

# Un DSL para definir y analizar sistemas de ferrocarriles

Víctor García-Bermejo Mazorra

Universidad Autónoma de Madrid

**Abstracto.** La primera línea ferroviaria del mundo fue entre las ciudades de Liverpool y Manchester en 1826. La idea principal era el transporte de forma simplista y lineal de un punto a otro de la forma más rápida posible. Actualmente las vías del tren son una herramienta de uso común entre diversos transportes. Sin embargo, donde inicialmente se hacían recorridos lineales ahora trenes o ferrocarriles se desplazan a la vez en extensos y complejos sistemas ferroviarios donde los raíles están interconectados entre sí. Por ello, el objetivo de este trabajo es crear un Lenguaje de Dominio Específico (DSL) en base a unas normas establecidas para crear redes ferroviarias con el fin de poder diseñarlas. Adicionalmente, mediante el desarrollo de las transformaciones “model-to-model” (M2M) y “model-to-text” (M2T) obtendremos redes de Petri donde se podrá analizar el riesgo de los trenes dentro de la línea de ferrocarril.

**Keywords:** Lenguaje de Dominio Específico, Model-to-model, Model-to-text, red ferroviaria, red de Petri.

## 1 Introducción

A finales del siglo XVIII y principios del siglo XIX, tuvo lugar un proceso de desarrollo económico y tecnológico de gran magnitud conocido como la Revolución Industrial. Uno de los inventos más importantes de este periodo, fue la máquina de vapor cuyo objetivo era transformar la energía de vapor creada mediante la combustión en energía cinética.

A partir de la máquina de vapor, en 1826 apareció el primer ferrocarril, un método de transporte que se movía por raíles o líneas ferroviarias. La idea inicial fue el movimiento de un lugar a otro transportando todo tipo de materiales pesados, aunque a lo largo del tiempo esa premisa ha cambiado puesto que en la actualidad se utiliza de manera muy habitual para el desplazamiento de cualquier elemento, como materiales, mensajería o personas. Otro factor que cambia desde la Edad Moderna a la actualidad son los propios sistemas de redes ferroviarias. El planteamiento utilizado inicialmente era unir un punto con otro mediante una línea ferroviaria en la que se desplazará un tren mientras que hoy en día los sistemas unen gran cantidad de estaciones entre sí, de tal forma que los trenes en base a su necesidad pueden coger desvíos o utilizar rutas alternativas. En el caso actual, es importante recalcar que en una red ferroviaria hay más de un tren circulando a la vez. Con esto sucede que habrá puntos críticos donde puedan coincidir varios trenes a la vez suponiendo inseguridades en la red ferroviaria.

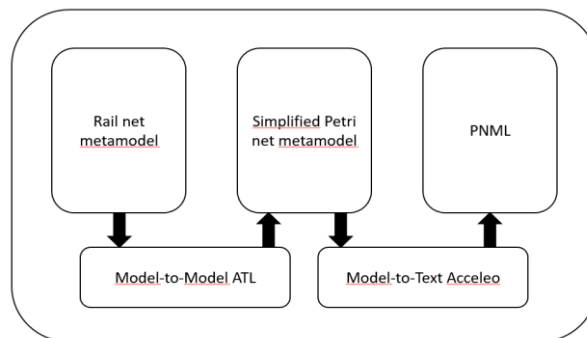
Por ello, el objetivo de este proyecto es crear una serie de aplicaciones para diseñar redes ferroviarias, y poder analizar su seguridad mediante redes de Petri.

Teniendo presente el objetivo, en el apartado 2 de este documento se plantearán las cuestiones del diseño que se han tenido en cuenta y como se distribuirán las aplicaciones que conforman el proyecto; en el apartado 3 se explicará los pasos que se han seguido para su desarrollo; en el apartado 4 se presentará una prueba de la aplicación con su correspondiente flujo; en el apartado 5 se propondrán trabajos relacionados y; finalmente en el apartado 6 se hará un breve repaso que incluirán unas conclusiones.

## 2 Enfoque

Puesto que el objetivo es generar un DSL [1] donde se pueda crear una red de ferrocarriles y efectuar su posterior análisis de flujo de los ferrocarriles o trenes mediante algún programa externo que tratase las redes de Petri, se marcaron los siguientes hitos dentro del proyecto:

- Creación de un meta-modelo para diseñar redes ferroviarias. Para ello, se implementará el lenguaje formal para la definición de sistemas ferroviarios propuesto por Bjørner [2]. En él se explica cómo tratar a los elementos de una red de la forma más sencilla posible, dejando un diseño abierto a nuevos elementos.
- Transformación de los modelos de redes ferroviarias creados a redes de Petri mediante la transformación M2M. Para completar este paso, se propuso un meta-modelo de para modelar redes de Petri, que sería una simplificación del formato Petri Net Markup Language (PNML) [4]. En este caso, la herramienta que se utilizó fue ATL [5].
- Transformación de las redes de Petri simplificadas al formato PNML mediante la transformación M2T. Como existen herramientas con las que se puede hacer un análisis en profundidad de redes de Petri, se debía dejar en un formato legible por ellos. En este caso esta transformación se hizo con la idea de utilizar el programa WoPeD [6].



