

VARIABILIDAD EN REQUISITOS: UN ENFOQUE BASADO EN META-MODELADO Y TRANSFORMACIÓN DE MODELOS¹

Bruno González-Baixauli¹, Miguel A. Laguna¹ y Julio Cesar Sampaio do Prado Leite²

1: Grupo GIRO. Departamento de Informática
Universidad de Valladolid
Campus Miguel Delibes, 47011 Valladolid, España
e-mail: {bbaixauli, mlaguna}@infor.uva.es, web: <http://www.giro.infor.uva.es>

2: Departamento de Informática,
PUC-RIO
Rua Marquês de São Vicente, 225 RDC; CEP 22453-900 Gávea; Rio de Janeiro RJ - BRASIL
e-mail: julio@inf.puc-rio.br, web: <http://www-di.inf.puc-rio.br/~julio>

Palabras clave: Ingeniería de requisitos, Ingeniería de requisitos orientada a metas, Aspectos, Variabilidad, Meta-modelado

Resumen. *En este trabajo se utiliza un meta-modelo para la descripción de la variabilidad durante la fase de requisitos. Dicho modelo se basa en un enfoque orientado a metas, con un especial énfasis en los requisitos no funcionales. Además, hemos aplicado parte de los enfoques de orientación a aspectos para solucionar principalmente dos problemas encontrados en este tipo de modelos: a) la integración de las soluciones de la parte no funcional en la parte funcional; y b) la falta de escalabilidad en el análisis de alternativas debido a la explosión combinatoria del número de variantes. Esta aplicación se resume en una separación explícita de los grafos de metas en modelos, que requiere la definición e implementación de un conjunto de operaciones para la composición automática de modelos. Estas composiciones se han definido como transformaciones de modelos, de forma similar a como se hace en el Desarrollo de Software Dirigido por Modelos (DSDM), pero con la particularidad de que tanto el modelo origen como el destino tienen el mismo meta-modelo.*

¹ Este trabajo ha sido parcialmente soportado por el proyecto MEC/FEDER (TIN2004-03145). El primer autor disfruta de una beca MEC-FPU.

1. INTRODUCCIÓN

Durante los últimos años ha aumentado el interés sobre la variabilidad en los sistemas software. Este interés se debe a la necesidad de proporcionar una mayor funcionalidad, extensibilidad y adaptabilidad al producto software de modo que mejore su competitividad (por ejemplo, en el desarrollo de Líneas de Producto). Inicialmente, este problema se ha abordado en la fase de diseño como una búsqueda de los mecanismos para permitir implementar la variabilidad que va apareciendo, pero sin valorar su necesidad. El estudio de la variabilidad desde la fase de requisitos trata de detectar esa necesidad bajo la premisa de que cuanto antes se detecte, mejor se podrá tratar. Así, existen trabajos que utilizan modelos de *features* [10], de casos de uso [9], o incluso ambos [7]. El problema es que fallan al no permitir el análisis completo de dicha variabilidad, fundamentalmente en el área de los requisitos no funcionales (RNF). Esta limitación en el análisis es muy importante puesto que los RNF son un factor clave en la definición de la variabilidad, por ejemplo al estudiar distintos niveles de seguridad o de fiabilidad. En cambio, nuestra propuesta opta por adaptar mecanismos y técnicas de la Ingeniería de requisitos a la variabilidad, en particular el enfoque orientado a metas (OM). La Ingeniería de requisitos OM se basa en estudiar los requisitos desde un punto de vista intencional, es decir desde los objetivos de los interesados. Este enfoque permite el estudio de alternativas de forma natural, y el análisis del impacto de cada una sobre los RNF utilizando el *NFR Framework* [2]. Además, su aplicación al campo de la variabilidad ha sido ya explorada con éxito en trabajos como [12][4], que utilizan un modelo funcional para explorar la variabilidad y un modelo no funcional (modelos de *softgoal* en el *NFR Framework*) para elegir una de las distintas variantes.

