

INTERACCIÓN 2.000



Actas de las Jornadas

Facultad de Psicología

Universidad de Granada

Granada, 19 y 20 de Junio del 2000



Interacción'2000

I Jornadas de Interacción Persona-Ordenador

Comité Organizador

José J. Cañas.
Dpt. Psicología Experimental.
Facultad de Psicología. Universidad de Granada

Miguel Gea.
Dpt. Lenguajes y Sistemas Informáticos.
ETSI Informática. Universidad de Granada

Comité de Programa

José J. Cañas (Universidad Granada)
Pablo Castells (Universidad Autónoma Madrid)
Miguel Gea (Universidad Granada)
Jesús Lorés (Universidad de Lleida)

Entidades Colaboradoras

Facultad de Psicología
Asesoría Antonio Gijón



Granada, 19-20 de Junio del 2000

Presentación

Interacción'2000 son las primeras Jornadas de ámbito nacional dedicadas al estudio de la interacción entre la persona y el ordenador analizando todos los factores que intervienen y afectan a este proceso. En este ámbito se van a abordar aspectos esenciales en la comunicación con los ordenadores tales como accesibilidad a discapacitados, factores humanos, estrategias de diseño, nuevas técnicas de interacción (realidad virtual, voz, gestos), mejora de la usabilidad en las aplicaciones, sistemas de ayuda, aprendizaje sobre ordenadores, etc.

Para cubrir todos estos aspectos relacionados con los sistemas interactivos, se van a realizar diferentes contribuciones desde áreas de investigación tales como la informática, psicología, Bellas Artes, empresas, etc. Estas aportaciones interdisciplinares permiten afrontar los aspectos que son necesarios para realizar un desarrollo tecnológico adecuado a las demandas de la sociedad: puestos de trabajo informatizados inteligentes, fáciles de usar y adaptados a las necesidades del usuario.

Estas Jornadas se organizan en el seno de la Universidad de Granada como resultado de una colaboración entre investigadores del Dept. de Lenguajes y Sistemas Informáticos (ETSI Informática), Dept. de Psicología Experimental (Facultad de Psicología) y la colaboración de AIPO (Asociación de Interacción Persona-Ordenador).

En este volumen recoge se las ponencias presentadas a las I Jornadas de Interacción Persona-Ordenador celebradas en Granada entre el 19 y 20 de Junio del 2000, y son el exponente de la calidad e importancia de las investigaciones que se llevan a cabo en España dentro de esta área de conocimiento. Queremos expresar nuestro agradecimiento a los autores, al comité de programa a los integrantes de la asociación AIPO, a las entidades colaboradoras y a la Facultad de Psicología, los cuales han hecho posible la celebración de estas Jornadas.

José J. Cañas

Miguel Gea

Indice

I. Análisis de los Factores Humanos en el Diseño

| | |
|--|----|
| 1. Evolución de los sentidos y el ordenador como una nueva prolongación del hombre I. López-Aparicio | 9 |
| 2. Sistemas de Interacción Persona-Computador para usuarios con discapacidad J. Abascal, N. Garay, L. Gardezabal | 15 |
| 3. Metodología y herramientas para una evaluación automática de la usabilidad J. Falgueras, A. Guevara | 21 |
| 4. Evaluación de un simulador para la estiba y desestiba portuaria A. Lucas, P.M. Valero, R. García-Ros, M. Pérez | 27 |
| 5. Evaluación del conocimiento adquirido durante la interacción J. J. Cañas, A. Antolí, J. F. Quesada, I. Fajardo | 34 |
| 6. Estudio Formal de Factores Humanos en el diseño de Sistemas Cooperativos N. Padilla, M. Gea | 41 |

II. Diseño de Sistemas Interactivos

| | |
|---|----|
| 7. Modelización y diseño interactivo de interfaces con estructura dinámica P. Castells, F. Saiz, R. Moriyón, F. García | 49 |
| 8. Aproximación Basada en Modelos en la especificación de Interfaces de Usuario M. D. Lozano, P. González, I. Ramos | 55 |
| 9. Modelo de Construcción de Sistemas Interactivos basado en técnicas Formales F.L. Gutiérrez, M. Cabrera, J.C. Torres, M. Gea | 61 |
| 10. Arquitectura para Agentes de Interfaz Inteligentes: el ordenador <i>sugerente</i> J.R. Balsas, M.C. Díaz, A. Montejo, F. Martínez, M. García, L.A. Ureña | 68 |
| 11. Descripción de la expresividad de agentes inteligentes mediante Alhambra D. Martín, M. Gea | 74 |

III. Docencia en Interacción Persona-Ordenador

| | |
|--|-----|
| 12. Interacción Hombre - Máquina en el C.P.S. de la Universidad de Zaragoza S. Baldassarri, F. Serón, P. Latorre, J. A. Magallón | 83 |
| 13. Interacción Persona Ordenador en la EUITIO A. B. Martínez, B. López, J. M. Cueva, I. Fernández | 90 |
| 14. El Perfil de Tecnologías Interactivas en la Ingeniería Informática de la UCLM Manuel Prieto y Francisco Ruiz | 95 |
| 15. Curso de Diseño y Desarrollo de Sistemas Interactivos Multimedia M. González, J. M. Cordero, M. Toro. | 99 |
| 16. Hacia un corpus docente virtual común en IPO. Experiencia del Curso Virtual de Introducción a la Interacción Persona-Ordenador J. Abascal, J. J. Cañas, M. Gea, J. Lorés, M. Ortega, L. A. Ureña, M. Velez | 105 |

IV. Nuevas Tecnologías aplicadas a la Educación

| | |
|---|-----|
| 17. Diseño de libros electrónicos educativos I. Aedo, P. Díaz | 113 |
| 18. Organización de objetos del sistema TANGOW: creación y seguimiento de cursos adaptativos a través de Internet R. Carro, R. Moriyón, E. Pulido, P. Rodríguez | 121 |
| 19. Nuevas tecnologías aplicadas a la educación mediante la interacción persona- ordenador: Proyecto experimental CIEV J. García Galera, F. L. Cintrano, J. D. Caravaca | 127 |
| 20. Descripción de una experiencia de evaluación de los Sistemas de Navegación en teleformación C. Gayà, I. Dolz, F. Alcantud | 135 |
| 21. Colaboración en entornos de aprendizaje basados en casos reales. Aplicación en ambientes de diseño y simulación. M.A. Redondo, C. Bravo, J. Bravo, M. Ortega | 143 |

V. Paradigmas de Interacción y Comunicación

| | |
|--|-----|
| 22. Interacción Hombre-Máquina mediante Sistemas Automáticos de Diálogo R. López-Cózar, A. J. Rubio, P. García, J. E. Díaz-Verdejo | 157 |
| 23. Diseño de sistemas basados en voz para la interacción hombre-máquina P.García, J. Díaz-Verdejo, J.C. Segura, R. López-Cózar, A. Rubio | 164 |
| 24. Interacción a través del Lenguaje Natural L. A. Ureña, M. García, J.R. Balsas, M.C. Díaz, A. Montejo, F. Martínez | 170 |
| 25. Una Técnica de Especificación Espacial para Diseño de Sistemas Interactivos M.L. Rodríguez, M. Gea | 176 |
| 26. Mas allá de la manipulación directa. Hacia nuevos paradigmas de interacción J. Lorés, M. Sendín, C. Aguiló | 182 |

VI. Aplicaciones y desarrollo

| | |
|--|-----|
| 27. Diseño Interactivo de Programas de Control Industrial Basado en Técnicas de Inteligencia Artificial L. Castillo, J. Fernández Olivares, A. González. | 193 |
| 28. Requisitos para una interfaz visual de un SGBDOO construido sobre un sistema integral OO A. Hernández , A. B. Martínez, J. M. Cueva, D. Álvarez | 205 |
| 29. Interfaz de Usuario en el Desarrollo de un Simulador de Conducción M. Sánchez, P. Valero, I. Pareja | 211 |
| 30. Un sistema de etiquetado para el Resaltado de Entornos Web M. Villarroel, P. De la Fuente | 217 |
| 31. Tirsus IV, Navegación Multidimensional en Aplicaciones Hipermedia sobre Acontecimientos Históricos A. M. García, B. Martín, J.M. Redondo, A. J. Sánchez | 225 |

Póster

| | |
|---|-----|
| 32. “Information Walkthrough”. Visualización 3D de Sistemas de Información Iván Fernández Lobo | 233 |
| 33. Cinemedia Astur: Herramienta Generadora de Títulos Hipermedia de Interfaz Flexible J. M. Redondo, A. J. Sánchez, A. M. García, B. Martín | 240 |
| 34. Tirsus II, Aplicación de Hipermedia para la Enseñanza de la Historia A. J. Sánchez, A. M. García, B. Martín, J. M. Redondo | 245 |
| 35. Creación de una herramienta multimedia para la distribución de actas en un congreso. Adaptación de las técnicas de comunicación a los soportes multimedia A.Gil, J. Muñoz. | 251 |
| 36. Interacción con mundos VRML utilizando Java C. Romero, S. Ventura, C. Castro | 260 |
| 37. Virtual-PRISMAKER: Juegos de ordenador, educación e interfaces. F. Montero, V. López, M. Lozano, A. Fernández, P. González | 266 |
| 38. Diseño y desarrollo de una aplicación informática para la gestión de laboratorios M. Francisco, P. Vega, F. J. Blanco | 272 |

Sesión I

Análisis de los Factores Humanos en el Diseño

Evolución de los sentidos y el ordenador como una nueva prolongación del hombre

Isidro López-Aparicio Pérez.

Departamento de Dibujo
Universidad de Granada, Av. Andalucía 71, Granada
e-mail isidro@goliat.ugr.es

Resumen

En esta ponencia pretendo realizar unas reflexiones sobre cómo el ordenador ha venido a alterar la forma de relación del hombre con el mundo.

En la sociedad actual se destaca el conocimiento de la realidad a partir de lo visual. Si cogemos un texto del XVIII en el que se describa una mujer "ideal" veremos como se destacan, en gran medida sobre otros factores, elementos asociados al olor y a lo táctil. En el conocimiento de las personas primaba la presencia física, un conocimiento real; en la actualidad, aparte de éste nosotros conocemos a un sin fin de otras personas a través de los medios de comunicación por lo que tendemos a percibir a una mujer "ideal" a partir de imágenes y tendemos a valorar principalmente la forma, obviando otros factores de indiscutible interés. En este sentido hay que destacar que los avances tecnológicos repercuten directamente en la manera de relacionarnos y percibir la realidad.

Presentación

En esta ponencia pretendo realizar unas reflexiones sobre cómo el ordenador ha venido a alterar la forma de relación del hombre con el mundo.

Es habitual que europeos después de un viaje a algún país de África destaquen a su vuelta las nuevas sensaciones que han tenido entre las que siempre destacan los olores: las

especies, los alimentos, las pieles... parece como si se despertara un sentido que tuviéramos dormido.

En la sociedad actual se destaca el conocimiento de la realidad a partir de lo visual. Si cogemos un texto del XVIII en el que se describa una mujer "ideal" veremos como se destacan, en gran medida sobre otros factores, elementos asociados al olor y a lo táctil. En el conocimiento de las personas primaba la presencia física, un conocimiento real; en la actualidad, aparte de éste nosotros conocemos a un sin fin de otras personas a través de los medios de comunicación por lo que tendemos a percibir a una mujer "ideal" a partir de imágenes y tendemos a valorar principalmente la forma, obviando otros factores de indiscutible interés. En este sentido hay que destacar que los avances tecnológicos repercuten directamente en la manera de relacionarnos y percibir la realidad.

El hombre a lo largo de la historia ha ido inventando artilugios que se han convertido en prolongaciones de sí mismo, de sus funciones, de sus capacidades. Por ejemplo, las lentes en el fondo vinieron a ser una prótesis que ayuda a corregir la visión del hombre, los prismáticos permiten agudizar su capacidad visual hasta extremos que su naturaleza no le permite, lo que le aporta la posibilidad de obtener nuevas sensaciones en su entorno y altera la forma de coexistir con éste. La utilización de telescopios le permite el descubrimiento de un mundo que le era inaccesible en sus limitaciones físicas, a la vez que activa su capacidad de deducción y razonamiento.

El hombre, en su búsqueda de elementos que faciliten sus actividades y respondan a las nuevas necesidades, encuentra en las computadoras, en un primer momento la posibilidad de memorizar gran cantidad de información de manera ordenada al igual que la realización de cálculos cada vez más complejos. Estas funciones vienen a simplificar ciertas tareas, pero conforme se introducen en la realidad social permiten la cesión de algunas operaciones. En este sentido los ejercicios de memorización se relajan, el hombre cede al ordenador una de sus capacidades, sin que éste tenga que implicar ningún detrimento de las capacidades de la persona, aunque sí una alteración de sus máximas.

En este sentido podemos hablar del paso de la cultura hablada a la escrita: Aún en la actualidad, existen pueblos que transmiten su herencia histórica a partir de narraciones nunca escritas, esta forma de transmisión no sólo desarrolla la capacidad de memorización del individuo sino que imprime un carácter particular del conocimiento, tanto de la forma de aprendizaje como de relación social. La transmisión escrita permite una forma distinta de relación y un mayor volumen de transmisión del contenido sin que implique la presencia física del comunicador. El volumen de contenidos que en la actualidad se pueden almacenar en información electrónica en espacios irrisorios y de enorme accesibilidad permite una nueva forma de entender la transmisión y almacenamiento de conocimientos. A su vez el desarrollo acaecido en la tecnología nos permite nuevas formas de narración que aparte de las posibilidades de incorporar texto, imagen, movimiento, voz... ofrece la posibilidad de que el receptor interactúe con el nuevo continente de conocimiento. Hemos visto como desde

las computadoras hemos ido progresando hasta un momento esencial que fue la incorporación de la imagen a la pantalla, lo cual junto a otros factores ha ayudado a popularizar el ordenador personal y dio paso un nuevo mundo visual.

La cantidad de información que una persona en la actualidad, inmersa en una sociedad avanzada maneja, no es comparable con la que tiene una persona de una sociedad primitiva o puede necesitar. Y no hablo tanto en cantidad, que indiscutiblemente existe, sino en los medios o sentidos que utiliza para adquirirla.

Un miembro de un pueblo indígena maneja una información considerable pero adquirida principalmente a partir de medios directos, comunicación hablada y utilización de todos los sentidos, en muchos casos demostrando una gran agudeza visual, olfativa, auditiva y táctil cercana en momentos al terreno de lo intuitivo. En comparación, un niño inmerso en una sociedad avanzada como la europea adquiere de manera indirecta una gran cantidad de información a partir de los medios de comunicación, ya sea la lectura o los medios audiovisuales. A su vez, maneja desde muy joven objetos creados por el hombre que necesitan del conocimiento de su manipulación y que, en muchas ocasiones suele ser altamente compleja. De esta forma, se desarrollan igualmente todos los sentidos pero supeditados no sólo al conocimiento de la realidad de forma directa, sino a través de medios que potencian la utilización de dos de ellos, principalmente los visuales y auditivos.

El ordenador viene a ser un nuevo estadio en el desarrollo y conocimiento del ser humano. Como hemos visto permite la relajación y la simplificación de tareas de almacenaje y de cálculo. En este sentido es una herramienta que sirve de ayuda al hombre en tareas quizás más sistemáticas que las suyas propias, entendido éste como animal que se diferencia de cualquier otro en la creatividad y en la capacidad de hacer bien o mal.

El ordenador ha venido a convertirse en una herramienta indispensable y habitual para la realización de un sin fin de tareas: procesado, contención, manipulación, ejecución... de cálculos, imágenes, sonidos, textos, datos... lo cual permite a cada usuario optimizar sus quehaceres. Se convierte en una prolongación del propio hombre que permite realizar funciones que serían harto complejas por otros medios y en muchos casos irrealizables. No creo que se deba mirar como una dejación de funciones habituales del ser humano que vaya en detrimento de sus cualidades sino, al contrario, la liberación de ciertas limitaciones, la eliminación de pérdidas de tiempo en trabajos sistemáticos para poder incorporar factores más humanos y la aportación de nuevas posibilidades imposibles hasta el momento. En este sentido, el desarrollo de los programas cada vez más accesibles permiten que las personas aborden y se desarrollen dentro de sus posibilidades de forma más intuitiva y empaticen con este nuevo entorno acumulativo, operativo y creativo.

No se abordará aquí la importancia del ordenador entendido como herramienta individual de trabajo, pero si algo es indiscutible es que la gran revolución del mundo de los ordenadores ha sido sacarlos de su aislamiento para comunicarlos entre sí y crear una trama de relaciones

e interacciones que desborda el mundo de la comunicación. Todo el potencial que permite un ordenador se pone al servicio de las redes electrónicas para transmitir contenidos, imágenes, movimiento, colores, textos, voz... y todo en tiempo real y con la posibilidad de interacción.

En este sentido, y salvando las distancias, se puede analizar el gran paralelismo existente entre la aparición de la imagen impresa y la divulgación del conocimientos a través de las redes:

Es fundamental la importancia e influencia que tuvo la acumulación y divulgación del conocimiento gracias a la aparición de los tipos móviles metálicos en el siglo XV para dar paso al avance del humanismo renacentista, de igual modo que la asociación de la imagen impresa al texto completó la información dando paso al mundo ilustrado, a la cultura enciclopédica.

Esta multiplicación permitió que existiera una difusión visual más allá de fronteras locales, lo cual divulgó una manera de entender e interpretar la realidad, un concepto de representación condicionado por el medio, se creó un cierto lenguaje común que hizo que algunas formas y figuras se estandarizaran. En este sentido la globalización del mundo de la red viene a desbordar los límites de la comunicación existente hasta el momento, que tuvo ya su primera gran transgresión en el Renacimiento. Este hecho ayuda a la construcción de un nuevo mundo, a un conocimiento global, aunque nunca sustituible por el conocimiento directo.

Es habitual ver como en toda Europa existen imágenes que se “asimilan, repiten, copian en incluso se alteran y plagian que representan un determinado modo de visualizar la realidad”. Esta práctica es habitual en la red, copiando, alterando, transmitiendo de manera totalmente arbitraria o voluntaria todo aquello encontrado y usado según nuestro deseo.

En el Renacimiento, cuando la ciencia alcanza un empuje nunca hasta el momento conocido al disponer de un medio de difusión que permite, por un lado, la transmisión de conocimiento ya existente (cuando su impacto había sido mínimo al ser mínima su disponibilidad) y por otro la realización de obra específica como reflejo de conocimiento contemporáneo, de la misma forma la red se ha hecho eco tanto de la divulgación de conocimientos elaborados previamente o específicos contruidos por y para el medio. Las imágenes xilográficas de épocas pasadas tienden a describir el silueteado o contorno de las formas limitándose a transmitir la esencia formal; poco a poco se van incorporando sombras, valoraciones tonales... todo a partir de una mayor cantidad de información, tramados sutiles, concentración de líneas... lo cual implicará problemas técnicos de impresión. La técnica encuentra sus límites y para obtener otras calidades, y responder a nuevas necesidades plásticas se debe pasar a procesos calcográficos que permiten un mayor virtuosismo plástico. La utilización de estos procesos no es algo vanal, sino que tiene una repercusión inmediata en el modo de transmitir la realidad.

Igual está sucediendo constantemente en la transmisión de información vía electrónica, ya que los condicionantes entre el hardware y el software son indiscutibles y uno condiciona al otro y así progresivamente al mismo tiempo que condicionan las apariencias resultantes.

Los procesos que en un primer momento se usan (xilografía principalmente) para reproducir de forma múltiple imágenes que ilustren el contenido vienen determinadas por los condicionantes que impone el sistema de reproducción lo cual conlleva un nuevo ordenamiento de lo visible. De igual modo las imágenes que se transmiten vía electrónica vienen condicionadas por el medio; su calidad visual y definición terminan construyendo una estética particular que la identifica.

La aparición de la imagen múltiple también vino a alterar el concepto de imagen original, ya que un original podía ser multiplicado. De esta forma es habitual encontrar imágenes de las que es difícil encontrar su origen e incluso no se pretendió una autoría cuando se generaron. Pero si el mundo de la imagen impresa es el que crea los primeros antecedentes sobre el cuestionamiento de la propiedad intelectual, en la actualidad es la red la que vuelve a plantear nuevas dudas.

Durante mucho tiempo se ha pensado que los ordenadores podrían llegar a hacerlo todo y en la actualidad es cada vez más habitual recurrir a equipos interdisciplinarios que permitan el completar informaciones y valorar tantos cuantos aspectos propiamente humanos puedan afectar ante cualquier decisión, aportando de esta forma el hombre aquellos aspectos propios e intransferibles de su naturaleza.

En este sentido en los comienzos de la imprenta aparecieron nuevos vínculos sociales que hicieron que se acercaran entre sí los artífices de oficios distintos y que favoreció nuevas formas de intercambio cultural. El mundo de la informática necesita en todo momento de la interacción con nuevos colectivos, del mismo modo que es incalculable el sin fin de perfiles que se acercan a las aplicaciones software para personalizar sus necesidades.

No podemos dejar de referirnos a los periféricos que acompañan y completan en todo momento al ordenador, verdaderos "gatchets" que multiplican las posibilidades tanto de recepción de la realidad (llegando a la utilización de escáner de olores o como de la reproducción de objetos volumétricos). El ordenador, de esta manera, se convierte en una prolongación más del ser humano como fueron los prismáticos y debemos de entender éstos no como algo que desvirtúa nuestra realidad sino que posibilita nuevas formas de desarrollo, un mayor entendimiento de lo que acontece a nuestro alrededor y una nueva forma de relacionarnos con éste.

Referencias

- [1] A.A.V.V., "Media Culture", ed. Cludia Giannetti, Barcelona, 1995
- [2] DE PEDRO, Antonio E. "El diseño Científico, siglo XV-XIX" en Historia de las Ciencia y de la Técnica n°37, ed Akal, Madrid, 1999.
- [3] EISENSTEIN, Elizabeth. "La revolución de la Imprenta en la Edad Moderna", ed. Akal, Madrid, 1994.
- [4] HANHARDT, G. Jhon,"Modelos de interacción: cine y vídeo en una nueva era de los medios de comunicación" en A.A.V.V., "El nuevo espectador" colección Debates sobre Arte. Volúmen V, Fundación Argenteria, Visor Dis., Madrid, 1988.
- [5] IVINS, William, "Prints and Visual Communication", en Da Capo Press Series in Graphic Art, A. Hyatt Mayor ed., The Mit Press, Cambridge, 1992. 1ª edición, Harvard University Press, 1953.
- [6] NOBLE, F. David. "La religión de la tecnología. La divinidad del hombre y el espíritu de invención", ed. Paidós, Barcelona, 1999. 1ª Edición: "The Religion of Technology". Alfred A. Knopf, Inc., Nueva York 1997.
- [7] RECK MIRANDA, Eduardo, "Música y nuevas tecnologías, perspectivas para el siglo XXI", L'ANGELOT, Barcelona, 1999.

Sistemas de Interacción Persona-Computador para usuarios con discapacidad

Julio Abascal, Nestor Garay y Luis Gardezabal

Laboratorio de Interacción Persona-Computador para Necesidades Especiales
Departamento de Arquitectura y Tecnología de Computadores
Universidad del País Vasco-Euskal Herriko Unibertsitatea
M. Lardizabal 1, 20009 Donostia-San Sebastián
Tlf: 94301806, Fax: 943219306, e-mail: [julio,nestor,luisg]@si.ehu.es

Resumen

En esta ponencia se exponen las ventajas de la aplicación de las técnicas de interacción persona-computador en el diseño de interfaces para usuarios con discapacidad y las aportaciones de esta actividad al desarrollo de nuevas técnicas de interacción; lo cual se ilustra con ejemplos de la experiencia obtenida por el Laboratorio de Interacción Persona-Computador para Necesidades Especiales (LIPCNE) en el desarrollo de sistemas de interacción para personas con graves discapacidades motoras y orales que les impiden la comunicación hablada.

Palabras clave: acceso universal, discapacidades, adaptabilidad

1 IPC y usuarios con discapacidad

Durante mucho tiempo conseguir que los ordenadores fueran accesibles a las personas con determinadas discapacidades fue una tarea artesanal que producía soluciones dependientes del dispositivo y del usuario concreto y, por tanto, de difícil generalización. La adopción de las técnicas de interacción persona-computador (IPC) en este área ha permitido formalizar la metodología y producir interfaces más accesibles y fiables y para un mayor número de usuarios. Estos avances han conducido a un nuevo enfoque, el *diseño para todos*, que más que a dispositivos específicos para las personas con discapacidad, se orienta a diseñar sistemas de uso general accesibles a todos los usuarios. Así, muchos autores prefieren hablar de *accesibilidad universal*, en vez de diseño para personas con discapacidad [1]. El éxito de este enfoque se debe a la constatación generalizada de que los sistemas diseñados

para ser *accesibles para todos* son más fáciles de usar también para las personas sin discapacidad.

A pesar de que la razón fundamental para aplicar las técnicas de IPC para satisfacer las necesidades de este colectivo de usuarios procede fundamentalmente de sus demandas de acceso a los computadores, la experiencia demuestra que esta actividad ha producido una serie de beneficios añadidos [2], inicialmente no previstos, que podríamos resumir en tres puntos:

- Obliga al diseñador a tener en cuenta aspectos que normalmente pasan desapercibidos, tales como las diferencias individuales, los gustos personales, las capacidades físicas y cognitivas requeridas por la interfaz
- Como consecuencia, permite el desarrollo de soluciones más imaginativas e innovadoras
- Además, permite ampliar el mercado de los productos de uso general, al hacerlos accesibles a una población más numerosa

De esta forma, muchos de los avances teórico-prácticos producidos en este campo han sido frecuentemente integrados en los métodos de diseño de las interfaces de propósito general.

En los siguientes apartados vamos a ver algunos de los desarrollos del LIPCNE en el área de la IPC, que se han derivado de la experiencia del diseño de interfaces para personas con discapacidad.

2 Sistemas de comunicación

La demanda primaria de las personas con graves discapacidades motoras y orales es la comunicación. Para satisfacerla se han desarrollado una serie de métodos y técnicas, que se agrupan en la denominada *Comunicación Aumentativa y Alternativa* (CAA), inicialmente basados en signos corporales y tableros con signos o pictogramas. Sin embargo, la difusión de los microprocesadores y de los ordenadores personales ha permitido diseñar dispositivos informáticos que suponen un gran avance en la ayuda a la comunicación [3].