**Fig. 1.** Sistema de flujo del desarrollo del proyecto.

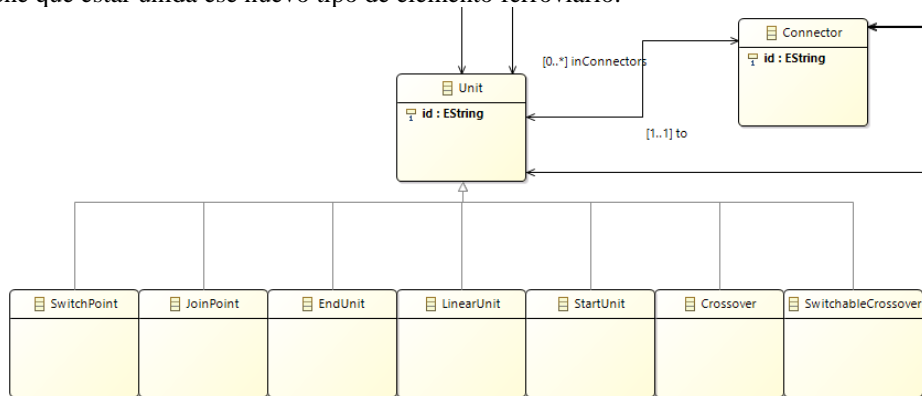
En la figura 1, podemos apreciar todos los elementos que conforman el proyecto, entrelazados paso a paso. En primer lugar se generó el meta-modelo de las redes de ferrocarril. Siguiendo el proceso, se crearon el meta-modelo de la red de Petri simplificada y el módulo de transformación M2M. Finalmente se creó la transformación M2T mediante Aceleo [7].

### 3 Desarrollo de las herramientas

Una vez comprendido el sistema que se iba a crear, empezó el desarrollo centrado en los hitos. En este apartado del documento se explicará cronológicamente el desarrollo de todas las partes que conforman el proyecto.

#### 3.1 Creación de un meta-modelo de la red ferroviaria.

Tal y como apunta Dines Bjørner en su descripción formal de los sistemas de ferrocarriles, un sistema tiene que estar mínimamente formado por dos estaciones y una línea que las una. Para que un tren circule de una estación a la otra, las estaciones y las líneas de alguna forma deben estar conformadas por railes y estos deben estar conectados de forma coherente. Por ello, en el desarrollo de este DSL se describe una clase básica y abstracta de la cual heredarán todas las posibilidades que se quieran generar para conformar el meta-modelo y así poder diseñar redes ferroviarias. En nuestro caso se definieron una serie de clases sencillas que casan con esta idea. Destacar que tal y como está creado el meta-modelo, si se quiere definir un nuevo tipo de rail como por ejemplo un cruce de 3 caminos, lo único que habría que hacer es definir la clase, que herede de ella y finalmente crear una restricción mediante el Object Constraint Language (OCL) con la finalidad de declarar a cuantos conectores de entrada y de salida tiene que estar unida ese nuevo tipo de elemento ferroviario.



**Fig. 2.** Clases que heredan de la unidad básica. Adicionalmente la clase Connector que unirá las unidades en el modelo.

Además de esta parte, hay que destacar que, tal y como se comentó en la propuesta de proyecto, las estaciones están compuestas por pistas y que tiene que haber como mínimo una en cada estación, las líneas y las pistas solo pueden componerse de unidades lineales (aquellas que solamente tienen un conector de entrada y otro de salida) y otras condiciones que se han ido tratando principalmente en OCL.

### 3.2 Creación de un meta-modelo de la red de Petri simplificada y transformación M2M

Una vez finalizado el diseño del meta-modelo de las redes ferroviarias, ya se podían diseñar modelos creando una instancia dinámica de la clase railNet y que fuesen coherentes si cumplían las restricciones impuestas por el diseño y por OCL. Sin embargo, con ello solo se puede controlar que sintácticamente estén bien. Para comprobar si su flujo de trenes dentro de la red ferroviaria era correcto, se debe transformar en red de Petri y analizarlo mediante un programa externo a Eclipse (WoPeD).

