

Aplicación de un Enfoque Intencional al Análisis de Variabilidad

Bruno González-Baixaui¹, Miguel A. Laguna¹, Julio Cesar Sampaio do Prado Leite²

¹ Departamento de Informática. Universidad de Valladolid
{bbaixauli, mlaguna}@infor.uva.es

² Departamento de Informática. PUC do Rio de Janeiro
<http://www.inf.puc-rio.br/~julio/>

Abstract. El interés por el estudio de la variabilidad, es decir la habilidad de cambio o adaptación de un sistema, ha aumentado significativamente durante los últimos años. Esto se debe a su utilidad en diversos campos, desde la personalización del software a las líneas de producto o la adaptación automática de sistemas. Dicho estudio ha estado centrado principalmente en cuestiones de diseño, y solo recientemente se ha empezado a abordar desde los requisitos del sistema. Las primeras propuestas, basadas en *features*, están claramente enfocadas a facilitar la obtención de la arquitectura del sistema. Debido a la necesidad de contemplar la elicitación de los requisitos del usuario han aparecido nuevas técnicas basadas en casos de uso, que facilitan la elicitación, pero fallan en la representación global de la variabilidad, dificultando la obtención de la arquitectura. También existen enfoques que unen ambas técnicas, compaginándose entre si. El problema común de estas técnicas es que ignoran el tratamiento de los Requisitos No Funcionales (RNF). Si bien los RNF son fundamentales para todo tipo de sistemas, en el caso de la variabilidad lo son aún más, puesto que explican la existencia de distintas variantes para una misma funcionalidad. En este artículo se presenta un ejemplo de aplicación de técnicas de la ingeniería de requisitos orientada a objetivos en el campo de la variabilidad, aplicada a la obtención de los requisitos y a la selección óptima de variantes, teniendo en cuenta los objetivos del usuario tanto funcionales como no funcionales.

1. Introducción

Durante los últimos años los sistemas software cada vez tienden a soportar una mayor variabilidad, definida como la habilidad de cambio o de personalización de un sistema [14]. Esta variabilidad se debe principalmente a dos causas: a las demandas de los usuarios de sistemas más adaptables a sus necesidades (sistemas personalizables [8]) y a la presión del mercado, que hace que los productos software se agrupen en familias para poder reutilizar la mayor parte posible de sus artefactos software [12]. De esta manera, el estudio de la variabilidad está logrando una creciente importancia.

La investigación en el campo de la variabilidad ha estado tradicionalmente ligada a la búsqueda de mecanismos de diseño para implementarla. Así, la mayor parte de las

propuestas no mencionan a los requisitos o hablan genéricamente de una fase de especificación de requisitos. En estos sistemas, los requisitos deben definir las cualidades del sistema con variabilidad. La principal diferencia respecto a los sistemas tradicionales es que deben prestar un especial interés al estudio de la parte común y variable, estableciendo las dependencias entre ellas.

Respecto a las propuestas que tienen en cuenta los requisitos, existen dos grandes familias, las basadas en modelos de características o *features* [10] y las basadas en casos de uso [9][7]. En cuanto a las primeras, utilizan *features*, que se definen como características o conceptos destacados y distinguidos visibles a varios interesados[10]. Estas *features* se organizan en diagramas jerárquicos Y-O, donde los nodos Y dan la parte común y los O la variable. De esta forma, son utilizados para definir la arquitectura de referencia del sistema y los componentes reutilizables instanciables durante el desarrollo de aplicaciones con variabilidad [10]. El problema de esta técnica es que requiere un gran conocimiento del dominio y que está más enfocada hacia la definición de la arquitectura que a la elicitación y definición de requisitos, por lo que son difíciles de aplicar directamente.

Las propuestas para utilizar casos de uso están más enfocadas hacia la elicitación de los requisitos. Por ejemplo, Jacobson [9] utiliza el mecanismo *extend* de los casos de uso tradicionales para representar gráficamente la variabilidad, describiendo los detalles de los puntos de variación en la descripción. En cambio, Halmans y Pohl [7] proponen indicar explícitamente los puntos de variación y las distintas variantes en el modelo de casos de uso para obtener una mejor visión de la variabilidad, para lo que extienden su notación. En ambos casos, las técnicas facilitan la elicitación de requisitos representando la variabilidad en cada caso de uso, pero fallan en la representación global de la variabilidad al estar contenida en diversos documentos (en cada caso de uso).