En los diez últimos años, el LIPCNE ha venido desarrollando múltiples sistemas de CAA, portátiles y fijos, con entrada por barrido controlado mediante una o varias teclas, o

Los *sistemas de barrido* ofrecen al usuario una matriz con las opciones seleccionables que son barridas secuencialmente (por filas y columnas). Cuando un determinado ítem es resaltado, el usuario puede seleccionarlo mediante la pulsación de una tecla.

mediante teclados reducidos, y con salida por *display*, voz sintética, impresora térmica incorporada, infrarrojos (para comunicación y control de otros dispositivos), etc. Estos sistemas permiten la comunicación personal directa o remota, el uso de PCs y *software* estándar, el acceso a redes telemáticas y el control del entorno. El diseño de la interfaz de usuario ha requerido el uso de una metodología orientada a la usabilidad, tal como *userfit*² [4] y la evaluación por parte de usuarios reales³. Vamos a describir sucintamente algunos de los aspectos más originales de estos sistemas de interacción.

3 Mejora de la velocidad de comunicación

Dado que los sistemas de comunicación alternativa resultan demasiado lentos para una conversación normal, se ha trabajado en la mejora la velocidad de comunicación utilizando diversos métodos.

3.1 Predicción de palabras

Los sistemas de predicción de palabras tratan de emular la capacidad humana de anticipar, bajo ciertas condiciones, lo que el interlocutor quiere decir, basándose en la redundancia del idioma y en el conocimiento de la estructura sintáctica y semántica de la frase. Con este propósito se han desarrollado diversos métodos de predicción que ayudan al usuario a seleccionar palabras completas en vez de tener que teclearlas letra a letra, lo que puede suponer la reducción del esfuerzo y la mejora de la velocidad de composición de mensajes. Estos sistemas utilizan la información estadística (frecuencia de uso), morfológica y sintáctica asociada a las palabras. Para su diseño se han aplicado métodos de procesamiento del lenguaje natural, tales como el análisis sintáctico mediante charts, que permiten ofrecer la palabra más frecuente entre las que posean la categoría sintáctica adecuada [5]. Dado que estos métodos son fuertemente dependientes del idioma, ha sido necesario desarrollar diferentes métodos para lenguas con bajo nivel de flexión (como el castellano, inglés, etc.) y con alto grado de flexión (como el euskara, finlandés, etc.).

3.2 Uso de teclados reducidos ambiguos

² *Userfit* es una metodología desarrollada por el *HUSAT Research Institute (Human Sciences and Advanced Technology. Science Faculty of the Loughborough University)*.

³ En colaboración en el CEAPAT/IMSERSO.

Algunos usuarios con discapacidad motora disponen de movilidad suficiente para manejar un teclado de tamaño reducido. Los teclados reducidos son aquellos que tienen menos teclas que las letras del alfabeto (por ejemplo el teclado telefónico). Para su uso, se pueden asignar varios caracteres a la misma tecla y establecer un protocolo con varias pulsaciones por carácter, lo que resulta bastante tedioso. El ideal (por cuanto es el que menos esfuerzo y tiempo requiere) es realizar una única pulsación por tecla, lo que hace que el mensaje que se produzca pueda ser ambiguo⁴ (ya que cada tecla puede tener varios significados)[6]. Para resolverlo, el LIPNE ha desarrollado un algoritmo genético⁵ que produce diversas asignaciones de letras en teclados de 12 a 14 teclas que con un bajísimo número de colisiones entre palabras [7]. Como además de por pulsación directa, estos teclados también pueden ser usados por barrido, la existencia de varias soluciones alternativas permite adecuar la disposición del conjunto de selección a las características de cada usuario.

4 Autoadaptación

Las características físicas y cognitivas de los usuarios con discapacidad son muy diversas, de modo que no es posible diseñar una interfaz estándar válida para todos ellos. Por esta razón, las interfaces deben ser altamente configurables⁶. Sin embargo, el ajuste de los parámetros personales, tales como la velocidad de barrido, forma y disposición de la matriz que contiene los elementos seleccionables, etc., no es una tarea fácil. Además, las características personales pueden cambiar con el tiempo, incluso a corto plazo. Por ejemplo la experiencia de uso y el aprendizaje permiten aumentar la velocidad de barrido y disminuir la necesidad de ayuda *on-line*. Por el contrario, la fatiga hace disminuir la capacidad de respuesta del usuario a lo largo del día.

La solución ideal para este tipo de problemas es el diseño de sistemas autoadaptables (*adaptive systems*) que se ajustan automáticamente a las necesidades de cada usuario. Para ello es necesario identificar los parámetros observables y relevantes, modelar la interacción y configurar las características que se modifican a lo largo de la misma.

4 En nuestro caso medimos la ambigüedad como el número de palabras distintas que *colisionan*, es decir que tienen la misma codificación.

5 Ya que se puede demostrar que el algoritmo determinista obvio para el problema de buscar una distribución de caracteres entre las teclas disponibles, que produzca una ambigüedad menor que un cierto valor dado, es NP-completo.

6 Mediante la modificación manual de ciertos parámetros o el uso de ficheros de *perfil de usuario*.

4.1 Adaptación de la velocidad de barrido

Estudios realizados por el LIPNE demuestran que la capacidad de reacción del usuario en los sistemas de barrido varía considerablemente en periodos de tiempo relativamente cortos. Por ello, si la velocidad de barrido es fija, o bien se producen tiempos muertos entre dos selecciones consecutivas o bien el usuario no tiene tiempo para reaccionar. Para evitarlo, se ha diseñado un sistema que adapta esta velocidad a la capacidad de reacción del usuario, tomando como parámetro de entrada el tiempo medio de respuesta del usuario. Con esta información, y partir de un modelo de la interacción realizado mediante lógica *fuzzy*, se adapta la velocidad de barrido dinámicamente [7]. Los resultados demuestran que el ahorro de tiempo de producción de mensajes es considerable.

4.2 Adaptación de diccionarios

Los sistemas de anticipación previamente mencionados cuentan con diccionarios autoadaptables que actualizan las frecuencias de las palabras, en función de su uso, e incluyen las palabras nuevas. En los diccionarios que contienen información puramente estadística la inclusión de palabras nuevas es trivial, pero si disponen de otros tipos de información (morfológica, sintáctica o semántica), su inclusión requiere un tratamiento especial [5]. Sin embargo, el esfuerzo añadido para que los diccionarios sean autoadaptables resulta rentable ya que las evaluaciones realizadas demuestran que, según se va adecuando el diccionario al léxico del usuario, los resultados de predicción mejoran notablemente.

5 Interfaces emocionales

Además de la limitación en la velocidad de comunicación, los sistemas de CAA dificultan la transmisión de la información que normalmente se expresa de modo no oral (mediante gestos y expresiones faciales, actitudes o sonidos no articulados). Este es un aspecto que interesa cada vez más a los diseñadores de sistemas interactivos en general [8], pero que es especialmente importante para los usuarios que utilizan el ordenador como medio de comunicación, ya que el comunicador que les representa, en cierto modo, ante los demás⁷ filtra sus emociones, sentimientos, estados de ánimo, etc.

⁷ Es lo que se conoce como *Affective Mediation*.

Actualmente el LIPCNE está trabajando en una interfaz basada en el modelado de la conversación que permite modelar los estados de ánimo del usuario y del interlocutor y el tono de la conversación. A partir de estos modelos ayuda al usuario a generar frases completas en estilo y tono adecuados.

6 Conclusiones

La necesidad de enfrentarse a una gran diversidad de necesidades de comunicación y la variedad de las características físicas, cognitivas, gustos y deseos de los usuarios con discapacidad, obliga a plantearse aspectos de los sistemas de interacción de usuario que pueden quedar ocultos en las interfaces estándar. Los sistemas así diseñados tienden a ser más imaginativos y novedosos, con la ventaja añadida de que los resultados de estos trabajos suelen ser ampliamente utilizables en otros ámbitos de la interacción persona-computador [9].

Referencias

- [1] C. Stephanidis et al. "Toward an Information Society for All: An International R&D Agenda". *International Journal of Human-Computer Interaction*, Vol 10, No 2, pp. 107-134, 1998.
- [2] A. F. Newell and P. Gregor "Human Computer Interfaces for People with Disabilities", in M. Helander et al (eds.) *Handbook of Human-Computer Interaction* 2nd edition, Elsevier 1997.
- [3] J. Abascal y L. Gardeazabal. "Technology to support alternative and augmentative communication". In A. Casals (edt.) *Technological Aids for the Disabled*. Societat Catalana de Tecnologia/Institut d'Estudis Catalans, pp. 233-261, Barcelona, 1998.
- [4] D. Poulson D., M. Ashby & S. Richardson (Eds.) "*Userfit*. A practical handbook on user-centred design for Assistive Technology". TIDE EC-DG XIII, ECSC-EC-EAEC, Brussels-Luxembourg, 1996.
- [5] N. Garay and J. G. Abascal "Intelligent Word-Prediction to Enhance Text Input Rate". *Proceedings of the Intelligent User Interfaces 97 Congress*. ACM Press, New York. 241-244, 1997.
- [6] G. W. Lesh et al "Optimal Character Arrangements for Ambiguous Keyboards". *IEEE Trans on Rehabilitation Eng.*, Vol. 6. pp. 415-23, 1998.
- [7] L. Gardeazabal. "Aplicaciones de la Tecnología de Computadores a la mejora de velocidad de comunicación en Sistemas de Comunicación Aumentativa y Alternativa", Tesis Doctoral. Universidad del País Vasco-Euskal herriko Unibertsitatea, 1999.
- [8] R.W. Picard. "Affective Computing". MIT Press, 1998.

- [9] Gregg C. Vanderheiden. “Presente y futuro de la accesibilidad a internet” Novática nº 136, pags. 6-11. Noviembre–Diciembre 1998.
[<http://www.ati.es/PUBLICACIONES/novatica/1998/136/grevan.html>]

Metodología y herramientas para una evaluación automática de la usabilidad

J. Falgueras, A. Guevara

Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga, Campus de Teatinos s/n
Tlf: 95213 2792/3254. FAX: 95213 1397
e-mail: {juanfc,guevara}@lcc.uma.es

Resumen

Esta ponencia presenta una metodología de evaluación automática de la usabilidad de interfaces de usuario. En ella se estudia y se presenta una herramienta para la obtención de una métrica estática de la usabilidad durante el proceso de desarrollo. Esta medida está basada en consideraciones heurísticas, análisis GOMS y medidas de campo de la complejidad de los elementos de interfaz (*widgets*) empleados en la codificación en lenguajes de alto nivel y entornos UIMS y la aplicación de heurísticas, como la similitud e interactividad de la combinación de aquéllos elementos. Posteriormente, esta técnica se muestra útil para el seguimiento y la evaluación en tiempo real del uso del sistema, sin necesidad de la intervención de observadores sobre usuarios como en la metodología WOz. Finalmente se muestra el uso de la metodología para la adaptabilidad semiautomática de las interfaces a distintos perfiles de usuario en sistemas de trabajo cooperativos.

Palabras clave: usabilidad, complejidad algorítmica, métodos formales, inspección de la usabilidad, CSCW, desarrollo participativo.

1 Introducción

La mayoría de los métodos de usabilidad en ingeniería podrían contribuir de manera sustancial a la usabilidad con el mero hecho de llegar a ser *usados* durante el ciclo de vida del proyecto. Pero aún el mejor de los métodos es inútil si no se usa.

Los sistemas actuales de evaluación de la usabilidad de las interfaces de usuario son tan costosos y lentos que no se llevan a la práctica o si se hace, se aplican sólo de forma incompleta [1,2]. El modelo mixto presentado en [3,4] presenta un sistema de evaluación semiautomatizada capaz de recoger datos recibidos en forma multimodal. Esta evaluación automática está basada en cuatro pasos: (1) definición del modelo de tareas (particularmente

siguiendo modelos análogos a los modelos GOMS se llega a un modelo conceptual de tareas); (2) adquisición de los datos conductuales del usuario (esta recogida incluye la grabación de imágenes del usuario para la posterior interpretación de expresiones y gestos; (3) identificación de patrones de comportamiento; la repetición de acciones dentro de una tarea, debida a la forma de implementación de la interfaz. Por ejemplo, el reajuste sistemático del tamaño de las ventanas cada vez que se abren puede indicar un mal diseño de esta función de apertura, que no adecua correctamente la presentación de las ventanas a las necesidades del usuario); (4) crítica de los mismos (bajo una taxonomía debida a [5] que incluye compatibilidad, homogeneidad, concisión, pertinencia de las respuestas, control explícito, sobrecarga cognitiva y control de errores).

Aunque este modelo observa la necesidad de un acercamiento a los detalles útil en el diseño de futuros interfaces, requiere de una abundante evaluación de campo y no define una metodología realista aplicable a una Ciclo de Vida de desarrollo en espiral basado en prototipos [6].

La propuesta que presentamos surge de la necesidad de un desarrollo del sistema en el que el usuario participe desde el principio describiendo su propio dominio de tareas, datos y, finalmente, su particular modo de uso de la interfaz [7]. Para ello, se consideran dos tipos de métricas de la usabilidad (complejidades de la interfaz): la complejidad estática y la complejidad dinámica. Estos conceptos surgen directamente del seguimiento real de la construcción de una interfaz de usuario. Durante el desarrollo, el diseñador se enfrenta al diseño de pantallas, diálogos cuya complejidad denominaremos estática dado que implican una representación estática, una interrelación de opciones y un estado. Estos diálogos conllevan decisiones respecto a la complejidad de la presentación que mediremos mediante consideraciones acerca de los elementos componentes y de ciertas relaciones que los aglutinan. Por otro lado estará el sistema en funcionamiento con el usuario actuando sobre cada uno de los elementos y saltando de un diálogo-estado a otro. Este proceso dinámico requiere de una continua toma de decisiones, claramente expresada en modelos como el GOMS [8], por ejemplo. La actividad del usuario implica claramente un nuevo parámetro, el tiempo. Tenemos una complejidad, cantidad necesaria de pasos para alcanzar un objetivo, tiempo requerido y trayectoria seguida. Todos estos datos nos informarán acerca del comportamiento del usuario real y de la idoneidad de los elementos presentados.

La complejidad estática se medirá en base a un modelo basado en la semántica de la interfaz y servirá al propio diseñador-programador durante el desarrollo como una medida fundamental de la usabilidad. Esta medida, al poder basarse en el propio código que construye el aspecto y determina las posibles acciones mediante construcciones algorítmicas, podrá ser tomada directamente del propio código, analizando el código de la interfaz, medidas de la complejidad y parámetros heurísticos relativos a la similitud y categoría de cada elemento de interfaz.

La complejidad dinámica recoge un conjunto de tablas de seguimiento que trazan tanto el camino seguido por el usuario hacia cada objetivo como los tiempos. Simultáneamente, se evalúan las frecuencias de acceso a los recursos presentados en la interfaz: de estas tablas es inmediato mejorar las estrategias seguidas respecto a la importancia prevista de cada elemento de interfaz.

3 Análisis estático de la complejidad: AEC

** [9,10] arguyen que no necesariamente una interfaz visualmente más compleja tiene por qué ser menos usable, formatos de pantalla más densos y ricos en información mejoran los resultados [10]. La evaluación de la comprensión de la interfaz se hacía en la primera etapa de uso de usuarios inexpertos, cuando aún éstos no habían captado la estructura de las pantallas. ** [11] indica que es la adecuación de la complejidad de la interfaz a la capacidad del usuario lo que es realmente decisivo. Podemos, pues, concluir que no es la complejidad visual, en el sentido clásico, de una gran cantidad de elementos interactivos simultáneos, la que hace más difícil el uso de la interfaz, obviamente una innecesaria complejidad en este sentido será negativa, pero, dada una cantidad N de decisiones a tomar por parte del usuario, la presentación escalonada de los diálogos dando lugar a un total de N decisiones es de mayor complejidad que la agrupación de todos esos N elementos de decisión en un único diálogo. Esto es especialmente cierto para tareas en las que la información está muy interacoplada. Será pues la complejidad, por tanto de los diálogos, junto al tipo de tarea, la que determine la mayor o menor usabilidad de la interfaz.

El Análisis Estático de la Complejidad se realiza durante la codificación mediante sencillos *parsers* (*analizadores sintácticos*) del código fuente en un lenguaje de alto nivel, preferentemente. Un ejemplo de este tipo de lenguaje es Tcl/Tk. Del análisis sintáctico se recogen el número de elementos de interfaz, su situación y agrupación jerárquica respecto a los demás. La creación repetitiva de elementos gráficos, por ejemplo, en secuencias de botones, disminuye la complejidad del conjunto, que adquiere así la categoría de lista de elementos. Inicialmente se realiza una jerarquización en categorías de los elementos gráficos, de sus complejidades de uso y de los coeficientes de similitud. El análisis semántico revela la coordinación existente entre los grupos de elementos gráficos, su jerarquía en la presentación. Por ejemplo, en grupos de botones o listas de campos de entrada de datos. La jerarquización es fácil de detectar en lenguajes orientados a objetos y en especial en el lenguaje Tk en el que los elementos gráficos han de ir obligatoriamente contenidos formando una jerarquía arbórea.

Uno de los aspectos clave de partida es la construcción de las categorías:

| | |
|-----------|------------|
| categoría | widgets Tk |
|-----------|------------|

| | | |
|----------|----------------|--|
| <i>P</i> | Pulsables | Button |
| <i>E</i> | Editables | entry, text |
| <i>S</i> | Seleccionables | listbox, menu, menubutton, checkbutton, radiobutton, scrollbar |
| <i>N</i> | Pasivos | label, image |

Tabla 0-1: Categorías de los elementos de interfaz

Prácticamente todos los entornos tipo WIMP (*windows, menus, icons pointer*) actuales utilizan este tipo de *widgets*. En cualquier otro lenguaje de comunicación gráfico, se dispone análogamente de elementos básicos que responde a un *modo de comunicación*. La existencia de estos modos es la que hace familiar, fácil y coherente el uso de cada sistema interactivo.

4 Evaluación dinámica paramétrica de la complejidad: EDPC

Si en cada momento de la actuación de una interfaz, podemos conocer el estado real de la complejidad $C(N)$, podemos construir una función mucho más informativa e interesante $C(N,t)$ que nos permita llevar una cuenta más realista del estado del sistema. Esta nueva posibilidad ha sido uno de los incentivos mayores a la hora de estudiar este mecanismo de análisis de la complejidad. En la mayoría de los métodos de evaluación de la complejidad se ignora, por completo esta posibilidad, ignorando así el fenómeno cambiante del comportamiento de los propios interfaces. Si, sin embargo, podemos conocer la forma de evolucionar la complejidad durante el desarrollo de una tarea, podemos dar una dimensión nueva a la estimación de la complejidad de la interfaz. Es fácil establecer, gracias al acúmulo de conocimientos en psicología cognitiva, cuál es el proceso de choque ante la complejidad más adecuado durante el proceso de interacción. Una interfaz excesivamente compleja al principio pero que se simplifique gradualmente puede ser adecuada para tareas delicadas que requieran la máxima atención. Un proceso en el que la complejidad vaya creciendo gradualmente y finalmente vuelva a decrecer parece el más aconsejado en general para tareas largas. Finalmente, los procesos cuya complejidad aumenta al final parecen más adecuados para usuarios inexpertos que van ganando confianza con el uso.

Básicamente la EDPC se realiza en base al sondado de los constructores de elementos interactivos. La aparición de estos elementos activa las correspondientes rutinas "espía" que así pueden registrar la actividad sobre ellas. En las rutinas de sondado se activarán conexiones con sistemas remotos en los entornos de desarrollo cooperativo que permitirán a los analistas reconocer el estilo de los usuarios y la actividad real en las interfaces. Un sencillo esquema básico de sondado, por ejemplo, sería:

```
set sondaFN "sondar.out"
proc sondar args {
  global sondaFN
  foreach p $args {
    if [regexp {\-([^\+])} $p n nn] {
      set sondaFN $nn
    } else {
      if {[info command _$p] == ""} {
        rename $p _$p
        proc $p args {
          global sondaFN
          set cmd [[index [info level 1] 0]]
          set f [open $sondaFN a]
          puts $f "$cmd $args"
          close $f
          uplevel _$cmd $args
        } } } } }
proc dessondar args {
  foreach p $args {
    if {[info command _$p] != ""} {
      rename $p ""
      rename _$p $p
    } } }
}
```

5 Conclusiones

La metodología de AEC y EDPC permiten abordar el difícil problema de la evaluación de la usabilidad directamente durante el proceso de construcción en un Ciclo de Vida orientado a prototipos. Este método resulta además idóneo para abordar el objetivo final del *diseño participativo*, en el que el usuario es el que define, dentro de entornos cooperativos, su propio sistema. A la vez que el constructor de las herramientas CASE necesarias es capaz de conocer el uso que se hace de su interfaz gráfica, puede dejar al usuario un entorno adaptado a sus costumbres, en el que los caminos más frecuentes y los elementos más cómodos sean los preponderantes. Con estas métricas tanto estáticas como dinámicas de la interfaz, es posible hacer un seguimiento de la usabilidad en sistemas UIMS de cuarta generación sin la prácticamente inabordable inspección posterior de la usabilidad.

Referencias

- J. Nielsen "Usability Engineering". Academic Press. 1993
- J. Nielsen "Usability Inspection Methods". John Wiley & Sons. 1994
- Balbo, S. y Coutaz, J. "Automatic Evaluation in Human Computer Interaction". En The Ergonomics Society, Annual Conference. 13-16 de Abril de 1993.
- Coutaz, J., Salber, D. y Balbo, S. "Towards Automatic Evaluation of Multimodal User Interfaces" (Informe Técnico). Amodeus Project Document: SM/WP32. 1993
- Scapin, D. "Guide ergonomique de conception des interfaces homme-machine" (Informe Técnico). INRIA. 1986.
- Mantei, M. y Teory, T. "Cost/benefit analysis for incorporating human factors in the software lifecycle". CACM 31 (428-439). 1988.
- Guevara, A., Aguayo, A. y Falgueras, J. "End-User Oriented Cooperative Design: A New Approach to Human-Human Intercommunication via Computer Resources". En Proceedings IFIP 8.2. 1995.
- Card, S.K., Moran, T.P. y Newell, A. "The Psychology of Human-Computer Interaction". Hillsdale, NJ: Lawrence Erlbaum Associates. 1983.
- Brooks, R.E. "The case for the specialized interface". IEEE Software, 10(2), 86-88. 1993.
- Staggers, N. "Impact of screen density on clinical nurses' computer task performance and subjective screen satisfaction". International Journal of Man-Machine Studies, 39,775-792. 1993.
- Norman, D.A. "Things That Make Us Smart: Defending Human Attributes in the Age of the Machine". Reading, Massachusetts: Addison-Wesley. 1993.

Evaluación de un simulador para la estiba y desestiba portuaria

A. Lucas, P.M. Valero, R. García-Ros [†], M. Pérez [‡]

[†] Instituto Universitario de Tráfico y Seguridad Vial (INTRAS)
Universitat de València, C/ Hugo de Moncada, 4. Bajo, 46010, Valencia
Tlf: 963.393.880 Fax: 963.393.881 e-mail: lucalba@uv.es

[‡] Instituto de Robótica
Universitat de València, Polígono de la Coma s/n, Paterna –46071. Valencia
Tlf: 963.983.586

Resumen

Diseñar y construir un simulador implica elaborar un modelo simplificado de la realidad. Esto conlleva el riesgo de eliminar elementos clave para la realización de las tareas, quedando degradado el potencial del simulador. MEVAFIS (Método para la EVALUACIÓN de la FIDELIDAD en Simuladores) es la metodología que hemos desarrollado para evaluar esta discrepancia entre simulación y realidad. Fundamentalmente, consiste en a) llevar a cabo un análisis de tareas, b) describir minuciosamente sus aspectos esenciales (Método GOMS) y c) contrastarlos después con el simulador. MEVAFIS ha sido utilizado para examinar el desempeño del Simulador de Grúa Pórtico desarrollado por la Universitat de València.

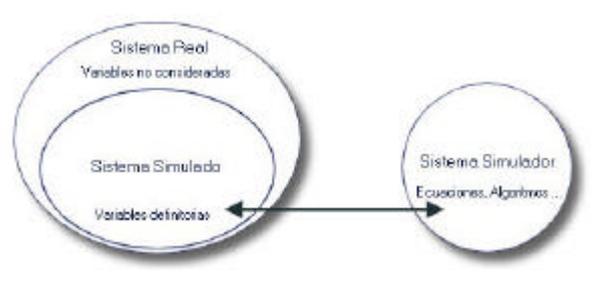
Palabras clave: simulación, análisis de tareas, evaluación de la fidelidad

1 Introducción

En esta comunicación se presenta el trabajo realizado con el fin de determinar la puesta a punto del Sistema Simulador orientado a la evaluación y el entrenamiento de los operarios del sector de estiba y desestiba que se está utilizando, a través del departamento de formación de SEVASA (Sociedad Estatal de Estiba y Desestiba del Puerto de Valencia), en el Puerto de Valencia.

En su sentido técnico habitual, entendemos por simulación el empleo de un modelo particular utilizado con el objetivo de obtener conclusiones acerca de un sistema del mundo real [1]. El modelo deberá involucrar todas las variables y/o factores consigan que el sistema evolucione de manera similar al real a través del tiempo. Podemos distinguir entre sistema simulador, sistema simulado y sistema real. Como muestra la Figura 1 [2], el sistema simulado

no es más que el compendio de variables o factores reales que tomaremos en consideración para construir el sistema simulador. Se trata de que este último represente dichos factores



con el mayor realismo posible.

Figura 1. Relación entre el sistema real (sistema simulado) y el sistema simulador

En cualquier simulación que se realice de un proceso físico real siempre arrastraremos un número finito de variables que no tendremos en consideración principalmente por dos razones: la primera, que el coste de su inclusión en el modelo es elevado y la segunda, que su repercusión o importancia en el mismo es mínima. Debemos considerar siempre cualquier simulador como una aproximación al sistema real, y nunca como el sistema real propiamente dicho. Esto tiene sus ventajas y sus inconvenientes. Lógicamente los sistemas simuladores explotarán dichas ventajas haciendo que el uso de simuladores llegue a ser, bajo determinadas condiciones, más aconsejable y útil que el uso del sistema real. Por ejemplo, un simulador puede contribuir a la seguridad, en especial cuando la práctica en la vida real conlleva un alto riesgo. En segundo lugar la simulación presenta un gran potencial para la evaluación, formación y entrenamiento [3]. Por último, el uso de simuladores es rentable, pues evita los cortes en la productividad por cuestiones relativas a la formación. Tales razones hacen particularmente conveniente usar un sistema de simulación en el ámbito portuario.

La consecución de un buen ajuste simulación/realidad debe considerar, entre otros, aspectos como la visualización, la fidelidad y la proporción en los tamaños, las sensaciones táctiles, auditivas, y dinámicas, etc. Con todo un simulador no es más que un modelo de la realidad: aún considerando multitud de elementos muchos de ellos se perderán, y con esta pérdida nos arriesgamos a convertir las tareas realizadas en el simulador en algo diferente a las tareas reales. La selección de variables es un momento crítico del proceso de diseño y debe incorporar una evaluación sobre la medida en que es seria esa discrepancia con la realidad.