Para esta transformación en red de Petri, se aprendió a utilizar la extensión de Eclipse denominada ATL. Ésta nos permitía mediante la implementación de una serie de reglas obtener de un modelo fuente, otro basado en un meta-modelo distinto. Como en el caso que se está tratando se quiere transformar a redes de Petri, se necesita un meta-modelo para ellas. Con la finalidad de crearlo, se creó una versión simple de PNML.

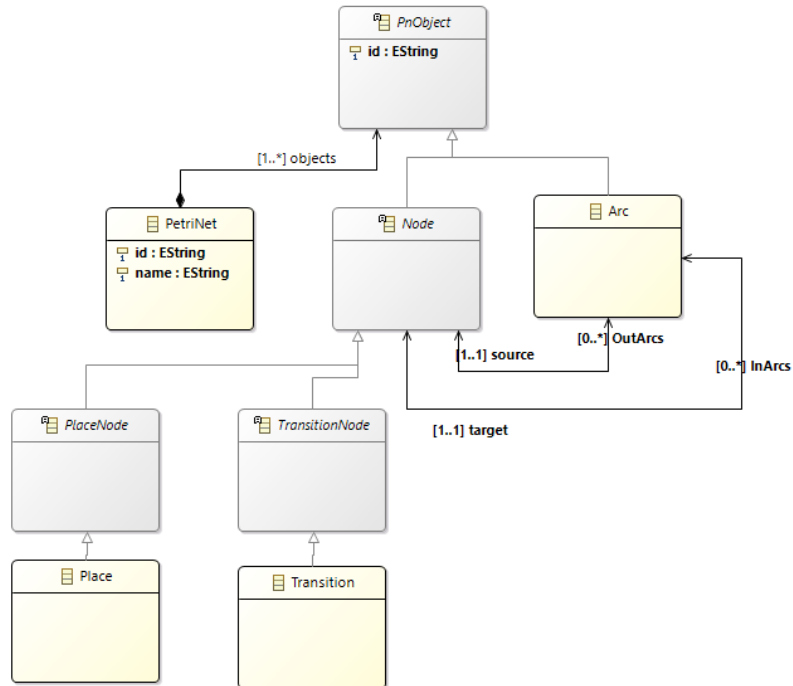


Fig. 3. Diagrama de clases de redes de Petri simplificadas basado en PNML.

Con el meta-modelo de redes de Petri creado, se implementó la transformación en ATL. Este fue un proceso tedioso puesto que no se comprendía el funcionamiento concreto de la transformación y se tuvo que aprender desde cero. Los elementos más destacables que hubo en la transformación fueron los siguientes:

- Generación de arcos y transiciones en sustitución de los conectores. Las dos unidades básicas que se iban a transformar de la red ferroviaria a la de Petri eran las unidades y los conectores. La correspondencia con los lugares y las transiciones respectivamente es casi directa. El problema reside en que los arcos no tenían una clase semejante dentro del meta-modelo de la red ferroviaria. Por ello, se decidió añadir su creación a la regla que transforma conectores en transiciones, aprovechando que los datos necesarios para su generación están definidos en los conectores.
- Generación de lugares para hacer semáforos. Cuando dos trenes llegan a una unión de carriles, uno debería esperar a que el otro pasara. Por ello cuando una unidad de rail tiene un mínimo de 2 conectores de entrada, debe tener un semáforo. Este semáforo se define como un lugar y en la endrule de ATL se le añaden los arcos que lo unen a las transiciones correspondientes.
- Adición de todos los elementos de la red de Petri al final de la transformación. Como debido a las situaciones anteriores había elementos que se quedaban fuera de la red de Petri si se intentaban añadir en la regla de la transformación de la red ferroviaria a la de Petri, se añadieron en la endrule de ATL.

### 3.3 Transformación M2T mediante Acceleo.

En este punto del desarrollo ya se obtenían redes de Petri simplificadas a raíz de las redes ferroviarias. Ya solo se debía generar un fichero legible por WoPeD en el formato PNML. Para ello se utilizó la extensión de Eclipse llamada Acceleo. Con ella, a raíz de modelos se pueden crear ficheros de texto con cualquier tipo de extensión. Como lo que se quería generar era simplemente una ampliación y una serie de etiquetas que envolvían a los modelos de redes de Petri, simplemente en el fichero generate.mtl se recorrerían los modelos con simples bucles for. Lo más destacable de esta parte del proyecto es la imposibilidad de crear variables no finales dentro de Acceleo. Esto supuso que, al pasarlo al formato PNML, no se pudiesen dar coordenadas distintas a los elementos de la red de Petri, y por ello, al leerlo desde WoPeD, se generasen todos los elementos en un mismo punto.