También existen otras técnicas que unen ambos enfoques como [5], pero en general todos fallan en una cuestión fundamental: el análisis de los requisitos no funcionales (RNF). Este campo ha sido ampliamente tratado dentro de la Ingeniería de Requisitos (IR), una rama de la Ingeniería del Software. Una de las propuestas más exitosas es la IR orientada a objetivos (*goals*), donde los objetivos proporcionan una visión intencional, que permite a los interesados expresar sus necesidades de una manera más natural, centrándose en lo que quieren, es decir sus objetivos, frente a la manera de alcanzarlos o requisitos, que se pueden derivar a partir de los objetivos. Este mecanismo ha demostrado varias ventajas como su mayor estabilidad, la posibilidad de analizar distintas alternativas y verificar la compleción de un conjunto de requisitos con respecto a los objetivos planteados, o un tratamiento de los RNF y los conflictos más natural [16].

En este artículo se muestra un ejemplo de aplicación de los modelos de objetivos al estudio de variabilidad mostrando la elicitación de requisitos, el análisis de variantes y la obtención de una herramienta personalizada para un usuario determinado. Creemos que mediante la introducción de un enfoque intencional se mejora la elicitación de requisitos del sistema variable, y el posterior estudio y selección de la mejor variante para cada caso. El ejemplo desarrollado se integra en el dominio de la Comunicación Aumentativa y Alternativa (CAA), es decir en la creación de herramientas que permitan la comunicación a personas discapacitadas.

El artículo está organizado como sigue: primero se muestra nuestra propuesta para introducir los modelos de objetivos en el análisis de variabilidad. En la sección 3 desarrolla dicha propuesta sobre un caso de estudio. Por último, se termina con las conclusiones y trabajo futuro.

2. Elicitación de variabilidad orientada a objetivos

El enfoque orientado a objetivos tiene varias características que permiten mejorar el análisis de la variabilidad. La primera de ellas es que está orientado hacia el estudio de distintas formas de alcanzar los objetivos. Para ello, se estructuran los objetivos en grafos Y-O, descomponiéndose según sean necesarios o suficientes para alcanzar el objetivo superior. Desde un punto de vista de un único producto, esto se explica porque se escoge la mejor alternativa al finalizar el análisis, en cambio, en sistemas variables esta selección se retrasa hasta fases posteriores. Por tanto, es necesario distinguir entre alternativa y variante, donde una variante es una alternativa que ha sido hallada útil para el sistema variable y que por tanto, implementará dicha variante. Consecuentemente, al considerar una alternativa como no válida, estamos limitando el ámbito (*scope*) del sistema.

La segunda característica que nos interesa es la posibilidad de aplicar las ideas del *NFR Framework* [2] para analizar los RNF. Dicho marco permite representar explícitamente los RNF, lo cual es fundamental en el estudio de la variabilidad, puesto que entendemos que toda alternativa es debida al deseo de satisfacer mejor un conjunto de RNF. De esta forma, una alternativa se convertirá en variante si sus cualidades son interesantes para el sistema con variabilidad. Además, estas relaciones entre la parte funcional y la no funcional son la base para la selección de variantes. Otro factor interesante es que el *NFR Framework* permite dar guías para el diseño de la arquitectura, lo que se traducirá en nuevas variantes durante el diseño.

Uno de los principales problemas para la adopción de una de las metodologías orientadas a objetivos son las diferentes semánticas de los distintos enfoques. En las siguientes subsecciones se describe el modelo de variabilidad que utilizamos, basado en GRL (*Goal-oriented Requirement Language*) [6] y el proceso de análisis de la variabilidad que proponemos.

2.1. Modelado de Variabilidad

GRL es un intento de normalización de los distintos conceptos utilizados en la IR orientada a objetivos que tiene la ventaja de dar soporte al *NFR Framework*, aunque está demasiado enfocado al análisis de negocio. Por todo esto, lo hemos utilizado como base, pero dando un enfoque más orientado al análisis del sistema. Para ello hemos seleccionado un subconjunto de sus relaciones, que en algunos casos hemos renombrado, como se muestra en la Figura 1.

Estos elementos dan lugar a dos modelos, un modelo funcional compuesto por los objetivos funcionales (*hardgoals*), las maneras de alcanzarlos (tareas) y los recursos necesarios, y un modelo no funcional, que describe los RNF y que está formado por objetivos cuya satisfacción no puede establecerse de forma clara (*softgoals*).