Nuestra respuesta a este problema queda incorporada en una metodología de trabajo para hacer frente a esta situación. MEVAFIS (Método para la EVALUACIÓN de la FIDELIDAD en

Simuladores) considera varias acciones para evaluar la importancia de la pérdida de fidelidad de un simulador:

1. Llevar a cabo un análisis pormenorizado de tareas (entrevistas a usuarios, observación de campo, consulta de manuales técnicos y de instrucciones, etc.) que permita abstraer sus características esenciales.
2. Elaborar una descripción formal de las tareas, quedando así registrado el resultado de la observación y el análisis anterior.
3. Llevar a cabo una comparación entre la descripción elaborada y el simulador.

2 MEVAFIS: Evaluando la fidelidad de un simulador.

2.1. Los modelos conceptuales del diseñador y el usuario.

Los modelos conceptuales que implícita, o explícitamente, intervienen en el diseño y la elaboración de un sistema simulador son de extrema importancia. Los seres humanos, con su capacidad limitada para el procesamiento de la información [4], atienden a un conjunto limitado de variables, creando así un modelo determinado de su entorno. Puede haber tantos modelos de un contexto dado como personas atienden al mismo, pero nos interesan dos tipos de persona en particular: el diseñador del sistema y el usuario del sistema. Como hemos podido ver (Fig. 1) el diseñador ha de dar dos pasos: seleccionar un conjunto de variables definitorias y presentarlas luego en el sistema simulador. La selección depende, por lo general, del sentido común del diseñador del sistema [5], mientras que la presentación depende de sus posibilidades técnicas. El primer paso suele llevarse a cabo mediante procedimientos que permiten “copiar la realidad”, por ejemplo empleando un vídeo o cámaras fotográficas, lo que permite al diseñador dedicar el tiempo que considere oportuno para observar el contexto físico y el desempeño de la tarea tantas veces como precise. Entonces elegirá los detalles que le parezcan imprescindibles y pasará al segundo paso: modelado a través de ecuaciones y algoritmos, etc. Primero decide *qué* se modela y entonces empieza a pensar *cómo*.

La cuestión es si el diseñador puede asumir que ha elegido reproducir todos los factores necesarios y pertinentes para la misma. El diseñador, en última instancia, pone en juego una serie de modelos mentales [6], más o menos afortunados y más o menos evidentes a la luz de lo que, por ejemplo el vídeo y su sentido común, le indican. Pero, por lo general, su percepción y sus sistemas de representación no están educados para hacer la tarea. Pensemos en la diferencia entre el lego y el experto en la construcción. Precisamente esta sensibilidad sobre el contexto pertinente es lo que diferencia a un experto de un lego. Donde

el lego no ve nada de particular, dejándose llevar por la impresión del conjunto o atendiendo a los detalles más obvios, el profesional ve un fallo de pintura, una ventana o un interruptor torcidos, un muro defectuoso, etc. Lo más normal es que el diseñador carezca de ese conocimiento sobre la realidad de la tarea y que no vea las cosas como otros –por ejemplo, los usuarios expertos- las ven, por lo que puede incurrir en errores de selección importantes. Por eso es de la máxima importancia conocer los modelos conceptuales del usuario que desempeña la tarea en el contexto real, y no conviene desdeñar su visión de las cosas.

2.2 El análisis de tareas

Aspectos preliminares

La obtención y la sistematización de la información necesaria para contrastar la fidelidad del simulador requiere una inmersión adecuada en el contexto en el que se realiza la tarea simulada. Aunque es posible emplear informes, manuales técnicos o de instrucciones, etc., para obtener la información necesaria, la parte substancial de esta información debe obtenerse, a través de la observación, en el medio natural para el que se preparan el simulador y el diseño de instrucción. Con este propósito se han establecido una serie de entrevistas de toma de contacto con los expertos en la tarea, y también con los aprendices de distintos niveles. El análisis del conocimiento experto es fundamental por una serie de motivos. En primer lugar, da a conocer el nivel de respuesta mínimo que debe exigirse al simulador, o si se prefiere, contribuye a la evaluación de la fidelidad mínima necesaria para llevar a cabo la tarea al máximo nivel. En segundo lugar, proporciona información relativa a la ejecución de la tarea (partiendo del supuesto de un conocimiento y práctica óptimos por parte del experto). Por último, gracias a su experiencia, el experto nos ofrece conocimiento específico sobre las situaciones atípicas y las excepciones en el procedimiento habitual en la tarea. También ayuda a identificar más fácilmente los errores y las soluciones que se suelen proporcionar a los mismos. En suma, la integración de los expertos, siguiendo los principios generales de la “evaluación cooperativa” [7], ha sido clave.

Gracias a la observación se ha tomado conciencia de los parámetros que afectan al desempeño: la experiencia del entrevistado, el contexto de trabajo, las variaciones en la tarea, las condiciones meteorológicas, etc. Durante este tiempo el analista ha observado el proceso de estiba/desestiba tratando de aprovechar al máximo el conjunto de perspectivas físicas presentes (desde la cabina de mandos, a pie de grúa, desde la cubierta del barco), las variaciones según el tipo de grúa que opera y los parámetros de carga. La observación y el establecimiento del desempeño de la tarea permite al analista absorber y componer el conocimiento de la tarea y el contexto en el que se desempeña. Una vez la percepción del analista está mínimamente afinada, y gracias a las observaciones y comentarios de los expertos y aprendices (“thinking aloud”) queda la parte más formal del análisis de tareas, que implica la elaboración de una ficción psicológica que sistematice y describa los pasos

fundamentales en la tarea. Esta descripción quedará reflejada fundamentalmente en el análisis GOMS.

El método GOMS

El modelo GOMS es una representación del conocimiento práctico, del *cómo se hace* que un usuario precisa para desempeñar tareas con un dispositivo o con un sistema determinado [5]. GOMS, acrónimo de G(oals) –objetivos –, O(perators) –operadores –, M(ethods) –métodos –, y S(election rules) –reglas de selección, es resultado de las investigaciones realizadas en el centro de Palo Alto subvencionadas por Xerox en la década de los 70. Sus orígenes pueden buscarse en los trabajos seminales de Newell y Simon [8] aunque el desarrollo y la formalización del modelo se debe a Card, Moran y Newell [9]. GOMS ha sido probablemente el método formal de representación de interfaces de usuario más influyente en el campo de la interacción hombre-ordenador [10].

El modelo GOMS parte de la idea de que la estructura cognitiva de un usuario a la hora de realizar una tarea con un sistema se compone de una serie de conjuntos de objetivos, operadores, métodos y reglas de selección:

- Un objetivo es algo que el usuario quiere conseguir hacer. El analista tratará de identificar y representar los objetivos de los usuarios normales que, en la medida en que impliquen subobjetivos, tendrán una disposición jerárquica.
- Los *operadores* son actos cognitivos, motores o perceptivos que el usuario lleva a cabo. Al igual que los objetivos, forman un par acción-objeto, pero en un modelo GOMS un objetivo es algo que debe llevarse a cabo, mientras que un operador simplemente se ejecuta. Esta división se basa en la intuición y depende del nivel de análisis. La conducta del usuario podría ser trazada como una secuencia de estas operaciones. Los operadores definen la fineza –o grano - de análisis con el que el modelo GOMS ha sido realizado.
- Los *métodos* describen una secuencia o serie de pasos que llevan a cumplir un objetivo. Es una de las maneras en las que un usuario almacena su conocimiento de una tarea. Se supone que ese conocimiento ya se posee (el usuario no debe implicarse en un proceso de solución de problemas para lograr sus objetivos).
- Las *reglas de selección* se formulan con el propósito de dirigir el control sobre el método apropiado para cumplir un objetivo. Las reglas de selección facilitan la elección idónea de cada método en cada situación.

GOMS es un modelo hipotético de cómo hacemos cosas, en el que el conocimiento práctico es descrito de forma que pueda ser, de hecho, llevado a la práctica. La idea con la que se construye el modelo es que alguien pueda seguir la descripción de GOMS, ejecutar las

acciones descritas y llevar realmente a cabo la tarea. La identificación y la definición de los objetivos del usuario suele ser difícil debido a que, como hemos visto, se debe examinar la tarea que el usuario trata de desempeñar con cierto detalle. Este examen implica ir más allá del dispositivo que se quiere utilizar para incluir, además, el contexto del trabajo en el cual se utiliza ese dispositivo.

Como hemos visto, llevar a cabo un análisis de tareas GOMS implica definir y describir con una notación formal los objetivos, operadores, métodos y reglas de selección del usuario. Precisamente uno de los problemas del modelo GOMS es la dificultad técnica que conlleva la escritura de las reglas de producción, proceso muy similar a la programación en lenguaje ensamblador. La literatura ofrece distintas alternativas para hacer más asimilable el proceso de elaboración de los modelos, entre las que destacaremos el NGOMSL [5], de "Natural GOMS Language" (Lenguaje GOMS Naturalizado). La ventaja de NGOMSL consiste en mantener una relación directa con el modelo de producción de reglas subyacente (es decir, GOMS), siendo a la vez relativamente fácil de leer y escribir.

La comparación entre la descripción formal y el simulador

Como consecuencia del trabajo anterior fue posible constatar la existencia de divergencias entre el funcionamiento del sistema simulador y el desempeño real de la tarea. De este modo, se establecieron un conjunto de objetivos informáticos dirigidos a incrementar la fidelidad del simulador. Los objetivos se han centrado en el concepto de fidelidad en el equipamiento (su apariencia general, la funcionalidad de la instrumentación), sensorio-perceptiva (dinámica -aceleraciones, vibraciones e inercias -, visual y auditiva), respecto a los parámetros del entorno portuario y de las situaciones concretas de la tarea (efectos atmosféricos, distribución de las cargas en el barco, tipo de contenedores, etc.) y psicológica (valoración de los monitores como duplicado del real).

3 Reflexiones y conclusión

El proceso seguido en MEVAFIS nos deja relativamente satisfechos en lo tocante a la evaluación de la fidelidad de un simulador. Uno de los problemas lo plantea la exhaustividad de GOMS con relación a la descripción de la acción de la que se deriva la estiba y la desestiba portuaria. Si consideramos que los objetivos del usuario, pero también el efecto de sus acciones en el entorno determinan, en una suerte de "interacción cíclica", la acción final del usuario [11], resulta evidente que GOMS aporta la mitad del proceso. Con GOMS queda cubierta la descripción de los objetivos y la manera en la que estos se generan en la mente del usuario a partir de sus intenciones más generales (en nuestro caso estibar o desestibar), los vectores generales de la acción. Sin embargo, ¿qué hay del contexto que genera los indicadores que contribuyen de manera específica a la acción? Es cierto que este contexto queda indirectamente reflejado en GOMS, a través de la incorporación de las reglas de

selección. Sin embargo, el contexto particular de la estiba/desestiba portuaria, en el que finalmente son la multiplicidad de referencias mecánicas, visuales, dinámicas, acústicas, etc., las que integran el “micro-avance”, los vectores inmediatos de la acción, precisa a nuestro juicio de una valoración aparte que no parece razonable integrar en el análisis GOMS.

Un enfoque empírico complementario a GOMS podría consistir en llevar a cabo pruebas de evaluación con usuarios de distintos niveles (novatos, intermedios, expertos). Por un lado, se trataría de tomar una serie de registros en condiciones controladas, observando en principio tareas muy simples, para comprobar la existencia y el grado de la discrepancia entre la grúa real y el simulador, dentro de cada grupo de usuarios. Por otro lado, y dado que es posible obtener información sobre la ejecución de la tarea programando el sistema para guardar los registros relativos a una serie de variables (número de traslados, tiempo de enganche, distancias de seguridad, colisiones, etc.), se podría asimismo comparar la ejecución de los usuarios de distintos niveles. De este modo, sorprende constatar que, sin demasiado esfuerzo adicional, la misma tecnología que da soporte a una interacción determinada con el usuario permite además evaluar la calidad de la dinámica interactiva que el usuario mantiene con la tecnología.

Referencias

- [1] McHaney, R. (1991). *Computer simulation: a practical perspective*. Academic Press.
- [2] Bayarri, S. (1995). Técnicas de visualización y simulación en tiempo real de entornos de conducción: nuevos algoritmos, estructuras de datos y su gestión,. Tesis Doctoral. Universitat de València.
- [3] García-Ros, R., Valero, P., Fernández, M., Lucas, A. (1999). Tecnología de simulación y diseño de instrucción. *III Congreso Internacional de Psicología y Educación*. Santiago de Compostela.
- [4] Fiske, S., y Taylor, S. (1991). *Social Cognition*. McGraw-Hill.
- [5] Kieras, D. (1991). *A guide to GOMS task analysis*. Taller presentado en CHI'91. Manuscrito no publicado. University of Michigan.
- [6] Norman, D.A. (1988). *La psicología de los objetos cotidianos*. New York: Basic Books.
- [7] Monk, A., Wright, P., Haber, J., y Davenport, L. (1993). *Improving your Human-Computer Interface*. Prentice Hall, London.
- [8] Newell, A. y Simon, H. (1972). *Human problem solving*. Englewood Cliffs. NJ: Prentice-Hall.
- [9] Card, S., Moran, T., y Newell, A. (1983). *The psychology of human computer interaction*. Hillsdale, New Jersey: Erlbaum Associates.
- [10] Valero, P.M. (1996). *Descripción de interfaces hombre computador por medio de métodos formales: aplicación de métodos para la evaluación de un interfaz simulado*. Tesis Doctoral no publicada. Universitat de València.

- [11] Monk, Andrew (1998). Cyclic interaction: a unitary approach to intention, action and the environment. *Cognition*, 68, pp. 95-110.

Evaluación del conocimiento adquirido durante la interacción

José J. Cañas, Adoración Antolí, José F. Quesada, Inmaculada Fajardo
Grupo de Ergonomía Cognitiva
Departamento de Psicología Experimental
Universidad de Granada

1 Introducción

Las investigaciones realizadas por nuestro grupo han tenido como objetivo explorar la relación entre representación del conocimiento y el aprendizaje del uso y la programación en ordenadores. Desde la Psicología Cognitiva, la mayoría de las definiciones de aprendizaje incluyen el concepto de 'representación' (Rumelhart y Norman, 1978; Bajo y Cañas, 1991). De esta manera, se considera que el aprendizaje no sólo da lugar a un cambio en la conducta sino que también produce un cambio en las estructuras representacionales internas. Por ello, de acuerdo con nuestra hipótesis de trabajo, la adquisición de las habilidades necesarias para interactuar con los ordenadores puede ser facilitada por la adquisición de un buen modelo mental del ordenador.

En primer lugar, haremos un resumen de las técnicas de elicitación del conocimiento que utilizamos para obtener una estimación del conocimiento adquirido por los usuarios y, a continuación expondremos tres de nuestras investigaciones donde estas técnicas han sido utilizadas.

2 Técnicas de elicitación del conocimiento

En nuestras investigaciones hemos utilizado un conjunto de técnicas llamadas genéricamente 'Técnicas de Elicitación del Conocimiento' que tienen como objetivo obtener una estimación del conocimiento adquirido por el usuario y como éste está representado en su memoria. Más concretamente, hemos utilizado los llamados 'Métodos Indirectos de Elicitación' que forman un conjunto de pruebas psicométricas que proporcionan datos sobre la proximidad, en la mente del usuario, de parte de los conceptos que componen su conocimiento.

El fundamento teórico de estas técnicas parte de que, desde una perspectiva psicológica, el aprendizaje de una herramienta informática requiere la adquisición de una representación mental de los conceptos relacionados con dicha herramienta. Además, las teorías psicológicas de representación del conocimiento generalmente suponen que la información

sobre un dominio de conocimiento está organizada y que, al menos, parte de esta organización está basada en las relaciones de similitud o proximidad entre conceptos. Se asume también en estas teorías que no todos los conceptos están igualmente relacionados y que es posible utilizar formalismos de representación para plasmar estas diferencias en la relación semántica. Por ejemplo, está claro que existe una mayor similitud o proximidad conceptual entre los conceptos de 'silla' y 'mesa' que entre este último y 'pájaro', y que este mayor grado de relación refleja el conocimiento que tenemos sobre la categoría 'mueble'. De la misma manera, una persona que usa una herramienta informática, por ejemplo, el procesador de textos MicrosoftWord, tendrá una representación mental de los conceptos de este procesador en la que la relación o proximidad entre los conceptos refleje su conocimiento de él.

La aplicación de estas técnicas de elicitación del conocimiento comienza por la selección de los conceptos y objetos de la interfaz que se consideran más importantes para interactuar con ella. Después se le pide a una persona que realice una tarea con los conceptos para obtener datos de proximidad conceptual. Dos de las tareas más usadas son la de *juicios de relación* y la de *clasificación*. En la tarea de juicios de relación se le presentan pares de conceptos y la persona da un valor de relación en una escala numérica, por ejemplo, de 1 a 6. Un valor de 1 significa que la persona piensa que los conceptos no están relacionados y un valor de 6 significará que están muy relacionados. En la tarea de clasificación, se le presentan todos los conceptos y tiene que hacer grupos con ellos. De los datos obtenidos con cualquiera de estas tareas se obtiene una matriz de proximidad donde las columnas y las filas representan los conceptos, y las celdas contienen los datos de proximidad entre ellos, es decir, los juicios de relación dados por la persona.

Finalmente, esta matriz de proximidad es sometida a un procedimiento matemático con el que se obtiene la representación conceptual de la persona evaluada. En los últimos años se han propuesto varios procedimientos para llevar a cabo esta transformación pero la investigación ha demostrado que no existe un procedimiento que permita extraer toda la información contenida en una matriz de juicios de relación conceptual y, en la mayoría de las ocasiones es necesario combinar varios de ellos. Esto es debido, fundamentalmente, a que los diversos procedimientos han sido desarrollados en el contexto de diferentes teorías psicológicas de representación de conocimiento y cada uno ofrece información sobre características diferentes del conocimiento de un experto. De entre todos los propuestos, los dos procedimientos más usados son el Escalamiento Multidimensional y el análisis Pathfinder. Estos dos métodos son los que en la literatura actual en Psicología e Inteligencia Artificial se consideran más apropiados por cubrir aspectos diferentes y complementarios de la representación del conocimiento.

El análisis Pathfinder (Schvaneveldt, 1990) está basado en la teoría matemática de grafos y ha sido desarrollado debido a la importancia emergente que en la Psicología y la Inteligencia Artificial ha adquirido la representación del conocimiento en estructuras de redes

semánticas. La teoría matemática de grafos es el procedimiento adecuado para trabajar con redes semánticas. Como en las teorías de redes, este procedimiento representa los conceptos en nodos de una red, y las relaciones semánticas entre los conceptos como punteros que unen estos nodos.

Básicamente, el procedimiento consiste en reducir un grafo completo, donde todos los nodos están relacionados con todos los demás, y que representa la matriz de juicios de relación, en un grafo donde solo están representadas las relaciones importantes, desde el punto de vista de la persona, entre los conceptos. Es utilizado cuando lo que se quiere obtener son las relaciones locales entre pares de conceptos. Si la relación entre dos conceptos es muy importante en la representación mental de una persona, esta relación se verá reflejada en un puntero directo entre ellos en el grafo. Por el contrario, si la relación no es importante y puede ser deducida a partir de las relaciones con otros conceptos, no existirá un puntero directo entre ellos, y la importancia relativa de esta relación se inferirá del número de punteros y nodos que hay que recorrer entre los dos conceptos en el grafo. En el diseño de interfaces, el procedimiento Pathfinder se ha revelado útil para investigar las estrategias de navegación de los usuarios (Cooke, y Schvaneveldt, 1987)..

En el escalamiento multidimensional el objetivo fundamental es representar un conjunto de conceptos en un espacio multidimensional de tal manera que la distancia euclidiana entre dos objetos en ese espacio se corresponda lo más fielmente posible con el juicio de proximidad conceptual asignado por una persona o un conjunto de personas a ese par de conceptos. Este tipo de análisis está estrechamente ligado a las teorías de representación dimensionales, y ofrece una forma de obtener una representación empírica de las dimensiones que relacionan los distintos conceptos de un área de conocimiento determinada.

A diferencia del procedimiento anterior, el escalamiento multidimensional encuentra una solución global considerando todas las relaciones entre los conceptos simultáneamente. Es decir, mientras que el procedimiento Pathfinder es útil para describir relaciones locales entre pares de conceptos, el escalamiento multidimensional ofrece las dimensiones globales que relacionan todos los conceptos con todos los demás.

En el diseño de interfaces, el Análisis multidimensional se ha revelado útil para descubrir las estructuras abstractas en la representación del conocimiento de los expertos usuarios de los sistemas. Son aquellas estructuras, más difíciles de adquirir por los nuevos usuarios, que determinan las estrategias globales de trabajo con el sistema.

3 Evaluación del conocimiento adquirido en un curso de procesadores de texto

En un estudio realizado por Cañas, Antolí y Quesada (2000) se combinaron los dos procedimientos para evaluar el conocimiento que un grupo de nuevos usuarios del procesador de textos MSWORD habían adquirido después de un curso de introducción ofrecido por la empresa donde trabajaban. Los participantes realizaron una tarea de clasificación con un conjunto de conceptos y objetos de la interfaz del procesador de textos antes y después del curso.

Una de las posibilidades que tienen estas técnicas es que permiten evaluar el conocimiento adquirido por los usuarios comparándolo con el conocimiento que tiene los expertos en ese dominio. Para ello, se suele pedir al grupo de expertos que realicen también la tarea de clasificación de conceptos y se someten sus matrices de similitud a los algoritmos matemáticos para obtener sus representaciones mentales. En el estudio de Cañas, Antolí y Quesada (2000) los expertos que realizaron también la tarea de clasificación fueron los monitores de los cursos. Los resultados de este estudio mostraron que al final del curso, los participantes habían adquirido un conocimiento sobre WORD similar al que tenían los monitores. De esta manera, las técnicas de elicitación del conocimiento se mostraron como válidas para detectar el aprendizaje que había tenido lugar en el corto período de tiempo que transcurrió entre las dos fases.

4 Evaluación del conocimiento adquirido por los programadores cuando usan un ‘trazador’

En los últimos años ha cobrado importancia la idea de que la tarea de programar puede ser facilitada con diversas herramientas que están a disposición de los programadores. Una técnica diseñada para facilitar la adquisición del modelo mental del ordenador es la conocida como ‘Trazador’ y está basada en las investigaciones iniciadas por Mayer (1994). Un trazador es una utilidad incorporada a un lenguaje de programación que tiene como finalidad permitir al programador visualizar el estado interno de la memoria del ordenador y la secuencia de acciones que éste realiza como consecuencia de la ejecución de las instrucciones del programa. La eficacia de los trazadores ha sido puesta a prueba por Cañas, Bajo y Gonzalvo (1994) en un experimento donde se utilizó la técnica de elicitación del conocimiento del escalamiento multidimensional. En el sistema utilizado por estos autores, los programadores de C podían ver en un diseño de ventanas una representación del programa, de las variables con sus valores y del “output”. El programador podía ejecutar, una a una, las instrucciones del programa y observar en las ventanas de variables y de “output” lo que ocurría con éstas cuando aquellas se ejecutaban. Puesto que el programador tenía control sobre el ritmo de ejecución podía observar como, por ejemplo en un bucle, unas instrucciones dependen de otras.

En este experimento, dos grupos de estudiantes sin experiencia en el uso de ordenadores participaron en un curso de programación en C. Un grupo utilizó el trazador durante las clases y otro grupo no lo utilizó. Todos los estudiantes pasaron por dos períodos de evaluación de sus conocimientos, uno a la mitad del curso y otro al final. En ambos períodos, los estudiantes tuvieron que realizar dos tareas de programación que sirvieron para medir las variables dependientes de ejecución. En una tarea debían encontrar un error y corregirlo en un programa defectuoso que se les proporcionaba, y en otra se les pidió que escribiesen un pequeño programa. También en ambos períodos realizaron una tarea de juicios de relación de conceptos. En consonancia con investigaciones previas (Mayer, 1981) estos resultados mostraron que el modelo mental del ordenador puede facilitarse con el uso de un trazador que hace visible la estructura y funciones de los componentes de éste. También mostraron la utilidad de la técnica de escalamiento multidimensional para medir el modelo mental adquirido por un estudiante.

5 Evaluación del conocimiento adquirido por los programadores cuando utilizan lenguajes visuales

En la investigación llevada a cabo por Navarro y Cañas (2000) se utilizaron las técnicas de elicitación del conocimiento para estudiar la posible ventaja que tienen los lenguajes de programación visuales sobre los lenguajes llamados textuales. En un lenguaje textual, como Pascal o C, el programador escribe las instrucciones del programa en un lenguaje cuasi-natural, generalmente inglés. Por el contrario, en un lenguaje visual se utiliza información pictórica y espacial para especificar el código de un programa. Existen tres tipos de lenguajes visuales: lenguajes con organigramas (flowcharts), lenguajes sin organigramas y lenguajes de hoja de cálculo.

La investigación tenía por objeto resolver una situación anómala en la llamada Psicología de la Programación. La popularidad de la que gozan actualmente los lenguajes visuales está basada en la creencia en sus ventajas sobre los lenguajes textuales pero no en investigación empírica que sustentase esta creencia. Además, la escasa investigación llevada a cabo sobre los lenguajes visuales se ha centrado en los lenguajes con organigramas (Whitley, 1997) y sin embargo, los lenguajes visuales más utilizados actualmente son las hojas de cálculo. Tanto es así que, entre todas las aplicaciones informáticas de las que disponemos, las hojas de cálculo son las más usadas después de los procesadores de texto (Panko, 1988).

La investigación se ha centrado en una de las tareas que los programadores realizan, la comprensión de un programa escrito por otra persona, donde se cree que es más evidente la supuesta ventaja que las hojas de cálculo tienen sobre los lenguajes textuales. Según las modernas teorías propuestas para explicar los procesos de comprensión con lenguajes

textuales (v.g. Pennington, 1987) cuando un programador intenta comprender un programa pasa por dos fases. En primer lugar, segmenta el programa sobre la base de las relaciones de flujo de control (Control Flow), que es la secuencia en las que las diferentes partes del programa son ejecutadas. En una segunda fase, adquieren la información más profunda sobre el Flujo de datos (Data flow) que es la secuencia de cambio de los valores de las estructuras de los datos como consecuencia de la ejecución de las instrucciones del programa. La información sobre flujo de datos es considerada información más profunda semánticamente que la información sobre flujo de control. Por tanto, se asume que los programadores expertos son capaces de adquirir más fácilmente la información sobre flujo de datos que los novatos. Además, un lenguaje será mejor desde el punto de vista de la comprensión si facilita la adquisición de esta información.

