

Integración de mecanismos de adaptación de contenidos y generación de IUs plásticas

Montserrat Sendín-Veloso¹, Nuria Medina-Medina², Marcelino Cabrera-Cuevas², Víctor López-Jaquero³, Lina García-Cabrera⁴

Dept. de Informática e
Ingeniería Industrial
Escuela Politécnica Superior
Univ. de Lleida
25001 Lleida
m.sendin@diei.udl.cat

Dept. de Lenguajes y
Sistemas Informáticos
ETS Ingeniería Informática
y de Telecomunicaciones
Univ. de Granada
18071 Granada
nmedina@ugr.es
mcabrera@ugr.es

Dept. de Sistemas
Informáticos
Escuela Politécnica Superior
de Albacete
Univ. de Castilla-La
Mancha
02071 Albacete
victor@dsi.uclm.es

Dept. de Informática
Campus Las Lagunillas
Univ. de Jaén
23071 Jaén
lina@ujaen.es

Resumen

La Web proporciona cada vez más aplicaciones y sitios destinados al usuario final cuyas características (por ejemplo, limitaciones físicas y/o psíquicas), conocimientos, preferencias, necesidades y objetivos pueden ser muy diversos. Por otro lado, se amplía cada vez más el tipo de dispositivos desde los que se puede acceder a Internet, los cuales tienen limitaciones computacionales y restricciones importantes como pantallas más pequeñas y teclados con menor funcionalidad. En consecuencia, se hace necesario proporcionar sitios Web más usables, que puedan adaptar tanto sus interfaces como sus contenidos teniendo en cuenta quién accede y qué dispositivo utiliza. En este trabajo, se presenta una arquitectura en la que se han integrado los modelos SEM-HP y AB-UIDE que personalizan y adaptan de forma dinámica el contenido y la interfaz de una aplicación Web respectivamente. Ambos modelos proponen una descomposición en diversos niveles de concreción, con el fin de separar las partes o capas de software que exhiben diferentes tipos o grados de adaptación (adaptación estática y dinámica) y establecen distintos niveles de abstracción para una misma capa software (adaptación en el proceso de diseño y durante la navegación). Finalmente se esboza la aplicación de nuestro trabajo en la plataforma hipermídia Sc@ut, la cual permite crear sistemas de comunicación alternativos y aumentativos para personas discapacitadas, como por ejemplo niños autistas.

1. Introducción

La audiencia potencial de un sitio Web es amplia y muy diversa. Cada uno de los usuarios busca algo diferente y puede tener necesidades únicas. Tradicionalmente se ha intentado dar respuesta a este problema generalizando con usuarios tipo. Sin embargo, esto puede desestimar a un pequeño porcentaje pero un gran número de navegantes que no encajan en ninguna categoría. Una forma más efectiva de llegar a estas audiencias en un entorno Web son los sistemas hipermídia adaptativos. De forma breve, un sistema adaptativo es un sistema que contiene un modelo de usuario y es capaz de adaptar el contenido a ese modelo de usuario. Por otro lado, son cada vez más el tipo de dispositivos con acceso a Internet que presentan limitaciones computacionales y limitaciones para la interacción (pantallas pequeñas, teclados con menor funcionalidad, etc.). Esta situación, hace necesaria la adaptación también de la interfaz a estos dispositivos con restricciones de espacio y modo de interacción.

El campo de la *plasticidad de las interfaces* es un área de la Interacción Persona Ordenador dedicada al estudio, descripción y catalogación de la problemática derivada de la diversidad de contextos de uso, en la medida que proporciona soluciones económicas y ergonómicas que facilitan la adaptación de los sistemas interactivos a dicha diversidad. Así, se define una *interfaz plástica* como una interfaz capaz de proporcionar adaptación a distintos contextos de uso mientras

preserva un conjunto predefinido de propiedades de usabilidad.