No obstante, al explorar esa línea en trabajos anteriores [4][5], hemos observado varios problemas al utilizar este enfoque. El primero surge al integrar el modelo de metas funcional con el no funcional, puesto que las soluciones de los modelos no funcionales (*operacionalizaciones* en el *NFR Framework*) modifican el modelo funcional, pero no hay un mecanismo para representarlo. Este es un problema muy parecido al que tratan los enfoques orientados a aspectos (OA) [11], donde los aspectos definen un comportamiento similar a aplicar en distintos lugares del código. El otro problema es que la combinación de alternativas provoca una explosión combinatoria, incluso en sistemas relativamente simples [4], lo que afecta a la escalabilidad del análisis de la variabilidad.

La solución que proponemos a estos problemas consiste en aplicar ideas de los enfoques OA. De esta forma, el mecanismo para representar las relaciones entre modelos funcionales y no funcionales, será una relación de tipo aspectual. En cuanto al problema de la escalabilidad, la separación del modelo funcional y no funcional en varios modelos (intereses) independientes, como se hace en los enfoques de aspectos tempranos (*early-aspects*) [1], permite una mayor independencia de cada parte. Esta independencia permite analizar por separado cada modelo, disminuyendo su complejidad, mientras que la composición de intereses permite un análisis global, pero más escalable, ya que se pueden simplificar los intereses (eligiendo entre las distintas alternativas) antes de la composición.

Este trabajo se enmarca dentro de nuestra propuesta para el análisis de la variabilidad teniendo en cuenta los objetivos y deseos (RNF) de los interesados. Para ello nos basamos en

modelos de metas, que representan los intereses del sistema, y en su composición automática en vistas, que son la base del análisis. La propuesta consta de tres fases: a) definición de un meta-modelo y de las reglas de composición de modelos en vistas; b) propuesta de técnicas de exploración y selección de la variabilidad; y c) propuesta de técnicas de configuración de la variabilidad.

Este artículo se centra en la primera fase, es decir en dar soporte a la representación de la variabilidad, para poder analizarla en fases posteriores. Para ello, en la siguiente sección se define un meta-modelo que describe los elementos de modelado y las posibles relaciones entre ellos. En la sección 3 se describen las reglas de composición necesarias para crear vistas a partir de los modelos iniciales y de otras vistas, utilizando para ello un enfoque de transformación de modelos. La sección 4 presenta un ejemplo que ilustra la utilización de la propuesta. Finalmente, se presentan las conclusiones del artículo y el trabajo futuro.

2. META-MODELO PARA EL ESTUDIO DE LA VARIABILIDAD

Nuestra propuesta está basada en modelos. Para definir claramente los elementos utilizados en el modelado, las relaciones entre ellos y las restricciones en su utilización, hemos construido un meta-modelo que permite:

1. Modelar los elementos básicos de la orientación a metas. En este caso, utilizamos como base los elementos definidos en el modelo *V-Graph* [15], que consideramos suficientes para el modelado de la variabilidad.
2. Relacionar las operacionalizaciones de la parte no funcional, con la parte funcional utilizando relaciones aspectuales.
3. Permitir la modularización de los modelos, donde una parte del modelo se modela por separado en otro modelo y el modelo “padre” mantiene una referencia.
4. Crear vistas que permitan integrar varios modelos por medio de la composición de módulos (modelos). Las vistas se componen de referencias a los elementos y relaciones de los modelos de interés, de forma que se puedan eliminar (por ejemplo para la selección de alternativas), sin modificar los modelos base.
5. Establecer relaciones de restricción entre tareas, que especifiquen su compatibilidad, como en los modelos de *features* [10].

En la Tabla 1 se describen de forma muy resumida los distintos tipos de elementos, mientras que la parte principal del meta-modelo se muestra en la Figura 1. Por razones de espacio no se muestran ni las especializaciones de las relaciones OM, ni las restricciones en los tipos de asociaciones, aunque sí se mencionan en la tabla. Una descripción más detallada del meta-modelo y sus elementos se puede encontrar en [6].