En Psicología Cognitiva se sabe que los procesos que intervienen en la creación de imágenes facilitan la comprensión (Paivio, 1977; Potter y Faulconer, 1975; Glasser, 1992). También existen datos que indican que las características visuales de las notaciones tienen un efecto facilitador en programación (Scalan, 1989; Saariluoma y Sajaniemi 1989, 1991). Por esta razón Navarro y Cañas hipotetizaron que la posible ventaja de los lenguajes visuales sobre los textuales se puede deber a que los primeros facilitan la adquisición de la representación de flujos de datos al poner en funcionamiento los procesos cognitivos que procesan imágenes.

En la investigación participaron dos grupos de programadores, uno de Hojas de Cálculo y otro del lenguaje de programación C. La sesión experimental constaba de dos partes. En una de ellas se les presentaba un pequeño programa escrito en el lenguaje en el que eran expertos y se les pedía que lo leyesen y memorizaran. En otra parte también se les presentaba un programa pero en este caso se les decía que había un error y que debían corregirlo.

Después de leer o modificar los programas, los programadores realizaron una tarea de elicitación del conocimiento en la que se les presentaron segmentos de los programas y se les pidió que hicieran grupos con ellos. Con los grupos se construyeron matrices de similitud entre los segmentos que fueron sometidas a un análisis con la técnica del Pathfinder (Schvaneveldt, 1990). Esta técnica permite obtener la representación mental (modelo mental) que los programadores han adquirido. Estas representaciones mentales fueron comparadas con representaciones teóricas creadas por programadores expertos. Una representación teórica correspondía a la representación del programa basada en el flujo de control y otra a la representación del flujo de datos. Los resultados de esta comparación mostraron, de acuerdo a la hipótesis de los investigadores, que los programadores de C adquirieron una representación del flujo de control mejor que la representación del flujo de datos. Por el contrario, los programadores de hojas de cálculo adquirieron igual de bien, y mejor que los programadores de C, ambas representaciones.

6 Bibliografía

- [1] Bajo, M.T. y Cañas, J.J. (1991) *Ciencia Cognitiva*. Madrid: Editorial Debate. ISBN-7444-490.
- [2] Cañas, J.J.; Bajo, M.T.; Gonzalvo, P. (1994) *Mental Models and Computer Programming*. *International Journal of Human-Computer Studies*. 40, 795-811.
- [3] Cañas, J.J., Antolí, A. y Quesada, J.F. (2000). *Evaluación del conocimiento adquirido en un curso de procesadores de texto*. [manuscrito en preparación]
- [4] Cooke, N.J. and Schvaneveldt, R.W. (1987). *Effects of computer programming on network representations of abstract programming concepts*. *International Journal of Man-Machine Studies*, 29, 407-427.
- [5] Glaser, W.R. (1992) *Picture Naming*. *Cognition*. 42, 61-105.
- [6] Mayer, R.E. (1994) *Visual Aids to Knowledge Construction: Building Mental Representations from Picture and Words*. IN W. Schnotz, R. Kulhavy (Eds.). *Comprehension of Graphics*. North-Holland. pp. 125-138.
- [7] Navarro, R., and Cañas, J.J. (2000). *Are visual programming languages better? The role of imagery in program comprehension*. *International Journal of Human-Computer Studies*. [en prensa]
- [8] Paivio, A. (1991) *Dual Coding Theory: Retrospect and Current Status*. *Canadian Journal of Psychology*, 45 (3), 255-287.
- [9] Panko, R. (1988) *Object-oriented Spreadsheets: The Analytic Spreadsheet package*. IN *Proceedings of OOPSLA'86*. pp. 385-390.
- [10] Pennigton, N. (1987) *Stimulus Structures and Mental Representation In Expert Comprehension of Computer Programs*. *Cognitive Psychology*, 19, 295-341.
- [11] Potter, M.C.; Falconer, B.A. (1975) *Time to Understand Pictures and Words*. *Nature*. 253, 437-438.
- [12] Rumelhart, D.E.; Norman, D.A. (1978) *Accretion, Tuning and Restructuring: Three Models of Learning*. IN J. W. Cotton, R. Klatsky (Eds.) *Semantic Factors in Cognition*. NJ: Erlbaum.
- [13] Saarilouma, P.; Sajaniemi, J. (1989) *Visual Information Chunking in Spreadsheet Calculation*. *International Journal of Man-Machine Studies*. 30, 475-488.
- [14] Saarilouma, P.; Sajaniemi, J. (1991) *Extracting Implicit Tree Structures in Spreadsheet Calculation*. *Ergonomics*. 34 (8), 1027-1046.

- [15] Scalan, D.A. (1989) Structured Flowcharts Outperform Pseudocode: An Empirical Comparison. *IEEE Software*, 6 (5), 28-36.
- [16] Schvaneveldt, R.W. (1990). *Pathfinder Associative Networks: Studies in Knowledge Organization*. Norwood, NJ: Ablex Publishing.
- [17] Whitley, K.N. (1997) Visual Programming Languages and the Empirical Evidence For and Against. *Journal of Visual Languages and Computing*. 8 (1), 109-142.

Estudio Formal de Factores Humanos en el diseño de Sistemas Cooperativos

N.Padilla[†], M. Gea[‡].

[†] Departamento de Lenguajes y Computación.
Universidad de Almería, Carretera de Sacramento S/N, Almería
Tlf: 950.215.424 Fax: 950.215.129. e-mail: npadilla@ualm.es

[‡] Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada, Av. Andalucía 38, Granada
Tlf: 958.242.812 Fax: 958.243.179. e-mail: mgea@ugr.es

Resumen

Los modelos formales se pueden usar para encapsular conocimiento acerca de la interacción persona-ordenador. Sobre esta base, se ha diseñado un modelo para poder especificar propiedades relacionadas con los factores humanos en un contexto de trabajo cooperativo.

Palabras clave: CSCW, métodos formales, factores humanos

1 Introducción

El objetivo principal del campo de la interacción entre la persona y el ordenador (IPO) es proporcionar una comprensión de cómo la motivación humana, las acciones y la experiencia restringen la usabilidad del ordenador [1]. Para ello se requiere de conocimiento del comportamiento humano. La mayoría de la práctica actual en el diseño de IPO ha sido criticada por confiar sólo en el conocimiento que se ha obtenido a través de la experiencia. Este conocimiento debe ser interpretado por expertos para poder ser aplicado satisfactoriamente. Sin embargo, en un gran proyecto software, no es normal que todos los ingenieros tengan el mismo nivel de conocimiento sobre IPO. Estos ingenieros necesitan algún medio para comunicar ese conocimiento en el contexto de un diseño particular, y una alternativa es mediante su especificación mediante métodos formales [2,3,4].

Partiendo de esta idea, hemos diseñado un modelo formal para el estudio de sistemas cooperativos, de modo que permita estudiar propiedades de estos entornos desde el punto de vista individual y de la organización social sin necesidad de prestar atención a su representación interna. Para ello, es necesario conocer la forma en la que trabaja un grupo,

los protocolos sociales presentes en el contexto del grupo y manejar las relaciones sociales que aparecen en el grupo (*roles/tareas*).

En el apartado 2 presentaremos el modelo abstracto de sistemas cooperativos. En el apartado 3 analizaremos algunas propiedades de estos sistemas que están directamente relacionadas con el modelo de razonamiento humano. El apartado 4 desarrolla formalmente una propiedad usando nuestra técnica de especificación. Esta propiedad nos servirá como ejemplo para examinar la conexión del modelo abstracto con la fase de diseño.

2 Un modelo abstracto para sistemas cooperativos

El modelo Extended-PIE [5] (figura 1) es un modelo de caja-negra para modelizar sistemas cooperativos. Se basa en el modelo PIE [2] de sistemas interactivos. PIE fue descrito sólo en función de las entradas realizadas por el usuario y las salidas hacia el usuario. Sin embargo, no consideraba la posibilidad de cooperación entre diferentes participantes para cualquier tarea dada. De hecho, el usuario en el proceso de computación tenía poco efecto en el sistema.

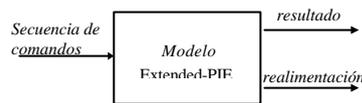


Figura 1. El modelo Extended-PIE

Nosotros hemos extendido el concepto de usuario a grupo incluyendo las siguientes características:

- a) Usuario. Es necesario ser conscientes que más de un usuario (artificial o real) existe en el sistema, donde cada uno puede tener un rol y por tanto distintos privilegios en el escenario. Los usuarios deben tener la capacidad de reconocerse así mismos en el entorno. Así, el dominio de los comandos de entrada son particionados para que cada partición describa el conjunto de comandos disponible para cada usuario, y por otra parte, junto con el comando de entrada se tiene en cuenta al usuario responsable de dicho comando.
- b) Medio. Se debe tomar en cuenta la forma de realizar estos comandos. Así, podemos ver si el participante es remoto, si usa diferentes dispositivos, etc. Además, podemos estudiar el dominio de los comandos con respecto a un dispositivo específico.

Considerando este modelo, veamos algunas de las propiedades abstractas de los sistemas cooperativos relacionadas con los factores humanos.

3 Factores Humanos en los sistemas cooperativos

El estudio de los factores humanos en el diseño de sistemas interactivos ha sido abordado ampliamente para el caso de la interacción entre persona y ordenador. Así por ejemplo, se debe tener en cuenta la diferencia entre habilidades humanas (cognitivas, sensoriales), personalidad y todos aquellos factores que pueden afectar a la comunicación (entorno, estrés, etc.) [6]. Sin embargo, existe poca bibliografía que estudie factores humanos en el contexto de grupos de trabajo, y este estudio es de gran importancia, ya que el éxito del groupware radica en la aceptación de este tipo de comunicación y colaboración por los participantes del sistema [7].

Cada persona establece un esquema conceptual (modelo mental) del funcionamiento de un sistema (computacional) que le sirve para comprender, analizar y predecir el comportamiento del sistema. Este modelo mental se sustenta bajo el paradigma del modelo de procesador humano [8] mediante el cual, se puede descomponer una actividad humana en una serie de etapas encaminadas a lograr un objetivo. Primeramente se establece un objetivo, indicamos nuestra intención, se especifica y ejecuta la secuencia de acciones necesarias, se percibe el efecto y finalmente se evalúa el resultado obtenido sobre el estado del sistema. Parte de estas actividades son mentales mientras que otras son motoras y sensoriales (mover un ratón, observar la pantalla, escuchar, etc.). La figura 2 muestra el proceso de percepción, análisis y ejecución.



Figura. 2 *Actividades cognitivas de un usuario*

Cuando nos encontramos en un sistema cooperativo, el usuario debe comprender tanto el sistema como la relación y las acciones de las personas que comparten el mismo sistema (ya sea síncrono o no). Desde este punto de vista, deberemos estudiar la percepción de grupo en un sistema cooperativo, cómo se pueden establecer acciones (intenciones) que impliquen a más de una persona, y cual es el grado de incertidumbre (en evaluación e interpretación) que provoca el sistema en los usuarios del mismo.

Las actividades de grupo se realizan a través de las tareas cooperativas. Una tarea, en este contexto, es una secuencia de entradas de usuario que permiten conseguir el resultado deseado. Estas tareas son equivalentes a la formulación de objetivos en el modelo de procesamiento humano teniendo en mente que se trabaja en grupo y por tanto debe considerarse ese cambio de mentalidad. De hecho, no todos los usuarios pueden realizar las mismas tareas en el sistema. Esto depende de los privilegios que cada usuario tiene en el sistema. El uso de roles en sistemas cooperativos permiten definir privilegios y, por tanto, sus tareas asociadas. Podemos asociar propiedades deseables del sistema (predecibilidad, alcanzabilidad, observabilidad [2]). Vamos a analizar la relación existente entre estas propiedades del sistema y los factores humanos que influyen en este proceso (desde un punto de vista cognitivo).

La propiedad *alcanzabilidad* asocia tareas con los estados posibles del sistema. Así, esta propiedad nos indica que estando en un determinado estado, podemos obtener cualquier otro estado a partir de una secuencia de entradas de usuarios. De igual forma que cuando definimos un objetivo necesitamos formular intenciones para alcanzar dicho objetivo. Estas intenciones expresan, en el modelo de procesamiento humano, la capacidad de poder expresar cualquier tipo de acción “disponible” dentro del sistema. Sin embargo, esta propiedad no nos informa de la complejidad para el usuario asociada con la secuencia de entrada que nos permita alcanzar el efecto deseado. Para ello se utilizan las *estrategias*. Las *estrategias* son los mecanismos elaborados para formular una intención. Éstas nos permitirán alcanzar un cierto estado, basándose en información del sistema, como puede ser la secuencia de entradas de usuarios que ya han sido presentadas, el estado actual del sistema, etc. En ocasiones, una determinada estrategia requiere que los usuarios retrocedan a determinados estados en el pasado (indicando qué deben deshacer) y/o que obtengan determinados estados en el futuro (indicando qué nuevas órdenes deben aplicar).

Para poder comprender el estado actual del sistema necesitamos de la propiedad *observabilidad*. Esta propiedad estudia la visión que cada usuario posee del comportamiento del sistema, y a menudo es diferente (corresponde a la percepción que el usuario obtiene del sistema que esta utilizando). Por ello, esta propiedad es subjetiva dado que depende de cada usuario y su rol en el escenario. Para obtener esta percepción del sistema necesitamos asociar los atributos visibles del estado con lo que actualmente ocurre

en el sistema. Esto nos lo ofrece la propiedad visibilidad. A partir de estas propiedades podemos definir la *predicibilidad*. Esta propiedad define la habilidad de ligar, desde el punto de vista del usuario, lo que se ha comprendido sobre el estado y el progreso actual del sistema con el comportamiento futuro del sistema. Es decir, relaciona la interpretación y evaluación con las nuevas intenciones en el futuro. En el caso de tareas cooperativas, el usuario puede llegar a interpretar el estado del sistema sobre acciones no realizadas por él mismo, pero, sin embargo, necesita de éstas para predecir comportamientos futuros. Otras tareas pueden ser analizadas en nuestro modelo, pero por razones de espacio no se desarrollan. Veamos como se pueden formalizar.

4 Formalización de propiedades y su aproximación al diseño

Propiedades como las anteriores pueden ser analizadas formalmente utilizando nuestro modelo. La formalización permite introducir requerimientos del usuario en la etapa de especificación del software. Analicemos brevemente la propiedad visibilidad a partir de la siguiente definición.

Definición 1. Dado un sistema cooperativo $S = \langle U, m, C, \tilde{O}, I, E \rangle$, donde U es el conjunto de usuarios del sistema, m asocia a cada usuario el conjunto de comandos que puede realizar, C es el conjunto de comandos, \tilde{O} es el conjunto de secuencias de entradas de usuario (formadas por los comandos y los usuarios que han realizado cada comando), I es la función interpretación que asocia cada entrada del usuario con el nuevo efecto del sistema y E es el conjunto de efectos del sistema (los cuales permiten definir los distintos estados del sistema). El usuario $u_i \in U$ ve cualquier acción realizada por él mismo en su pantalla (su retroalimentación) sii:

$$display_i(I(p \circ (c_m, u_i))) \neq display_i(I(p)), \quad p \in \tilde{O}, \quad c_m \in C / c_m \in m(u_i)$$

siendo la función *display* aquella que asocia el efecto actual del sistema con los elementos visibles de ese efecto en la pantalla del usuario. Sin embargo, si las acciones son realizadas formando parte de tareas cooperativas, unos usuarios podrían ver las acciones de otros. Esto puede ser definido de la forma siguiente:

Definición 2. Dado un sistema cooperativo $S = \langle U, m, C, \tilde{O}, I, E \rangle$, el usuario $u_i \in U$ puede percibir las acciones de $u_k \in U$ si cualquier comando de entrada realizado por el usuario- k , puede ser visto en la pantalla del usuario- i . Esto es:

$$display_i(I(p \circ (c_m, u_k))) \neq display_i(I(p)), \quad p \in \tilde{O}, \quad c_m \in C / c_m \in m(u_k)$$

Como se ha comentado anteriormente, una de las ventajas del modelo abstracto Extended-PIE es que permite expresar una gran variedad de propiedades de sistemas cooperativos. Estas propiedades abstractas pueden tener varias connotaciones en el proceso de diseño. Por ejemplo, la propiedad visibilidad, cuando se realiza una tarea cooperativa, tendrá varias implicaciones en el diseño. Por un lado, afectará al proceso de interacción. Esto es debido a que la entrada de un comando por parte de un usuario generará una retroalimentación en la pantalla de ese usuario y del resto de usuarios que forman parte de la tarea. Por otro lado, dado que la acción realizada forma parte de una tarea cooperativa, aparece un componente de coordinación entre las acciones de los distintos usuarios. También aparece un aspecto funcional relacionado con los privilegios que cada usuario tiene según su rol en el sistema. Resumiendo, con este ejemplo podemos apreciar que estas propiedades no son inmediatas de “traducir” a una propuesta de diseño.

5 Conclusiones

A menudo, los factores humanos se han estudiado “a posteriori” en los sistemas mediante una evaluación de los mismos. El uso de técnicas formales junto con el alto nivel de abstracción permiten analizar propiedades deseables de un sistema cooperativo en una etapa temprana y de forma adecuada. Este análisis puede ser realizado por un grupo de trabajo interdisciplinar que integre a psicólogos, sociólogos e informáticos. A partir de este análisis podremos obtener los requerimientos que formarán el punto de partida en el diseño de estos sistemas.

Referencias

- [1] J.M. Carroll et al. “*Mental Models in human-computer interaction*” Handbook of human-computer interaction. 1987
- [2] A.J. Dix, *Formal methods for Interactive Systems*. Academic Press, 1991
- [3] G.D. Abowd et al. “*Giving undo attention*”. Int. with Computers. 4:3. 1992
- [4] H. Thimbleby. “*User Interface Design*”. ACM Press. 1990.
- [5] M. Gea et al. “*Modelisation of Co-operative Work*”. Proc. of DSVIS'99. 1999
- [6] B. Shneiderman, “*Designing the user interface, 2ª ed.*”. Addison Wesley. 1992
- [7] J. Grudin, “*Groupware and cooperative work: problem and prospect*”. The art of human Computer Interface Design, Addison Wesley, 1990.

- [8] S. Card et al. *"The psychology of Human-Computer Interaction"*.Lawrence Erlbaum. 1983

Sesión II

Diseño de Sistemas Interactivos

Modelización y diseño interactivo de interfaces con estructura dinámica

P. Castells, F. Saiz, R. Moriyón, F. García

E.T.S. de Informática, Universidad Autónoma de Madrid

Ctra. de Colmenar Viejo km. 17, 28049 Madrid

{pablo.castells | roberto.moriyon | francisco.saiz | federico.garcia}@ii.uam.es

Resumen

El actual panorama tecnológico demanda aplicaciones interactivas cada vez más flexibles y complejas, capaces de responder a una creciente dispersión y heterogeneidad de la información, de las plataformas y de los usuarios. En esta ponencia se describe el desarrollo de nuevas herramientas de diseño y soporte de interfaces de usuario capaces de definir y ajustar su propia estructura en función de diferencias en la información, los usuarios u otros factores, que no se conocen hasta el momento de la ejecución del software. Nuestro trabajo incluye el desarrollo de dos sistemas de modelización y soporte de presentaciones y tareas de usuario, una herramienta de autor interactiva para presentaciones avanzadas, así como la aplicación de las técnicas desarrolladas a la construcción de tutores interactivos.

Palabras clave: interfaces de usuario, diseño de sistemas interactivos, herramientas de diseño, interfaces basadas en modelos, programación mediante ejemplos.

1 Introducción

La rápida expansión de las tecnologías de la información plantea nuevas demandas para el software que ha de mediar entre las aplicaciones y las personas. Las nuevas aplicaciones tienen que manejar una información sujeta a variaciones, dispersa y heterogénea, y responder a la creciente diversificación de usuarios y plataformas. A nivel de las interfaces de usuario, esto significa que en muchos casos no es posible concebir de antemano soluciones satisfactorias en términos de una presentación gráfica fija y una estructura predeterminada de diálogo con el usuario.

Las herramientas de desarrollo de interfaces de usuario de uso común hoy en día no dan un soporte adecuado para este tipo de necesidades, por lo que o bien se producen soluciones obsoletas, o bien una gran parte del desarrollo se lleva a cabo con lenguajes programación de propósito general y frameworks de bajo nivel, dando lugar a un desarrollo difícil, costoso, y propenso a errores. La creación de herramientas de alto nivel para el desarrollo de interfaces que soporten expresamente los aspectos dinámicos es un problema difícil en general, por lo que sigue siendo un campo activo de investigación.

En este artículo se describe el trabajo realizado por nuestro grupo en la línea del desarrollo de nuevas herramientas de diseño y soporte de interfaces que, en uno u otro aspecto, son capaces de definir, ajustar o ampliar su propia estructura y funcionalidad en función de diferencias en la información, los usuarios, sus acciones, el estado del programa u otros factores que se concretan y pueden cambiar en el momento de la ejecución del software. Nuestro trabajo en esta dirección se ha centrado en los siguientes aspectos (ver figura 1):

- Un sistema de presentación, para la modelización y generación de la componente visual de las interfaces, que soporta el flujo dinámico de datos, así como la construcción de presentaciones iterativas, recursivas y condicionales.
- Una herramienta de autor complementaria al sistema de presentación, para el diseño interactivo de modelos de la presentación mediante manipulación directa y programación basada en ejemplos.
- Un sistema de representación de tareas de usuario para una descripción de alto nivel de la funcionalidad de las interfaces, que permite el análisis de la actividad del usuario en tiempo de ejecución.
- Una herramienta basada en el modelo anterior, para la generación de tutoriales de aplicaciones arbitrarias a partir de la modelización de tareas.

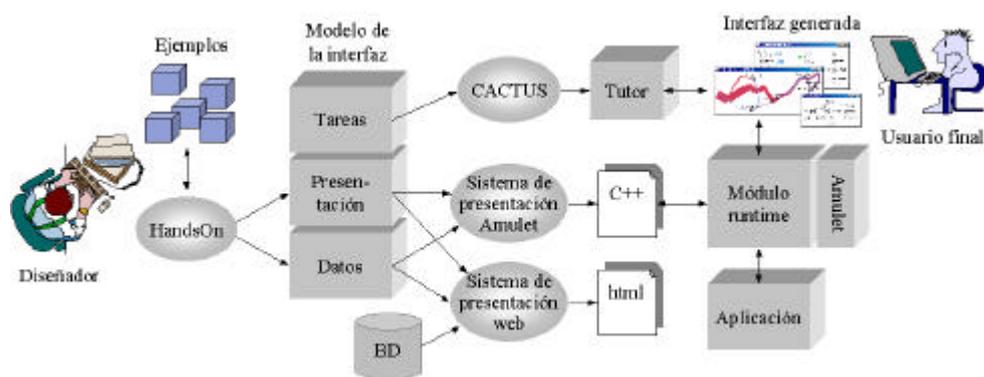


Figura 1. Componentes para la construcción de interfaces dinámicas

2 Interfaces basadas en modelos

Una interfaz con capacidad para ajustarse automáticamente a un entorno cambiante requiere una representación adecuada de sus propias componentes, tanto para razonar sobre ellas como para construirlas o modificarlas. Frente a la realización de las interfaces como una suma de componentes estándar y eventos de bajo nivel, a partir de la cual las características del diseño resultan difícilmente reconocibles, el llamado *paradigma basado en modelos* [1] propone la modelización las interfaces mediante descripciones declarativas (modelos) que incluyan explícitamente información semántica de alto nivel sobre el diseño, y relativa, en particular, a los aspectos que han de ser objeto de actualización y construcción dinámica.

Siguiendo esta filosofía, hemos desarrollado un sistema de presentación [2], a partir de los sistemas de presentación de Humanoid [3] y Mastermind [4], para la modelización, generación y ejecución de la componente visual (presentación) de las interfaces. En nuestro sistema, el diseñador construye modelos de las interfaces en un lenguaje textual de alto nivel, a partir del cual el sistema genera código C++ basado en Amulet [5], una toolkit que incorpora un sistema de programación con restricciones sobre campos de objetos. El sistema de presentación incluye así mismo un módulo propio de soporte a la ejecución para las interfaces generadas.

Nuestro modelo de la presentación permite la especificación de interfaces en las que el número, aspecto, y tipo de las componentes varían en función del espacio disponible, las capacidades del hardware, o las características de los datos que afectan a la interfaz. Concretamente, permite establecer dependencias en las propiedades de los objetos gráficos respecto de cualquier dato de la aplicación, así como la definición de componentes basadas en iteraciones, recursión y condicionales, de tal forma que estas estructuras de control se recomputan automáticamente cuando cambian los datos de los que dependen. Todo ello permite reflejar en el diseño datos de distinto tipo con un alto grado de libertad. Con este sistema es posible diseñar, por ejemplo, un plano dinámico del metro que refleje la posición de los trenes y otro tipo de información en tiempo real, mostrando mayor o menor cantidad de información y detalle en función de la resolución de la pantalla del dispositivo desde el que se está operando.

Así mismo hemos desarrollado una versión del sistema de presentación para la visualización de datos relacionales a través de la web [6], con capacidades similares a las descritas, añadiendo además la posibilidad de adaptar la presentación de la información a las preferencias y perfil de los usuarios. Este sistema genera páginas HTML dinámicamente a partir de a) los datos almacenados en una base de datos, b) un metamodelo de datos que

permite extender el modelo relacional con nuevas propiedades y descripciones asociadas a las estructuras de datos, c) una descripción de la presentación, mediante una extensión del lenguaje HTML que permite expresar distintos tipos de selección y procesamiento de los datos, d) un perfil de usuario en función del cual se personaliza automáticamente la presentación de los datos y e) un conjunto de reglas de estilo para definir presentaciones condicionales. El sistema permite establecer, por ejemplo, que los valores de un campo de texto ocupen una columna de una tabla, o por el contrario se muestren como una lista HTML, dependiendo de la extensión del texto visualizado, utilizando al mismo tiempo distintos colores de fondo para destacar o agrupar visualmente los datos que cumplen ciertas propiedades o relaciones.

3 Modelización de tareas

Junto con el sistema de modelización de la presentación, hemos desarrollado un sistema para la modelización del diálogo con el usuario, basado en una descripción de las tareas que el usuario final puede llevar a cabo con la interfaz. Este modelo es similar al que utilizan otras herramientas como Mastermind [4] y MOBI-D [1], y consiste en una descomposición jerárquica que pone en relación las tareas abstractas del dominio de aplicación (emitir una factura, dar de alta a un paciente, planificar calendarios laborales), con subtareas y acciones elementales de interacción básica propias de la interfaz (pulsar botones, escribir texto, arrastrar iconos).