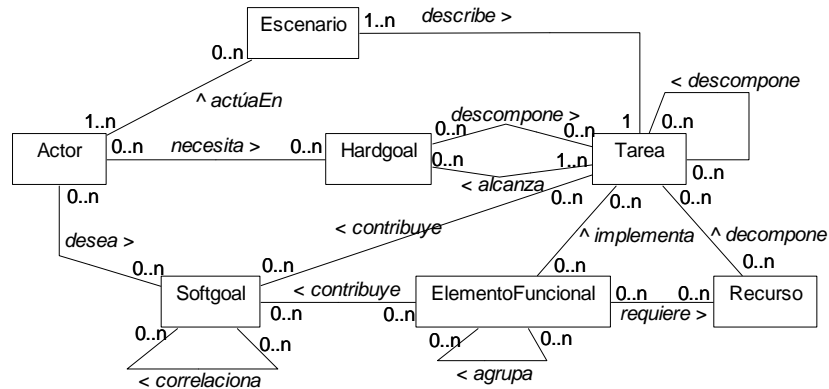


Figura 1. Elementos de modelado y relaciones entre ellos

En cuanto a las relaciones, en el aspecto funcional, GRL no utiliza una descomposición en árbol Y-O sino que sólo permite descomponer *hardgoals* en tareas por medio de la relación de tarea a *hardgoal* *alcanza*. Posteriormente, dichas tareas pueden ser descompuestas en sub-tareas, nuevos objetivos y recursos. Estas dos relaciones complican bastante el modelo (frente a la opción de utilizar árboles Y-O), pero ayudan a mejorar el proceso de elicitación obligando al analista por una parte a pensar siempre en las distintas formas de alcanzar un objetivo, y por la otra, a buscar los objetivos de una tarea si se pretende dar maneras alternativas de realizarla. Además, esta diferenciación permite representar explícitamente la variabilidad, dada por la transición *hardgoal* – tarea (relación *alcanza*).

Para el modelo no funcional utilizamos el enfoque del *NFR Framework* [2]. Dicho enfoque utiliza árboles Y-O que descomponen los *softgoals* en otros *softgoals* según sean necesarios o suficientes. Finalmente, los *softgoal* se operacionalizan en tareas, que representan elecciones de diseño. La última pieza de este modelo son las relaciones que permiten relacionar los *softgoal* entre si (correlaciones) y cómo afectan las tareas a los distintos *softgoals* (contribuciones). Estas relaciones pueden ser positivas o negativas y fuertes o débiles como se muestra en la Tabla 1. Analizando las relaciones entre los distintos elementos es posible determinar si un *softgoal* se cumple y en que grado, incluso de forma automática como se describe en [3].

↑ -	↑ -	↑ ?	↑ +	↑ ++
BREAK	HURT	UNKNOW	HELP	MAKE

Tabla 1. Relaciones entre *softgoals* definidas por el *NFR Framework*

Es importante señalar que la propuesta del *NFR Framework* no contempla la existencia de un modelo de requisitos funcional, por lo que solo existen relaciones dentro del modelo de *softgoals*. En cambio, en esta propuesta se relacionan fácilmente al existir un elemento común entre ambos modelos, las tareas, que por un lado son maneras de alcanzar los objetivos funcionales y por el otro afectan a los *softgoals*.

Existen dos elementos de modelado que todavía no se han descrito: los *escenarios* que se utilizan para describir las tareas, y que también se pueden utilizar para refinar el análisis como se describe en [13][1]; y los *elementos funcionales*, que representan a

los elementos de menor grado de abstracción que implementan las tareas proporcionando de esta forma un mecanismo de trazabilidad. Esta trazabilidad es fundamental para obtener la variante específica. Además, dan una visión más exacta de cómo son afectados los requisitos de calidad frente a las estimaciones hechas durante los requisitos, al permitir establecer nuevas contribuciones a los *softgoal*.

2.2. Proceso de Análisis de Variabilidad

En cuanto al proceso a seguir para realizar el análisis de variabilidad, proponemos los siguientes pasos:

1. Realizar un análisis global del sistema. Este análisis se realiza mediante técnicas de análisis de negocio, en particular, creemos que i* [15] es la más apropiada porque utiliza también un enfoque orientado a objetivos, y básicamente los mismos elementos que los propuestos para el análisis del sistema (actores, objetivos, tareas, y recursos).
2. Analizar los objetivos (funcionales y no funcionales) de los interesados, obteniendo los modelos de objetivos iniciales.
3. Refinar los objetivos iniciales. Para ello se realiza un proceso iterativo por el cual para cada objetivo funcional se obtienen las tareas que los alcanzan (tantas como sea posible); y para cada tarea, se descompone en: las subtareas que deben llevarse a cabo, recursos necesarios y nuevos objetivos. Esto se hace hasta obtener tareas que no se pueden seguir descomponiendo. En cuanto a los objetivos no funcionales, los *softgoals* se descomponen en nuevos *softgoals*, como se describe en [2]. En este punto se deben recalcar varias cuestiones, específicas al análisis de variabilidad:
 - Rol de las aplicaciones existentes o legadas: dichas aplicaciones facilitan el análisis puesto que la descomposición puede obtenerse a partir de cómo se alcanzan los objetivos en cada una.
 - Desarrollar los escenarios que describen cada tarea facilita la descomposición de los objetivos funcionales. Los escenarios deberían desarrollarse incluso para las tareas intermedias, donde serán más abstractos o incluso parametrizados.
 - El análisis de cómo afectan las tareas que alcanzan un determinado objetivo a las distintos *softgoals* ayuda a obtener *softgoals* que no habían sido previamente considerados.
4. Obtener las partes comunes y variables. El análisis de objetivos permite obtener formas alternativas de alcanzar los objetivos de una forma sistemática, pero no qué parte es común o variable, ni sus dependencias. Creemos que la mejor manera de realizar este análisis es mediante el análisis de *features*. Para pasar de un modelo de objetivos a otro de *features*, se deben analizar las distintas tareas y determinar si existen partes comunes o no. Este análisis se realiza a partir de las descripciones de las tareas, realizadas con escenarios. En este análisis es fundamental mantener una trazabilidad entre objetivos/tareas y *features*, lo que nos permitirá obtener las *features* a partir de los objetivos