Elemento [<i>Meta-clase</i>]	Descripción
Proyecto [<i>Project</i>]	Proyecto de variabilidad orientado a metas. Es una agrupación de modelos.
Modelo [<i>Model</i>]	Modelo del proyecto.
Modelo de Interés [<i>Concern</i>]	Una parte de interés del sistema a analizar. Es una estructura jerárquica de Elementos con un nodo raíz.
Vista [<i>View</i>]	Modelo generado a partir de uno o más modelos.

Elemento [<i>Meta-clase</i>]	Descripción
Elemento [<i>Element</i>]	Elementos del Modelo de Interés. Se describen por un tema (<i>Subject</i>) y cero o más tópicos (<i>Topic</i>).
Meta [<i>Goal</i>]	Objetivo del sistema. Se descomponen en sub-metas hasta llegar a las tareas. Valores: alcanzada / denegada.
Tarea [<i>Task</i>]	Forma de alcanzar una meta / satisfacer un <i>softgoal</i> . Puede ser un algoritmo, proceso, restricción, decisión de diseño... Su presencia o no en una variante determina los valores de metas y softgoals.
Softgoal [<i>Softgoal</i>]	Meta que no tiene un criterio claro y definido de satisfacción. Valores desde claramente satisfecho a claramente insatisfecho.
Descriptor [<i>DescriptionElement</i>]	Unidad de descripción de un elemento. Se organizan en jerarquías de herencia que pueden utilizarse para sugerir relaciones aspectuales.
Relación OM [<i>GO_Relationship</i>]	Relaciones propias de los modelos orientados a metas.
Descomposición [<i>Decomposition</i>]	Relación de jerarquía entre elementos de mismo tipo o de tarea a meta. Subtipos: <i>And</i> , <i>Or</i> , <i>Xor</i> .
Operacionalización [<i>Operationalization</i>]	Indica una posible solución a un <i>softgoal</i> (representada como una tarea). Subtipos: <i>Strong</i> , <i>Weak</i> .
Contribución [<i>Contribution</i>]	Indica como contribuye un elemento a un softgoal. Subtipos: <i>Strong Positive</i> , <i>Weak Positive</i> , <i>Weak Negative</i> , <i>Strong Negative</i>
Relación Aspectual [<i>AO_Relationship</i>]	Indica que un softgoal puede afectar a un elemento funcional, modificando su comportamiento.
Restricción [<i>Constraint</i>]	Relaciona una o más tareas. Subtipos: <i>Requires</i> (1 tarea necesita otras tareas) y <i>MutuallyExclusive</i> (las tareas no pueden estar a la vez)
Referencia a Elemento [<i>Element_Ref</i>]	Referencia a un elemento dentro de una vista.
Referencia a Relación [<i>Relationship_Ref</i>]	Referencia a una relación dentro de una relación. También mantienen los <i>source</i> de la relación, para poder eliminar parte (seleccionar una alternativa).

Tabla 1. Resumen de los elementos del meta-modelo.

3. COMPOSICIÓN DE MODELOS

En esta sección se describen los pasos necesarios para crear nuevas vistas a partir de modelos existentes (intereses o vistas ya creadas). La creación de vistas (o composición de modelosⁱⁱ) es fundamental por dos razones: a) para obtener un modelo global del sistema, puesto que los intereses permiten centrarse en un problema concreto, pero no dan la visión global; y b) para dar visiones particulares a partir de varios intereses que sirvan como base para decidir entre una alternativa u otra.

En nuestro caso, la creación de una vista se realiza a partir de uno o dos modelos. Una vista sobre un único modelo tiene sentido si queremos modificar un modelo de interés antes de componerlo con otros (por ejemplo para seleccionar una alternativa). Como se ha explicado, las vistas contienen referencias a los elementos y relaciones de los intereses, para poder modificarlos sin alterar dichos intereses, pero además pueden tener nuevas

ⁱⁱ Se utiliza “composición” como traducción del término “*weaving*” de la OA puesto que en el caso de modelos no existe tanto un entrelazamiento, que tiene más sentido en el código, sino una composición.

relaciones OM. Estas relaciones pueden aparecer debido a las relaciones aspectuales, ya que la composición de dos modelos entre los que haya una relación aspectual produce la inserción de las operacionalizaciones del softgoal dentro del modelo funcional.

