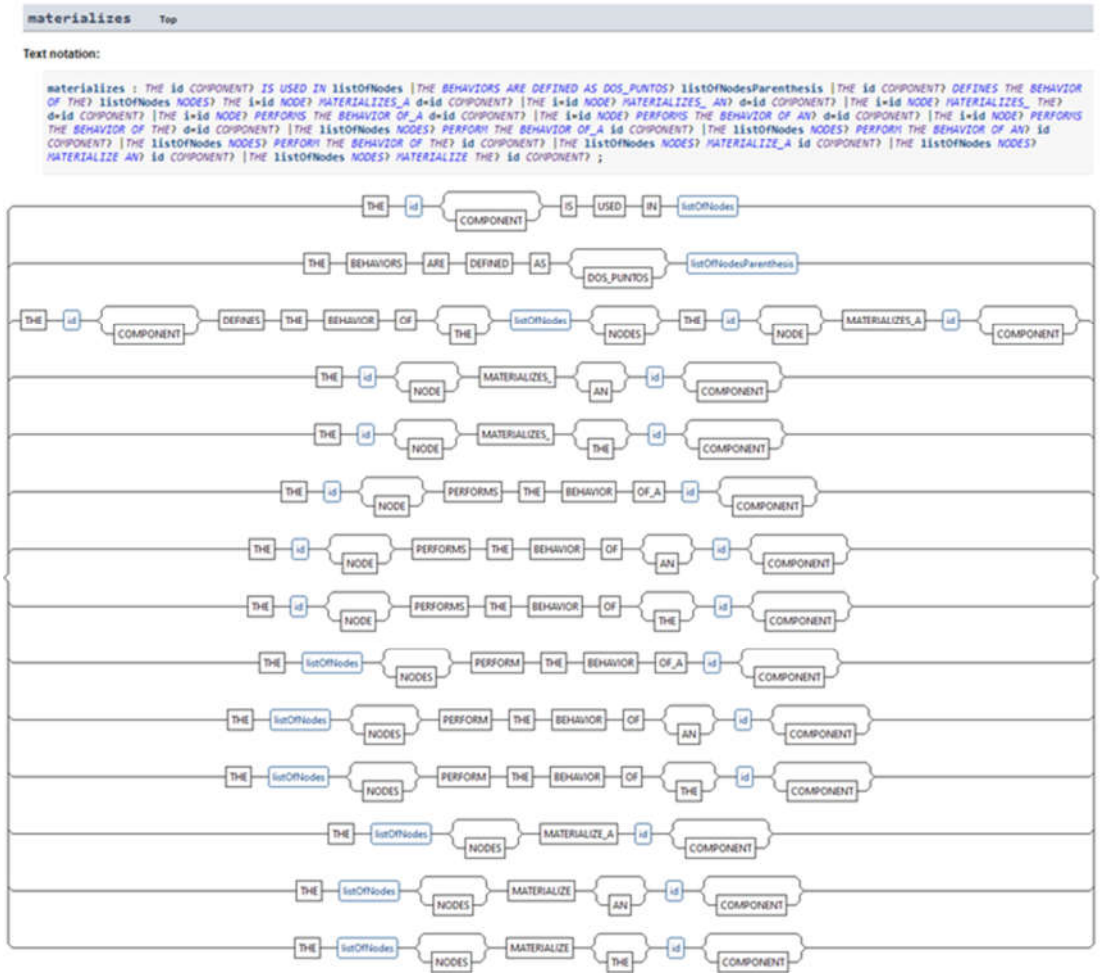


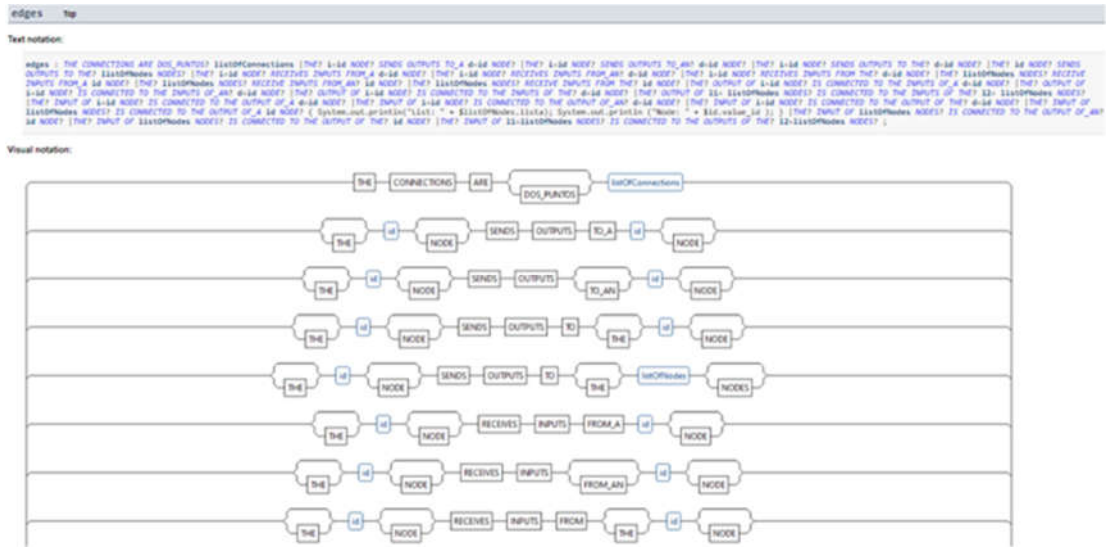
Figura 1. Esquema del proceso de definición de modelos de simulación a partir del uso de la gramática RDEVSNL.



(a)



(b)



(c)

Figura 2. Reglas de producción y diagrama de sintaxis de los tres principales símbolos no terminales: (a) network, (b) materializes y (c) edges.

3.1. Diagrama de Sintaxis (Definición en Inglés)

Los diagramas de sintaxis representan la estructura que aplica una gramática a las cadenas de su lenguaje [10]. En la especificación en inglés, se pueden identificar tres bloques de construcción primarios (o tres principales símbolos no terminales): *network, materializes* y *edges.* En la Figura 2 se pueden observar las reglas de producción y el diagrama de sintaxis de cada uno de ellos.

Cuando un modelo de red restringido es utilizado para estructurar un proceso de enrutamiento, una red es definida para modelar dicho proceso. El modelo es definido sobre un conjunto de nodos que denotan componentes. A su vez, incluye un conjunto de enlaces, que definen interacciones directas entre componentes.