Estos pasos no deben ser considerados secuenciales, sino más bien iterativos, Así, al refinar los objetivos puede ser necesario a replantearse los objetivos de los interesados

o, más frecuentemente, el análisis de las partes comunes y variables de las tareas puede conducir a nuevas maneras de alcanzar los objetivos.

Una vez analizados los requisitos del sistema, el modelo de *features* sirve como guía para el desarrollo de la arquitectura y el posterior desarrollo del sistema variable. Si se mantiene la trazabilidad entre objetivos y *features* y entre éstas y los assets que las implementan se pueden utilizar los modelos de objetivos como modelo de decisión aplicable a la selección de variantes. Este modelo proporciona un mecanismo de decisión de alto nivel, los objetivos de los usuarios (incluyendo aspectos no funcionales), y permite la aplicación de algoritmos de razonamiento sobre modelos de objetivos como los definidos en [3] para el análisis de las variantes.

De esta forma, el modelo funcional permite seleccionar la funcionalidad requerida, lo que restringe el espacio de variabilidad, de todas las posibles variantes, a las variantes capaces de implementarla. A partir del espacio de variabilidad reducido, el modelo de *softgoals* se utiliza para determinar la mejor variante a partir de los deseos del usuario. En un trabajo previo, se muestra el potencial de utilizar modelos de objetivos para la selección de variantes [4]. Dicho trabajo describe un prototipo que, a partir de un modelo de objetivos Y-O, donde se define la variabilidad funcional; un modelo no funcional (*softgoals*) sobre el que se definen prioridades; y las contribuciones de los elementos funcionales a los no funcionales, se muestra una serie de gráficas que comparan las variantes permitiendo seleccionar la más adecuada.

La utilización del modelo presentado en la Figura 1 permite, mediante el mecanismo de trazabilidad entre los elementos intencionales y los elementos que implementan la variabilidad, obtener una variante implementada del sistema y no sólo una colección de objetivos.

3. Caso de Estudio: comunicación aumentativa y alternativa

Para ilustrar nuestra propuesta, hemos trabajado sobre el dominio de aplicación de la Comunicación Aumentativa y Alternativa (CAA). Este dominio se enfoca hacia la mejora de la capacidad de comunicación de personas discapacitadas con problemas para pronunciar palabras, ya sea parcialmente (aumentativos) o completamente (alternativo).

La elección de este dominio se debe a su evidente interés social y al alto grado de variabilidad debido a las diversas discapacidades que puede tener el usuario y a las maneras de mitigarlas. En este enfoque se intenta dar una visión global, encontrando los objetivos tanto funcionales como no funcionales de los usuarios, teniendo en cuenta tanto las discapacidades de los usuarios como sus preferencias, y los distintos mecanismos, tanto software como hardware, disponibles.

En este artículo reducimos la magnitud del problema centrándonos en las discapacidades de los alumnos del Colegio de Educación Especial Obregón, debidas a la parálisis cerebral. En cuanto al tipo de dispositivos, nos centraremos en ordenadores de mano (PDAs) que proporcionan una gran movilidad a los alumnos, sin requerir una gran inversión.

3.1. Visión general del sistema

En este caso el modelo es muy simple, con tres actores: la persona discapacitada (emisor), el comunicador y la persona que recibe el mensaje (receptor).

La Figura 2 muestra el modelo para el sistema. El emisor depende del comunicador para comunicarse con otras personas (*hardgoal*) por medio de un mensaje (*recurso*) y desea que el sistema sea fácil de usar y tenga la mayor movilidad posible (*softgoals*). El comunicador depende del emisor para obtener el mensaje y del receptor para entregar el mensaje. Por último, el receptor depende del comunicador para que dicha entrega sea lo más sencilla posible.