2. Arquitectura general de adaptación

El objetivo de nuestro trabajo es, por tanto, permitir una adaptación integral que contemple tanto la adaptación de la interfaz a través de la cuál el usuario accede a la información, como la manera en que dicho usuario manipula y percibe esa información.

Para ello, proponemos una arquitectura en capas que permite separar los diversos aspectos que son necesarios para realizar la adaptación, aumentando así las posibilidades de la misma. Tal y como se muestra en la figura 1, la arquitectura propuesta se divide verticalmente en dos niveles paralelos: Interfaz y Contenido. En cada nivel

Concreta del nivel correspondiente, interfaz o contenido, teniendo en cuenta los factores concretos mencionados anteriormente y otros similares. Esta separación nos permite, por ejemplo, obtener a partir de una presentación abstracta diferentes presentaciones concretas.

Finalmente, la unión de una presentación concreta de la interfaz con una presentación concreta del contenido supone un sistema totalmente adaptado que, además, va a ser capaz de cambiar dinámicamente si los aspectos concretos citados cambian durante su uso. Por ejemplo, si el usuario cambia de dispositivo, elige un modo de interacción diferente o modifica su conocimiento, preferencias o intereses respecto a alguna información, la presentación concreta que usa será modificada consecuentemente en ambos niveles.

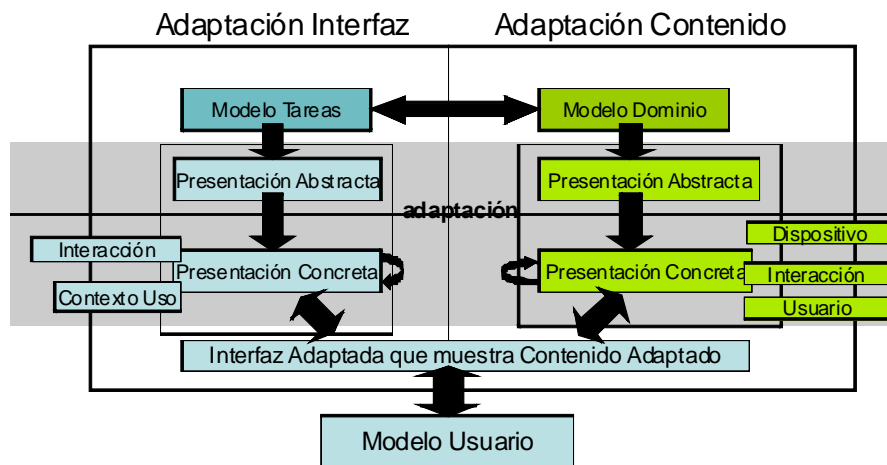


Figura 1. Arquitectura en capas de doble nivel.

existen tres capas: Modelado, Presentación Abstracta y Presentación Concreta. La capa de Modelado permite representar convenientemente los elementos de la interfaz o del contenido que van a ser susceptibles de adaptación. La capa de Presentación Abstracta realiza una primera aproximación a la adaptación independiente de detalles concretos como la plataforma, el dispositivo, el modo de interacción, o las características y preferencias del usuario (contempladas en un modelo de usuario común). Una presentación abstracta se refina posteriormente en la capa de Presentación

2.1. Adaptación de la interfaz

Con el fin de sistematizar la generación de interfaces surgen las primeras técnicas de modelado de la interfaz, reduciendo el esfuerzo requerido por el desarrollador para obtener distintas versiones de una interfaz. Estas técnicas han ido evolucionando hacia lo que se conoce comúnmente como MB-UIDE (del término inglés *Model-Based User Interface Development Environment*) o desarrollo de interfaces basado en modelos. Con el tiempo, las herramientas MB-

UIDE van incorporando un conjunto más diverso de metas y aspectos a considerar, explotando un rango mucho más amplio de modelos –usuario, plataforma, etc.–, convirtiéndose en el enfoque por excelencia para el desarrollo sistemático de interfaces plásticas. Además, se va experimentando un incremento progresivo en el nivel de abstracción.