Una red (n*etwork*, Figura 2(a)) se especifica utilizando un *id* y siempre incluye una lista de nodos, que puede ser expresada como una única especificación (es decir, detallando la lista completa de nodos como *lisfOfNodes*) o en múltiples líneas (donde cada nodo, con un *id*, es identificado como parte de *network*). En un proceso de enrutamiento, cada componente exhibe una operación interna, que es identificada como un componente (*component*) en la Figura 2(b), y define el comportamiento de un nodo o una lista de nodos. Los enlaces (*links*) definen interacciones directas entre nodos, que son especificadas en la Figura 2(c) como *edges.* La gramática permite su definición de múltiples formas, aunque por razones de espacio solo se presentan algunas de las estructuras admitidas. Por ejemplo, un nodo envía salidas a un nodo/una lista de nodos, un nodo recibe entradas desde un nodo/una lista de nodos, y la entrada de un nodo/una lista de nodos está conectada con la salida de un nodo/una lista de nodos, entre otras.

3.2. Editor de Texto

El editor de texto que permite definir modelos de red restringidos haciendo uso de la sintaxis se implementó como un plugin para Eclipse. En dicho editor, se cuenta con resaltado de sintaxis y sugerencias de escritura, para facilitar al modelador la edición de sus especificaciones.