A partir este diagrama se obtiene una primera idea de donde puede presentarse la variabilidad del sistema. En este caso, en las formas de comunicación, en la manera en que el sistema obtendrá o entregará el mensaje, en el mensaje o en las maneras de permitir que el sistema sea más fácil de usar o permita una mayor movilidad.

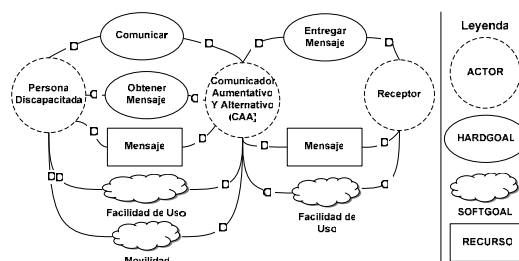


Figura 2. Visión global del sistema

3.2. Análisis de los actores

La principal fuente de variabilidad de este sistema viene dada por las discapacidades de los usuarios, que por un lado obligan a utilizar el CAA y por el otro, pueden dificultar la utilización del sistema. Por ello, lo primero que hay que hacer es analizar los usuarios para hallar sus habilidades y preferencias. En el Colegio Obregón encontramos que las siguientes discapacidades de los alumnos:

- Auditivas: desde problemas leves hasta la sordera total.
- Visuales: desde baja visión a ceguera.
- De manipulación: con un mayor rango de problemas:
 - Parálisis, necesitando incluso dispositivos especiales para señalar.
 - Poca precisión en los movimientos.
 - Poca amplitud.
 - Escasa velocidad.
 - Movimientos involuntarios.
- Habla: desde problemas en la pronunciación de fonemas, ritmo,... hasta la imposibilidad de emitir sonidos.

Para tratar tanto habilidades como preferencias (en este caso movilidad, rapidez de uso, facilidad de aprendizaje y flexibilidad) utilizamos un enfoque similar a [4], donde se incluyen en el modelo de *softgoals*. De esta forma, las distintas discapacidades del usuario son contempladas en el *softgoal* accesibilidad.

3.3. Análisis de softgoals

A partir de los *softgoals* iniciales y de la información de los actores se definen los *softgoals*. Por ejemplo la Figura 3 muestra la descomposición inicial del *softgoal* Facilidad de Uso centrada en el emisor.

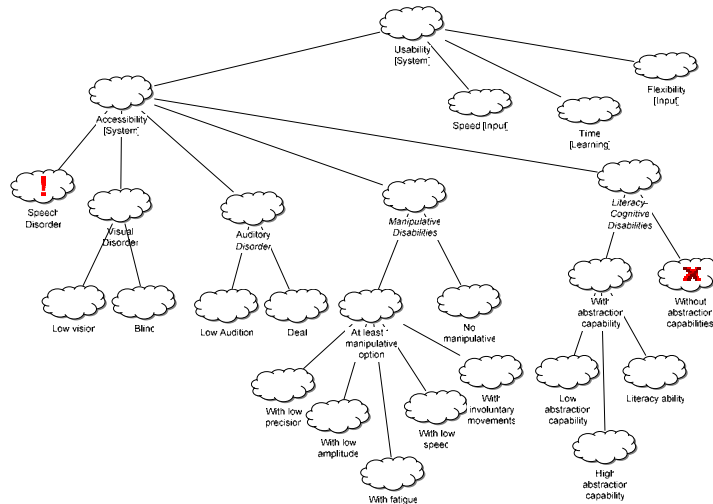


Figura 3. Descomposición del *softgoal* facilidad de uso (centrada en el emisor)

En este ejemplo hay que distinguir entre el emisor y el receptor. Para el emisor, lo fundamental es tener en cuenta la accesibilidad, entendida como facilidad de uso para personas con algún tipo de discapacidad, aunque también se encuentran otros elementos como la rapidez de entrada, el tiempo de aprendizaje o la flexibilidad (que sea capaz de transmitir el mayor número de mensajes posible). En cambio, para el receptor la adaptabilidad no será tan importante y nos podemos centrar en conceptos como su comodidad, la calidad de la salida,...

Hay que tener en cuenta que el diagrama anterior es una descomposición de bastante alto nivel. Si seguimos descomponiendo, muchos de los *softgoals* de más bajo nivel tendrán que ver con decisiones de diseño como se puede ver en la Figura 4. Estas opciones de diseño definidas durante el análisis de requisitos permiten razonar mejor las decisiones de diseño, e incluso introducir nuevas variantes durante las fases posteriores a la definición de los requisitos.