Los métodos basados en modelos [12] proporcionan un mecanismo para diseñar y desarrollar interfaces partiendo de una especificación única de la misma a un alto nivel de abstracción, suficientemente flexible para afrontar múltiples variaciones. Esa especificación genérica se construye utilizando modelos declarativos que describen tanto los aspectos estáticos como los dinámicos de una interfaz, así como todos los factores relevantes de la interfaz, reuniendo los distintos requisitos de cada **contexto de uso**. La interfaz es derivada a partir de la explotación de esos modelos declarativos hasta la obtención de las distintas interfaces requeridas.

Uno de los pilares del proceso de diseño que caracteriza un enfoque basado en modelos es que se fomenta la separación de conceptos aplicando un diseño progresivo de la interfaz a través de sucesivos niveles o capas de abstracción.

La herramienta desarrollada, denominada AB-UIDE [7][8], se estructura en cuatro capas de abstracción, tal y como se propone en el proyecto Europeo CAMELEON [1]. De la más abstracta a la más concreta son:

(1) *Capa superior de abstracción* en la que interviene el **modelo de tareas**, donde se consideran las actividades a llevar a cabo por los usuarios. En AB-UIDE se conjugan tres notaciones: los diagramas de estado [5], los ConcurTaskTrees [11] y las herramientas abstractas de interacción [2].

(2) *Capa abstracta*, que maneja una especificación de la interfaz de alto nivel independiente de la modalidad y de la plataforma, la denominada **presentación abstracta**. En AB-UIDE se utilizan los *Objetos Abstractos de Interacción* (OAI) propuestos para UMLi en [13]. Estos objetos han sido enriquecidos con un nuevo tipo de OAI: el denominado *selector*. La presentación abstracta obtenida se almacena en usiXML [6], un lenguaje de descripción de interfaces basado en XML.

(3) *Capa concreta*, que proporciona una descripción lógica más refinada, y por lo tanto dependiente de la modalidad denominada **presentación concreta**. Constituye una instancia o representación concreta suficientemente detallada de la IUA, de bajo nivel, independiente de la plataforma destino. En AB-UIDE se ha elegido el modelo de presentación concreto propuesto para usiXML. Esta última descripción no consiste en una versión operativa, sino que es una descripción independiente de la plataforma. Todavía se requiere un último paso de concreción para seleccionar las componentes de interfaz finales propias de la plataforma objetivo, las cuales van a formar parte del código ejecutable.

(4) *Capa física*, que es el que obtiene la interfaz final o **interfaz adaptada**, directamente ejecutable. Actualmente se han desarrollado tres *renderers* que permiten generar las presentaciones finales con las que el usuario interactuará.

2.2. Adaptación del contenido

Para adaptar el contenido que se muestra en la aplicación se siguen las bases teóricas propuestas y formalizadas en el modelo semántico y evolutivo para sistemas hipertexto adaptativos, SEM-HP [9][10].

Para ello, el contenido, esto es, el conjunto de información que se mostrará, es previamente dividido en trozos de información a los que llamamos unidades de contenido o ítems. Por ejemplo, en una aplicación web, cada unidad de contenido será finalmente un nodo o página web.

Puesto que la adaptación que se va a realizar es una adaptación semántica, el autor debe crear, en la primera capa, un mapa conceptual del **modelo de dominio**. En dicho mapa se hacen explícitos los conceptos subyacentes a la información y las relaciones existentes entre dichos conceptos. Por su parte, las unidades de contenido o ítems de información se catalogan indicando el concepto o conceptos que tratan. De manera que tenemos relaciones entre conceptos y también relaciones entre conceptos e ítems. Estas últimas relaciones se enriquecen indicando la función o rol del ítem de información respecto al concepto, por lo que se denominan relaciones funcionales. Por ejemplo, un ítem de información puede explicar un concepto o definirlo, ejemplificarlo o verter una opinión personal sobre

él, etc. Por supuesto, un ítem puede tratar varios conceptos y estar relacionado con todos ellos, en cada caso con el rol correspondiente.

