

**(C-168)**

**UNA HERRAMIENTA LIBRE PARA LA DOCENCIA DE  
SISTEMAS DOMÓTICOS.**

*Francisca Rosique*

*Pedro Sánchez, Manuel Jiménez*



## (C-168) UNA HERRAMIENTA LIBRE PARA LA DOCENCIA DE SISTEMAS DOMÓTICOS.

*Francisca Rosique, Pedro Sánchez, Manuel Jiménez*

**Afiliación Institucional:** Universidad Politécnica de Cartagena, DSIE (División de Sistemas e Ingeniería Electrónica)

**Indique uno o varios de los siete Temas de Interés Didáctico:** (Poner x entre los [ ])

Metodologías didácticas, elaboraciones de guías, planificaciones y materiales adaptados al EEES.

Actividades para el desarrollo de trabajo en grupos, seguimiento del aprendizaje colaborativo y experiencias en tutorías.

Desarrollo de contenidos multimedia, espacios virtuales de enseñanza- aprendizaje y redes sociales.

Planificación e implantación de docencia en otros idiomas.

Sistemas de coordinación y estrategias de enseñanza-aprendizaje.

Desarrollo de las competencias profesionales mediante la experiencia en el aula y la investigación científica.

Evaluación de competencias.

### **Resumen.**

En este artículo se presenta una herramienta libre y gratuita pensada principalmente para mejorar los métodos de enseñanza en el desarrollo de sistemas domóticos. Esta herramienta permite a los estudiantes comprender las principales propiedades de un sistema domóticos y los conceptos genéricos, así como modelar una aplicación real de una forma más fácil e intuitiva. Este artículo no sólo se centra en la herramienta también se presenta la metodología que deben seguir profesores y estudiantes. Además siguiendo con las pautas marcadas por Bolonia en cuanto a auto estudio y trabajo en casa, se han incorporado tareas que los estudiantes son capaces de desarrollar como parte de su trabajo en casa y sin necesidad de realizar inversiones en licencias de herramientas comerciales.

**Keywords:** Domótica, herramienta libre, innovación docente

### **Abstract.**

This paper presents a free tool that can be used to improve the teaching methods of smart home systems development. This tool allows students to understand the main properties and the generic concepts of home automation systems. In addition, students can use the tool to develop home automation systems in a more easy and intuitive way. The article is not only concerned with the description of the tool but also with the methodology to be followed by teachers and students. Furthermore, according to the guidelines of Bolonia Declaration with regard to autonomous-learning and homework, tasks have been incorporated that can be developed by students as part of their homework without the need of commercial tool licenses.

## 1. Introducción.

Las universidades tienen una responsabilidad y unos intereses mucho más amplios que una empresa o un particular cuando adquieren tecnologías de la información. Deben tener una vocación de consumo responsable. En este sentido, el ahorro que representan las herramientas libres es atractivo para las universidades, ya que les permite la mejor y más eficiente utilización de sus recursos.

Por otra parte, no hay que olvidar que la universidad cumple con una función de servicio público. Este es un motivo determinante para respetar e impulsar los estándares abiertos, cuando se dirige a los miembros de la comunidad universitaria. No hacerlo supone favorecer a un determinado fabricante y a sus clientes, discriminando al resto de usuarios. Además, en el nuevo marco de educación estas herramientas pueden resultar de gran utilidad pues el Espacio Europeo de Educación Superior (EEES)[1]:

- Ratifica al estudiante como el centro de la tarea docente.
- Incluye en los créditos ECTS horas de trabajo no presencial.
- Considera relevantes el trabajo reflexivo y colaborativo.
- Entiende que la enseñanza y aprendizaje equivalen a participar de procesos.