En nuestro sistema, el modelo de tareas no se utiliza para generar las interfaces. La capacidad expresiva del modelo funcional descendente, propio de las tareas, resulta limitada para describir por completo la funcionalidad de una interfaz, definida mediante programación basada en eventos, por poco compleja que sea. El modelo de tareas desempeña en nuestro sistema un papel descriptivo, con vistas a su utilización por otras herramientas que generan servicios añadidos en tiempo de ejecución, y que requieren un análisis del diálogo que lleva a cabo el usuario con la interfaz.

A partir de este modelo, hemos desarrollado una herramienta, CACTUS [7], para el diseño de tutores de aplicaciones, que basan la enseñanza en a) unos contenidos explicativos transmitidos en forma de hipertexto mediante una metáfora de libro de texto, y b) la instrucción práctica utilizando la propia interfaz de la aplicación, bajo el control del programa tutor. En el segundo caso, el tutor utiliza el modelo de tareas para el seguimiento automático de las actividades de los usuarios, permitiendo proporcionar feedback a los usuarios y evaluar sus conocimientos de manera continua. CACTUS incorpora además la capacidad de emular, esto es, ejecutar por sí mismo, tareas abstractas de alto nivel de la interfaz sin la intervención del usuario.

4 Herramientas de autor

Los sistemas descritos en los apartados anteriores permiten la construcción de un amplio rango de interfaces, presentaciones dinámicas y servicios en tiempo de ejecución, utilizando un conjunto relativamente pequeño de primitivas de alto nivel. No obstante, la utilización de estas herramientas sigue requiriendo un esfuerzo importante por parte del diseñador, ya que éste tiene que aprender un lenguaje de especificación, comprender el funcionamiento del sistema soporte y la forma en que se generan las interfaces en tiempo de ejecución.

Por este motivo, hemos complementado el sistema de presentación descrito en el apartado 2 con una herramienta de autor interactiva, HandsOn [8], más intuitiva y ergonómica, de una facilidad de uso comparable a la de los constructores de interfaces como los que integran Visual Basic o Delphi. Con HandsOn el diseñador trabaja en un entorno gráfico similar al de estas herramientas. La salida de HandsOn es un modelo de la interfaz que sirve de input para el sistema de presentación.

En general las herramientas de autor son más fáciles de usar cuanto menor es la diferencia entre la representación manipulada por el diseñador y el resultado final. La dificultad estriba, en nuestro caso, en que la interfaz como tal no existe en tiempo de diseño, puesto que depende de condiciones que no se conocen hasta el momento de la ejecución. Para resolver este problema HandsOn, utilizando técnicas de *programación mediante ejemplos* [9], permite que el diseñador construya, en lugar de presentaciones genéricas y abstractas, ejemplos concretos de la interfaz utilizando ejemplos concretos de los datos, creados por él mismo en el propio entorno de edición. La herramienta examina las acciones del diseñador, analiza los ejemplos creados, infiere la intención del autor, y generaliza los ejemplos, generando un modelo abstracto listo para ser procesado por el sistema de presentación, que genera y gestiona la interfaz ejecutable.

Las técnicas de programación mediante ejemplos han sido aplicadas también para la construcción de modelos de tareas. En concreto, hemos completado el sistema CACTUS, descrito en el apartado anterior, con un entorno de edición para los diseñadores, en el que tanto los libros de texto como las prácticas se pueden diseñar bien con un lenguaje textual, o bien de forma interactiva mediante técnicas de programación mediante ejemplos. Este entorno se ha utilizado para la enseñanza de sistemas de simulación digital continua, como el sistema solar o las ecuaciones de Volterra. También se ha usado para generar cursos tutores para un entorno de diseño orientado a objetos asistido por ordenador. Además, se han hecho pruebas con una agenda electrónica y, finalmente, hemos generado cursos acerca de *Schoodule*, una aplicación que utiliza una base de datos y un resolvidor de restricciones para planificar horarios de centros docentes.

Nuestros planes de trabajo para el futuro próximo incluyen la ampliación de estas ideas para manejar en este tipo de entornos una representación gráfica de un modelo del usuario. Actualmente estamos desarrollando una herramienta de autor para la construcción de

cursos adaptativos, que integra en un mismo entorno un modelo de tareas que describe las actividades de aprendizaje del curso, un modelo de la presentación de los cursos, y un modelo del usuario, de tal forma que el diseñador, mediante la manipulación gráfica de estos tres modelos, puede establecer dependencias entre la interfaz del curso y las características de los estudiantes.

Referencias

- [1] A. R. Puerta, "A Model-Based Interface Development Environment", IEEE Software, v. 14, n° 4, 1997, pp. 40-47.
- [2] P. Castells, P. Szekely and E. Salcher, "Declarative Models of Presentation", proc. International Conference on Intelligent User Interfaces (IUI'97), Orlando (Florida), 1997, pp. 137-144.
- [3] P. Szekely, P. Luo, and R. Neches, "Beyond Interface Builders: Model-Based Interface Tools", proc. Conference on Human Factors in Computing Systems (INTERCHI'93), Amsterdam, 1993, pp. 383-390.
- [4] P. Szekely, P. Sukaviriya, P. Castells, J. Muthukumarasamy and E. Salcher, "Declarative Interface Models for User Interface Construction: The Mastermind Approach", en "Engineering for Human-Computer Interaction", L. Bass and C. Unger (eds), Chapman & Hall, 1996, pp. 120-150.
- [5] B. A. Myers et al, "The Amulet Environment: New Models for Effective User Interface Software Development", IEEE Transactions on Software Engineering, v. 23, n° 6, 1997, pp. 347-365.
- [6] F. Saiz, P. Szekely, D. Devang, "Customized Web-based Data Presentation", proc. World Conference on the WWW, Internet and Intranet (WebNet'98), Orlando (Florida), 1998, pp. 792-798.
- [7] F. García, "CACTUS: Automated Tutorial Course Generation for Software Applications", proc. International Conference on Intelligent User Interfaces (IUI'2000), New Orleans (Louisiana), 2000.
- [8] P. Castells and P. Szekely, "Presentation Models by Example", en "Design, Specification and Verification of Interactive Systems '99", D.J. Duke and A. Puerta (eds), Springer-Verlag, Viena, 1999, pp. 100-116.

- [9] A. Cypher (ed.), “Watch What I Do: Programming by Demonstration”, the MIT Press, 1993.

Aproximación Basada en Modelos para la Especificación de Interfaces de Usuario

María Dolores Lozano[†], Pascual González[†], Isidro Ramos[‡]

[†]Departamento de Informática
Escuela Politécnica Superior de Albacete
Universidad de Castilla-La Mancha 02071-ALBACETE
Tlf: 967.59.92.00 Fax: 967.59.92.24 [mlozano, pgonzalez]@info-ab.uclm.es

[‡]Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n 46020-VALENCIA
Tlf: 96.387.93.59 Fax: 96.387.73.59. iramos@dsic.upv.es

Resumen

La importancia del diseño y creación de interfaces de usuario es cada vez mayor, siendo considerado como un campo de investigación de gran relevancia y tal vez uno de los que producen un mayor impacto en los diferentes usuarios de los sistemas de información. Podemos considerar varias disciplinas dentro de esta área de investigación, que pasan desde la definición de métodos de análisis y diseño de interfaces de usuario, la creación de nuevas alternativas de representar o acceder a la información, hasta el estudio de los factores humanos que están detrás del manejo de dichas interfaces. En este trabajo vamos a realizar una revisión de las investigaciones más interesantes realizadas hasta el momento en el ámbito de los métodos de definición de interfaces de usuario, basados en la utilización de diferentes modelos semánticos. Indicando finalmente nuestra aportación en este terreno, un entorno de desarrollo de interfaces de usuario dentro del proceso de desarrollo de OO-Method y siguiendo la filosofía de OASIS.

Palabras clave: Interfaces de Usuario, Diseño de Sistemas Interactivos.

1 Introducción.

Existen varios motivos que justifican la complejidad del diseño e implementación de los IU [1], Pero tal vez la principal es la dificultad de comprender las tareas que el usuario desea realizar sobre el sistema y las características de los diferentes usuarios que lo manejarán [2]. En este sentido parece necesario definir nuevos modelos que permitan recoger los requerimientos y especificar las diferentes características de la interfaz asociada a una determinada aplicación. Estos modelos deben permitir que se defina una especificación de la interfaz independiente [3]. Es decir, los modelos utilizados para especificar la interfaz deben ser distintos de los utilizados para especificar la aplicación.

Diferentes estudios han demostrado que el 48% del código de una aplicación se dedica a la interfaz de usuario, y que cerca del 50% del tiempo dedicado a implementación es para implementar la parte de Interfaz de Usuario [4]. A medida que la interfaz de usuario es más fácil de usar, implica que es más difícil de construir. Los desarrolladores de IUs necesitan herramientas que proporcionen un buen soporte para el desarrollo de IUs avanzadas.

En este trabajo, se realiza una presentación de los principales estudios sobre definición y utilización de modelos en la especificación de interfaces de usuario. Después de dicha presentación, se indican las características básicas de los nuevos entornos de desarrollo de interfaces de usuario basados en modelos. Y finalmente se expone el modelo de interfaz de usuario aportado en el contexto de OO-Method y como enriquecimiento del lenguaje de especificación OASIS, así como la ontología asociada a este modelo.

2 Entornos de desarrollo de interfaces de usuario basada en modelos. Estado del arte.

En este apartado se analizan los diferentes usos de modelos declarativos como parte del Modelo de Interfaz en un entorno de desarrollo de interfaces de usuario basadas en modelos, de ahora en adelante, EDIBAM, para abreviar.

Durante los últimos años se han creado distintas herramientas para soportar el desarrollo de IUs, por ejemplo, Toolkits, User Interface Management Systems, Interface Builders, UIDE, etc. En un report sobre el estado del arte, B. Myers presenta una clasificación de estas herramientas software de Interfaz de Usuario [5]. Esta clasificación se basa en la manera en la que los desarrolladores de IUs pueden especificar la forma y el comportamiento dinámico de una IU. La clasificación es la siguiente:

- *Herramientas basadas en lenguajes*: Estas requieren que el desarrollador programe en un lenguaje de propósito especial.
- *Herramientas de especificación gráfica interactiva*: Estas permiten un diseño interactivo de la IU.
- *Herramientas de generación basadas en modelos*: Estas hacen uso de un modelo o especificación de alto nivel para generar la IU de forma automática.

De estos tres tipos, las últimas parecen ser las más adecuadas para abordar el desarrollo de la IU. Las dos primeras permiten la especificación tanto del comportamiento dinámico como de la estructura de la IU de forma sencilla, pero no ambas partes a la vez. Y al igual que la mayoría de las herramientas actuales soportan solamente la fase final de desarrollo del ciclo de vida de la IU, y las abstracciones que proporcionan no tienen una conexión directa con los resultados del análisis de tareas.

2.1 Arquitectura genérica de los EDIBAMs

Los componentes típicos y el procedimiento de desarrollo principal de un EDIBAM se muestran en la Fig. 1

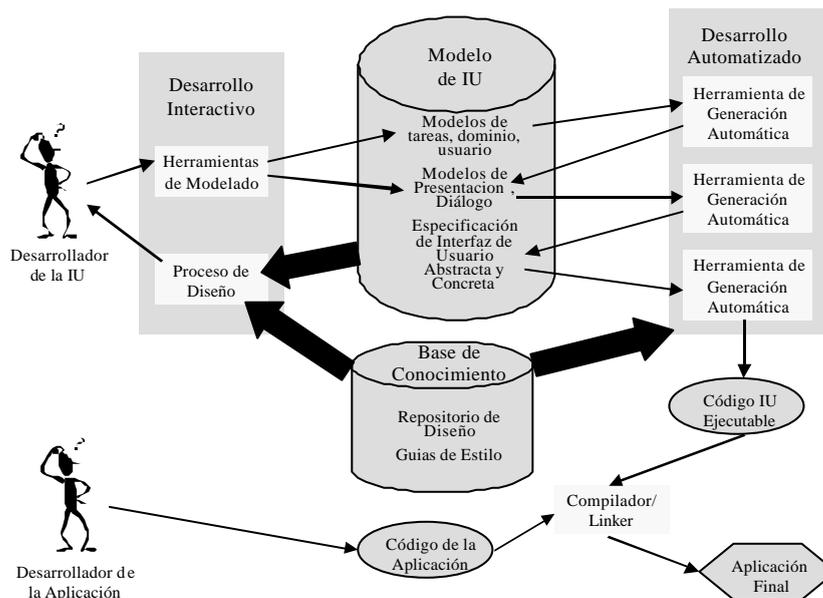


Figura 1: Arquitectura Genérica de un Entorno de Desarrollo de Interfaces de Usuario Basado en Modelos.

El componente central del EDIBAM es el Modelo de Interfaz que incluye diferentes modelos declarativos. En el siguiente apartado se detallarán como son estos modelos declarativos. En este entorno también intervienen herramientas para el desarrollo interactivo (Herramientas de Modelado y de Diseño) y para el desarrollo automático. Las herramientas para la generación automática se encargan de las transformaciones entre los diferentes modelos declarativos y la representación implementada y ejecutable de la IU deseada. La información requerida por las herramientas de generación automática y de diseño está representada en la Base de Conocimiento.

3 IDEAS: Desarrollo de Interfaces de Usuario en un Entorno de Producción Automática de Software Orientado a Objetos.

El objetivo de este trabajo es la integración de un Modelo de Interfaz de Usuario dentro de OASIS a partir del cual se pueda generar la interfaz de la aplicación de forma automática. El proceso de especificación de la Interfaz de Usuario se realiza de forma paralela al desarrollo de lo que es la aplicación en sí, y de acuerdo a los principios de los EDIBAMs estudiados previamente. Este proceso de desarrollo se puede ver en la siguiente figura:

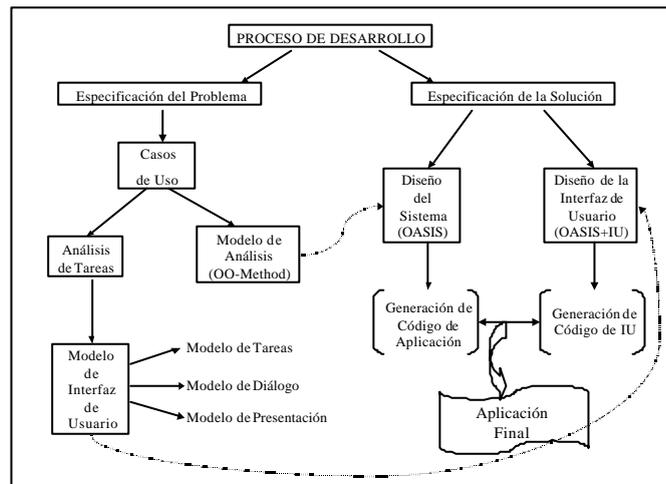


Figura 2: Proceso de Desarrollo de Sistemas en el Entorno de Producción Automática de Software de OASIS enriquecido con el Modelo de IU

De esta forma IDEAS (**I**nterface **D**evelopment **E**nvironment within **O**ASIS), pretende ser un sistema de desarrollo automático de Interfaces de Usuario integrado en el marco de trabajo de OASIS [6] y OO-Method [7], [8], para soportar de forma automática la producción de IUs de alta calidad. Esto es posible porque está basada en modelos declarativos, con las ventajas ya comentadas de estas aproximaciones. Los modelos declarativos utilizados en IDEAS son tres: el modelo de tareas, el modelo de diálogo y el modelo de presentación.

El **modelo de tareas** define el conjunto ordenado de actividades y acciones que tiene que llevar a cabo el usuario para conseguir un objetivo concreto. El **modelo de diálogo**[9], que se utiliza para describir la "conversación" hombre-sistema. Describe cuando el usuario puede invocar comandos, seleccionar o especificar entradas de información y cuando el computador puede requerir información del usuario y presentar la información de salida. Este modelo incluye la generación de tres tipos de diagramas que representan estos aspectos, el Diagrama de Estructura de Diálogos (DED), el Diagrama de Especificación de Componentes (DEC) y la Tabla de Definición de Comportamiento de Componentes (TDC), [10].

El **modelo de presentación**, que describe las componentes que pueden aparecer en la pantalla del usuario final, sus características de diseño y las dependencias visuales existentes entre ellas. Este modelo está estrechamente relacionado con el modelo de análisis del sistema, ya que la información que se muestra en este modelo es aquella descrita en el modelo de análisis, es decir, los atributos característicos de las clases identificadas en el sistema serán accesibles al usuario a través del modelo de presentación.

A partir de los modelos declarativos que componen el modelo de interfaz de usuario, se genera automáticamente el código de IU, que junto con el código de la aplicación generado en el proceso de desarrollo del sistema paralelo formará la aplicación resultante final. Como paso intermedio a esto, en la fase de diseño la especificación de la IU se lleva a cabo en el lenguaje de especificación OASIS enriquecido para especificar la parte de interfaz de usuario, de la misma forma que se viene utilizando para la especificación del sistema.

4 Conclusiones

Con el objetivo de conseguir un entorno de producción automática de software orientado a objetos, que no sólo genere sistemas software de calidad, sino que además permita ser utilizado con la sencillez y eficacia que requiere el usuario a través de una interfaz de usuario amigable pero a la vez robusta, se ha presentado IDEAS, un entorno de desarrollo de interfaces de usuario con la filosofía de OASIS e integrado en el proceso de desarrollo de OO-Method.

IDEAS, al estar basada en modelos declarativos, proporciona información descriptiva de alto nivel de abstracción de los futuros componentes de la interfaz de usuario, a la vez que

permite estudiar de forma precisa cómo se llevará a cabo la interacción del usuario con el sistema en las primeras etapas de desarrollo. Y en las fases finales, permite la generación automática de código a partir de los modelos. Destacar por último la importancia que va cobrando cada vez más, sobre todo en los sistemas interactivos, la interfaz de usuario como medio para interactuar y sacar el máximo partido de un sistema. Esta es la principal motivación de la investigación que se está llevando a cabo.

Referencias

- [1] B.A. Myers: *Why are Human-Computer Interfaces Difficult to Design and Implement*. Tech. Report CMU-CS-93-183, CMU, Jul 1993.
- [2] J. Gould, C. Lewis: *Designing for Usability – Key Principles and What Designers Think*. Commun. ACM, Vol 28, 1985, pp.300-311.
- [3] W. Swartout, R. Blazaer: *The Inevitable Intertwining of Specification and Implementation*. Commun ACM, Vol 25, No. 7, Jul 1982, pp.438-440.
- [4] B. A. Myers, M. B. Rosson: Survey on User Interface Programming. In Striking a Balance. Proceedings CHI'92. Monterey, May 1992, New York: ACM Press, 1992, 195-202
- [5] B. A. Myers: User Interface Software Tools. ACM Transactions on Computer-Human Interaction 2 (1995), 1, 64-103
- [6] P. Letelier, I. Ramos, P. Sanchez, O. Pastor. *OASIS versión 3.0: A Formal Approach for Object Oriented Conceptual Modelling*. SPUPV-98.4011. Edited by the Technical University of Valencia, Spain, 1998.
- [7] O. Pastor, E. Insfran, V. Pelechano, J. Romero, J. Meseguer. *OO-Method: An OO Software Production Environment Combining Conventional and Formal Methods*. Procs. of 9th. International Conference, CAiSE'97. Barcelona, Catalonia, Spain, June 16-20, 1997.
- [8] M.D. Lozano, I. Ramos, P. Cuenca. "Application of an Object-Oriented Methodology for Automatic Software Production: OO-Method". Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'97). Pags. 985 – 989. Victoria, Canadá. 20-22 de Agosto, 1997.
- [9] M.Lozano, I. Ramos: An Object Oriented Approach for the Specification of User Interfaces. PACRIM'99, Victoria -Canadá- Agosto, 1999.
- [10] M. Lozano, I. Ramos: Integration of the User Model and Human Factors in an Object Oriented Software Production Environment. ECOOP'99, Lisboa -Portugal-, Junio, 1999.

Modelo de Construcción de Sistemas Interactivos basado en técnicas Formales.

F.L. Gutiérrez, M. Cabrera, J.C. Torres, M.Gea.

Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Granada, Av. Andalucía 38, Granada
e-mail (fgutierr,mcabrera,jctorres,mgea)@ugr.es

Resumen

En este artículo se presenta un modelo de construcción de sistemas interactivos basado en la utilización de un lenguaje de especificación formal y una herramienta de prototipado que juntos facilitan la creación tanto de la parte del sistema que interactúa con el usuario como de los elementos propios de la aplicación. Este modelo se basa en la posibilidad de estudiar propiedades importantes del software a partir de la especificación y de un prototipo generado de forma automática a partir de la propia especificación. La herramienta de prototipado usada permite añadir a los elementos de la aplicación, que forman parte de la interfaz de usuario, características propias de este tipo de sistemas. (Diseño de su apariencia gráfica, Comunicación con el usuario, interacción con otros elementos, ...)

Palabras clave: Técnicas formales, herramientas, prototipado, Técnicas de evaluación de sistemas Interactivos.

1 Introducción

El coste de construcción de las aplicaciones interactivas se ve elevado considerablemente debido a los problemas que genera la especificación y el diseño de la comunicación entre el usuario y la aplicación.

Se puede reducir este coste partiendo de especificaciones lo suficientemente robustas como para que el resto del desarrollo se realice sobre una base firme. La generación de prototipos facilita en gran medida la obtención de este tipo de especificaciones.

Se han construido numerosas herramientas para que el diseñador de interfaces de usuario pueda generar de forma rápida prototipos del sistema [1]. Desafortunadamente es difícil comprobar que los prototipos generados con estos sistemas se comportan como se desea, en algunos casos el poder probar propiedades importantes del sistema sobre el propio

prototipo permite asegurar aspectos como la usabilidad del software o la validación de la especificación de partida. En este artículo proponemos un modelo de construcción de software interactivo que pueda solucionar el problema planteado.

Nuestro modelo se basa en la construcción de software a partir de la especificación formal de este, sobre la que podemos realizar verificaciones de propiedades del software, y en la inclusión de una herramienta de prototipado que automatice el proceso de creación de los prototipos partiendo de la especificación inicial del mismo.

La elección de un lenguaje formal como herramienta de especificación permite utilizar el marco matemático sobre el que se construye el lenguaje como herramienta para la verificación de propiedades. Hemos elegido un lenguaje de especificación Algebraico, que posibilita la generación de especificaciones con un alto grado de abstracción, evitando en gran medida la inclusión en etapas iniciales de construcción del software de elementos mas cercanos al diseño que a la propia especificación.

2 Modelo del software

Para la descripción del software vamos a partir de un modelo orientado a los componentes, donde el software se va a componer de un conjunto de elementos independientes que interactúan entre sí para la realización de las acciones del sistema. Dentro de este conjunto de elementos se incluyen los elementos que van a interactuar tanto con la aplicación como con el usuario.

El modelo que utilizamos para la descripción de los sistemas es similar a los modelos usados para representar la arquitectura de los sistemas reactivos, en nuestro caso los componentes van a ser elementos que proveen una serie de funciones al resto del sistema, y la comunicación entre ellos, en vez de realizarse mediante canales de comunicación, se realiza por medio de llamadas a las distintas funciones que puede realizar cada elemento.

Para modelizar sistemas interactivos vamos a utilizar el concepto de Objetos Activos (OA) como una mezcla de las nociones de objeto y proceso. Un OA podría verse como un objeto que oferta una serie de funciones a su entorno y que puede realizar algunas de ellas sin la intervención del resto de los objetos del sistema.

Basándonos en el concepto de OA se podría describir un sistema como “un conjunto de elementos autónomos que pueden realizar acciones, comunicarse unos con otros por medio de mensajes y sincronizar las distintas acciones que realizan”.

Los OAs poseen una serie de características que es importante describir y que les hacen diferentes del resto de los objetos:

- Un OA permite recibir, de forma simultanea, varias llamadas a sus funciones lo que puede generar mas de una actividad interna (Concurrencia IntraObjeto)
- Pueden ser activados por la recepción de una llamada a una de sus funciones (Objeto activo reactivo), o por alguna acción interna que realice el objeto (Objeto activo autonomo).
- Los distintos mensajes que un OA puede aceptar varían en el tiempo en función de distintas condiciones.
- Los OA pueden cooperar para realizar alguna acción mediante la sincronización de sus actividades.

Cuando modelizamos un sistema que interactua con el usuario aparece un tipo de objeto que por sus características especiales puede ser destacado del resto, nos referimos a los objetos que van a representar el dialogo entre el usuario y el software.

En este tipo de Objetos es importante que el estado del objeto pueda ser consultado por otros objetos del sistema o por el usuario. Esto permite que el usuario, en cada momento obtenga una realimentación de las distintas actividades que se realizan en el sistema.

La realimentación del estado de los objetos, ya sea hacia el usuario o hacia otro objeto, debe poder ser controlada por el propio objeto en función de los cambios de estado o de las funciones que se estén aplicando.

La representación del estado de un objeto puede ser cualquier otro objeto, incluso en algunos casos sería interesante poder usar un objeto gráfico que vaya modificando su apariencia en función del estado del objeto.

En el caso de representar la parte del software que describe la interfaz de usuario se utilizarían objetos del tipo anterior a los que se les puede asignar representaciones gráficas en función de la apariencia que se desee. Sobre un modelo que describa el comportamiento del software se podría aplicar distintos diseños de la interfaz, sin perder la funcionalidad definida en la especificación.

Una de las ventajas de la utilización de un modelo de este tipo es la facilidad existente para trasladar el modelo a una implementación basada en objetos. Algunos modelos que recogen parcialmente esta estructura son el modelo arquitectónico de los Interadores [2], el modelo MVC (Model View Controller) [3] de Smalltalk o la arquitectura de sistemas interactivos PAC [4].

3 Lenguaje de Especificación

Para la especificación del modelo del software vamos a usar un lenguaje Algebraico, el cual permite la especificación del conjunto de objetos que modelan nuestro sistema a un nivel de abstracción alto, centrándonos sólo en una serie de propiedades importantes que deben satisfacer cada uno de ellos.

Hemos desarrollado un lenguaje de especificación, basado en técnicas algebraicas, llamado GRALPLA [5][6], al que se le han añadido elementos para facilitar la especificación de las características especiales que poseen los objetos de nuestro modelo. El lenguaje permite la definición de jerarquías de tipos de datos abstractos, la especificación de información gráfica, la especificación de procesos y la descripción de la sincronización entre ellos.

Se ha utilizado una semántica basada en la secuencia de acciones que se va realizando sobre cada uno de los objetos, lo que nos permite generar un prototipo a partir de una herramienta de prototipado, así como verificar propiedades [7] usando técnicas de inducción estructural sobre los axiomas de la especificación. La especificación de un sistema usando GRALPLA consiste en una colección de módulos donde cada uno especifica un tipo de dato. Cada una de las especificaciones esta formada por una interfaz que describe las características generales del tipo (otros tipos que usa, herencia, parametrización, ...) , el conjunto de funciones y el conjunto de condiciones (axiomas) que debe satisfacer y que van a ser utilizadas para la descripción del comportamiento abstracto de cada una de sus funciones.