El mapa conceptual se representa mediante una red semántica, en la que se distinguen gráficamente dos tipos de nodos: los conceptos modelados como círculos y los ítems representados como rectángulos. Las relaciones conceptuales se dibujan con flechas que indican el sentido de la relación entre dos conceptos y las relaciones funcionales con líneas que unen un ítem a un concepto. Ambos tipos de relaciones se etiquetan semánticamente sobre la red.

En la segunda capa, el autor puede preparar una o varias presentaciones del contenido que el modelo de dominio recoge y estructura en forma de red semántica. Cada una de estas presentaciones se denomina **presentación abstracta** y se centra en una parcela o subdominio del modelo de dominio completo, por lo que supone una visión parcial del contenido. En definitiva, una presentación es un subconjunto de la red semántica formado sólo por aquellos conceptos, ítems y relaciones que el autor selecciona para esa vista.

A partir de las presentaciones abstractas definidas en la capa anterior, en la tercera capa, el autor crea presentaciones concretas que serán utilizadas por los usuarios de la aplicación. Cada **presentación concreta** se basa en una presentación abstracta, que es elegida según el subdominio que interesa al usuario y su experiencia previa en el mismo. La presentación elegida es adaptada en tres fases: adaptación al dispositivo, adaptación al tipo de interacción y adaptación a las características dinámicas del usuario.

La adaptación al *dispositivo* se centra principalmente en el tamaño de la pantalla donde van a visualizarse las unidades de contenido. Por ello, cada ítem ha sido previamente fragmentado por el autor, asociando a cada fragmento una condición de visualización que indica el tipo o tipos de dispositivos en los que ese fragmento se debe mostrar. Así, para dispositivos con limitaciones importantes de espacio, como por ejemplo una PDA, el autor incluirá sólo los fragmentos más significativos de cada unidad de contenido.

La adaptación al tipo de *interacción*, hace referencia a la forma en que el usuario accede a la información y puede ser: por contenido o por

conceptos. En el primer caso, se proporciona al usuario la red semántica de la presentación, mientras que en el segundo sólo los conceptos y relaciones conceptuales existentes en la misma. En ambos casos, se usa una representación gráfica o textual según las limitaciones del dispositivo. Pero, en el primer caso, el usuario selecciona un ítem y obtiene el contenido del mismo, mientras que en el segundo caso el usuario selecciona un concepto y obtiene una composición con todas las unidades de contenido que lo tratan. Dentro de cada tipo de interacción existen dos posibilidades: interacción libre e interacción restringida. En el primer caso, cualquier selección del usuario se satisface, mientras que en el segundo sólo se satisfacen aquellas para las que el usuario cumple las restricciones impuestas. Las restricciones pueden ser a su vez de dos tipos: de orden o de conocimiento.

La adaptación al *usuario* interviene cuando la interacción elegida es restringida, de modo que se decide si el usuario puede o no ver un ítem o concepto según los ítems o conceptos que ha visto previamente (restricciones de orden) o según el conocimiento que tiene sobre ellos (restricciones de conocimiento). Así, entre otra información, el modelo de usuario almacena y refleja los elementos vistos por el usuario y el conocimiento que ha adquirido con ello. También durante la interacción por conceptos, la composición de las unidades de contenido de un concepto se realiza de acuerdo a las preferencias del usuario. Por ejemplo, el usuario puede indicar que prefiere que se muestren primero los ítems que describen, explican o definen el concepto y que no se muestren los que sean opiniones personales. Puesto que esta adaptación depende de características dinámicas del usuario, una presentación concreta cambia constantemente durante su uso.