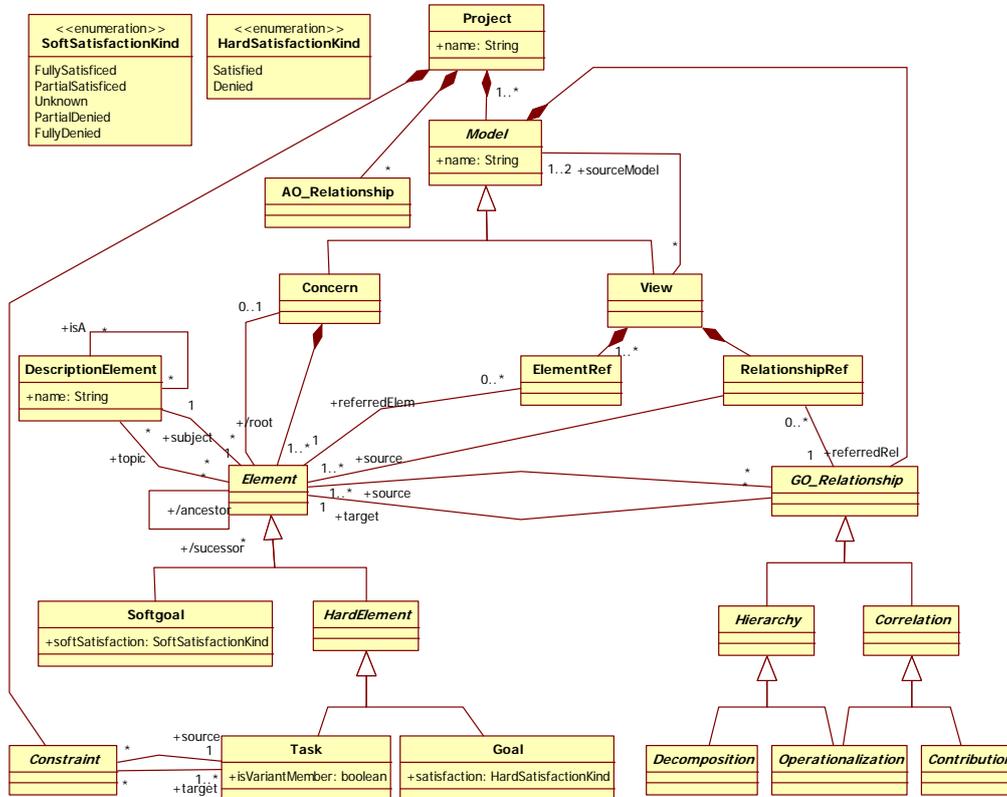


Figura 1. Meta-modelo para el análisis de variabilidad basado en metas y aspectos.

Los pasos necesarios para crear una nueva vista se resumen en cinco:

1. Creación de la vista: se crea una instancia de la meta-clase *View* y enlaces a los modelos que se están componiendo, para mantener la trazabilidad de la vista.
2. Creación de las referencias a elementos: se crean instancias de la meta-clase *ElementRef* para cada elemento en los modelos origen, que refieren a los elementos originales (relación *ReferredElem*).
3. Creación de relaciones a partir de relaciones aspectuales: si existe una relación aspectual entre un *softgoal* de un modelo y un elemento funcional del otro, se crea una nueva relación *OR* donde se incluyen las operacionalizaciones del *softgoal* (elemento fuente).
4. Creación de relaciones *OR*: si los dos modelos tienen relaciones *OR* diferentes (causadas por las relaciones aspectuales), se crea una nueva relación *OR* con los elementos de las dos relaciones para tener una única relación *OR*. En el caso de que ya se haya creado en el paso anterior, simplemente se añaden los elementos que no estuvieran.