El trabajo con herramientas libres es muy conveniente en este marco de trabajo y con una gran ventaja: se adapta y avanza al ritmo que marca la comunidad. Pero, a la hora de elegir y seleccionar estas herramientas, hay que tener presente las necesidades por cumplir y asegurarse que cumplen unas condiciones mínimas [2]:

- La mayoría de las herramientas libres están disponibles en distintos países y en distintos idiomas, pero hay que asegurar que al menos se encuentre en inglés o en el idioma oficial del país.
- Debe poder garantizar el acceso a la información en todo momento, y en la medida de lo posible utilizar estándares y formatos abiertos, ya que en caso contrario el usuario se puede ver obligado a comprar productos de software a empresas que se verían beneficiadas de una situación de monopolio. Si el particular no quiere o no puede pagar este software sería una discriminación flagrante. En resumen, no se puede forzar a los miembros de la comunidad universitaria a adquirir software de una determinada marca.
- No ha de permitir que personas no autorizadas tengan acceso a los datos confidenciales de los particulares o a información reservada (seguridad). Pero, sobre todo, el usuario ha de poder comprobar si el software empleado por la universidad cumple objetivamente las funciones de tratamiento de sus datos personales. Es decir, el código fuente ha de estar totalmente disponible y se ha de poder compilar para comprobar que el programa que se utiliza coincide con el código fuente entregado. La trazabilidad ha de ser total para que no se puedan producir fraudes.

Estos puntos aseguran que se pueda realizar una innovación docente de forma clara a través de herramientas libres, convirtiéndose en un magnífico objetivo, pero sin olvidar que la innovación tecnológica no significa de forma automática innovación docente. No es suficiente con el empleo de la tecnología en las clases, si éstas no se dominan, para ofrecer un buen servicio a los estudiantes. El empleo de la tecnología debe tener claramente un fin pedagógico, nunca el empleo por comodidad o moda.

Se debe tener presente que el estudiante debe ser el eje de actuación, lo que interesa es su aprendizaje, en la medida que las nuevas tecnologías permitan llegar a ese objetivo se estará haciendo un uso racional de dichas tecnologías. Se necesita de una planificación coherente y cuidada, en la que se mezclen hábilmente enseñanza, aprendizaje y tecnología, pero sin perderse en lo tecnológico.

En este artículo se presenta una herramienta libre y gratuita que puede utilizarse para mejorar los métodos de enseñanza en el desarrollo de sistemas domóticos. La principal contribución de esta herramienta es que permite a los usuarios

desarrollar sistemas domóticos utilizando elementos gráficos independientemente de la plataforma de implementación. Los profesores pueden utilizar esta herramienta para introducir el desarrollo de aplicaciones domóticas y demostrar los conceptos y teorías relacionadas con este dominio. Además, los estudiantes pueden utilizar la herramienta como un medio autónomo (individual o colaborativo) de aprendizaje, sin necesidad de realizar inversiones en licencias de herramientas comerciales.

A continuación se describe en la sección 2 las principales características de herramienta desarrollada, en la sección 3 la metodología docente adoptada y finalmente las conclusiones.

## 2. Una herramienta libre para el desarrollo de sistemas domóticos

El grupo de investigación DSIE (División de Sistemas e Ingeniería Electrónica) de la Universidad Politécnica de Cartagena, ha desarrollado una herramienta libre totalmente gratuita, pensada principalmente para mejorar la enseñanza de la domótica.

Las principales características de la herramienta son:

- Herramienta libre y gratuita. En la web del grupo de investigación están disponibles los plug-ins para descargar e instalar la herramienta ([http://www.dsie.upct.es/proyectos/web\\_explore](http://www.dsie.upct.es/proyectos/web_explore)).
- Tanto la herramienta como la documentación está disponible en inglés y español.
- Entorno amigable, con conceptos comunes a todos los protocolos, estándares y plataformas domóticas. Gracias a esto, los estudiantes tendrán una mejor comprensión de los conceptos en el dominio domótico, ya que estos conceptos se ponen en práctica sin la necesidad de aprender conceptos de plataformas específicas (KNX, Lonworks, X10, etc.) (para más detalles sobre los elementos del editor véase [3]). Dispone de un editor para el modelado de sistemas domóticos que utiliza descripciones fáciles e intuitivas a cualquier dominio domótico y totalmente independiente de cualquier plataforma de implementación. El editor dispone de dos posibles perfiles:
  - Editor gráfico que facilita la comprensión dada su simplicidad y autocontención.
  - Editor textual que permite definir o consultar un modelo del sistema domótico en texto plano.
- Dispone de un repositorio instalado en un servidor CVS (Concurrent Versions System) que proporciona un mecanismo de almacenamiento compartido de todos los elementos que componen un proyecto (documentos, aplicaciones, ficheros de configuración, etc.), así como un histórico de las modificaciones que se han realizado sobre los mismos, de manera que las distintas personas que comparten el proyecto puedan estar al tanto de las modificaciones que se van realizando en el proyecto y puedan colaborar en él. Lo que permite una enseñanza colaborativa tanto en el laboratorio como a distancia.
- Proporciona libertad de uso, libertad de estudio, libertad de adaptación (se ofrece el código fuente) y libertad de redistribución.
- Permite realizar una evaluación continua del estudiante.
- La mayoría de herramientas existentes en el dominio son herramientas comerciales específicas para ciertas tecnologías, protocolos o estándares, son difíciles de utilizar y tiene un coste económico elevado.