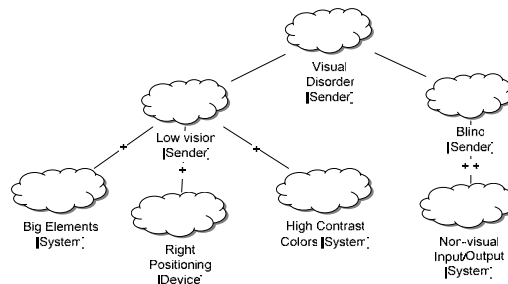


Figura 4. Descomposición del *softgoal* Adaptabilidad para Discapacitados Visuales.

Por último, a estos *softgoals* específicos se pueden añadir otros más generales como coste, rendimiento,... o relativos a la variabilidad como escalabilidad, flexibilidad,... que ayuden a tomar decisiones durante el diseño. En todo caso, existen catálogos de *softgoals*, que indican descomposiciones típicas y que son fáciles de reutilizar gracias al alto nivel de abstracción de los *softgoals* [2].

3.4. Análisis funcional

Paralelamente al estudio de los *softgoals* debemos determinar los requisitos funcionales por medio de las tareas que debe realizar el sistema. Para ello se analizan las maneras de alcanzar los distintos objetivos del sistema. En nuestro caso, obtener mensaje y entregar mensaje. Una parte del modelo, centrada en la entrada del mensaje (*hardgoal* obtener mensaje) se muestra en la Figura 5 (los elementos grises indican que no muestran los subárboles), donde también se muestran parte de las contribuciones de las tareas a los *softgoals*.

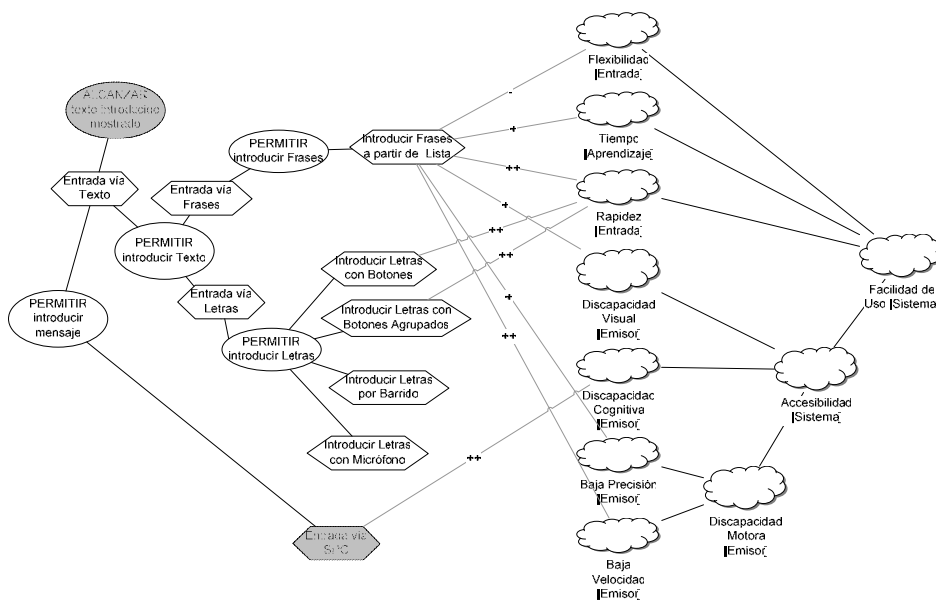


Figura 5. Parte del modelo de objetivos para el objetivo obtener mensaje y contribuciones a los RNF

Como se puede observar, el objetivo principal se va descomponiendo, obteniendo cada vez tareas más específicas, hasta llegar a las tareas básicas del sistema. Al mismo tiempo, se va relacionando cada transición *goal*-tarea con los *softgoals* que explican la existencia de distintas tareas para alcanzar un *goal*. Por ejemplo, la entrada de datos vía texto se explica por una mayor rapidez, pero requiere un mayor nivel cognitivo (capacidad de leer y escribir), en cambio, la entrada vía símbolos pictográficos para la comunicación (SPC) permite a un usuario con una mínima capacidad de abstracción comunicarse, pero a costa de una menor velocidad.

3.5. Modelo de *features*

El siguiente paso es el análisis de la parte común y variable de las tareas. A partir del modelo anterior y mediante el estudio de las descripciones de las tareas obtenemos el siguiente subárbol de *features* (Figura 6), que separa por un lado el mecanismo de entrada (botones, barrido, lista,...) y por el otro el elemento a introducir (letras, frases,...). La trazabilidad con el modelo de objetivos se obtiene por medio de los la agrupación de elementos funcionales (en este caso *features*) representado por el símbolo de tarea gris, que implementa una tarea de dicho modelo de objetivos, en este caso *Introducir Letras con Botones Agrupados*.