3. La Plataforma Sc@ut

La plataforma Sc@ut [11] es un conjunto de herramientas que permite crear sistemas de Comunicación Alternativos y Aumentativos destinados a ayudar a personas con problemas de comunicación temporales o permanentes. En la actualidad, nos estamos centrando en niños autistas pero se puede aplicar a otras discapacidades como parálisis cerebral y disfasia.

Puesto que el sistema de comunicación que se crea con Sc@ut está basado en plantillas hipermedia, hemos considerado interesante evaluar sobre Sc@ut la utilidad de la arquitectura de doble nivel que proponemos para adaptar interfaz y contenido, y que ha sido introducida en la sección 2. De este modo, los resultados obtenidos sirven como una primera aproximación o ejemplo concreto de la adaptación integral en aplicaciones Web.

3.1. La Adaptación en Sc@ut

Esta plataforma está basada en una arquitectura de doble nivel: El comunicador y el meta-comunicador. El meta-comunicador crea y adapta el comunicador con el que interactúa el diseñador (el docente o los padres). El comunicador lo usa el discapacitado (niño autista) que interactúa con él para comunicarse.

Con el meta-comunicador (Figura 2) que se ejecuta en un PC se intenta modelar al usuario/niño teniendo en cuenta sus necesidades y destrezas comunicativas, sus conocimientos, los ambientes en los que se desenvuelve y sus objetivos educativos. Partiendo de este perfil de

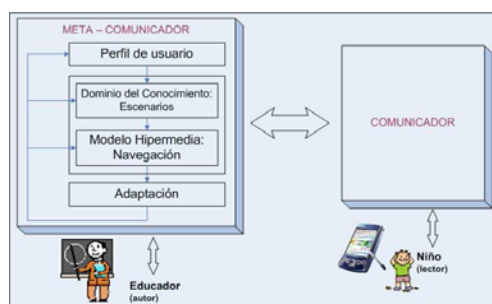


Figura 2. Arquitectura de Sc@ut.

usuario se representa el modelo de dominio mediante una red semántica, formada por conceptos y relaciones entre conceptos, que captura su dominio de conocimiento. El diseñador puede también seleccionar una presentación concreta.

El comunicador es un sistema hipermedia sencillo que se genera traduciendo la presentación concreta teniendo en cuenta prerequisites, conceptos permanentes, tipo de asociaciones, etc. (en [4] se explica este proceso). Se implementa en

una PDA y consiste en distintos escenarios que el niño puede navegar, esto es, el niño selecciona componentes (conceptos) de cada página para expresar lo que quiere o siente. Cada niño tiene su propio comunicador basado en su perfil de usuario.

Por tanto, la construcción del comunicador supone varias fases de refinamiento y filtrado en cuyo proceso de adaptación intervienen: el perfil de usuario, el modelo de dominio, las restricciones de navegación, las características del dispositivo y el modo de interacción.

Por otro lado, el uso del comunicador puede suponer un proceso de adaptación una vez analizada e interpretada la información recogida durante el proceso de interacción del niño con el comunicador. Para ello, el comunicador recoge información sobre la navegación realizada por el niño en ficheros log (qué escenarios, componentes o enlaces entre escenarios se han visitado y cuándo, cuántas veces, el orden de navegación, etc.).

El análisis de la información recogida en el proceso de navegación les permitirá a los educadores hacer evolucionar/adaptar el dominio conceptual representado por la red semántica. Estos cambios deberán cumplir ciertas restricciones de integridad y consistencia.

Por tanto, la plataforma Sc@ut requiere adaptación/evolución tanto en el proceso de creación (en el nivel meta) como en el proceso de interacción o navegación (en el nivel comunicador).

3.2. Adaptación del contenido en Sc@ut

De acuerdo al modelo SEM-HP, en la figura 3 podemos ver un ejemplo sencillo de una red hipermedia, en la que se reflejan tres escenarios: Colegio, Desayuno y Comida. Dentro de estos escenarios encontramos diversos conceptos asociados a cada uno de ellos: Ordenador, Patatas fritas y Leche.

