MOSKITT, UNA HERRAMIENTA LIBRE QUE DA SOPORTE A LA APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

Martín García Hernández – Jefe del Servicio de Organización e Informática de la Conselleria de Infraestructuras y Transporte de la Generalitat Valenciana
Begoña Bonet Pérez de León, Javier Muñoz Ferrara

La herramienta libre MOSKitt es el resultado del proyecto gvCASE promovido y liderado por la ***Conselleria de Infraestructuras y Transporte (CIT)*** de la ***Generalitat Valenciana (GVA)***. El principal de objetivo del proyecto gvCASE es el desarrollo de una herramienta que soporte y automatice en la medida de los posible el método definido por la metodología de desarrollo gvMétrica, la adaptación de Métrica III realizada por la CIT.

La búsqueda de un Método para producir el software y la necesidad de dotar a los profesionales de las TIC en las Administraciones Públicas de herramientas para poder aplicarlo ha sido el origen de este proyecto. Es conocida por todos los la dificultad propia del desarrollo del software y su grave impacto en los sistemas a los que debe dar soporte. Estos dos factores (dificultad e impacto), han puesto de manifiesto la necesidad de aplicar una *metodología formal* para llevar a cabo los proyectos TIC en la CIT, de una forma más predecible y eficiente.

Un aspecto clave para implantar con éxito en una organización una *metodología de desarrollo de software*, es la mecanización de determinadas prácticas mediante el uso de *herramientas específicas*. Sin embargo, a pesar de las herramientas, construir Software es de por sí una tarea muy compleja, por lo que el método de trabajo que se aplique también lo será. Uno de los grandes retos de MOSKitt es el de "crear la ilusión" de que parezca más sencillo algo que no lo es en absoluto. Para conseguir este objetivo, MOSKitt debe facilitar a los ingenieros del software:

- La ejecución de un procedimiento de desarrollo adaptable a la tipología de proyectos identificados en una organización por parte de todos y cada uno de los roles participantes en ellos.
- La construcción de los modelos propuestos por la metodología haciendo uso de los métodos, técnicas y prácticas definidos en ella
- La automatización en la elaboración de estos modelos y productos de salida definidos en cada una de las tareas establecidas por la metodología.
- El trabajo colaborativo entre todos los miembros de los equipos de trabajo considerando los diferentes roles a los que puedan pertenecer.
- Establecer interfaces estándares con el resto de las herramientas empleadas en la organización durante todo el proceso de desarrollo para integrar el trabajo realizado por todos los miembros del Servicio durante el desarrollo de sus Sistemas de Información.

El objetivo final de la aplicación de la metodología a través de las herramientas adecuadas es conseguir la ***racionalización del uso de los recursos TIC en la organización***, a través del cual se quiere obtener como

beneficio un ***incremento de la calidad y un abaratamiento de los costes en la producción de software***. Más adelante, una vez finalizada la implantación de la automatización de los procesos de desarrollo, en una fase ya más madura del proceso se podrán aplicar métricas con el objetivo de mejorar el procedimiento establecido por gvMétrica.

MOSKitt (www.moskitt.org) se ha desarrollado siguiendo una arquitectura muy modular con el objetivo de poder ser configurado y/o extendido para adaptarse a las necesidades de otros métodos de desarrollo y, en general, organizaciones. Por ello, según las necesidades específicas de cada caso, MOSKitt puede considerarse :

- Un conjunto de herramientas para el análisis, diseño e implementación de sistemas software.
- Una herramienta de soporte a procesos de desarrollo y, en concreto al Método propuesto por gvMétrica.
- Una infraestructura para el desarrollo de herramientas de modelado y generación de código.

1. MOSKitt como un Conjunto de Herramientas de Análisis, Diseño e Implementación de Sistemas Software.

MOSKitt proporciona una serie de ***editores gráficos*** para el análisis y diseño de sistemas de información. Estos editores se basan, cuando es posible, en técnicas estándar en el ámbito del modelado de sistemas software.