Una de las características que se va a describir del tipo es si su naturaleza es gráfica, esto va a implicar la existencia de un serie de funciones que permitan obtener, en cada momento, la apariencia gráfica del objeto. Podemos tener tipos que posean una representación explícita del estado del objeto en cada momento usando para ello un conjunto de objetos. Esta representación puede ser de naturaleza gráfica.

Dentro de la especificación de las funciones que contendrá el tipo, permitimos la definición de unas funciones con unas características especiales:

- **Process:** describe una función que se va activar sin la intervención de ningún agente externo al objeto.
- **Asynchronous:** Se usa para describir funciones que realizan la recepción de llamadas de forma asíncrona.

Dentro del conjunto de axiomas vamos a definir una serie de ecuaciones que van a describir el comportamiento del tipo por medio de propiedades, que deben ser ciertas en todo

momento del estado de los objetos del tipo. Si el tipo posee representación debemos describir como va a evolucionar ésta en función del estado del objeto, para ello se utilizaran nuevos axiomas sobre sus funciones. Existe una ultima sección dedicada a la descripción de restricciones sobre el comportamiento de las funciones, esto permite indicar mediante expresiones guardadas cuando pueden realizarse cada una de ellas.

4 Herramientas de Prototipado

Junto con el lenguaje de especificación se han desarrollado varias herramientas que facilitan la creación de prototipos del sistema. Una primera herramienta permite generar un prototipo en C++ partiendo de una especificación en GRALPLA del modelo del sistema. Utilizando este prototipo podríamos realizar un estudio de la funcionalidad del sistema.

Así mismo vimos la necesidad de desarrollar una herramienta interactiva [8] que nos permitiera la definición del layout de la Interfaz, ya que a nivel de especificación no debemos entrar en este tipo de detalles más cercanos a la fase de diseño del sistema, sin embargo siempre es útil poder estudiar la aplicación de distintos diseños de la interfaz a los prototipos generados con la herramienta.

La figura 1 representa el proceso de generación del prototipo del sistema partiendo de la especificación. El proceso de generación del prototipo se subdivide en dos partes, por un lado la parte no interactiva de la aplicación se procesa con la herramienta de prototipado inicial, y se genera el código C++ correspondiente.

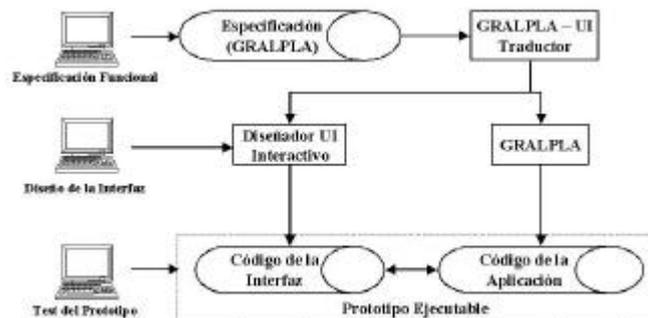


Figura 1. Esquema del proceso de generación del prototipo

Por otro lado, se ha modificado GRALPLA para que genere un fichero con los datos de necesarios para la generación de la parte interactiva de la aplicación. Este fichero sirve como entrada a la herramienta interactiva, generando una primera aproximación al diseño de la interfaz del sistema.

Una vez que se ha terminado la definición del layout según las preferencias del diseñador, es posible la generación del código del prototipo correspondiente a esta parte de la aplicación. El diseño que se propone no impone el uso de un lenguaje concreto, el único requisito que debe cumplir el lenguaje elegido es poder ser enlazado con el lenguaje C++, que es en el que se genera el código del resto de la aplicación. En la actualidad se está desarrollando una versión que genera lenguaje Java.

Referencias

- [1] B.A. Myers, "User Interface Software Tools". ACM Transactions on Computer Human Interaction. Vol 2, No 1, Pag 64-103.
- [2] F. Paterno, G. Faconti: "On the use of LOTOS to describe Graphical Interaction". HCI, 1992.
- [3] Goldberg: "Smalltalk-80-the interactive programming environment". Addison Wesley, 1984.
- [4] J. Coutaz: "PAC-ing the architecture of your interface" J.C. Torres (Ed.). EG Workshop on Design, Specification and Verification of Interactive System (DSVIS'97), Springer Verlag, Granada, 1997.
- [5] M. Gea, "Especificación formal de sistemas gráficos", Tesis Doctoral. Granada, 1997
- [6] J.C. Torres; B. Clares: "Using an Abstract Model for the Specification of Interactive Graphics Systems". F. Paterno (Ed.): Design, Specification and Verification of Interactive Graphic Systems (DSVIS'94). Springer Verlag, 1994.
- [7] Gutiérrez, F.L.; Gea, M.; Torres, J.C.: "Verification of Interactive Systems using Algebraic Specification". Proceedings 5th Eurographics Workshop on Design, Specification and Verification of Interactive Systems, DSV-IS'98. Abingdon, United Kingdom, 1998.
- [8] M. Cabrera, J.C. Torres, M. Gea, "Towards Users Interfaces Prototyping From Algebraic Specification", Proceedings 6th Eurographics Workshop on Design, Specification and Verification of Interactive Systems, DSV-IS'99. Braga, Portugal, 1999

```
Graphic Object Ventana;
// Especificacion parcial de una Ventana.
// -----
Import icono, punto, ventana_grafica, manejador_ventanas;

Ventana: punto, punto, str → Ventana;

Functions

Asynchronous E_raton_pulsado: Ventana, punto → Ventana;
Asynchronous E_raton_soltado: Ventana, punto → Ventana;
Asynchronous E_raton_arrastre: Ventana, punto → Ventana;
Mover: Ventana, punto → Ventana;
Escalar: Ventana, punto, punto → Ventana;
Minimizar: Ventana → Ventana;
Private Cambio_estado: Ventana → Bool;
Private pos<p1,p2>: Ventana → punto,punto;
Titulo: Ventana → str;

Process feedback: Ventana → graphic_objetc;
Graphic_rep: Ventana → graphic_object;

Axioms
.....
Cambio_estado (mover(v,p))=true;

graphic_rep (minimizar(v)) = icono(v);
graphic_rep(ventana(p1,p2,s)) = ventana_grafica (p1,p2,s);
graphic_rep (mover(v,p)) = ventana_grafica(pos_p1(v)+p, pos_p2(v)+p,titulo(v));

feedback (V) = draw (graphic_rep(V));

synchronization

Do feedback(V) when cambio_estado(V);
```

En esta especificación parcial de un objeto que representa una ventana dentro de un interface de usuario podemos observar algunas de las características que se le han añadido al lenguaje GRALPLA para permitir describir de forma sencilla las características importantes de un objeto interactivo.

Arquitectura para Agentes de Interfaz Inteligentes: el ordenador *sugerente*

J.R. Balsas, M.C. Díaz, A. Montejo, F. Martínez, M. García, L.A. Ureña
Departamento de Informática
Universidad de Jaén, Av. De Madrid 35, Jaén
e-mail: laruena@ujaen.es
tlf: 953 212 445 – fax: 953 212 420

Resumen

Desde los asistentes hasta los tutores interactivos podemos identificar una nueva forma de entender la interfaz de usuario. La interfaz inteligente se hace necesaria desde el momento en que la complejidad de las aplicaciones crece, así como la valiosa capacidad de reducir el tiempo de aprendizaje. Tras una clasificación de las soluciones comerciales más extendidas, identificamos las características deseables por parte de estos agentes inteligentes encargados de la interfaz con el usuario. Después proponemos una nueva arquitectura para el desarrollo de sistemas basadas en este tipo de interfaces.

Palabras clave: interfaz de usuario, agentes inteligentes, arquitectura, interacción persona-ordenador

1 Introducción

El uso de interfaces inteligentes para todo tipo de aplicaciones es una realidad creciente. Desde los asistentes hasta los tutores interactivos podemos identificar una nueva forma de entender la comunicación entre el usuario y la herramienta. La idea de un *diálogo interactivo* libera al usuario de la necesidad de conocer a fondo el uso de la herramienta en cuestión. La interfaz inteligente se hace necesaria desde el momento en que la complejidad de las aplicaciones crece, así como la valiosa capacidad de reducir el tiempo de aprendizaje.

Como se refleja en el trabajo de Brown y Santo [1], es difícil diseñar la interfaz debido a los diferentes sistemas informáticos, la carencia de estándares en cuanto a interfaces, la necesidad de que la interfaz se adapte a cierto estándar preestablecido y la falta de una metodología apoyada en una teoría sólida para la especificación de los requisitos de la interfaz. Entendemos, por tanto, que la mejor solución consiste en analizar aquellos sistemas que realmente han funcionado en el mundo comercial y, a partir de ellos, abstraer todo lo posible sus características para la forja de una nueva arquitectura.

2 Paradigma actual

Se hace necesario, primero, establecer una definición clara de qué consideramos como agente. Usaremos para ello la definición dada por Lieberman [2], quien nos dice que “*un agente es cualquier programa que puede ser considerado por el usuario como un asistente o un ayudante en su forma de actuar, en lugar de una mera herramienta al estilo de las interfaces convencionales de manipulación directa*”.

Si bien en un principio se hablaba de interacción entre la persona y el ordenador, debemos abstraer esta realidad como un modelo de interacción entre persona y sistema. Esto nos da una visión más amplia de lo que hoy en día ocurre en los entornos distribuidos, cooperativos y multitarea. Las interfaces de última generación vienen acompañadas de asistentes y otros elementos bajo los cuales se ocultan agentes de interfaz. Los denominados *agentes software* están presentes en casi todas las interfaces de las últimas versiones de muchos programas.

3 Clasificación

Atendiendo a las soluciones comerciales actuales, identificamos la clasificación de las interfaces inteligentes como sigue:

- **Basados en menús.** Desde hace algún tiempo, los menús han dejado de ser estáticos, para presentarse con capacidades tales como sensibilidad al contexto, personalización, crecimiento adaptable... (*Windows, MacOS, ...*)
- **Asistentes.** Este tipo usa la metáfora del “experto” que nos acompaña en nuestra sesión de trabajo (*Microsoft Office*). Con apariencia antropomorfa, el asistente nos sugiere formas de actuación y ayuda.
- **Basados en plantillas/ejemplos.** Son los determinados “wizards” (*WinZip, InstallShield, ...*), donde, paso a paso, vamos completando la información necesaria en la consecución de una determinada tarea. Existen muchas propuestas de este tipo, desde las plantillas sin más, hasta tutores completos.
- **Mixtos.** Es difícil poder encuadrar las propuestas actuales en una sola de estas categorías. Hoy día es palpable que los interfaces intentan dertar los tres tipos descritos anteriormente.

4 Características deseables

Como en todo proceso de ingeniería del software, pasamos a identificar los requisitos deseables de nuestro futuro sistema. Considerando contempladas las características exigidas a toda interfaz [3], vamos a centrarnos sobre estos requisitos: los deseables en los agentes y los deseables en las aplicaciones.

En cuanto a los agentes. De todas las propuestas estudiadas, se extraen las siguientes características principales:

- *Colaboración*: capacidad de los agentes para comunicarse entre sí en forma de llamadas-respuesta.
- *Autonomía*: capaces de realizar *ciertas* tareas de forma independiente, sin necesidad de comunicación con otros agentes y/o usuario/s.
- *Adaptable*: aprenderán de la evolución que la interacción experimente y serán transportables a diferentes entornos y contextos.

Existen más características, también contempladas. Para más información ver [4]. Las aquí incluidas nos parecen las más críticas para un agente de interfaz.

En cuanto a las aplicaciones. En un marco como el que se pretende implantar el sistema a proponer, se hace necesaria la determinación de ciertas características que las aplicaciones deberían contemplar para su total integración en nuestro esquema:

- *Examen*: Las aplicaciones deben facilitar la supervisión de la interacción.
- *Memorización*: Debe poder registrarse esta interacción (ver [5]).
- *Descriptible*: La comunicación con la aplicación debe ser posible a través de algún lenguaje o protocolo (no sólo mediante la interfaz que la aplicación misma posee); por ejemplo, *AppleScript* en entornos *Macintosh*, lenguaje *Visual Basic* en *Windows*... ([6])

5 Arquitectura

En este apartado vamos a presentar el diseño más simple de nuestra arquitectura, aquella en la que un único usuario se enfrenta a una sola aplicación. La idea de agentes de interfaz no es nueva, pero aunque se han implementado muchas soluciones, basadas sobre todo en prototipos “forzados” sobre sistemas previos [6], nuestra propuesta plantea un sistema nuevo, de principio a fin, que proporcione una solución definitiva a los problemas a los que se enfrentan los intentos seguidos hasta el momento y que, al mismo tiempo, satisfaga todos los requisitos identificados para un agente de interfaz inteligente. Básicamente consiste en una maduración del modelo multi-agente PAC de Joelle Coutaz [10]. Como

podemos ver en la *figura 1*, en nuestro modelo el usuario se comunica con la aplicación a través de un *agente de interfaz* [2].

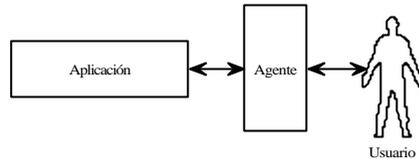


Figura 1. Esquema básico de un agente de interfaz

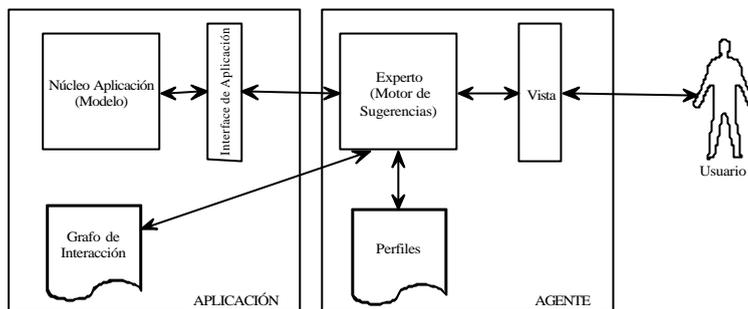


Figura 2. Esquema de un agente de interfaz monousuario y monoaplicación

La *figura 2* muestra con más detalle el modelo propuesto. En este esquema el usuario interactúa con una *vista* (similar a la *presentación* del modelo PAC y a la *vista* del modelo MVC (*Modelo-Vision-Control*)). Esta *vista* transfiere y recibe información de interacción con el núcleo del agente, que nosotros hemos denominado *motor de sugerencias*. Este motor registra el comportamiento del usuario en una base de datos de *perfiles* y propone una interfaz de comunicación a la *vista* en función de los datos del *perfil* y del *grafo de interacción*. Una vez determinada una comunicación, se notifica a la aplicación de la acción a realizar mediante una interfaz cuyo protocolo sea estándar.

Los *perfiles* son, básicamente, visiones parciales enriquecidas del grafo de interacción de la aplicación. El grafo de interacción es la estructura fundamental del sistema. Un grafo de interacción refleja todos posibles estados de una aplicación y su paso a otros estados en función de la comunicación que se establece con el usuario. Podemos considerarlo como un subgrafo dentro de un espacio de fases de dimensión igual al número de todas las posibles comunicaciones que se establecen con el usuario. Aquí NO nos interesa el estado de la aplicación en cuanto a procesamiento, sino sólo en cuanto a su comunicación con el

usuario. Cada aplicación posee su propio grafo de interacción. Los grafos de interacción de distintas aplicaciones pueden estar conectados. Es un modelo para lo que se denomina *base de conocimiento* en [7].

El *perfil* de un determinado usuario es un grafo que crece a partir de los caminos que el usuario va siguiendo por el grafo de interacción de la aplicación. Similar al *modelo mental* propuesto por Lieberman [6]. Toda esta dinámica de flujo del usuario hacia la aplicación es simétrica, es decir, también se aplica en la dirección de la aplicación hacia el usuario. Se establece así un sistema de mensajes de doble sentido: *orden-notificación*.

6 Conclusiones y trabajo futuro

Intentamos con nuestro modelo superar los problemas que plantea el trabajo con sistemas previos, rediseñando un nuevo sistema que permita integrar fácilmente todos los requisitos, modelos y protocolos diseñados en el mundo de las interfaces de usuario basadas en agentes.

Toda esta estructura tiene múltiples extensiones que amplían el ámbito de actuación de esta arquitectura. Una de estas extensiones es crear un sistema *multiaplicación* para que un solo agente gestione la interacción con varias aplicaciones. También contemplamos la posibilidad de agentes multiusuario, que permita la comunicación y la colaboración entre usuarios, además de realizar tareas cooperativas entre usuario y agentes (como en [8]).

La arquitectura propuesta permite integrar un *Servidor de Aplicaciones*, de forma que el agente se comunica con él en lugar de con la aplicación. Esta estructura nos permitiría tener un sistema distribuido de aplicaciones. Cuando integramos varias aplicaciones en el sistema hemos de tener en cuenta que los grafos de interacción de cada aplicación no son elementos aislados, por lo tanto, debemos establecer algún método de relación entre grafos.

Para la comunicación de los agentes sería interesante utilizar el formato *ACL (Agent Communication Language, [9])*, el cual utiliza *KQML (Knowledge Query and Manipulation Language)* para la comunicación del agente hacia el exterior (p.e. otro agente) y *KIF (Knowledge Interchange Format)* para comunicaciones internas.

En la comunicación entre el motor y la aplicación se estudia la utilización de DCOM o CORBA que son las interfaces más extendidas en la actualidad.

Para la implementación del agente se puede utilizar Java, ya que son múltiples las ventajas que ofrece. Además de la portabilidad, el lenguaje Java soporta el uso de comunicaciones y el desarrollo de aplicaciones multitarea y multiplataforma.

La evaluación del sistema se basará en el desarrollo de perfiles mediante un entrenamiento masivo. En esta fase el sistema registrará la interacción de un número determinado de

usuario durante un tiempo prefijado. Una vez obtenidos varios perfiles se analizarán las diferencias de las métricas satisfacción y productividad del usuario.

Referencias

- [1] Scott M. Brown, Eugene Santo Jr. "IaDEA: A Development Enviroment Architecture for Building Generic User Interface Agents", 1998
- [2] Henry Lieberman, "Autonomus Interface Agents", ACM Conference on Human-Computer Interaction [CHI-97], Atlanta, March 1997.
- [3] L. Bartfield. "The User Interface. Concepts & Design". Addison Wesley. 1993
- [4] Todd Sundsted. "An introduction to agents". Java World on-line publication, Junio 1998.
- [5] Yezdi Lashkari, Max Metral, Patti Maes, "Collaborative Interface Agents". IT Media Laboratory, Autonomous Agents Group.
- [6] Henry Lieberman, "Integrating User Interface Agents with Conventional Applications". Proceedings of the ACM Conference on Intelligent User Interfaces, San Francisco, January 1998.
- [7] Robert J.K. Jacob James G. Schmolze , "A Human-Computer Interaction Framework for Media-Independent Knowledge". Department of Electrical Engineering and Computer Science
- [8] M. Tambe, D.V.Pynadath, N. Chauvat y otros. "Adaptive Agent Integration Architectures for Heterogeneous Team Members". Information Sciences Institute and Computer Science Department, University of California.
- [9] R. Genesereth, S.P. Ketchpel. "Software Agents". Standford University. 1994
- [10] Gaelle Calvary, Joëlle Coutaz, Laurence Nigay. "From Single-User Architectural Design to PAC: a Generic Software Architecture Model for CSCW". CHI 97 Conference Proceedings, Atlanta, Georgia, 242-249. March 1997. ACM/Addison-Wesley 1997, ISBN0-201-32229-3

Descripción de la expresividad de agentes inteligentes mediante Alhambra

D. Martín, M.Gea
Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Granada, Av. Andalucía 38, Granada
e-mail: <dmartin,mgea>@ugr.es

Resumen

El desarrollo y uso de agentes es un área de gran interés en estos últimos años. Uno de los componentes principales en el diseño de agentes es la forma de gráfica, esto es, su aspecto visual. Se presenta el sistema *Alhambra* como herramienta abierta para la definición tanto de las componentes estéticas como expresivas de agentes, aportando las ventajas de permitir la generación de agentes 2D y 3D indistintamente, y la generación de acciones mediante un modelado jerárquico de las mismas.

Palabras clave: agentes, transformaciones, animación.

1 Introducción

En el desarrollo de los interfaces de usuario, se observa en los últimos años la aparición de un gran interés por los llamados agentes inteligentes. Los agentes están formados por distintas partes: la captura de datos, el almacenamiento de dichos datos, la posibilidad de inferir nueva información y patrones de comportamiento, generar repuestas acordes a los estímulos y transmitir dicha información al usuario. Es esta última componente es en la que nos vamos a centrar: la interfaz gráfica de los agentes.

La información visual del agente tiene dos componentes principales: la componente estética y la componente expresiva. La *componente estética* se relaciona con el aspecto visual que tendrá el agente, antropomorfo o no, ser real o imaginado, bidimensional o tridimensional, etc. En general, existe una tendencia a la producción de agentes con características antropomorfas, con elementos faciales muy simplificados, 2D o 3D. La *componente expresiva* se relaciona con las posibilidades de transmitir información mediante movimientos.

Mientras que la producción de modelos tridimensionales de apariencia realista o de personajes está bastante desarrollada, no ocurre lo mismo para la obtención de objetos con una apariencia bidimensional. Esta característica resulta importante debido a que en general

se prefieren agentes con un conjunto simplificado de elementos expresivos, ya que presentan varias ventajas: son más fáciles de crear y modificar, son más fáciles de entender, y no resultan tan intimidatorios como los modelos muy realistas [1]. Es por ello por lo que se suelen usar gentes con aspecto de personajes de dibujos animados, con apariencia bidimensional.

Frente a las ventajas que aporta las imágenes bidimensional, nos encontramos con el problema de la dificultad para su generación totalmente automática mediante ordenador. En general, se dispone de secuencias de animación que son encadenadas o bien se realiza interpolación entre movimientos clave. El primer caso, implica una limitación en cuanto a la posibilidad de modificación y creación de las acciones, mientras que en el segundo caso, se debe simplificar demasiado el modelo para permitir una correcta interpolación. Nuestro trabajo está encaminado hacia un sistema flexible para generación de modelos con apariencia definible por el usuario (bidimensional, tridimensional, esquemática, etc.).

2 El sistema *Alhambra*

Si bien el desarrollo de modelos tridimensionales permite la creación de seres tanto reales como imaginados con una visualización muy realista, reproduciendo de forma fidedigna los efectos de luz y movimiento, no ocurre lo mismo con la creación de animación e ilustración con la estética del dibujo animado clásico. *Alhambra* [2] es un programa que se ha desarrollado para permitir, principalmente, ilustraciones y generación de animación de estética bidimensional con las características estéticas y expresivas que se pueden encontrar en la animación clásica. Para la componente estética se ha desarrollado el modelo *Luces Virtuales* que permite la generación de imágenes de aspecto bidimensional a partir de modelos tridimensionales. Para la componente expresiva se ha desarrollado el método de las Transformaciones No-lineales Extendidas Jerárquicas que permiten, mediante una aproximación tipo cinematográfica directa, la producción de movimientos de forma fácil y flexible.

El sistema, al usar modelos geométricos tridimensionales, aporta todas las características de estos, pero además añade la posibilidad de producir imágenes con una estética de dibujo animado a partir de dichos modelos. Esto resuelve los problemas que ocurren en los sistemas de animación bidimensional, principalmente el del intercalado. Dada esta orientación hacia la producción de animación de estética de animación clásica, se ha desarrollado un método que permite una fácil definición de movimientos de los personajes.

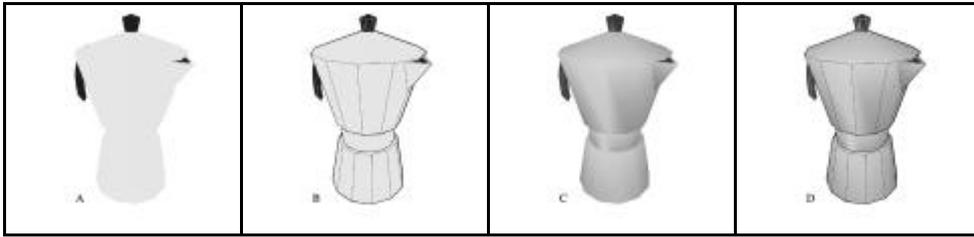


Figura 1. Ejemplo de las distintas formas de visualizar del sistema. Modo no realista sin trazos (A) y sin trazos (B). Modo realista sin trazos (C) y con trazos (D)

2.1 Luces Virtuales

Dado que el sistema parte de modelos geométricos tridimensionales y se desea obtener una estética de animación clásica, debe haber un proceso de conversión. Para ello hemos de definir cuales son los elementos estéticos que hacen que una imagen pueda ser considerada dentro de dicho tipo. Los componentes principales son el uso de trazos y de un modelo de color simplificado. Los trazos son líneas que describen la forma del objeto. Los trazos pueden ser de dos tipos: líneas de borde y líneas de forma. Las líneas de borde delimitan el borde exterior de un objeto. Un subconjunto de las líneas de borde son las siluetas. Las líneas de forma indican cambios de color o tono. Los trazos son un elemento indispensable, tanto en animación como en ilustración, ya que la mayoría de las veces son el único elemento visual que permite definir la forma de los objetos, ya que el color que se usa suele ser plano.

Para la obtención de los trazos se ha desarrollado el modelo *Luces Virtuales*. La idea principal de este método es que los trazos, tanto líneas de borde como líneas de forma, pueden considerarse como límites de características visuales, producidas por luces. Dado que se usa un mecanismo parecido al modelo simplificado de iluminación, pero sin producir efectos en el color, sino el producir los trazos, se denominan luces virtuales.

2.3 Transformaciones No-Lineales Extendidas Jerárquicas

Las Transformaciones No-Lineales Extendidas Jerárquicas (TNLEJ) son una extensión de las Transformaciones No-Lineales de Barr[3], añadiendo mayores posibilidades en la obtención de deformaciones, basándose para ello en el uso de las funciones de control, el eje de aplicación y el eje de selección. Además se pueden crear estructuras jerárquicas en las cuales se pueden combinar transformaciones geométricas y Transformaciones No-Lineales. La definición de la transformación depende de la función de control. Mediante la combinación de varias TNLEJs se pueden crear deformaciones complicadas de una manera sencilla e intuitiva. Al estar definidas de forma paramétrica, las funciones de control se

pueden hacer depender de otras funciones, las cuales a su vez dependen del tiempo, lo que permite crear animaciones.

3 Diseño de agentes

Una vez se han expuesto los componentes principales del sistema *Alhambra*, vamos a desarrollar una capa de comportamientos que permita describir las transformaciones que se deben aplicar a los modelos para que sus acciones estén relacionadas con las acciones del usuario sobre el sistema. Como ventaja, podremos crear y editar la componente visual de los agentes, estética y expresividad de una forma fácil e intuitiva, añadiendo la ventaja de poder utilizar modelos con apariencia tridimensional o bidimensional indistintamente.