Para lograr el resaltado de sintaxis, las palabras válidas se clasifican en: artículos, símbolos no terminales, acciones, comentarios e interacciones. La Tabla 2 muestra las principales palabras que se resaltan, para ambos idiomas, detallando para cada clasificación, su color. De esta forma, al escribir la especificación, el modelador puede identificar claramente los componentes de cada sentencia a través de sus colores. Además, puede incluir comentarios que no serán analizados en el proceso de validación de la especificación. Los mismos pueden ser definidos en una sola línea o, por el contrario, en múltiples líneas. Si desea escribir un comentario de una sola línea, el modelador deberá colocar al inicio los caracteres "//". En cambio, si se trata de un comentario de múltiples líneas, se deben incluir los caracteres "/*" en la línea que da comienzo y "*/" al finalizar.

Tabla 2. Palabras principales para ambos idiomas.

Tipo de Palabra	*Color*	*Palabra en Inglés*	*Palabra en Español*
Artículo	Violeta	the	el
		from	desde
		a	un
		to	a
		of	de
		as	como
Símbolo no terminal	Azul	network	red
		node	nodo
		component	componente
Comentario	Verde	//	
		/* … */	
Acción	Rojo	is part of	es parte de
		materializes	materializa
		includes	incluye
		sends	envía
		receives	recibe
		performs	ejecuta
Interacción	Magenta	inputs	entradas
		outputs	salidas

Con respecto a las sugerencias de escritura, las mismas pueden obtenerse presionando CTRL+SPACE. El modelador podrá elegir de a una palabra a la vez, haciendo clic sobre ella o navegando con las flechas de dirección del teclado, y luego presionando ENTER sobre la palabra deseada. Las palabras que el modelador visualiza y elige son almacenadas en archivos de configuración (existiendo un archivo para cada idioma disponible).

Tanto el resaltado de sintaxis como las sugerencias de escritura se corresponden con el idioma seleccionado por el modelador durante el proceso de creación del archivo que contiene su especificación. Cada vez que un archivo es creado, el idioma seleccionado por el modelador es almacenado junto con sus metadatos como un atributo definido por el usuario. Luego, cada vez que el archivo es abierto, se lee de sus metadatos el valor almacenado en el atributo idioma a fin de conocer su definición.

4. Semántica "Modelo de Red Restringido"

La Figura 3 presenta el metamodelo que conceptualiza procesos de enrutamiento, donde los estereotipos son

utilizados para indicar el componente de red al que refiere el elemento de ruteo.

Una especificación RDEVSNL (*NLSpecification*) incluye un proceso de enrutamiento (*Routing Process)*. Este concepto refiere al modelo de red (*Network Model*). Cuando un modelo de red restringido es utilizado para estructurar un proceso de enrutamiento, el modelo es definido sobre un conjunto de nodos (*Node*). A su vez, incluye un conjunto de interacciones (*Link*). Los nodos denotan componentes (*Component*) y los enlaces definen interacciones dirigidas entre componentes (*Interaction*). Dos componentes están vinculados a través de una interacción. Un componente actúa como fuente (*source*), desde donde se lleva a cabo la interacción, y el componente restante actúa como destino (*destination*), hacia donde la interacción se dirige.

El diagrama presentado en la Figura 3 ha sido restringido con restricciones OCL (específicamente, invariantes) para asegurar la definición del proceso de enrutamiento.

4.1. Metamodelo Ecore (EMF)

Para lograr la validación de la semántica de una definición RDEVSNL, una versión Ecore del modelo que se observa en la Figura 3 fue implementada. Para esto, se utilizó el proyecto EMF de Eclipse [13] (que consiste en una herramienta que provee un marco de trabajo para la definición de modelos, junto con un mecanismo de generación de código para la construcción de herramientas de software basadas en modelos de datos estructurados). La Tabla 3 muestra las restricciones OCL incluidas en dicho modelo. Las mismas tienen relación con las descripciones presentadas previamente.