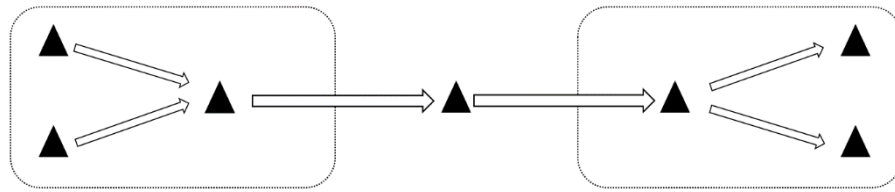
## 4 Evaluación

A lo largo de todo el proyecto, los módulos que se generaron fueron probándose por separado puesto que, todos ellos tenían una gran independencia de los otros. Más concretamente el orden de las pruebas que se hicieron fue el siguiente:

1. Generación de varios modelos basados en el meta-modelo de las redes de ferrocarriles. Se intentó exprimir al máximo los casos que se podían cumplir las restricciones impuestas y los que no.

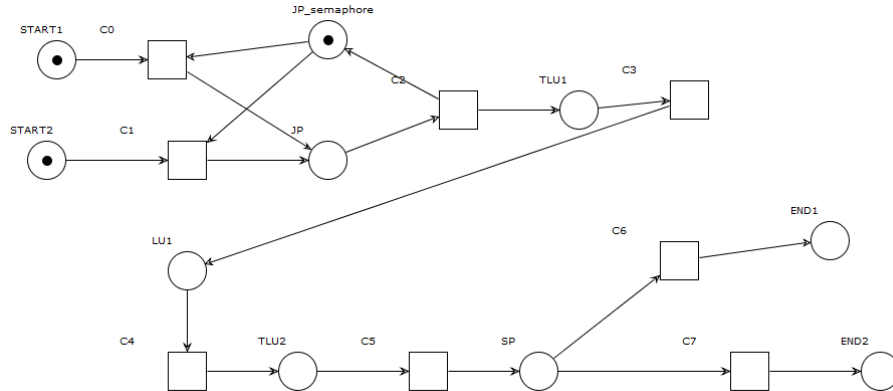
2. Generación de varios modelos basados en el meta-modelo de las redes de Petri. En este caso, para saber simplemente si la simplificación del formato PNML era la correcta.
3. Diversas pruebas de transformación con los modelos de redes ferroviarias. Se intentó exprimir la funcionalidad de los semáforos y buscar casos que fueran concretos como los raíles de inicio y final dentro de la red de ferrocarril.
4. Diversas pruebas de transformación de los modelos de redes de Petri salientes de ATL al formato PNML mediante Acceleo. En este caso se transformaban al texto correspondiente y se veía si la red de Petri obtenida era la correcta.

Adicionalmente en la entrega de este proyecto y esta memoria, se añadirá una grabación de video donde se mostrará paso a paso se ejecutan todas las partes y finalmente se prueba en WoPeD la red de Petri resultante. En él se planteará un sistema sencillo donde habrá una unión de raíles y una separación en un recorrido casi lineal.



**Fig. 4.** Sistema planteado en la demostración del video. Los triángulos son unidades y las flechas son conectores. Los cuadrados representan las estaciones.

El resultado final de la ejecución de todos los módulos en su correcto orden sobre la red ferroviaria propuesta en la figura 4 es la red de Petri propuesta en la figura 5.



**Fig. 5.** Red de Petri resultante con tokens añadidos a modo de trenes y semáforos.

## 5 Conclusiones

En este proyecto se ha desarrollado una serie de módulos con el objetivo de ver si los sistemas ferroviarios que se generan tienen fallos de diseño. Para ello, primero se ha generado un meta-modelo para definir un DSL con el que se puede crear modelos de redes de ferrocarriles de forma coherente. Más adelante se ha implementado una transformación M2M de estos modelos a redes de Petri simplificadas mediante ATL. Finalmente con estas redes de Petri simplificadas se obtiene un archivo de extensión PNML mediante la transformación M2T de Acceleo que servirá para analizar la red resultante en WoPeD.

## Referencias

1. <https://www.jetbrains.com/es-es/mps/concepts/domain-specific-languages/>
2. BJØRNER, Dines. Formal software techniques for railway systems. IFAC Proceedings Volumes, 2000, vol. 33, no 9, p. 101-108.
3. <https://projects.eclipse.org/projects/modeling.mmt>
4. <http://www.pnml.org/>
5. [https://wiki.eclipse.org/ATL/User\\_Guide\\_-\\_The\\_ATL\\_Language](https://wiki.eclipse.org/ATL/User_Guide_-_The_ATL_Language)
6. <https://woped.dhbw-karlsruhe.de/>
7. [https://wiki.eclipse.org/Acceleo/User\\_Guide](https://wiki.eclipse.org/Acceleo/User_Guide)