- Creación de referencias a relaciones: se crean instancias de la meta-clase *RelationshipRef* para cada relación de los modelos que se están componiendo, excepto para las relaciones OR que ya se hayan creado en los pasos anteriores.

A la hora de definir los pasos anteriores, hemos utilizado transformaciones de modelos, pero con una perspectiva ligeramente distinta a la habitual en DSDM. En nuestro caso no queremos transformar un modelo definido por un meta-modelo en otro distinto (definido a su vez por otro meta-modelo), sino transformar el modelo origen en otro igual al que se añade la nueva vista. Por tanto, los meta-modelos de los modelos origen y destino son el mismo. Aquí es importante distinguir entre el modelo que se transforma, que es una instancia del meta-modelo definido en la Sección 2 y que llamaremos *proyecto*, y los modelos a componer, que son instancias de la meta-clase modelo (intereses o vistas).

En la Figura 2 se muestra un ejemplo de definición de una regla, de una forma similar a QVT [13]: la regla 3.1, que compone modelos de interés con relaciones aspectuales, creando una nueva relación *OR*. Para ello se crea una nueva relación dentro de la vista (*ViewOrRel*) que relaciona uno de los elementos que operacionalizan al elemento fuente (*E_Source*) con el elemento destino de la relación aspectual (*E_Target*). Es necesaria otra regla, la 3.5 (definida en [6]) para crear el resto de relaciones fuente (*source*) entre la relación creada y las operacionalizaciones. La restricción asegura que no se vuelve a crear la descomposición *OR* para ese elemento comprobando que si existe ya una relación *OR* para el elemento destino (*E_Target*), en cuyo caso no se ejecutaría. Las reglas de pasos anteriores habrán creado la vista y referencias a los elementos. El resto de reglas están definidas en [6].

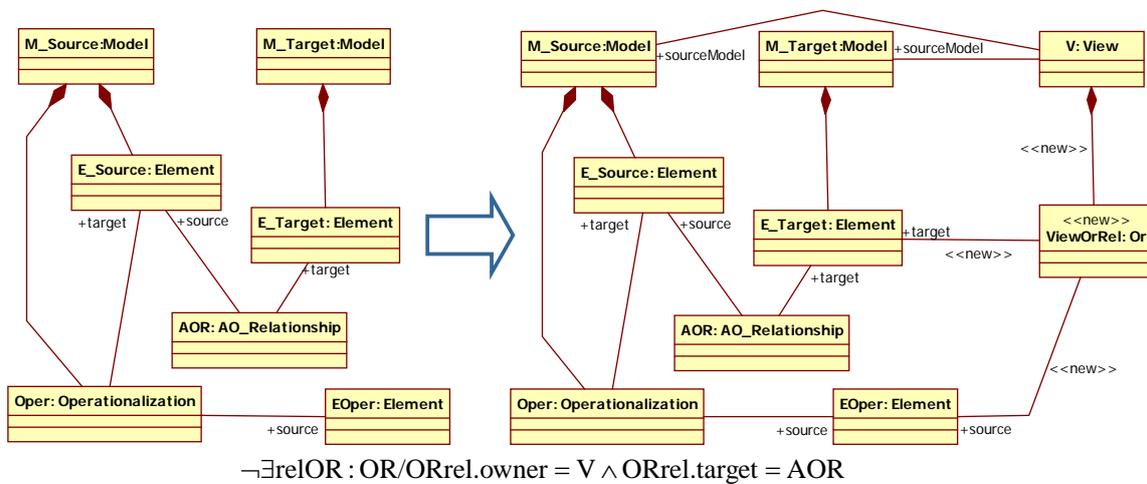


Figura 2: Regla 3.1 para la composición de modelos de interés cuando existe una relación aspectual entre ellos

4. EJEMPLO

En esta sección se utiliza el meta-modelo y las reglas propuestas en el modelado de un comunicador alternativo y aumentativo. La elección de este dominio de aplicación se debe a que tiene un alto grado de variabilidad debido a las distintas discapacidades sobre las que se

También se podría analizar como interaccionan varios *softgoals*, en ese caso se compondrían vistas del estilo a las mostradas en la figura 4, es decir con sus operacionalizaciones.