La herramienta se ha creado usando el entorno de desarrollo Eclipse [4], lo que hace que la distribución de las zonas de trabajo de la herramienta sea muy similar a Eclipse. Las partes más características de la herramienta son:

- **Zona de dibujo:** zona de trabajo donde se arrastran las figuras desde la paleta para construir un modelo gráfico concreto, tanto para la vista de Catálogo como de Aplicaciones. También existe la opción de instanciar elementos desde un menú flotante en lugar de utilizar la paleta.

- **Paleta gráfica:** contiene las primitivas que se pueden arrastrar a la zona de dibujo, se puede apreciar en la parte derecha de la figura 1.
- **Zona de propiedades:** zona donde se visualizan y se modifican aquellas propiedades (atributos, parámetros, etc...) disponibles para la primitiva seleccionada. Para acceder a estas propiedades basta con seleccionar un elemento de la zona de dibujo. Las propiedades se muestran como campos editables o desplegable para facilitar un acceso rápido e intuitivo.



Figura 1: Principales áreas de trabajo de la herramienta.

Para el desarrollo de la herramienta descrita en este artículo se ha considerado el enfoque de Desarrollo Dirigido por Modelos (MDD en sus siglas inglesas) [5]. Este enfoque permite generar automáticamente el código de la aplicación gracias a la transformación automática de modelos a distintos niveles de abstracción desde la captura de requisitos hasta el código de implementación final. La definición del editor gráfico ha sido realizada utilizando técnicas de definición de Lenguaje Específico de Dominio (DSL en sus siglas inglesas) [6]. Este DSL desempeña un papel muy importante en la herramienta, ya que proporciona directamente los conceptos y elementos a utilizar por el sistema.

La definición de este lenguaje tiene por objetivo ayudar a los estudiantes a describir los sistemas domóticos utilizando únicamente conceptos del dominio. En este sentido, este DSL facilita la captura de requisitos propios de un sistema domótico de forma visual e intuitiva. De esta forma, los estudiantes ven facilitada la posibilidad de expresar y entender su conocimiento y experiencia en el dominio. Por esta razón la primera premisa es la de disponer de una capacidad de expresar de manera directa y a alto nivel de abstracción los conceptos del dominio, que permita la visualización del conocimiento, de una manera concisa y común a las distintas plataformas domóticas. Antes de entrar en detalles es

necesario exponer los principales conceptos con los que se trabaja en el dominio domótico y que deben tenerse en cuenta a la hora de crear el DSL.

En cualquier sistema domótico existe una serie de elementos (denominados "Unidades Funcionales") que aparecen en todas las tecnologías y estándares domóticos. Estos elementos se diferencian en la arquitectura, protocolos utilizados o módulos disponibles, pero son iguales en cuanto funcionalidad. Con el fin de promover la reutilización de estas unidades funcionales y evitar tener que definir múltiples veces la misma unidad para cada aplicación (incluso varias veces dentro de una misma aplicación), se ha optado por definir un Catálogo de unidades funcionales reutilizable, de manera que una vez definido dicho catálogo se pueda utilizar en cualquier aplicación y sólo sea necesario obtener ejemplares de dicho catálogo.