- **Editor de Modelos de Procesos**: Basado en el estándar *Business Process Modeling Notation (BPMN)* y extendido para soportar la definición de procedimiento administrativos. Además, se integra con otros editores para permitir la gestión de tipos y familias documentos, estructuras organizativas de recursos humanos y definición de las estructuras de datos de la información que en ellos se maneja.
- **Editor de Esquemas de Bases de Datos Relacionales**: Basado en el proyecto *DataTools Platform Project (DTP) de Eclipse*, en el que participan distintos proveedores de BBDD (Sybase, Oracle, etc.) para proporcionar un modelo de base de datos común sobre el que construir herramientas de gestión de bases de datos. Este editor incluye también ***ingeniería inversa*** ya que ha sido enriquecido con la capacidad poblar estos modelos de bases de datos a partir de esquemas ya existentes (Sybase, Oracle, etc.).
- **Editor de Modelos de Sistemas de Software**: Basado en el estándar de modelado *Unified Modeling Language (UML2)*, proporciona editores gráficos para la mayoría de los distintos tipos de diagramas que define el estándar (clases, secuencia, actividad, máquinas de estado y casos de uso). También permite la definición y aplicación de perfiles para decorar y extender los modelos.
- **Modelado de Interfaces de Usuario**: Ante la falta de estándares en este área, se ha definido con la supervisión del Centro ProS de la Universidad Politécnica de Valencia un lenguaje para la

especificación de manera abstracta de interfaces de usuario. Este lenguaje permite describir la navegación entre los distintos puntos de interacción de los usuarios con el sistema, así como los datos que se deben visualizar y las acciones disponibles en cada uno de estos puntos de interacción.

Los editores gráficos de modelos han sido desarrollados utilizando las tecnologías del proyecto *Eclipse* y pueden modificarse o extenderse para adaptarse a las necesidades específicas de una organización o grupo de trabajo.

En la aplicación de la metodología gvMétrica han sido definidas toda una serie de plantillas que deben ser completadas por los diferentes participantes en el proceso de desarrollo de una forma homogénea para todos los proyectos. Se ha definido el metamodelo subyacente a la metodología gvMétrica y que da soporte a toda la información requerida en dichas plantillas metodológicas. Actualmente se está trabajando en la integración en MOSKitt de todas ellas de forma que puedan ser editadas desde la herramienta en lugar de tener que hacerlo sobre documentos de texto. Ocurre que, no en todos los casos el modelo que subyace tras esta documentación tiene una representación gráfica (como UML2, BPMN etc..) sino que en muchos de ellos, tiene más sentido su edición siguiendo una aproximación más tradicional como lo es un formulario típico de introducción de datos. En MOSKitt se ha desarrollado toda una infraestructura para dotar de ***editores en formato formulario*** a aquellos modelos que así lo requieran. Por ejemplo, se sigue esta estrategia en el caso de la Definición del Alcance de los Proyectos, su Entorno Tecnológico, el Glosario de Términos asociado, el Diccionario de Datos etc...

A partir de los modelos construidos con MOSKitt para los diferentes proyectos de desarrollo, actualmente se encuentran disponibles ***informes*** que presentan la información en diferentes formatos (PDF, HTML, RTF, etc.) preparados para ser impresos y distribuidos a los distintos interesados en la información que contienen. Adicionalmente a los informes disponibles, es posible desarrollar nuevos informes que se ajusten a las necesidades específicas de una organización o grupo de trabajo.

1.1. Las Transformaciones: el valor añadido en MOSKitt

En gvMétrica se apuesta por una novedosa estrategia de desarrollo de aplicaciones como es el Desarrollo Dirigido por Modelos en el cual las piezas clave son los modelos y las transformaciones de dichos modelos. Los modelos pasan a ser el principal mecanismo de comunicación entre todos los miembros de un proyecto y su evolución a lo largo del mismo (desde el análisis hasta la implantación), se automatiza al máximo incluso llegando a la generación del código de las aplicaciones si es necesario.

Por tanto, para dar soporte a esta estrategia dentro del contexto de gvMétrica, MOSKitt proporciona una serie de transformaciones entre modelos que permiten generar total o parcialmente otros modelos. Por ejemplo, de modelos UML2 a modelos de base de datos (tomando en consideración las clases) o de modelos de procesos a modelos UML2 (convirtiendo las actividades en casos de uso).