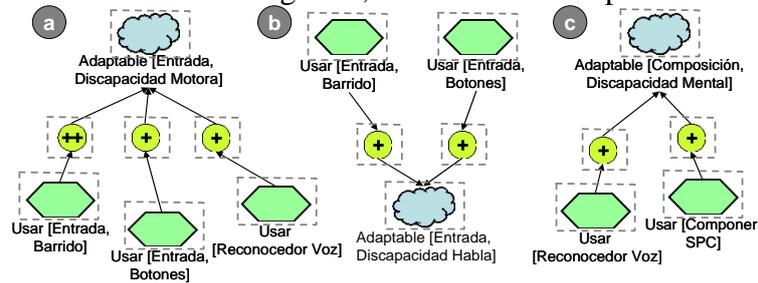


Figura 4: Ejemplos de vistas que muestran las operacionalizaciones para cada *softgoal* mostrada en la figura 3b).

Si hay relaciones aspectuales entre los modelos, también se aplica el paso 3 del algoritmo presentado en la sección anterior (*creación de relaciones a partir de relaciones aspectuales*). La Figura 5 muestra el resultado de componer la parte funcional de la Figura 3 con la vista resultado de componer el resto de intereses. En los primeros pasos, se crean la vista y las referencias y en el tercer paso se crean nuevas relaciones (en este caso las dos relaciones *Or*) para dar una visión global del sistema.

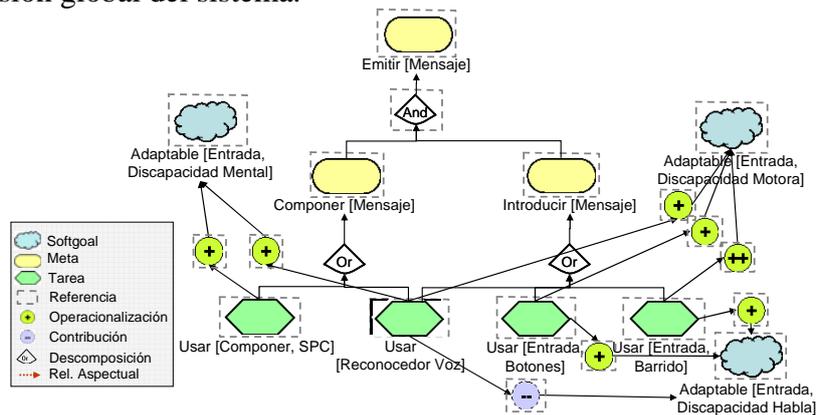


Figura 5: Ejemplo de vista, resultado de componer los modelos de la Figura 3.

Para representar los ejemplos anteriores, hemos utilizado GME (*Generic Modeling Environment*) [3], una herramienta de meta-modelado desarrollada en la Universidad de Vanderbilt. Esta herramienta permite obtener rápidamente un entorno de modelado a partir de un meta-modelo, y definir restricciones que deben cumplir los modelos utilizando OCL sobre dicho meta-modelo. Los modelos desarrollados con esta herramienta nos han permitido observar como se comporta el meta-modelo en la creación de modelos de intereses y en la generación de vistas, puesto que permite exportar a formato XML. En la Figura 6 se muestra una pantalla de ejemplo del entorno resultante de la herramienta, donde se puede ver el proyecto con los distintos Intereses y Vistas y las relaciones aspectuales representadas por relámpagos; la parte funcional (presentada previamente en la Figura 3a); y una vista.