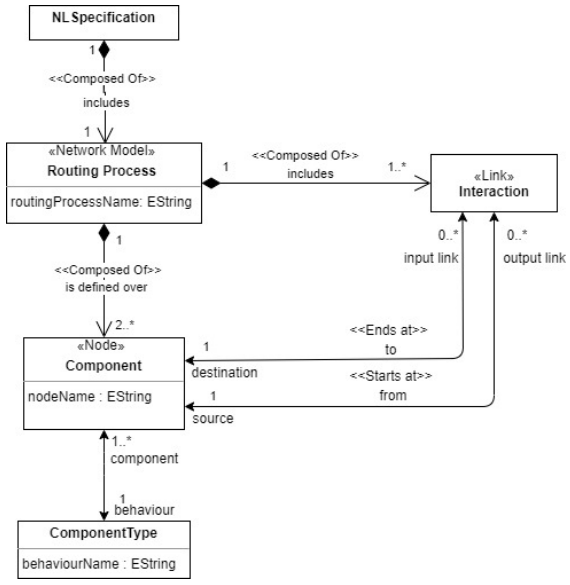


Figura 3. Metamodelo del dominio utilizado para validar la especificación a partir de la sintaxis RDEVSNL.

Tabla 3. Restricciones OCL incluidas en el diagrama de la Figura 3.

Id	*Restricción OCL*
i	context RoutingProcess invariant existsStartingComponent: self.component-> select(c|c.inputLink->size()=0 and c.outputLink-> size()>0)->size()>0
ii	context RoutingProcess invariant existsEndingComponent: self.component-> select(c|c.inputLink->size()>0 and c.outputLink-> size()=0)->size()>0
iii	context Component invariant notIsolated:(self.inputLink->size() + self.outputLink->size()) > 0
iv	context Component invariant multipleInteractions: self.outputLink-> forAll(e1,e2|e1<>e2 implies e1.destination <> e2.destination)
v	context Interaction invariant notSelfInteraction: self.source<>self.destination

5. Plugin de Eclipse para la Definición de Especificaciones en RDEVSNL

Un archivo que cuente con una especificación de RDEVSNL tendrá extensión ".rdevsnl". Se implementó una herramienta específica (wizard) para la plataforma Eclipse que indica los pasos a seguir para crear archivos con dicha extensión. A través de esta herramienta, el modelador puede dentro de un proyecto Java, asignar un nombre al archivo que contendrá su especificación RDEVSNL y, además, seleccionar el idioma con el que desea trabajar. Si no selecciona un idioma, por defecto trabajará en inglés. La Figura 4 muestra la pantalla de configuración.

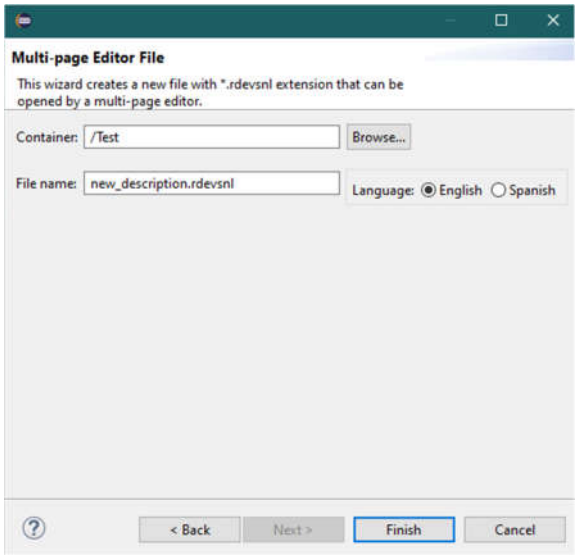


Figura 4. Wizard de creación de archivos con extensión ".rdevsnl".

La Figura 5 presenta el proceso de enrutamiento a definir. La red está conformada por 8 nodos (representados por máquinas) que se relacionan a través de interacciones dirigidas, y cada uno se corresponde con un tipo de máquina (es decir, ejecuta el comportamiento de un tipo de máquina). Luego, en la Figura 6 puede observarse su definición, que es editada por el modelador luego de finalizar la creación del archivo.