Estas transformaciones son configurables de manera que es posible producir distintos resultados atendiendo a las necesidades de los analistas/diseñadores. Todas las transformaciones generan trazas que permiten relacionar los elementos de los modelos de entrada (por ejemplo, una clase UML2) que fueron tomados como

partida para producir los elementos de los modelos de salida (por ejemplo, una tabla en el modelo de base de datos). Con estas trazas y la infraestructura proporcionada por MOSKitt, se produce una sincronización entre los modelos derivados de manera que al modificar el modelo de entrada (por ejemplo, al añadir una nueva propiedad de una clase UML2) se actualiza el modelo de salida (por ejemplo, añadiendo una nueva columna en una tabla).

1.2.- Herramientas para la Generación Automática de Código

El objetivo último de los proyectos de desarrollo de software es la obtención de una aplicación informática en funcionamiento que satisfaga las necesidades de los usuarios. Para facilitar esta tarea, MOSKitt pretende proporcionar herramientas que generen total o parcialmente el código fuente de la aplicación a partir de los modelos de análisis o diseño.

Actualmente MOSKitt permite generar los scripts DDL de creación de la base de datos para varios gestores de datos (MySQL, PostgreSQL y Oracle). Esta generación de DDL es configurable, de manera que pueda adaptarse a las necesidades específicas de cada proyecto u organización.

La generación del código fuente de la funcionalidad de la aplicación para algún lenguaje o plataforma de programación necesita tener asociado una arquitectura en concreto y/o framework de implementación. Es decir, la generación de la funcionalidad de la aplicación a partir de modelos para, por ejemplo, Java puede hacerse siguiendo diversas arquitecturas y varios frameworks de implementación. Actualmente, en la Conselleria de Infraestructuras y Transporte se está trabajando en la generación de aplicaciones para los siguientes frameworks:

- **PHP/gvHidra**: Es un entorno de trabajo para el desarrollo de aplicaciones de gestión en entornos web con PHP siguiendo la guía de estilo de la CIT. Tiene arquitectura web usando el patrón modelo-vista-controlador (MVC) aplicando las tecnologías Phrame, Smarty y Pear.

- **Java/gvNix**: Entorno de trabajo compuesto de distintas herramientas open source Java para desarrollar aplicaciones web J2EE de forma rápida y eficiente de acuerdo a las necesidades particulares de la CIT. La idea central gvNIX es automatizar y simplicar la creación de un esqueleto de proyecto web totalmente funcional y con las características propias de las aplicaciones en la CIT de tal forma que el trabajo del desarrollador consiste en personalizar y ampliar este esqueleto de aplicación. El framework esta formado por un conjunto de documentos y buenas practicas, componentes específicos y ampliaciones o contribuciones al proyecto AppFuse 2 (hasta la versión 0.2) y en el futuro al proyecto SpringRoo (0.3 en adelante).

Aunque el resultado deseable sería llegar a producir el 100% del código necesario para satisfacer las necesidades de los usuarios, los objetivos a corto plazo se centran en generar la mayor parte de la aplicación posible de manera que se facilite la posterior modificación del código generado por parte de programadores que implementen los requisitos específicos de los usuarios (principalmente, reglas de negocio y aspecto de la interfaz de usuario).

2.- MOSKitt como herramienta de Soporte a los Métodos de Desarrollo

Para producir de una manera sistemática y reproducible sistemas software es necesario aplicar métodos de trabajo que definan de manera precisa las tareas, técnicas y herramientas que deben ser utilizadas en la ejecución de un proyecto y los resultados que deben producirse (documentos, modelos, código fuente, etc.). MOSKitt permite la definición de estos métodos de trabajo proporcionando mecanismos para facilitar su ejecución por los participantes en un proyecto.

Gracias a las herramientas que proporciona MOSKitt, los participantes de un proyecto pueden conocer las tareas pendientes de llevarse a cabo y los resultados que debe producir cada tarea. También facilita la ejecución de las tareas, abriendo directamente la herramienta (por ejemplo, un editor de modelos o un editor de textos) que debe utilizarse para llevar a cabo una tarea en concreto.

Además, utilizando la infraestructura de Eclipse, es posible definir guías y asistentes integrados en el entorno para indicar a los participantes de un proyecto los pasos que deben llevar a cabo o las políticas que deben aplicar para realizar una tarea dentro de un método de desarrollo.