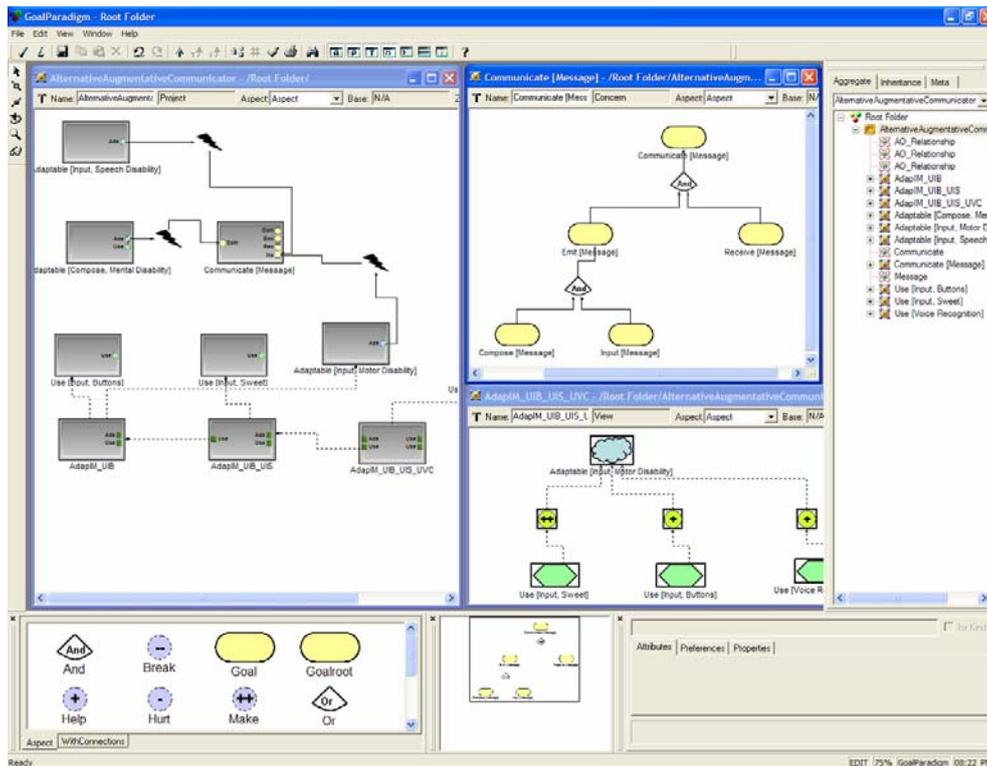


Figura 6: Entorno de modelado obtenido con GME a partir del meta-modelo presentado en la sección 3.

5. CONCLUSIONES

Se ha presentado una propuesta para el modelado de la variabilidad a partir de modelos de metas, donde se aplica un enfoque orientado a aspectos. Para ello se han utilizado técnicas propias del DSDM, como el meta-modelado, definiendo un meta-modelo basado en modelos de interés y vistas, y la transformación de modelos, utilizada para describir las reglas que componen dichos modelos y generan nuevas vistas de forma automática.

Aunque el objetivo principal de la propuesta es constituir un marco de trabajo para el análisis de variabilidad, también se puede aplicar al modelado de requisitos orientado a metas en general, proporcionando un mecanismo para relacionar la parte funcional y no funcional. Otra ventaja es que permite una mejor estructuración y un análisis más preciso, separando el modelado en distintos intereses, que pueden ser compuestos de forma automática. Esta mejor estructuración permite también una mayor reutilización, de forma que intereses lo suficientemente genéricos pueden utilizarse en varios proyectos, por ejemplo utilizando catálogos de RNF como los propuestos en el *NFR Framework* [2].

Las vistas son una mejora substancial para el análisis de la variabilidad, puesto que permiten eliminar elementos, por ejemplo para seleccionar una variante de un modelo, antes de componer con otros, obteniéndose una mejor escalabilidad. Además, se pueden componer sólo ciertos modelos, según los requisitos del usuario final.

Por otra parte, las composiciones se han implementado mediante hojas de estilo XML (XSLT). La hoja de estilo recibe el proyecto en XML y el nombre de los dos modelos a componer como parámetro y crea una nueva vista en el mismo proyecto con el resultado de la composición.