Cada concepto tiene asociada una imagen o un texto y un sonido que pueden ser adaptables al usuario que utiliza el comunicador. Así, sin alterar los conceptos podemos representarlos por una fotografía, un símbolo o su nombre escrito. Dependerá de las capacidades del usuario (cognitivas y sensoriales) la elección de uno u

otro. También existe un apartado subjetivo en cada sujeto, pues pueden tener predilección por una voz (su madre/padre, voz sintética) o una foto o dibujo para representar a un concepto.

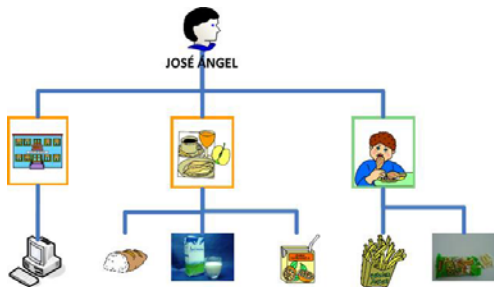


Figura 3. Ejemplo red hipermedia

En conclusión, la definición de la red hipermedia nos permite la adaptación o personalización del comunicador para cada usuario sin necesidad de alterar los conceptos a utilizar o entrenar con el mismo.

Además, con la red definimos las precondiciones y postcondiciones que deben cumplir cada escenario y concepto, a través de las relaciones que se definen entre los conceptos representados. Por ejemplo, antes de comer deberíamos lavarnos las manos.



Figura 4. Interfaz de Sc@ut.

Siguiendo la arquitectura propuesta, en nuestro modelo de dominio definimos todos los conceptos y escenarios, y en la capa de presentación abstracta indicamos qué elementos queremos utilizar para un ambiente concreto sin

fijar la representación que van a adoptar en cada comunicador.

En una tercera fase se define, entre otros aspectos, la representación concreta de cada elemento; por lo que hemos de fijar qué imagen, sonido o texto concreto representa a cada concepto en un comunicador concreto. En la figura 4 podemos ver la apariencia del comunicador Sc@ut para un niño llamado José Ángel cuya red hipermedia se muestra en la figura 3.

3.3. Adaptación de la interfaz en Sc@ut

Para la adaptación de la interfaz de Sc@ut el sistema hace uso del modelo de dominio construido y dinámicamente personalizado usando SEM-HP. Para ello será necesario relacionar el modelo de dominio dinámico con el modelo de tareas usado como base para la generación de la interfaz en AB-UIDE. Puesto que SEM-HP es una red semántica, se ha definido una interfaz entre la red semántica y un modelo de dominio “estándar” basado en orientación a objetos como el uso en AB-UIDE.

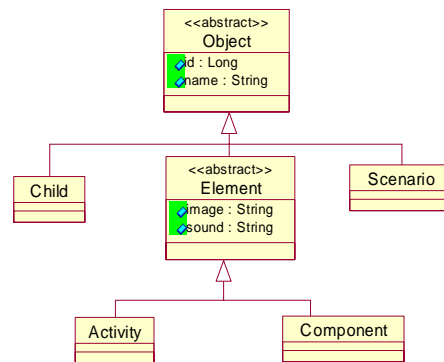


Figura 5. Interfaz entre el modelo de dominio de AB-UIDE y SEM-HP.

En la figura 5 se muestra dicha interfaz. En ella se observa como en Sc@ut de cara a la generación de la interfaz existen cuatro conceptos que son manejados. Los niños que usan la aplicación (por ejemplo, *José Ángel*), los escenarios donde dichos niños desean realizar alguna actividad (por ejemplo, el *colegio*), las

propias actividades que los niños desean realizar (por ejemplo, *comer*), y finalmente los componentes que permiten refinar las actividades anteriormente descritas (por ejemplo, *patatas*).