3.- MOSKitt como Infraestructura de Desarrollo de otras Herramientas

Durante el desarrollo de las herramientas que proporciona MOSKitt se han utilizado diversas tecnologías en el ámbito del proyecto Eclipse pero, debido al carácter innovador de MOSKitt, también ha sido necesario diseñar e implementar nuevas tecnologías o modificar tecnologías existentes. Estas tecnologías, que constituyen una infraestructura para el desarrollo de nuevas herramientas, se pueden englobar principalmente en las áreas de la construcción de editores gráficos y basados en formularios de modelos, y las transformaciones de modelos.

Haciendo uso de las tecnologías libres desarrolladas o mejoradas en el proyecto MOSKitt es posible incrementar notablemente la productividad en la construcción de herramientas de soporte al modelado de sistemas o, en general, a métodos de producción de software.

4.- Beneficios que aporta MOSKitt y el uso de una Metodología como gvMétrica

MOSKitt ayuda a las organizaciones en sus proyectos de desarrollo de software, tanto en el caso de que se realice la ejecución internamente como en situaciones de subcontratación. Gracias a su enfoque orientado a facilitar la aplicación de métodos de desarrollo de software, MOSKitt proporciona una serie de beneficios que se describen a continuación.

4.1.- Estandarización del trabajo (tareas, recursos producidos, etc.)

Gracias al soporte a la ejecución de métodos de desarrollo es posible definir las tareas que deben llevarse a cabo y los recursos (modelos, documentos, etc.) que se deben producir. Esta estandarización de los procedimientos de trabajo permite encapsular el conocimiento de la organización y, por lo tanto:

- Facilitar la validación del trabajo realizado para comprobar que se ha llevado a cabo correctamente.
- Facilitar el mantenimiento y actualización de los proyectos, ya que toda la organización conoce los recursos que se producen y el contenido de cada uno de ellos.
- Facilitar la formación de los recursos humanos de la organización y la incorporación de nuevo personal a los equipos de desarrollo.

4.2.- Estandarización de las aplicaciones

Habitualmente la estructuración de las aplicaciones dependen en gran medida del desarrollador encargado de llevarla cabo. Gracias a la sistematización del procedimiento de desarrollo y al uso de técnicas de generación automática de código que proporciona MOSKitt, las aplicaciones construidas tienen una estructura interna homogénea, tanto a nivel arquitectural como de diseño. Esta estandarización de las aplicaciones:

- Permite embeber buenas prácticas y evita tener que solucionar en cada sistema desarrollado problemas similares
- Facilita enormemente la depuración y la mantenibilidad del código, ya que se conocen de manera clara las responsabilidad de cada parte del código.

4.3.- Incremento de la productividad, la calidad y la Satisfacción de los usuarios

La aplicación de técnicas de generación de código en MOSKitt proporciona a los equipos de desarrollo un incremento drástico en la productividad y en la calidad de los sistemas que construyen. Gracias a la producción del código de manera automática es posible:

- Focalizar los esfuerzos del proyecto en entender las necesidades de los usuarios y garantizar que el sistema las satisface adecuadamente.
- Evitar que los desarrolladores dediquen una gran cantidad de tiempo en tareas repetitivas, de poco valor añadido y propensas a errores.
- Dedicar más tiempo a la ejecución de pruebas funcionales y a la validación con los usuarios.
- Proporcionar prototipos a los usuarios de manera temprana para recibir *feedback* lo antes posible

5.4.- Adaptación 100% a las necesidades de la organización

Los métodos de producción de software son, en muchos casos, una característica diferenciadora de cada organización. El tamaño de la organización, las características de los sistemas que necesita desarrollar, los

perfiles y experiencia del equipo de trabajo, son factores determinantes que condicionan el método de desarrollo más adecuado en cada entorno. MOSKitt tiene esto en cuenta y permite adaptarse completamente a las necesidades específicas de una organización gracias a que:

- Es un proyecto de software libre y, por lo tanto, puede ser utilizado y extendido sin ningún problema legal
- Está diseñado siguiendo una estructura modular basándose en la arquitectura y mecanismos de extensión que proporciona Eclipse.
- Se ha desarrollado una infraestructura tecnológica para facilitar el desarrollo de nuevas herramientas y la configuración o extensión de las herramientas existentes