3.1. Visualización de los agentes

Los agentes inteligentes a menudo presentan una apariencia visual para comunicar información al usuario. La forma que normalmente se usa es la de seres antropomorfos, y en particular los rostros de los mismos. Esto es así debido a la riqueza semántica que puede presentar la gesticulación facial. Dentro de la creación de rostros, se observan dos líneas divergentes: una en la cual se opta por versiones bastante detalladas, y casi siempre de aspecto realista y tridimensional, y otra en la que se opta por la simplificación de los mismos, con apariencia normalmente bidimensional.

La visualización de agentes mediante *Alhambra* tiene un gran nivel de flexibilidad ya que definido el modelo geométrico que represente al agente, el mismo puede ser visualizado de distintas maneras, desde seres reales con apariencia realista a personajes imaginados con apariencia de dibujo animado, esto es, seres antropomorfos, por ejemplo, un cohete con cara, etc. Esto permite adaptar las características visuales del agente a las capacidades y necesidades del usuario. Así por ejemplo, para niños resulta más conveniente utilizar agentes sencillos y divertidos, mientras que para un adulto las necesidades pueden ser muy distintas.

3.2. Expresividad de los agentes

La otra gran componente en la visualización es la expresividad. Es aquí donde se realizamos nuevos planteamientos frente a los que había desarrollados en *Alhambra*. Como se ha comentado previamente, la forma de los movimientos de los objetos vienen determinados por las funciones de control, las cuales indican cómo se va a producir la transformación. Las funciones de control, dependen a su vez de otras funciones que permiten la evolución de los movimientos, normalmente, dependiendo del tiempo.

Dadas las capacidades inherentes de las Transformaciones No-Lineales Extendidas Jerárquicas para la descripción de acciones y su evolución, se plantea su utilización para la generación de acciones con agentes. Se parte de una herramienta que permite crear escenas que son secuenciales, para obtener otra que permita la definición, de una forma fácil e intuitiva, de acciones que puedan ser utilizadas como átomos expresivos con agentes.

3.3. Modelado jerárquico de las acciones

La principal característica que se plantea para el diseño de las acciones es que posee una estructura jerárquica. Esta estructura es a la vez un reflejo de la estructura del modelo, el cual es también jerárquico, ya que las acciones de más alto nivel afectan a un mayor número de objetos que las de más bajo nivel. Estamos interesados en representar acciones que se identifican con verbos, pero con la capacidad de modularlas mediante la velocidad con la que se produzcan y la intensidad de la misma. Así, nos planteamos, por ejemplo, la acción "abrir ojo", la cual se puede realizar más o menos rápidamente, y además puede variar entre una ligera abertura, a una gran abertura.

Para la ejecución de las acciones se definen intervalos de tiempo en las cuales se deben realizar. La variación de la velocidad se consigue modificando la duración de dicho intervalo. La modificación de la intensidad se logra haciendo que los parámetros que definen las transformaciones no estén formados por valores fijos sino por funciones, que como ya hemos comentado, es el mecanismo que se usa en las TNLEJ. Esto es, permitir al diseñador la creación de acciones de forma jerárquica, de tal manera que acciones de más alto nivel sean semánticamente más ricas, las de más bajo nivel.

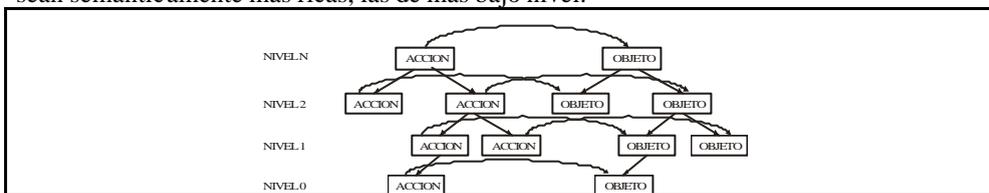


Figura 2. Estructura jerárquica mostrando la relación entre geometría y acciones

3.3. Comportamiento de los agentes

Los agentes representan entes autónomos dentro de la aplicación que asisten al usuario a lograr una serie de objetivos. En esta finalidad, la expresividad del agente puede ayudar significativamente a comprender los resultados de las acciones realizadas por el usuario. Una vez que se hemos diseñado un modelo expresivo de los gestos del agente, deberemos asociarle un comportamiento dinámico relativo al dominio de la aplicación.

Un agente determina el grado de cumplimiento de las acciones que produce el usuario dentro de una aplicación, por lo que deberá estimar el grado de cumplimiento de un objetivo. Por tanto, ya que se debe realizar estimaciones del comportamiento humano, vamos a conectar los gestos expresivos del agente con un modelo híbrido (cognitivo y predictivo [4]) de especificación de la aplicación. Mediante un modelo cognitivo podemos modelar el conocimiento que debe poseer el usuario acerca de una determinada tarea, mientras que los modelos predictivos nos permite estimar el tiempo de respuesta para llevar a cabo una acción. Vamos a describir un ejemplo sencillo de tarea especificada con GOMS y KLM. El modelo KLM (KeyStroke Level Model) consiste en un modelo GOMS simplificado que reduce el conjunto de operaciones disponibles a la pulsación de teclas y movimiento de ratón. Se centra en estudiar el rendimiento del procesamiento observable: pulsación de teclas (K), movimiento ratón (P), preparación mental (M), teniendo en cuenta que estos parámetros varían en cada usuario.

```

GOAL: FORMATEAR_DISCO
. GOAL: SELECCIONAR_ORDEN                1 M
. . [select GOAL: ORDEN_FORMATEAR
. . . MOVER-A-SHELL                      1 P (opcional)
. . . . ESCRIBIR-ORDEN-FORMATO          6*K ("format")
. . . . ESCRIBIR-UNIDAD                 2*K ("A:, C:")
. . . . ACEPTAR                       1K ("enter")
. . . GOAL: MENU_FORMATEAR
. . . . MOVER-MENU                      1 P
. . . . IDENTIFICAR-OPCION-FORMATO      1 K
. . . GOAL: SELECT-UNIDAD
. . . . MOVER-A-VENTANA-SELECCIÓN-UNIDAD 1 P
. . . . SELECCIONAR UNIDAD              1 P
. . . . PULSAR-BOTON-MOUSE]            1 K
. GOAL: CONFIRMAR                        1 M
. . [select GOAL: CONFIRMAR_TECLADO
. . . PULSAR-S                          2K ("s + enter")
. . . GOAL: CONFIRMAR_RATON
. . . . MOVER-VENTANA_CONFIRMACION      1 P
. . . . MOVER-BOTON-OK                  1 P
. . . . PULSAR-BOTON-MOUSE]            1 K
    
```

Selection Rule for GOAL: FORMATEAR_DISCO

Si es un sistema grafico, usa el metodo MENU-FORMATEAR, else usa ORDEN_FORMATEAR

Selection Rule for GOAL: CONFIRMAR

Si usas el método ORDEN-FORMATEAR then usa el metodo CONFIRMAR_TECLADO, else usa CONFIRMAR-RATON

Vamos a asociar gestos y expresividad tanto en la consecución de metas parciales como en la estimación predictiva del comportamiento humano. En el primer caso obtendremos una expresión acorde al estado actual de cumplimiento de una acción mientras que en el segundo caso podremos analizar el comportamiento del humano.

on(ORDEN_FORMATEAR) -> Agente preocupado, frunce ceño (imagen b)
on(ORDEN_CONFIRMAR) -> Agente interrogando, mirar fijamente (imagen c)

Por ejemplo, desde el punto de vista predictivo se ha estimado que el objetivo CONFIRMAR debe realizarse en $M+2K$ o bien $M+2P+K$. Un promedio menor puede denotar que es un usuario experto y está seguro de lo que quiere realizar, o bien que se ha precipitado en confirmar la acción. Un gesto preocupado puede describir ambas situaciones. Un promedio mayor del estimado puede connotar dudas e incertidumbre. En este caso podemos utilizar un gesto de temblor (asustado). Nótese que los operadores K, M, P, están en función de cada usuario, por lo que se deben ajustar tras un entrenamiento con el usuario.

time(CONFIRMAR_TECLADO) $\ll M+2K$ -> Agente preocupado (imagen a)
time(CONFIRMAR_TECLADO) $\cong M+2K$ -> Agente aprueba acción(imagen d)

Ambas expresiones (relativas a modelos de usuario diferente) son compatibles en el tiempo y cada una de ellas pondera un aspecto diferente del comportamiento del humano (capacidad y rendimiento).

A partir del lenguaje de descripción de acciones que se ha especificado se han creado algunas acciones muy sencillas, las cuales han podido ser reutilizadas para la creación de otras acciones de mayor nivel. En la Figura 3 se muestra un personaje con distintas expresiones. El paso de una expresión a otra se realiza mediante una interpolación. Esta forma de operar es inherente al sistema ya que se usan funciones para definir las acciones.

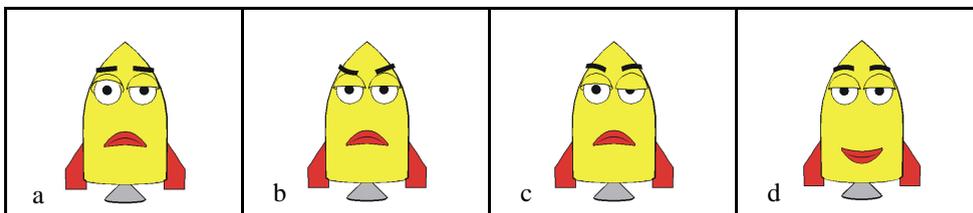


Figura 3. Ejemplo de un personaje mostrando distintas expresiones.

4 Conclusiones

En el presente trabajo se ha presentado una herramienta para la obtención de la parte visual de un agente, aportando la ventaja de poder utilizar indistintamente una estética

bidimensional o tridimensional, y sobre todo, permitiendo la creación de acciones para su posible utilización como forma de expresión de los agentes.

Aunque se plantea como una herramienta de bajo nivel, la posibilidad de construir de forma jerárquica, tanto los modelos geométricos, como especialmente, las acciones de los mismos, abre la posibilidad de convertirla en una herramienta de carácter genérico, permitiendo el que se puedan crear conjuntos de acciones que puedan ser reutilizados sin intervención del diseñador, o solamente para permitir una mayor personalización de las mismas.

Referencias

- [1] P. Maes, "Agents that reduce the Work and Information Overload", Readings in Intelligent User Interfaces. Morgan Kaufman, 1998.
- [2] D. Martín, Alhambra, Un Modelo Para la producción de Animación Bidimensional". Tesis Doctoral ETSI Informática, 1999.
- [3] A. Barr, "Global and Local Deformations of Solid Primitives", Proceedings of SIGGRAPH, no3, vol 18, 1984.
- [4] R. Ebert: "*User Interface Design*". Prentice Hall, 1994.

Sesión III

Docencia en Interacción Persona- Ordenador

Interacción Hombre - Máquina en el Centro Politécnico Superior de la Universidad de Zaragoza

Sandra Baldassarri, Francisco José Serón, Pedro Latorre, Juan Antonio Magallón

Grupo de Informática Gráfica Avanzada
Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior – Universidad de Zaragoza
Tlf: 976.761.939 Fax: 976.761.914 e-mail: seron@posta.unizar.es

Resumen

Este artículo presenta la experiencia docente del Grupo de Informática Gráfica Avanzada (GIGA) en la asignatura de Interacción Hombre-Máquina (IHM), incluida en el plan de estudios de Ingeniería Superior Informática. El GIGA es un grupo de investigación y desarrollo (I+D) integrado dentro del Área de Lenguajes y Sistemas Informáticos del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza.

La asignatura de IHM ofrece una perspectiva del proceso del diseño de sistemas interactivos centrado en el usuario. En el artículo se expone el planteamiento didáctico utilizado en la asignatura, desarrollando los objetivos, metodología docente y control del aprendizaje. Asimismo se desarrollan sus contenidos, prácticas y sistema de evaluación. Posteriormente se realiza un análisis de los resultados académicos obtenidos por los alumnos y se formulan las conclusiones.

Palabras clave: docencia, interacción hombre-máquina, entornos educativos.

1. Entorno de trabajo

En el Centro Politécnico Superior de Ingenieros de la Universidad de Zaragoza se imparten cuatro titulaciones de Ingeniería Superior: Ingeniero Industrial, Ingeniero en Telecomunicaciones, Ingeniero en Informática e Ingeniero Químico, así como cursos de postgrado y programas de doctorado.

En cuanto a las carreras de Ingeniería, los planes de estudio están estructurados en dos ciclos, cada uno de los cuales consta de cinco cuatrimestres. El primer ciclo proporciona una formación básica mientras que la posibilidad de especialización se centra en el segundo

ciclo. El alumno podrá optar por cursar un segundo ciclo con una formación diversificada o centrarse en una determinada especialidad. En estas carreras, para superar las asignaturas, el alumno dispone de tres convocatorias, de las cuales puede llegar a utilizar, en caso necesario, dos de ellas.

La asignatura que nos ocupa, IHM, se ofrece como optativa cuatrimestral de 2º ciclo para los alumnos de la carrera de Ingeniería Informática. La carga docente de la asignatura es de 6 créditos, que se corresponden con 4 horas semanales de clase. De los 6 créditos, 4.5 corresponderían a la parte de teoría y 1.5 a las clases prácticas, aunque en la realidad, debido a la imposibilidad existente hasta el momento para realizar las clases prácticas, todas las horas que se imparten son de teoría. Los alumnos pueden acceder a la asignatura a partir del tercer año de carrera, aunque como la asignatura pertenece al 10º cuatrimestre, generalmente lo hacen en 5º año de carrera.

2. Planteamiento didáctico

2.1. Objetivos

Los alumnos llegan a esta asignatura sin haber visto en ninguna otra parte de la carrera los conceptos y metodologías que tienen que ver con la interacción persona-ordenador. Debido a esta falta de conocimientos previos, a la amplitud de temas abarcados y a la rápida y continua evolución de los mismos, en esta asignatura se intenta dar una visión amplia y global de las nuevas tecnologías para que los alumnos interesados en estos temas puedan investigar y trabajar en ellas posteriormente por su cuenta (contando con una base, terminología, referencias para búsqueda de información, etc.).

El objetivo general de la asignatura de Interacción Hombre-Máquina (IHM) es, por lo tanto, ofrecer al alumno una perspectiva del mundo de la ingeniería del interfaz de usuario, describiendo sus posibilidades, principios y métodos de análisis, diseño e implementación.

2.2. Metodología

Esta asignatura se caracteriza por su diversidad temática y su continua evolución, por lo cual se decidió dividirla en cuatro bloques temáticos, impartido cada uno de ellos por un profesor diferente, especializado en dicho campo. Estas características específicas hacen que las clases deban impartirse siguiendo el modelo magistral.

Por otra parte, debido a la constante evolución de estos temas resulta casi imposible hacer referencias a la misma documentación en más de un año. Por lo tanto, en cuanto al material

de estudio, se ha optado por utilizar capítulos de libro para temas muy concretos y básicos, y artículos de revistas de actualidad para lo referente a hardware, software, etc.

2.3. Control del aprendizaje

La evaluación del rendimiento se hace por medio un examen teórico en el cual se pretende que el alumno demuestre los conocimientos básicos de cada una de las cuatro partes que componen la asignatura. Asimismo, el alumno debe realizar un trabajo en el cual se consideran importantes la innovación, la investigación realizada por el alumno, su capacidad para resolver problemas, ...

3. Contenidos

Los contenidos de Interacción Hombre-Máquina se describen en la Tabla 1.

| IHM | |
|--|---|
| Bloque 1: <i>Principios del Diseño de Interfaces para la Interacción Hombre – Computador</i> | Introducción: humano, computador, interacción Principios prácticos de diseño de IU: Análisis, Diseño, Implementación Evaluación: normativas estándar y su aplicabilidad Realización de interfaces de usuario para el trabajo en red: Lenguajes, Guías de estilo y evaluación de interfaces en red |
| Bloque 2: <i>Funcionamiento e Implementación de Interfaces Gráficas de Usuario</i> | Sistemas de Interacción 2D basados en ventanas Arquitectura Software de los Sistemas de Ventanas: Cliente-Servidor Paradigmas de Programación: Bucles de eventos y notificación APIs Estándar y Programación de IU en 2D Sistemas de Desarrollo para Ius Temas Avanzados: Interfaces de Usuario 3D e Interfaces en red |
| Bloque 3: <i>Sistemas y Aplicaciones Multimedia</i> | Introducción y descripción general de los elementos multimedia Interactividad, Texto, Gráficos, Bases de Datos Imagen, sonido y vídeo: Adquisición, compresión y tratamiento. Diseño y producción de aplicaciones multimedia Desarrollo de proyectos Campos actuales de desarrollo y futuro |
| Bloque 4: <i>Sistemas</i> | Introducción histórica Definición de Realidad Virtual |

| | |
|--|---|
| <i>Multisensoriales, Realidad Virtual y Realidad Aumentada</i> | Periferia de la Realidad Virtual Arquitecturas para los sistemas de RV Evaluación de los sistemas de RV Aplicaciones Realidad Aumentada |
|--|---|

Tabla 1: Programa de la asignatura de Interacción Hombre-Máquina.

4. Prácticas

La diversidad y amplitud de temas impartidos en la asignatura, sumado a la carencia de infraestructura necesaria, plantea una dificultad a la hora de realizar sesiones de prácticas que comprendan la totalidad de los contenidos. El dominio práctico de los conceptos estudiados se adquiere al realizar el trabajo práctico final, tal como se describe en el siguiente apartado.

5. Evaluación

Para realizar la tarea de control y poder medir la información recibida por el alumno se ha optado por realizar dos pruebas:

Un examen sobre los contenidos básicos que se califica con Apto o No Apto. Este examen incluye preguntas genéricas sobre cada uno de los bloques explicados en clase.

Un trabajo realizado por el alumno al final del curso en una de las 4 áreas. Los trabajos realizados pueden clasificarse en dos grandes grupos:

- Prácticos: Utilización e integración de alguna de las herramientas descritas en clase.
- Teóricos: Trabajos de investigación y documentación sobre nuevas técnicas y productos relacionados con el temario.

El material utilizado para la realización de las prácticas varía notablemente dependiendo del tipo de trabajo (teórico o práctico) y del bloque elegido. Los trabajos teóricos se caracterizan por un gran trabajo de investigación del alumno en bibliografía actualizada (libros y revistas) y en Internet, mientras que los trabajos prácticos se caracterizan por la utilización de alguna de las herramientas o métodos de trabajo descritos en clase. En ambos casos los temas de los trabajos a realizar pueden ser propuestos tanto por el alumno como por el profesor.

En la Tabla 2 se enumeran algunos de los trabajos realizados dentro de cada bloque a lo largo de los tres cursos impartidos de la asignatura.

| IHM | |
|--|--|
| <p>Bloque 1: <i>Principios del Diseño de Interfaces para la Interacción Hombre - Computador</i></p> | <p>Realización de interfaces utilizando una guía de estilo y un método de evaluación previamente diseñado. La norma ISO 9241 y su aplicación a entornos basados en Internet. Aplicación de las técnicas Lo-Fi al diseño de la interfaz de un sistema electrónico doméstico Diseño de websites</p> |
| <p>Bloque 2: <i>Funcionamiento e Implementación de Interfaces Gráficos de Usuario</i></p> | <p>Comparación entre sistemas de desarrollo: realización de una aplicación simple sobre varios toolkits (Qt, Gtk, Jx, Vx, ViewKit) Estudios de portabilidad: toolkits disponibles para distintas plataformas. Implementación de un interfaz gráfico sobre aplicaciones existentes orientadas a texto</p> |
| <p>Bloque 3: <i>Sistemas y Aplicaciones Multimedia</i></p> | <p>Utilización de herramientas de autor para la realización de CD-ROMs Comparación y ejemplificación de técnicas de animación Comparación de sistemas de audio y métodos de compresión Implementación métodos de compresión</p> |
| <p>Bloque 4: <i>Sistemas Multisensoriales, Realidad Virtual y Realidad Aumentada</i></p> | <p>Simulación y entornos inmersivos Visualización en tiempo real sobre grandes escenarios Realidad virtual y educación Ocio y entornos inmersivos Medicina y los entornos inmersivos Arquitectura, previsualización y realidad virtual Experiencias de realidad aumentada</p> |

Tabla 2: Trabajos prácticos en la asignatura de Interacción Hombre-Máquina.

El trabajo realizado por el alumno permite que el profesor pueda evaluar los conocimientos adquiridos sobre un tema como así también la originalidad en la resolución de un problema y los procesos seguidos por el alumno en relación con la atención, abstracción y codificación de datos, aplicación, análisis, síntesis o generalización del conocimiento, memoria, etc.

Cada trabajo es evaluado en primer lugar por el profesor del bloque correspondiente para revisar los contenidos, la profundidad alcanzada y la claridad de exposición. Posteriormente

se realiza una revisión conjunta entre todos los profesores de la asignatura para poder tener una visión global de los trabajos presentados y determinar la nota obtenida por el alumno.

6. Análisis de los resultados académicos

En la Tabla 3 se muestra la evolución histórica (en porcentajes) de los resultados obtenidos por los alumnos en el período de 1996/97 a 1998/99. Los alumnos eligen la convocatoria en la que se presentan con arreglo a la siguiente distribución:

| | Matric. | 1ª Convocatoria | 2ª Convocatoria | 3ª Convocatoria | Total |
|-------|---------|-----------------|-----------------|-----------------|-------|
| 96-97 | 58 | 25.86 | 15.52 | 39.65 | 81.03 |
| 97-98 | 66 | 16.67 | 30.30 | 25.76 | 72.73 |
| 98-99 | 70 | 17.14 | 22.86 | 38.57 | 78.57 |

Tabla 3: Resultados en el bloque de Interacción Hombre-Máquina.

La tabla de resultados refleja la tendencia creciente en cuanto a la matriculación de alumnos en esta asignatura.

En estos tres años de experiencia se ha observado que se han presentado a examen un promedio del 77.44 % sobre el total de matriculados. Aunque en este promedio hay que tener en cuenta que el porcentaje de la columna "Total" se encuentra ligeramente sesgado dado que puede haber alumnos que se presenten a más de una convocatoria. Sin embargo, sobre estos porcentajes es interesante resaltar la distribución de las calificaciones obtenidas con respecto al total de alumnos presentados en cada curso, tal como se detalla en la Tabla 4:

| | Suspense | Aprobado | Notable | Sobresaliente |
|-------|----------|----------|---------|---------------|
| 96-97 | 0 | 21.28 | 34.04 | 44.68 |
| 97-98 | 12.5 | 18.75 | 37.5 | 31.25 |
| 98-99 | 20 | 18.18 | 32.72 | 29.09 |

Tabla 4: Resultados en el bloque de Interacción Hombre-Máquina.

Como se puede observar, el número de suspensos fue aumentando a partir del primer año, en gran parte debido a una insuficiencia en los conocimientos teóricos por parte de los

alumnos. Sin embargo, una vez superado el examen teórico, los trabajos prácticos presentados son generalmente de alta calidad, lo que se refleja en el tipo de notas obtenidas ya que un alto porcentaje de alumnos obtienen una calificación entre notable y sobresaliente.

7. Conclusiones

La asignatura de IHM se caracteriza principalmente por la diversidad de temas tratados y por su continua evolución, por lo tanto ha resultado adecuada la división de la misma en cuatro bloques temáticos, cada uno de ellos impartido por un profesor especializado.

La distribución vertical de los temas (una hora por semana para cada tema) parece más adecuada que la horizontal (un tema tras otro) aunque los alumnos manifiestan opiniones divergentes.

A pesar de que en el aspecto práctico probablemente sería ideal proponer un solo trabajo que englobara todos los contenidos vistos en la asignatura, este planteamiento presenta dos dificultades: el número de horas necesario excede notablemente la carga de trabajo asignado en la asignatura y, además, es preciso contar con un conocimiento previo de numerosas herramientas. Por lo tanto, se ha optado por una solución de compromiso consistente en proponer el trabajo práctico centrándose en un área específica.

La evaluación teórica hace que el alumno tenga una visión general de la asignatura, mientras que el desarrollo de un trabajo práctico permite que se especialice en aquel bloque que más le interesa.

Finalmente, teniendo en cuenta el contexto general de la carrera de Ingeniería Informática consideramos que con esta asignatura se ha transmitido al estudiante la necesidad de contar con el usuario como elemento central en todo trabajo de diseño de aplicaciones informáticas, además de obtener una visión general de las nuevas tecnologías, indispensables para un informático.

8. Bibliografía recomendada

En la asignatura se utilizan varios libros de los cuales se extraen las ideas básicas, pero debido a la continua evolución de los temas tratados en clase, la mayor parte del material bibliográfico se centra fundamentalmente en artículos de actualidad.

Dix A., Finley J., Abowd J., Beale R.: "*Human-Computer Interaction*", Ed. Prentice-Hall, 2ª edición, 1998.

Schneiderman B: "*Designing the User Interface*", Addison-Wesley, 3ª edición, 1998.

Marcus A.: "Graphic Design for Electronic Documents and User Interfaces", Tutorial series, ACM Press, 1992.

Burger J.: "*La Biblia del Multimedia*", Ed. Addison-Wesley Iberoamericana, 1994.

Ozer J.: "*Video Compression for Multimedia*", Ed. AP Professional, 1995.

Burdea G., Coiffet P.: "*Tecnologías de la Realidad Virtual*", Ed. Paidós Hipermedia 3, 1996.

Vacca J.: "VRML Bringing Virtual to the Internet", Ed. AP Professional, 1996.

Interacción Persona Ordenador en la EUITIO

Ana Belén Martínez, Benjamín López, Juan Manuel Cueva, Iván Fernández

Departamento de Informática, Área de Lenguajes y Sistemas
Universidad de Oviedo, Calvo Sotelo s/n 33007, Oviedo
Tlf: 985.10.33.94 Fax: 985.10.33.54 e-mail: belen@lsi.uniovi.es

Resumen

En esta comunicación se pretende exponer como está organizada actualmente la enseñanza de Interacción Persona Ordenador (IPO) en la Escuela Universitaria de Ingeniería Técnica Informática de Oviedo (EUITIO). Para ello se hace una descripción de los objetivos, metodología y contenidos de la única asignatura de IPO impartida en el plan de estudios actualmente en vigor. Finalmente, se hace una breve exposición de la organización por la que se apuesta para la enseñanza de IPO en el nuevo plan de estudios y que formará a los futuros ingenieros técnicos.

Palabras clave: Interacción persona ordenador, plan de estudios.