El proceso de adaptación de la interfaz en AB-UIDE parte de la relación entre el modelo de tareas creado y el modelo de dominio. El modelo de tareas es expresando la notación de ConcurTaskTrees. En la figura 6 se muestra el modelo de tareas que corresponde a los requisitos de Sc@ut.

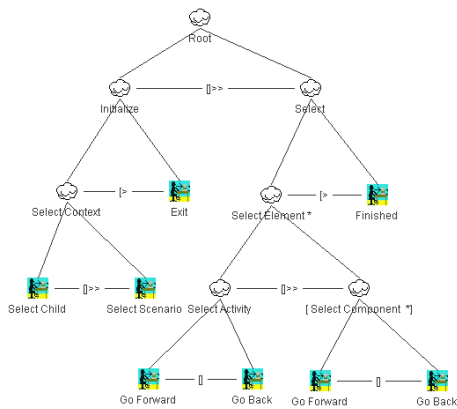


Figura 6. Modelo de tareas para Sc@ut.

La relación entre ambos modelos determina qué objetos de interacción serán seleccionados. La relación entre ambos modelos se describe en la figura 7. En dicha figura se observa como, por ejemplo, la tarea *seleccionar niño* (*select child*) se relaciona con uno o muchos niños en la interfaz del modelo de dominio, por lo que al ser una tarea de entrada, el usuario de la aplicación (el niño autista), tendrá que realizar una operación de selección de entre el conjunto de niños disponibles en la red semántica de SEM-HP. De forma análoga cuando un niño desee seleccionar una actividad lo hará de entre aquellas dinámicamente generadas adaptadas al niño actualmente seleccionado y al escenario en el que el niño se está desenvolviendo.

A partir de este momento se generará una interfaz de usuario abstracta basándose en las relaciones establecidas entre el modelo de tareas y el de dominio, el tipo de tarea (entrada, salida, control) y el tipo de datos de los atributos de las clases de la interfaz del modelo de dominio.

Dicha interfaz de usuario abstracta será transformada en una interfaz de usuario concreta para una modalidad determinada, que finalmente será transformada en la interfaz de usuario final con la que el niño interactuará.

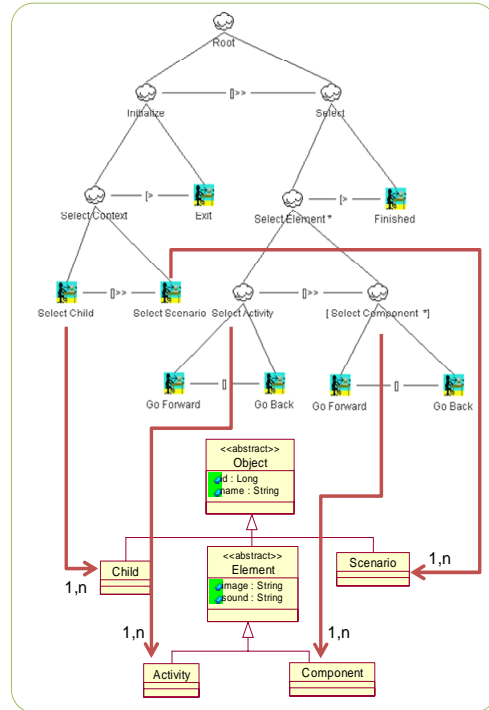


Figura 7. Relación entre el modelo de tareas y la interfaz con el modelo de dominio en Sc@ut.

4. Conclusión

El trabajo realizado supone la integración de dos trabajos previos, el modelo AB-UIDE y el modelo SEM-HP, con un objetivo muy concreto: proporcionar una adaptación integral que satisfaga la diversidad de los usuarios en sus necesidades especiales de interfaz y de contenido. Concretamente AB-UIDE se encarga de la adaptación de la interfaz y SEM-HP de la adaptación del contenido.