Finalmente, el framework Eclipse se ha utilizado para la implementación del DSL y como herramienta de gestión de modelos. En particular, se ha utilizado el plug-in Eclipse Modeling Framework (EMF) [7] para crear el editor de modelos y como base tecnológica para otras herramientas. Para la parte gráfica del DSL se ha utilizado Eclipse Modeling Framework (GMF) [8], plug-in que permite generar automáticamente editores gráficos como plug-ins para Eclipse. Las transformaciones de los modelos hasta la generación de código se han realizado con plug-ins adicionales, como la EMT [9] y Java emisor de plantillas (JET) [10].

Se puede acceder a un video de demostración de la herramienta en el siguiente enlace:  
<http://repositorio.bib.upct.es/dspace/bitstream/10317/854/6/has.swf>

### 3. Metodología de enseñanza con la herramienta

La nueva metodología de enseñanza aquí propuesta consta de dos roles principales, el del profesor y el del estudiante. Si atendemos a las distintas tareas de cada uno de estos roles podemos distinguir las siguientes fases (véase figura 3).

- El profesor impartirá los conceptos teóricos del dominio domótico en diferentes sesiones teóricas.
- El profesor planteará una serie de problemas a resolver por los estudiantes. Los problemas planteados se corresponden con un caso práctico de implementación de un sistema domótico real el cual debe satisfacer una serie de requisitos.
- En el laboratorio los estudiantes contarán con un ordenador, con la herramienta libre instalada, un catálogo de unidades funcionales reutilizables y un maletín que contiene una maqueta de una vivienda domótica (véase figura 2). Esta maleta es totalmente portátil y permite por un lado programar los dispositivos domóticos y por otro realizar cualquier tipo de cableado de la instalación.



Figura 2: Parte superior e inferior de la maqueta de una vivienda domótica.