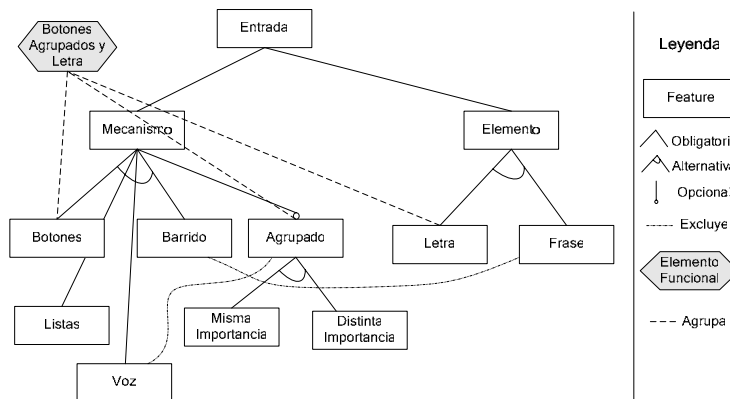


Figura 6. Árbol de *features* correspondiente a Figura 5

El análisis de *features* puede ayudar a encontrar nuevas tareas no contempladas anteriormente, así por ejemplo se podrían agrupar los elementos en el mecanismo de barrido o en el de las diagonales.

3.6. Selección de Variantes

El último paso es obtener las variantes óptimas para el sistema, para ello utilizaremos la aplicación desarrollada en [4]. Esta aplicación utiliza un modelo muy simple para la parte funcional es un árbol Y-O, donde no se distingue entre *hardgoals* y tareas, ni se permiten recursos ni elementos funcionales. En cualquier caso, la conversión para este ejemplo es muy simple y solo se necesita convertir todos los elementos a *goals* y, en cuanto a las relaciones, las *alcanza* se convierten en O y las *descomposición* en Y.

En cuanto al usuario, el alumno elegido presenta una gran discapacidad motora que le obliga a usar una silla de ruedas, aunque puede utilizar la mano de forma básica, con suficiente precisión y velocidad para señalar, pero no para realizar movimientos complejos. En relación a la capacidad cognitiva, es capaz de leer sin problemas, y tiene un ligero problema visual, que tampoco es demasiado grave para la utilización del sistema. Hablando con su tutor, llegamos a la conclusión de que el sistema tenía que implementar dos formas de comunicación, una flexible, para lograr una comunicación potente y otra rápida para expresar cosas importantes. Debido a que las discapacidades relacionadas con el uso del comunicador no son muy graves se dio prioridad a las preferencias del usuario. Las prioridades se pueden ver en la Tabla 2.

Accesibilidad [Emisor]			Rapidez [Entrada]	Tiempo [Aprendizaje]	Flexibilidad [Entrada]
Discapacidad Visual	Baja Precisión	Baja Velocidad			
10	15	15	30	15	15
			15	15	30

Tabla 2. Prioridades sobre los softgoal en %

Con estas prioridades y aplicando la herramienta, se llega a la conclusión de que las mejores variantes son utilizar botones con letras agrupadas según uso en el caso de flexibilidad y frases para cuando se requiere velocidad, como se muestra en la Figura 7.

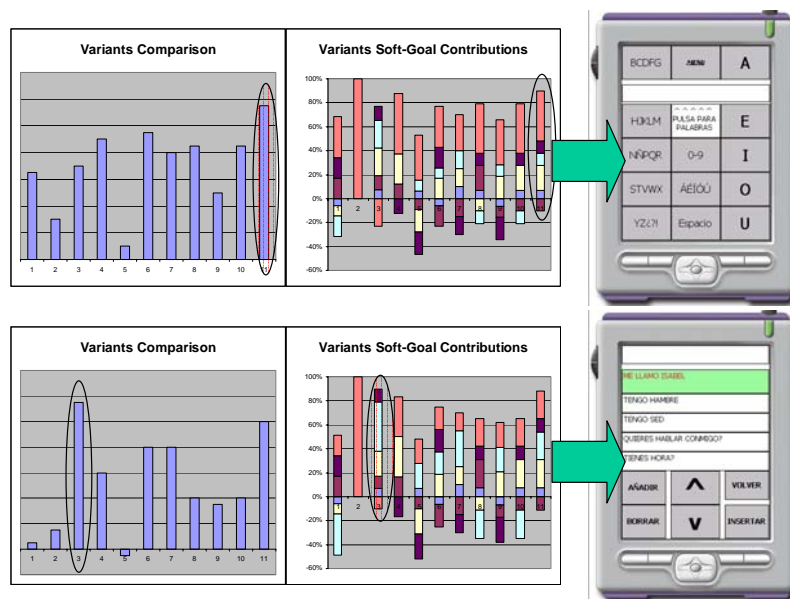


Figura 7. Resultados de la herramienta para las variantes y pantallas de la herramienta implementada (entrada de datos por botones agrupados por uso y por frases predefinidas)