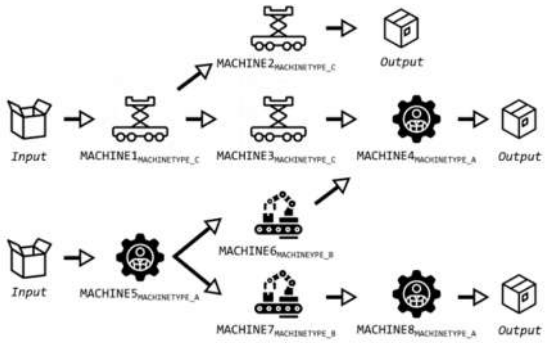


Figura 5. Ejemplo de proceso de enrutamiento.

5.1. Validación de la Especificación

El plugin implementado cuenta con la opción de validar el archivo de especificación, utilizando el analizador sintáctico (o parser) creado con ANTLR4. Cuando el modelador activa el proceso de validación (Figura 7), el análisis sintáctico de RDEVSNL es ejecutado sobre el contenido actual del archivo "*.rdevsnl", que fue previamente guardado en forma automática para asegurar que el análisis sintáctico se realiza sobre la versión más reciente del mismo. Luego, el parser trata de reconocer las estructuras de las sentencias a partir de un flujo de tokens, dado por el contenido actual de la especificación.

Si dicho análisis es exitoso (es decir, el parser reconoce como válidas todas las sentencias que el modelador introdujo en su especificación), utilizando los tokens identificados por el parser, se crea automáticamente una instancia del metamodelo Ecore (Figura 3). En esa instancia, cada elemento definido en la especificación es mapeado a un concepto o relación. Por ejemplo, la línea 3 de la Figura 6 es sintácticamente correcta. Por lo tanto, durante el proceso de validación, se creará una instancia del concepto *NL Specification* que contendrá los siguientes elementos: *i)* una instancia de *RoutingProcess* denominada "RoutingProcess", *ii)* ocho instancias de *Component* (cada una denominada "Machine1" hasta "Machine8), y *iii)* ocho relaciones *isDefinedOver* para vincular a cada *Component* con *RoutingProcess*. Este proceso se repite para todas las líneas válidas a fin de obtener una instancia del metamodelo a ser validada.

Por otro lado, si se identifican errores sintácticos en el archivo, serán mostrados al modelador, quien tendrá la posibilidad de corregir la descripción del proceso de enrutamiento y analizarlo nuevamente a fin de lograr generar la instancia del metamodelo. Haciendo referencia al ejemplo que se observa en la Figura 6, el análisis sintáctico arroja un error en la línea 13 (Figura 8). Esto es así porque la sentencia que identifica el parser como correcta, incluye la palabra "receives" en lugar de "receive".