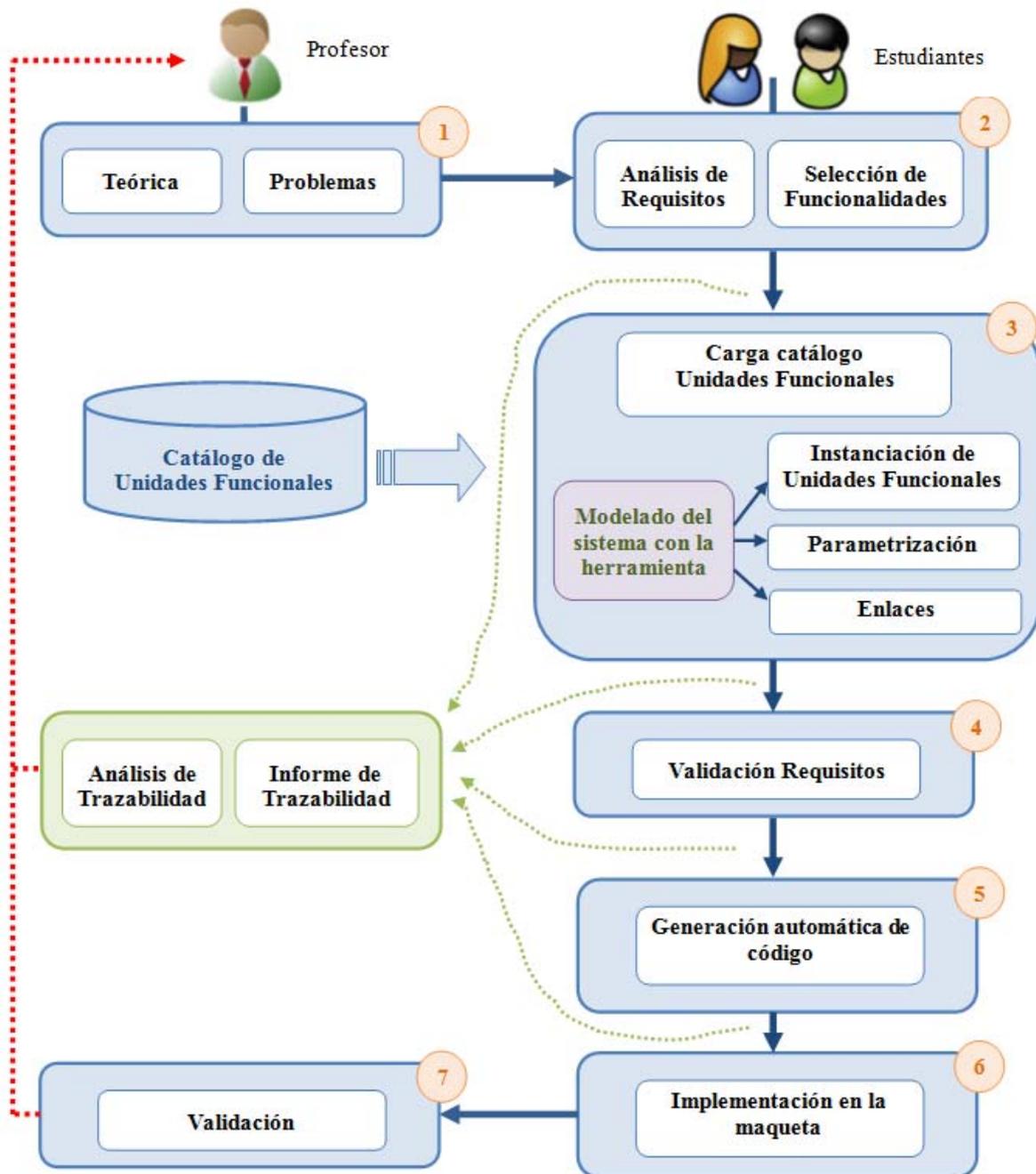


Figura 3: Metodología docente.

- Los estudiantes deben resolver el problema práctico con la ayuda de la herramienta e implementar el sistema domótico final correctamente en la maqueta. Para la resolución del problema propuesto, los estudiantes realizarán inicialmente un análisis de requisitos que debe cumplir el sistema domótico atendiendo a los requisitos de seguridad, confort, comunicación y ahorro energético y decidirán que funcionalidades son necesarias para satisfacer dichos requisitos.
- A continuación los estudiantes diseñarán el sistema haciendo uso de la herramienta.
  - Primero deben cargar el catálogo de unidades funcionales para poder utilizar y reutilizar dichas unidades en un modelo de aplicación.
  - A continuación deben modelar el sistema utilizando unidades funcionales reutilizables y dando los valores necesarios a sus parámetros.
  - Cuando estén seleccionadas todas las unidades funcionales y configurados sus parámetros, los estudiantes podrán enlazar dichas unidades según su criterio, indicando en cada enlace qué servicios se utilizan.
  - Finalmente, en el momento en el que el estudiante guarde su modelo de aplicación domótica, la herramienta realizará una validación automática para comprobar que no se han incumplido ninguna restricción. Por ejemplo se comprueba que cada unidad funcional tenga un identificador único dentro de la aplicación o que los servicios seleccionados en los enlaces sean compatibles entre sí.
- La herramienta realiza una serie de transformaciones internas totalmente transparentes al estudiante generando automáticamente el código del sistema domótico que el estudiante ha modelado. Este código está listo para ser cargado directamente a los dispositivos, permitiendo implementar el sistema domótico modelado por el estudiante en un caso real.
- Los estudiantes podrán comprobar el correcto funcionamiento del sistema domótico en las maquetas que se les proporciona en el laboratorio.
- Durante todo el proceso la herramienta genera una serie de trazas que permite tanto al estudiante como al profesor revisar de una manera rápida cada una de los pasos que el estudiante ha dado hasta llegar a la obtención del código. Del mismo modo, permite realizar un seguimiento para saber en qué dispositivo se ha implementado cada requisito.
- Finalmente, todo el proyecto generado, desde el primero hasta el último paso, queda almacenado en un repositorio alojado en el servidor CVS, donde existe un directorio por cada estudiante. Este directorio es accesible remotamente desde cualquier ordenador tanto por el estudiante como por el profesor, permitiendo de este modo dotar a la herramienta de una capacidad colaborativa.