4. Conclusiones

En este artículo se ha mostrado un ejemplo de cómo se pueden aplicar técnicas de la IR orientada a objetivos al estudio de la variabilidad. Para ello se ha desarrollado un modelo que define los elementos a utilizar, prestando especial hincapié en los RNF, por lo general no tratados en otros enfoques. El coste extra derivado de un análisis más detallado se compensa porque: facilita el proceso de elicitación al proporcionar un mecanismo sistemático para el estudio y organización de las distintas alternativas para cumplir los objetivos del sistema y utiliza un punto de vista más abstracto (los objetivos). Además, la aplicación del *NFR Framework* [2] ayuda en la toma de decisiones durante el diseño. Por último, proporciona un modelo de decisión para la selección de variantes a partir de un nivel de abstracción más comprensible por los usuarios, los objetivos, que además integra la parte no funcional.

La utilización de un enfoque orientado a objetivos, que no requiere extensiones para analizar la variabilidad como en los casos de uso, permite la aplicación directa de nuevas técnicas recién desarrolladas dentro de IR orientada a objetivos, como la presentada por Y. Yu en [17] para obtener aspectos.

Las ideas expresadas en este artículo constituyen el soporte para al desarrollo de herramientas que, siguiendo la idea de [4], permitan la selección óptima de variantes. En este sentido, debido a que el modelo está basado en GRL es posible utilizar herramientas que permiten el modelado con GRL, como el entorno libre *openOME* [11] como base.

Por último, estamos estudiando la ampliación del campo de estudio para comprobar la escalabilidad del análisis, así como la aplicación a otros dominios de aplicación.

Agradecimientos

Este artículo ha sido parcialmente financiado por el proyecto MEC/FEDER TIN-2004-03145. El primer autor disfruta de una beca financiada por la Junta de CyL y fondos FEDER. También deseamos agradecer el apoyo prestado por los responsables del Colegio de Educación Especial Obregón en el estudio del dominio y a los alumnos Fermín Juan y Virginia Vilorio por implementar la aplicación final.

Referencias

- [1] Antón, A.I., Carter, R., Dagnino, A., Dempster, J., and Siegel D. "Deriving Goals from a Use-Case Based Requirements Specification." *Requirements Engineering Journal*, Springer-Verlag, Volume 6, pp. 63-73, May 2001.
- [2] Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. "Non-Functional Requirements in Software Engineering" Kluwer Academic Publishers 2000.
- [3] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastián, R. "Reasoning with goal models" 21st Int. Conf. Conceptual Modeling (ER 2002), Tampere, Finland, Oct. 2002.
- [4] González-Baixauli, B., Leite J.C.S.P., and Mylopoulos, J. "Visual Variability Analysis with Goal Models". RE'04. Sep. 2004. Kyoto, Japan. IEEE Computer Society. pp: 198-207.
- [5] Griss, M., Favaro, J., and d' Alessandro, M. "Integrating feature modeling with the RSEB". 5th Int. Conf. on Soft. Reuse, pp: 76-85. IEEE Computer Society Press.
- [6] G.R.L.; Goal-oriented Requirement Language. University of Toronto, Canada. Available at <http://www.cs.toronto.edu/km/GRL/>.
- [7] 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.
- [8] Hui, B., Liaskos, S., Mylopoulos, J., "Requirements Analysis for Customizable Software: a Goals - Skills - Preferences Framework", RE'03, Monterey Bay, USA, Sep. 2003, pp.117-126
- [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". CMU/SEI-90-TR-21, SEI (Carnegie Mellon), 1990.
- [11] OpenOME tool project web page. <http://www.cs.toronto.edu/~yijun/OpenOME.html/>.
- [12] Parnas, D.L. "On the Design and Development of Program Families," *IEEE Trans. on Soft. Eng.* 2(1), Mar. 1976, pp: 1-9
- [13] Rolland, C. and Prakash, N. "From conceptual modelling to requirements engineering". *Annals of Soft. Eng.*, 10. pp: 151-176, 2000.
- [14] van Gorp, J., Bosch J. and Svahnberg, M. "On the notion of variability in software product lines", Working Int. Conference on Software Architecture, 2001. pp: 45-54.
- [15] Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", 3rd IEEE Int. Symp. on Requirements Engineering (RE'97) Jan 1997, USA. pp:226-235.
- [16] Yu, E., Mylopoulos, J., "Why goal-oriented requirements engineering", 4th Int. Workshop on Requirements Engineering, Jun. 8-9 1998, Pisa, Italy. pp. 15-22
- [17] Yu, Y., Leite, J.C.S.P., and Mylopoulos, J. "From goals to aspects: discovering aspects from requirements goal models". RE 04. Sep, pp: 38-47.