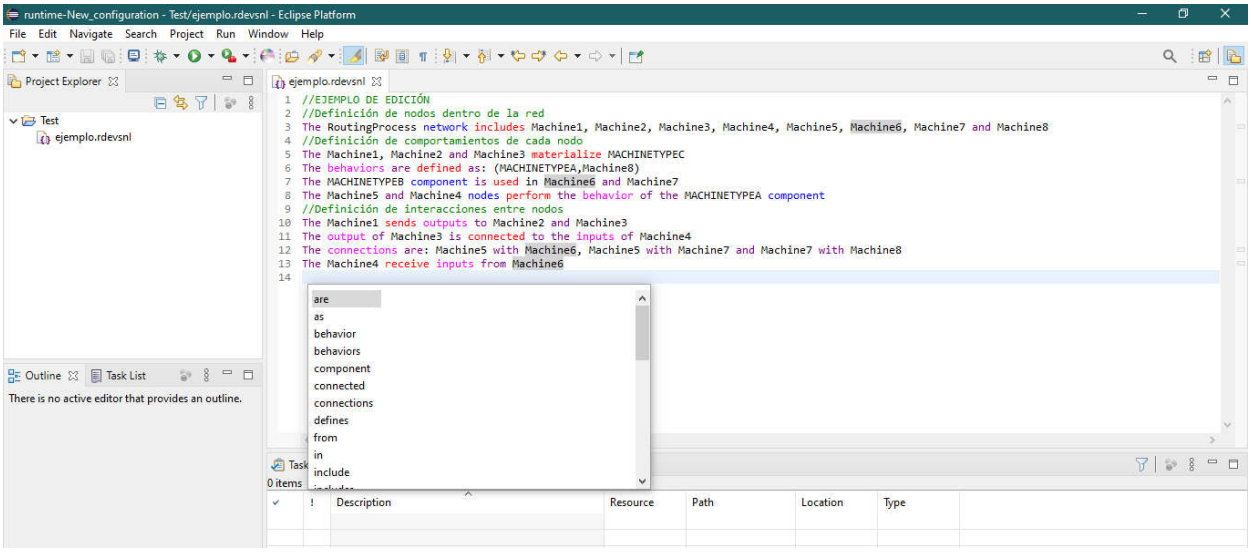


Figura 6. Captura de pantalla del Editor implementado en Eclipse para admitir la sintaxis. Sugerencias de escritura son ofrecidas y el resaltado de sintaxis puede observarse en las palabras clave, siguiendo la configuración del idioma (inglés en este caso). Las líneas 5 a 8 muestran diferentes formas de asignar a cada nodo un comportamiento, así como las líneas 10 a 13 presentan las distintas maneras de definir interacciones entre los nodos.

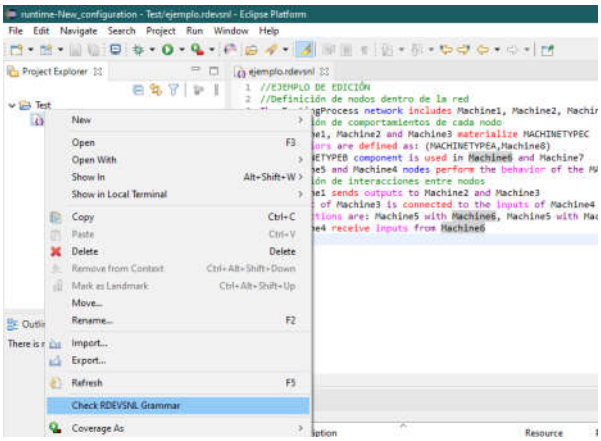


Figura 7. Opción del editor de texto utilizado para ejecutar el proceso de validación.

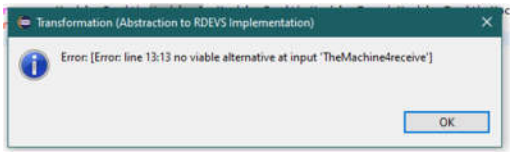


Figura 8. Mensaje de error que es mostrado cuando se detecta un problema durante el análisis sintáctico.

Una vez superado el proceso de validación sintáctica, teniendo ya creada la instancia del metamodelo, el plugin ejecuta la validación del metamodelo Ecore para asegurar su correctitud. Esta validación verifica los conceptos, relaciones, multiplicidades y restricciones de OCL del metamodelo sobre la instancia obtenida del análisis sintáctico. De esta manera se lleva a cabo el análisis semántico del contenido especificado. Si ambos análisis tienen éxito (es decir, el análisis sintáctico y semántico), el modelador recibirá un mensaje de éxito. De lo contrario, si se detectan problemas, el modelador podrá visualizar un mensaje de error (similar al de la Figura 8). De esta forma, con los errores identificados, podrá corregir su descripción del proceso de enrutamiento y volver a realizar las comprobaciones.

6. Conclusiones y Trabajos Futuros

Se ha introducido una gramática basada en un modelo de red restringido para abordar la definición de procesos de enrutamiento a través del formalismo RDEVS. El uso de una especificación textual como mecanismo de definición de un escenario a simular (en este caso, basado en un modelo de red restringido), facilita el proceso de M&S ya que abstrae al modelador de las complejidades matemáticas propias del formalismo RDEVS.