#### 4. Conclusiones

En este artículo se ha presentado una nueva herramienta para la docencia de la domótica. Esta herramienta permite a los estudiantes comprender las principales propiedades de un sistema domótico y los conceptos genéricos, así como modelar una aplicación real de una forma más fácil e intuitiva. Además siguiendo con las pautas marcadas por Bolonia en cuanto a auto estudio y trabajo en casa. Asimismo, se han incorporado tareas que los estudiantes son capaces de desarrollar como parte de su trabajo en casa. Esto sería imposible con herramientas comerciales como ETS (Ingeniería de Software Herramienta para KNX) o LonMaker (la herramienta de desarrollo para LonWorks) que requieren licencias de uso.

Además, la herramienta propuesta no requiere instalación. Para poder utilizarla basta con descargarla en un ordenador, ya sea por internet o desde cualquier otro soporte físico (CD, memoria usb), y ejecutarla desde el acceso directo que incluye. Para distribuir la herramienta es suficiente con copiar la carpeta que contiene los ficheros al ordenador donde se desee ejecutar, no hay que preocuparse por ningún tipo de licencia. Incluso se puede copiar en una memoria usb y

ejecutar la herramienta desde allí. Esta característica resulta muy ventajosa para los estudiantes que ven facilitado su trabajo sin necesidad de realizar pesadas instalaciones ni tener que preocuparse por licencias.

Todo esto junto con la ventaja del control de versiones colaborativo, permite a los estudiantes trabajar con su propia copia local y guardar las modificaciones en un servidor remoto. En caso de que varios estudiantes coincidan en la modificación de un mismo fichero, se creará una nueva revisión que contiene una mezcla de toda la información identificando qué y quién modificó cada elemento (en este caso se guarda la versión de cada estudiante y la versión mezcla), posteriormente se puede elegir con qué fichero trabajar (si con el mezcla o con el personal). Es posible que un mismo proyecto se ramifique dando varias soluciones finales.

Como trabajos futuros se plantea un conjunto de experimentos que permitan validar la integridad de la herramienta. En estos experimentos se pretende analizar varios factores, como el tiempo de desarrollo y la usabilidad entre otros. Toda esta información será muy valiosa para ampliar y ajustar las futuras versiones de la herramienta a las necesidades del estudiante. Como bien dijo Einstein “Nada sucede hasta que algo se mueve“, y en nuestro caso hemos intentado dar el primer paso.

### **Agradecimientos**

Este trabajo ha sido parcialmente financiado por el proyecto español CICYT EXPLORE (TIN2009-08572).

### **Referencias.**

- [1] Benito, A. y Cruz, A.; Nuevas claves para la Docencia Universitaria en el Espacio Europeo de Educación Superior. Ed. Narcea, 2005.
- [2] R.J. López, A. Pérez, Innovación docente y Software libre. II Encuentro sobre innovación en docencia universitaria. Universidad de Alcalá. Junio 2007.
- [3] M. Jimenez, F. Rosique, P. Sánchez, B. Álvarez, A. Iborra, Habitation: A Domain-Specific Language for Home Automation, IEEE Software, Vol. 26, No. 4, 2009, pp. 33-38.
- [4] Eclipse project home. <http://www.eclipse.org>.
- [5] B. Selic, The Pragmatics of Model-Driven Development, IEEE Software, Vol. 20, No. 5, 2003, pp. 19-25.
- [6] M. Mernik, J. Heering, A. M. Sloane, When and How to Develop Domain-Specific Languages, ACM Computing Surveys, Vol. 37, No. 4, 2005, pp. 316-344.
- [7] Eclipse Modeling Framework Project. <http://www.eclipse.org/modeling/emf>
- [8] Graphical Modeling Framework Project. <http://www.eclipse.org/gmf>
- [9] E. Biermann, K. Ehrig, C. Ermel, C. Köhler, G. Kuhns, G. Taentzer, Tiger EMF Model Transformation Framework (EMT). <http://fs.cs.tu-berlin.de/emftrans/papers/userdoc.pdf>.
- [10] Eclipse Consortium, Java Emitter Templates (JET). <http://www.eclipse.org/modeling/m2g/?project=jet>.