En cuanto a trabajo futuro, se pretende definir técnicas que mejoren el análisis de la variabilidad analizando diferentes RNF y priorizando unos sobre otros. Asimismo, hay que adaptar el trabajo desarrollado para el análisis de la configuración [4] al nuevo meta-modelo y al enfoque OA. Por otro lado, a pesar de que el prototipo desarrollado con GME nos ha permitido utilizar el meta-modelo con ejemplos simples, nuestro objetivo es la implementación de una herramienta de modelado, que incluya la aplicación transparente e integrada de las composiciones definidas, y de soporte a las técnicas de análisis y configuración de la variabilidad.

REFERENCIAS

- [1] Brito, I., & Moreira, A. “*Integrating the NFR Framework in a RE Model*”. Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design, Lancaster, UK. 2004
- [2] Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “*Non-Functional Requirements in Software Engineering*” Kluwer Academic Publishers 2000.
- [3] Generic Modeling Environment (GME 5) Universidad de Vanderbilt (Nashville). Disponible en <http://www.isis.vanderbilt.edu/projects/gme/> (Última visita en mayo de 2006).
- [4] González-Baixauli, B., Leite J.C.S.P., and Mylopoulos, J. “*Visual Variability Analysis with Goal Models*”. Proc. of the RE’2004. Sept. 2004. Kyoto, Japan. IEEE Computer Society, 2004. pp: 198-207.
- [5] González-Baixauli, B., Laguna M., Leite, J.C.S.P. “*Aplicación de un Enfoque Intencional al Análisis de Variabilidad*”. Anais do WER05, Porto, Portugal, Junho, 2005. ISBN 972-752-079-0, pp:100-111
- [6] González-Baixauli, B., Laguna M., Leite, J.C.S.P. “*Meta-modelo para el Análisis de Variabilidad Guiado por Metas*”. TR-GIRO-. Grupo GIRO. Universidad de Valladolid. Disponible en <http://www.giro.infor.uva.es>.
- [7] Griss, M., Favaro, J., and d' Alessandro, M. “*Integrating feature modeling with the RSEB*”. Fifth International Conference on Software Reuse (ICSR), pages 76--85. IEEE Computer Society Press, 1998.
- [8] Halmans, G., and Pohl, K., “*Communicating the Variability of a Software-Product Family to Customers*”. Journal of Software and Systems Modeling 2, 1 (2003), 15--36.
- [9] Jacobson I., Griss M., and Jonsson P.: Software Reuse. Architecture, Process and Organization for Business Success. ACM Press. Addison Wesley Longman (1997)
- [10] Kang, K. C., Cohen, S., Hess, J., Nowak, W. and Peterson, S. “*Feature-Oriented Domain Analysis (FODA) Feasibility Study*”. Technical Report, CMU/SEI-90-TR-21, SEI (Carnegie Mellon), Pittsburgh,
- [11] Kiczales, G. et al. “*Aspect-Oriented Programming*”. In: 11th ECOOP, LNCS 1241, Springer-Verlag, 1997. p. 220-242.
- [12] Liaskos, S., Lapouchnian, A., Wang, Y., Yu, Y., and Easterbrook, S.: “*Configuring Common Personal Software: a Requirements-Driven Approach*”. RE’05. pp. 9-18, Paris, France. 2005
- [13] Object-Management-Group, QVT-Merge-Group: MOF 2.0 Query/View/Transformation Specification version 2.0 – Final Adopted Specification. Object Management Group doc. ptc/05-11-01. (2005)
- [14] Obregón, Centro de Educación Especial. Página Web: <http://www.asprona-valladolid.es/obregon/> (última visita en mayo de 2006).
- [15] Yu, Y., Leite, J.C.S.P., and Mylopoulos, J.: “*From goals to aspects: discovering aspects from requirements goal models*”. 12th IEEE International Requirements Engineering Conference (RE’04). Sep. 6-10, pp:38-47