El plugin final RDEVSNL se compone de la sintaxis (definida utilizando ANTLR4), el editor de texto RDEVSNL y el modelo semántico (definido como un modelo Ecore). Esta herramienta facilita la definición y validación de procesos de enrutamiento empleando lenguaje natural. Aquí es importante destacar que, aunque nos referimos a "lenguaje natural", nuestra gramática sigue un conjunto de reglas de producción definidas que deben validarse para poder procesarla. Es decir, no se realiza procesamiento de lenguaje natural basado en técnicas de inteligencia artificial. Aun así, esta definición facilita a futuro, que el modelador desde la especificación textual de un proceso de enrutamiento como una red, obtenga un modelo de simulación RDEVS basado en Java sin contar con conocimientos de programación ni con los conocimientos matemáticos propios del formalismo.

El trabajo futuro está dedicado a la traducción del proceso de enrutamiento descrito en la instancia del modelo de red (obtenido de la gramática) a los modelos de simulación RDEVS. Dicha traducción será desarrollada como la ya implementada en [14]. Además, se utilizará la representación gráfica existente para mostrar la descripción en lenguaje natural en forma de grafo. Así, la herramienta de software final para el M&S en RDEVS admitirá definiciones tanto gráficas como textuales.

Referencias

[1] Newman, M., Barabasi, A.-L. y Watts, D.J. *The Structure and Dynamics of Networks*, Princeton University Press, 2006.

[2] Borgatti, S.P. y Halgin, D.S. "On network theory", *Organization Science*, 22, 5, Abril 2011, pp. 1168-1181.

[3] Wen, L., Kirk, D. y Dromey, R.G. "Software Systems as Complex Networks". En Actas de *IEEE International Conference on Cognitive Informatics*, 2017, pp. 106-115.

[4] Pan, W. "Applying Complex Network Theory to Software Structure Analysis", *International Journal of Computer and Systems Engineering*, 5, 12, 2011, pp. 1634- 1640.

[5] Zakari, A., Lee, S.P, y Chong, C.Y. "Simultaneous Localization of Software Faults based on Complex Network Theory", *IEEE Access*, 6, 2018, pp. 23990-24002.

[6] Blas, M., Gonnet, S. y Leone, H. "Routing Structure over Discrete Event System Specification: A DEVS Adaptation to Develop Smart Routing in Simulation Models", En Actas de *Winter Simulation Conference,* Las Vegas, USA, Diciembre 2017, pp. 774-785.

[7] Blas, M., Espertino, C. y Gonnet, S. "Modeling Routing Processes through Network Theory: A Grammar to Define RDEVS Simulation Models", En Actas de *3rd Workshop on Modeling and Simulation of Software-Intensive Systems*, 2021, https://doi.org/10.5753/mssis.2021.17255.

[8] The Eclipse Foundation. Eclipse. Disponible: https://www.eclipse.org/. Accedido por última vez el 11/9/2021.

[9] Zeigler, B., Muzy, A. y Kofman, E. *Theory of modeling and Simulation: Discrete Event & Iterative System computational Foundations*, Academic Press, 3era ed., 2018.

[10] Hopcroft, J.E., Motwani, R. y Ulman, J.D. Introduction to Automata Theory, Languages and Computation. 3era ed. Madrid: Pearson, 2007, pp. 143-184.

[11] OMG. Meta Object Facility (MOF) Specification, versión 1.4, 2002.

[12] ANTLR4 IDE Eclipse Plugin para ANTLR4. ANTLR. Disponible: https://www.antlr.org/tools. Accedido por última vez el 11/9/2021.

[13] The Eclipse Foundation: Eclipse Modeling Project. Eclipse Modeling Framework. Disponible: https://www.eclipse.org/modeling/emf/. Accedido por última vez el 11/9/2021.

[14] Blas, M. y Gonnet, S. "Computer-aided Design for Building Multipurpose Routing Processes in Discrete Event Simulation Models", *Engineering Science and Technology, an International Journal*, 24, 2021, pp. 22–34. https://doi.org/10.1016/j.jestch.2020.12.006.