Dicha integración es posible gracias a que ambos modelos siguen una estructura paralela en capas que separa fundamentalmente tres niveles

de abstracción: modelado, presentación abstracta y presentación concreta.

En la capa de modelado, AB-UIDE obtiene una representación de las tareas usando diagramas de estados, ConcurTaskTrees y herramientas abstractas de interacción. Por su parte, SEM-HP modela el dominio conceptual y de información de la aplicación usando redes semánticas.

Ambos modelos, el modelo de tareas y el modelo de dominio son enlazados convenientemente, lo cuál constituye el punto de partida para la capa de presentación abstracta. En dicha capa, tanto a nivel de interfaz como de contenido, se realiza una primera adaptación independiente de aspectos concretos, tales como: plataforma, modalidad, dispositivo, usuario, etc. Esos aspectos son tenidos en cuenta en la última capa para generar las presentaciones concretas que finalmente serán utilizadas por los usuarios.

Por último, se presenta un ejemplo concreto de aplicación de la propuesta, donde se describe cómo se utilizaría la arquitectura planteada para adaptar los sistemas de comunicación aumentativa generados con la plataforma Sc@ut, como un caso particular de sistema hipermedia y una primera aproximación al uso de esta arquitectura en aplicaciones Web.

Referencias

- [1] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. *Journal of interacting with computers*. Elsevier Science B.V., Vol. 15 (3), pp. 289-308, (2003)
- [2] Constantine, L.: Canonical Abstract Prototypes for Abstract Visual and Interaction Design. *Proc. of DSV-IS* (2003)
- [3] García, L.; Rodríguez, M^aJ.; Parets, J. *Evolving Hypermedia Systems: a Layered Software Architecture*. *Journal of Software Maintenance and Evolution: Research and Practice*. John Wiley & Sons, Ltd. Vol.: 14(5), pp: 389-405. (2002)
- [4] Gea-Mejías, M., Rodríguez Almendros, M.L., Medina, N., López-Cózar Delgado, R., Paderewski Rodríguez, P., Gutiérrez Vela, F.L. *Adaptive And Context-Aware Hypermedia Model For Users With Communication Disabilities*. Interacción. 2004.
- [5] Harel, D.: Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8, pp 231-274. (1987)
- [6] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López Jaquero, V.: UsiXML: a Language Supporting Multi-Path Development of User Interfaces. *Proc. of EHCI-DSVIS'2004*. LNCS, Vol. 3425. Springer-Verlag (2005)
- [7] López-Jaquero, V.: *Interfaces de Usuario Adaptativas Basadas en Modelos y Agentes de Software*. Tesis doctoral. Universidad de Castilla-La Mancha (2005)
- [8] López Jaquero, V., Montero, F., Molina, J.P., González, P. Fernández-Caballero, A. *A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties*. *Proc. of EHCI-DSVIS'2004*. LNCS, Vol. 3425. Springer-Verlag (2005)
- [9] Medina, N.; Molina, F.; García, L. *Diversity of Structures and Adaptive Methods on an Evolutionary Hypermedia System*. *Journal IEE Proc.-Software*. Vol.: 152 (3). Pp: 119-126. (2005)
- [10] Paredes-Garrido, M.D., Pino-Morillas, O., Rodríguez-Fórtiz, M.J. y otros. *A platform for creating adaptive communicators*. *Lecture Notes in Computer Science*, 4061: 847-854, July, (2006)
- [11] Paternò, F., Mancini and Meniconi, S. "ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models". *Interact' 97*, Chapman & Hall, pp.362-369, July, (1997)
- [12] Paternò, F.: *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag, ISBN 1-85233-155 (1999)
- [13] Pinheiro da Silva, P.: *Object Modelling of Interactive Systems: The UMLi Approach*. Tesis doctoral. Universidad de Manchester, N.W. Paton. Reino Unido (2002)