1 Interacción Persona Ordenador en el Plan Actual

La disciplina de Interacción Persona Ordenador (IPO) se ha estado impartiendo en la EUITIO, bajo el nombre de comunicación hombre-máquina, como asignatura optativa tanto para gestión como para sistemas desde la entrada en vigor del plan de estudios actual en 1992.

La presencia de esta única asignatura en el plan de estudios (Figura 1) fuerza a que sus contenidos se traduzcan en una visión general de lo que es la IPO, viéndose por tanto los objetivos de la misma reducidos al asentamiento de los conceptos básicos de interacción persona ordenador, tratando a la vez de concienciar al futuro ingeniero de la importancia cada vez mayor que tienen las interfaces de usuario, muy a menudo las grandes olvidadas, en el desarrollo de software y por tanto en su formación.

diálogo, barras de herramientas, aplicaciones de interfaz de documento múltiple, etc. En cada clase práctica se irán explicando estos conceptos y realizando ejercicios prácticos que permitan fijar los conocimientos previamente presentados. Especial atención se le dedica en las clases prácticas a la construcción de un buen soporte al usuario así como a la internacionalización de aplicaciones.

1.2 Evaluación

Para la evaluación de los conocimientos adquiridos se realiza una prueba teórica (a modo de test), y el proyecto práctico anteriormente mencionado. Dicho proyecto será individual y consistente en el desarrollo de una aplicación con una interfaz de manipulación directa (actualmente la más predominante). En esta aplicación se considera obligatorio: la incorporación de soporte para el usuario en forma de ayuda en línea (tutorial opcional), la internacionalización (con dos localizaciones) y la inclusión de un instalador/desinstalador.

El contenido eminentemente práctico de la asignatura fuerza a que en la nota tenga un mayor peso el ejercicio práctico que el teórico por considerar que es dónde podrán realmente plasmar la mayoría de los conocimientos adquiridos.

1.3 Herramientas

A la hora de seleccionar la herramienta para el desarrollo del proyecto, se le deja al alumno cierta libertad. El hecho de que el alumno tenga unos conocimientos adquiridos en determinados lenguajes de programación tales como C++ y Object Pascal, hace que la mayoría se decanten por entornos como Delphi, C++ Builder o Visual C++. Aunque en este último año se ha incrementado el número de proyectos que emplean el lenguaje Java.

1.4 Realización conjunta de la práctica con otras asignaturas

La realización de prácticas en las asignaturas de tercer curso de la EUITIO se suele traducir en el desarrollo por parte del alumno de una aplicación práctica. Para que ésta no sea excesivamente trivial debido al reducido número de créditos prácticos disponibles para cada asignatura y para permitir al alumno la integración de los conocimientos adquiridos en diferentes asignaturas, se permite la realización de un proyecto más o menos complejo que involucre los diferentes aspectos impartidos en distintas disciplinas informáticas. La evaluación de la práctica del alumno es totalmente independiente por parte de cada asignatura.

Este proyecto conjunto se ha venido realizando en los últimos años, de forma opcional por parte del alumno, implicando asignaturas como Informática Gráfica, Ingeniería del Software, Bases de Datos e Interacción Persona Ordenador. Los resultados son bastante positivos proporcionando al alumno una visión de conjunto que de otra forma le es bastante difícil de adquirir.

2 Apuesta por la IPO en el Nuevo Plan

En la mayoría de los sistemas informáticos hoy en día el usuario tiene que interactuar con el sistema, y la masificación en el empleo de los ordenadores hace que no todo el mundo que utiliza estos sistemas esté ampliamente preparado. Luego la adecuación de las interfaces en la medida que estas suponen el medio de comunicación entre el usuario y el sistema, será cada vez más un factor determinante en el éxito o fracaso de los productos. Es por esto, por lo que se demandan un mayor número de asignaturas que tengan en cuenta la interacción.

Como se ha mencionado anteriormente es necesaria una asignatura de introducción en la que se fijen los conceptos básicos, pero son necesarias otras asignaturas que tengan en cuenta las necesidades básicas actuales, tales como: diseño y construcción de aplicaciones web usables, diseño de aplicaciones multimedia, etc.

Teniendo en cuenta lo anterior y con el objetivo de conseguir futuros profesionales con una amplia formación en tecnologías interactivas se ha realizado una propuesta de modificación del plan de estudios en la que se apuesta fuerte por estas disciplinas y cuyas características más destacadas son (Figura 2):

- **Interacción Persona Ordenador** se convierte en una asignatura de tercer curso obligatoria de 4.5 créditos (2 Teóricos y 2.5 Prácticos) tanto para ingeniería técnica de gestión como de sistemas.
- Creación de una intensificación relacionada directamente con la interacción persona ordenador: **Tecnologías Interactivas e Hipermedia (TIH)** de la que se destacan tres asignaturas:
 - *Interacción de Mundos Virtuales* (entornos de desarrollo de mundos virtuales y realidad virtual)
 - *Desarrollo de aplicaciones Hipermedia* (diseño de productos multimedia)

- [8] B. Laurel. "Computers as Theatre". Addison-Wesley, 1993.

El Perfil de Tecnologías Interactivas en la Ingeniería Informática de la UCLM

Manuel Prieto y Francisco Ruiz

Departamento de Informática.
Universidad de Castilla-La Mancha
Ronda de Calatrava 5, 13071 Ciudad Real
Tlf: 926.295.300 Fax: 926.295.351 e-mail: mprieto@inf-cr.uclm.es

Resumen

En este artículo se presentan los objetivos que llevaron a la creación de un perfil o itinerario de salida dedicado a las *Tecnologías Interactivas* en el Plan de Estudios de la Ingeniería Superior en Informática en la Universidad de Castilla-La Mancha, así como una breve descripción de las asignaturas que se imparten. Se incluye también un comentario sobre la experiencia de su impartición desde Octubre de 1998.

Palabras clave: Interacción Persona-Computador. Tecnologías Interactivas. Plan de Estudios

1 Introducción

El plan de estudios de la nueva Ingeniería en Informática fue instaurado en la Escuela Superior de Informática (E.S.I.) de Ciudad Real en el curso 1998-1999. Con ello, la Universidad de Castilla-La Mancha dio prioridad a los estudios de segundo y tercer ciclos en Informática que comenzaron simultáneamente en Albacete y Ciudad Real. En cada campus se definieron tres perfiles, en principio diferentes, adaptados a la demanda social y a las características y experiencia del profesorado.

Cada perfil lo conforma un grupo de seis asignaturas optativas de cuatro créditos que se ofertan en quinto año, mas una obligatoria de nueve créditos que imparte en cuarto año. La asignatura obligatoria sirve de introducción a los temas del perfil y pretende informar suficientemente a los alumnos que inician el segundo ciclo para decidir su elección de optativas de quinto. Claro que la elección es libre, pero cada perfil se centra en un tema principal lo que da la oportunidad de especialización.

Los perfiles que se han definido en La E.S.I de Ciudad Real son los siguientes:

- Sistemas de Información (Diseño, desarrollo y mantenimiento de sistemas de información integrales para las organizaciones: Bases de Datos. Ingeniería del Software. Calidad del Software. Auditoría Informática.)
- Informática Industrial (Estudio y soluciones de las necesidades informáticas de las industrias: Arquitectura. Tratamiento de la señal. Automatización. Robótica. Codiseño.)
- Tecnologías Interactivas

2 El Perfil de Tecnologías Interactivas.

El Plan de estudios [1] establece que los ingenieros formados en este itinerario serán capaces de utilizar e integrar las diversas tecnologías que facilitan la comunicación y la interacción entre las personas y las nuevas tecnologías de la información:

- Diseñando, desarrollando e implementando sistemas interactivos, interfaces de usuario, sistemas de trabajo en grupo y aplicaciones con entornos hipermedia y multimedia.
- Adquiriendo los instrumentos necesarios para elaborar modelos de aprendizaje, de cooperación y de simulación que permiten una interacción mas avanzada con los computadores.
- Conociendo las técnicas actualizadas de la inteligencia artificial y en particular las aplicables a los sistemas de adquisición, representación y recuperación del conocimiento.
- Haciendo un uso creativo de los medios técnicos que facilitan la comunicación entre los seres humanos y los sistemas informáticos.

Con este itinerario se forman los especialistas en el nuevo campo tecnológico y económico considerado unánimemente una de las grandes áreas del desarrollo socioeconómico actual y futuro.

En el diseño del itinerario de Tecnologías Interactivas se ha intentado conciliar la dualidad tecnológica y científica ya que el aspecto propiamente tecnológico (Lenguajes de marcas, realidad virtual, herramientas, . . .) correspondería al nivel de la Ingeniería Técnica, mientras que los científicos (Colaboración, aprendizaje, diseño de interfaces, . . .) son propios del segundo ciclo.

En la definición de las asignaturas se han tenido en cuenta el curriculum formal de CHI (Computer-Human Interaction) de ACM y la Taxonomía Informática desarrollada por ACM/IEEE.

Las asignaturas incluidas en el perfil son:

- Sistemas de Interacción Persona Computador. (Obligatoria en el cuarto año)
- Interfaces de usuarios
- Dispositivos físicos para la interacción Persona-Computador
- Multimedia, Hipermedia y Realidad Virtual
- Sistemas para la colaboración
- Sistemas de aprendizaje
- Procesamiento de datos multimedia

Para dar una idea de los contenidos que se imparten en éstas asignaturas, se presenta un resumen del programa de la asignatura obligatoria (9 créditos) que se cursa en el cuarto año, **Sistemas de Interacción Persona Computador**. Debe comentarse que la tercera parte de las horas lectivas se dedica a clases prácticas en laboratorio y el resto a conferencias (magistrales).

Los **Objetivos** de la asignatura son: Presentar las técnicas básicas en la producción multimedia e hipermedia. Desarrollar pequeños proyectos multimedia, y que utilicen realidad virtual. Saber los conceptos fundamentales sobre los sistemas de trabajo en grupo. Presentar los conceptos, componentes, modelos, herramientas y ejemplos importantes de los sistemas de IPC. Desarrollar pequeños proyectos de trabajo colaborativo. Conocer los principios de los sistemas de aprendizaje humano y de máquina así como de los sistemas de aprendizaje colaborativo.

La **Estructura del temario** es como sigue:

- Introducción a los sistemas de Interacción Persona-Computador. : 8 horas
- Sistemas para la colaboración: 12 horas
- Introducción a los sistemas de aprendizaje : 12 horas
- Hipermedia, Multimedia y Realidad Virtual : 20 horas
- Interfaces de usuario: 8 horas

En **las prácticas** se realizan fundamentalmente trabajos de diseño o implementación de métodos, Pero sobre todo las prácticas se orientan a la realización de dos proyectos. El primero será la elaboración de una aplicación de trabajo en grupo o la realización de un

modelo de aprendizaje supervisado. El segundo consiste en la elaboración de algún material hipermedia usando herramientas como Toollbook, Macromedia Director o HTML.

La **Evaluación** contiene también dos actividades; una práctica (que viene de las clases prácticas) y otra conceptual (examen de teoría)

La **bibliografía** que se utiliza es la siguiente:

- B. Shneiderman, *Designing the user interface*, Addison-Wesley, 3ª Ed., 1998.
- J. Preece et al, *Human-computer interaction*, Addison-Wesley, 1994. 1997.
- D. Stampe, *Realidad virtual. Creaciones y desarrollo*, Anaya Multimedia, 1993.
- T. Hardy y Jackson, R.: *Aprendizaje y cognición*. Prentice Hall. 1998
- O'Malley, C *Computer Supported Collaborative Learning*. Springer V., 1995
- Cohen,P. y Feigenbaum: *The handbook of Artificial Intelligence*. Vol 3. Addison-Wesley, 1989.

3 La experiencia docente.

Durante el curso 1999-2000, se completó por primera vez el plan de estudios con alumnos en todos los cursos. La asignatura obligatoria de cuarto año cumple su papel de introducción a la especialidad, si bien su conceptos se presentan de manera inconexa. En los dos cursos en que se ha impartido se ha conseguido combinar adecuadamente el carácter informativo con el énfasis en la práctica, al dejar caer una parte importante del peso de la evaluación sobre los proyectos. Los estudiantes valoran positivamente el conjunto de técnicas que se imparten por su nivel de actualidad y encaje a las exigencias frecuentes del mercado laboral.

En el primer curso en que los alumnos realizaron selección de asignaturas optativas de quinto año, las asignaturas con mas demanda en conjunto, corresponden a las del perfil de Tecnologías Interactivas (120 matriculas). Le siguen Sistemas de Información (105) e Informática Industrial (87). Nuevamente el carácter práctico, la actualización de contenidos y adecuación a las demandas laborales son los aspectos que los alumnos refieren como determinantes.

Conviene destacar que como consecuencia del plan de inversiones de la UCLM para las nuevas titulaciones, la especialidad de Tecnologías Interactivas cuenta con un laboratorio docente especializado con equipos y software que garantizan el trabajo en las prácticas.

Referencias

- [1] E.S.I. (Escuela Superior de Informática. UCLM) "Guia Docente. Curso 1999-2000", Ciudad Real 1999

Curso de Diseño y Desarrollo de Sistemas Interactivos Multimedia

Mariano González, Juan Manuel Cordero, Miguel Toro.

Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Sevilla, Av. Reina Mercedes s/n, 41012 Sevilla
Tlf: 954.552.768 Fax: 954.557.139. e-mail: mariano@lsi.us.es

Resumen

Esta ponencia presenta la experiencia docente en el desarrollo del Curso de Doctorado titulado "Diseño y Desarrollo de Sistemas Interactivos Multimedia", impartido por primera vez en la Universidad de Sevilla durante el curso académico 1999/2000. Se describen los objetivos del curso, la metodología seguida, el temario, los trabajos desarrollados por los alumnos y las herramientas utilizadas. La ponencia concluye con algunas ideas sobre posibles mejoras y extensiones del curso.

Palabras clave: multimedia, diseño de sistemas interactivos

1 Motivación

El Plan de Estudios de Ingeniería Informática vigente actualmente en la Universidad de Sevilla no contempla una asignatura específica de multimedia. Aunque ciertas áreas concretas de la multimedia son tratadas en asignaturas tales como *Procesamiento de Imágenes Digitales*, *Tratamiento de Voz e Imágenes* o *Diseño de Interfaces Hombre Máquina*, no existe una asignatura única que trate todos estos temas de forma conjunta. Debido a ello los alumnos finalizan sus estudios careciendo de una visión global y unificadora de esta vasta área de conocimiento que es la multimedia. Esto motivó la idea de impartir un Curso de Doctorado destinado a proporcionar esta visión.

Los objetivos del curso son los siguientes:

- Proporcionar una visión general del campo de la multimedia, comenzando por los conceptos básicos y siguiendo con el estudio de los distintos tipos de medios, los sistemas multimedia, las aplicaciones presentes y futuras, la multimedia distribuida y la integración de la multimedia con internet.

- Mostrar la situación actual y las futuras direcciones de investigación en el campo de la multimedia.

2 Contenido del curso

El mayor problema a la hora de definir el contenido de este curso es la cantidad de campos que abarca la multimedia. Se puede optar por dar una visión en anchura, sin entrar en excesivo detalle en los temas estudiados, o en profundidad, limitando en total caso el número de temas. Al no existir en el Plan de Estudios una asignatura específica de multimedia se ha optado por la primera opción. Además, por tratarse de un Curso de Doctorado se ha añadido una revisión del estado actual del arte en la investigación en el campo de la multimedia. De esta forma se proporciona a los alumnos una visión global del área y posibles líneas hacia las cuales enfocar su trabajo de investigación. Para definir el temario concreto del curso se procedió a recopilar información bibliográfica, fundamentalmente en forma de libros, artículos en revistas y congresos y recursos en internet, sobre multimedia. En esta búsqueda se encontró que la mayor parte de la información disponible se centra en temas muy concretos, si bien se encontraron algunos libros más generales como [1], [2] y [3] (este último más divulgativo, pero interesante para conocer el mercado de la multimedia). También se encontraron artículos que describen la experiencia en el desarrollo de cursos similares ([4], [5]) y que además proporcionan una interesante documentación sobre el tema. En la página web del curso, <http://www.lsi.us.es/docencia/doctorado/smm.html>, se ofrece un listado con algunos cursos sobre multimedia existentes en internet.

Con toda la documentación recopilada se ha desarrollado un temario que es el que se expone a continuación:

TEMA 1. INTRODUCCIÓN

- Definición
- Historia
- Conceptos generales
- Características de los sistemas multimedia
- Aplicaciones

TEMA 2. MEDIOS Y TIEMPO

- Texto
- Gráficos e imágenes
- Video y animación

- Sonido
- Hipertexto e Hipermedia
- Técnicas de compresión
- Sincronización de los medios en el tiempo

TEMA 3. SISTEMAS MULTIMEDIA

- Arquitecturas de los sistemas multimedia
- Dispositivos multimedia
- Sistemas de almacenamiento
- Redes de comunicaciones
- El software multimedia

TEMA 4. APLICACIONES MULTIMEDIA

- Clasificación de las aplicaciones multimedia
- Aplicaciones locales
- Aplicaciones de comunicación
- Trabajo colaborativo
- Transmisión de información multimedia
- Aplicaciones basadas en servidores multimedia
- Sistemas de Realidad Virtual

TEMA 5. MULTIMEDIA E INTERNET

- Internet
- Hipertexto e Hipermedia
- La World Wide Web
- Páginas web multimedia
- Mundos virtuales (VRML)

TEMA 6. DESARROLLO DE UNA APLICACIÓN MULTIMEDIA

- Diseño de una aplicación multimedia
- Herramientas de autor
- Caso práctico

TEMA 7. FUTURO

- Aplicaciones de la multimedia
- Líneas de investigación en multimedia

En este temario se pueden apreciar tres bloques distintos: un primer bloque donde se explican los conceptos básicos sobre la multimedia y sus aplicaciones (temas 1 a 5), un segundo bloque donde se aborda el diseño práctico de una aplicación multimedia (tema 6) y un tercer bloque donde se exponen temas de investigación actuales en el campo de la multimedia (tema 7).

3 Metodología

El curso se imparte durante un cuatrimestre a razón de 2 horas semanales (lo cual equivale a 3 créditos). Las clases se imparten en un aula de teoría con la ayuda de un ordenador multimedia conectado a un cañón de proyección. No hay pues clases prácticas, debido al enfoque que se ha dado al curso. No obstante, los alumnos deben realizar un trabajo práctico para aprobar el curso. Este trabajo debe ser elegido de entre una lista de temas propuestos y debe ser presentado en público a final de curso.

Los alumnos tienen a su disposición toda la documentación del curso a través de la página web del mismo. También pueden obtener esta misma documentación en un CD-ROM proporcionado por el profesor. Este CD-ROM contiene además una selección de artículos relacionados con los trabajos propuestos, junto con las herramientas necesarias para el desarrollo de los mismos (herramientas de autor multimedia, editores de imágenes, sonido y vídeo, etc.).

3.1 Trabajos realizados por los alumnos

Los trabajos propuestos a los alumnos son de dos tipos distintos:

- a) Realización de un estudio donde se muestre el estado del arte de un campo específico dentro de la multimedia. Los trabajos deben adoptar el formato de una comunicación científica, siguiendo un esquema similar al siguiente:
 1. Introducción
 2. Fundamentos teóricos
 3. Estado del arte
 4. Líneas de investigación abiertas
 5. Bibliografía
- b) Realización de un tutorial sobre un tema relacionado con la multimedia. Este tutorial debe ser realizado mediante una herramienta de presentación o de autor multimedia. En particular, en el tema 6 se explica someramente el funcionamiento de Macromedia Director

Para cada tema propuesto se proporciona una bibliografía básica, así como una lista de fuentes de información sobre el mismo (revistas, congresos, asociaciones, cursos, grupos

de investigación, sitios web, etc). En particular se cuenta con los artículos de las bibliotecas digitales de la ACM (www.acm.org/dl) y del IEEE (www.computer.org). La lista de trabajos propuestos es la siguiente:

Trabajos de tipo a)

- Interfaces de usuario multimedia inteligentes
- Búsqueda de información basada en contenido en bases de datos multimedia: imágenes
- Búsqueda de información basada en contenido en bases de datos multimedia: música
- Búsqueda de información basada en contenido en bases de datos multimedia: vídeo
- Estándar MHEG (*Multimedia and Hypermedia Information Coding Expert Group*) para la especificación de documentos multimedia
- Estándar Hytime (*Hypermedia Time-based Structuring Language*) para la especificación de documentos multimedia
- La metodología de diseño de documentos hipermedia OODHM (*Object Oriented Design of Hypermedia Documents*)
- Compresión de vídeos
- Trabajo colaborativo (CSCW, *Computer Supported Collaborative Work*)
- Uso de VRML en interfaces de usuario multimedia
- Vídeo bajo demanda
- Cine interactivo
- Libros electrónicos
- Bibliotecas digitales
- Presentaciones multimedia en la web: el language SMIL (*Synchronized Multimedia Integration Language*)

Trabajos de tipo b)

- Historia de la multimedia: sistemas pioneros
- Técnicas de compresión
- Sistemas de realidad virtual
- Herramientas de autor multimedia
- Tratamiento digital de imágenes
- Gráficos y animaciones en 3D
- Edición digital de vídeo con *Adobe Premiere*
- Música electrónica
- Dispositivos multimedia
- Tecnología de realidad virtual aplicada al ocio: los parques temáticos virtuales
- La multimedia en el hogar del futuro
- Arte multimedia
- VRML (*Virtual Reality Modeling Language*)

4 Posibles extensiones y nuevas direcciones

La realimentación obtenida por parte de los alumnos de un curso es siempre importante, y más si cabe cuando éste se imparte por primera vez. También hay que considerar las sensaciones experimentadas por el propio profesor al impartir las clases. De todo ello se extraen una serie de conclusiones importantes para futuras ediciones del curso. Por ejemplo, incorporar al curso material desarrollado por los alumnos; atender las sugerencias sobre temas de interés por parte de los alumnos, que por lo general se encuentran ya trabajando y en algunos casos realizando tareas relacionadas de alguna forma con la multimedia; o incorporar nuevas ideas fruto de la incesante evolución que experimenta un área tan dinámica como la multimedia.

Por otra parte, y de cara a una futura revisión del Plan de Estudios de Ingeniería Informática, resulta muy interesante ir desarrollando una propuesta de asignatura sobre multimedia para ofrecer en los nuevos planes. Esta asignatura debería tener un mayor contenido práctico, por lo que la experiencia obtenida en el Curso de Doctorado puede ser muy útil a la hora de plantear los requisitos en cuanto al equipamiento (*hardware* y *software*) necesarios para la nueva asignatura, que sin duda alguna resultaría muy popular entre los alumnos por lo atractivo del tema y por las oportunidades de empleo que ofrece.

Referencias

- [1] J.F. Koegel Buford, "Multimedia Systems", Addison Wesley, 1994.
- [2] F. Fluckiger, "Understanding Networked Multimedia", Prentice Hall, 1995.
- [3] O. Peña de San Antonio, "Multimedia. Edición 2000", Anaya Multimedia, 1999.
- [4] Douglas S. Reeves, "A Course on Multimedia Technology for Computer Science and Computer Engineering Students", Proc. of the 28th SIGCSE Technical Symposium on Computer Science Education, 1997, pp.174-178.
- [5] Edward A. Fox y Linda M. Kieffer, "Multimedia Curricula, Courses, and Knowledge Modules", ACM Computing Surveys, Vol. 27, No. 4, dic 1995.

Hacia un corpus docente virtual común en IPO. Experiencia del Curso Virtual de Introducción a la Interacción Persona-Ordenador

Julio Abascal, José J. Cañas, Miguel Gea, Jesús Lorés†, Manuel Ortega, Luis
Alfonso Ureña, Manuel Velez

† Departamento de Informática e Ingeniería Industrial.

Universidad de Lleida, C/ Jaume II 69, Lleida

Tlf: 973.702.700 Fax: 973.702.702

e-mail:jesus@eup.udl.es

Resumen

En esta ponencia se presenta la iniciativa nacida en el seno de la asociación de edición de un corpus docente común en IPO, que pretende realizar un estudio de cómo debe ser la docencia de IPO y las contribuciones que se deben recoger desde un punto de vista interdisciplinar, utilizando el soporte virtual como medio de compartición de contenidos y como una iniciativa de promover y fomentar la difusión de IPO a toda la comunidad iberoamericana (por el idioma) de "forma libre" y abierta a contribuciones de todos los docentes interesados. Como ejemplo se presenta la experiencia realizada del curso virtual de Introducción a la interacción Persona-Ordenador.

Palabras clave: docencia virtual, interacción persona ordenador

1 Introducción

El nacimiento de la Asociación de Interacción Persona-Ordenador (AIPO) ha supuesto un fuerte impulso a la visualización de esta disciplina como fuerza emergente dentro de la Universidad española. Como toda disciplina universitaria hemos de considerar siempre la docencia y la investigación. Es pues un tema al que le hemos de dedicar un apartado importante dentro de nuestras líneas de trabajo. AIPO supone pues un marco idóneo de debates y contactos sobre los aspectos docentes de la interacción. Dentro de nuestras primeras reuniones surgió la idea de intentar la realización de un corpus docente común. Una propuesta lógica por tanto a partir de estas conclusiones es el uso de Internet como soporte de esta información común. Dentro de estas inquietudes surgió la iniciativa de la

confección de un curso de Introducción a la Interacción Persona-Ordenador basado en Internet.

2 La docencia virtual

La utilización de herramientas virtuales permite basar el proceso de aprendizaje no sólo en las actividades de tipo presencial que se desarrollan en el aula, sino también ofrecer un sistema de formación continua en el que en todo momento, se podrá transmitir información, material, recursos, experiencias.

Para desarrollar formación virtual de una manera efectiva es importante contar, por un lado, con una serie de capacidades técnicas en relación con la web y las herramientas interactivas que ofrece, y por otro, con capacidades pedagógicas y organizativas. La Web es un medio extraordinariamente flexible que se adapta a las necesidades pedagógicas y de comunicación a través de un sistema basado en su simplicidad, ya que no requiere grandes conocimientos. Es por lo que pretendemos utilizar esta plataforma como media para la creación de los contenidos y su utilización ubicua.

3 La docencia de la Interacción Persona-Ordenador. Una justificación

El Informe de ACM/IEEE-CS joint Curriculum Task Force Computing Curricula de 1991 [2] establece nueve áreas temáticas para cubrir la materia de la disciplina de informática.

La interacción Persona-Ordenador es una de las nueve áreas temáticas definidas en el curricula de ACM, aspecto importante en el momento de situar esta docencia en los planes de estudio de las Universidades españolas.

En 1988 el Grupo de Interés Especial en Interacción Persona ordenador (ACM-SIGCHI) puso en marcha un grupo con el objetivo de hacer un diseño curricular[3]. La tarea de este comité fue la de redactar una serie de recomendaciones para educación en IPO. Este comité redactó el documento ACM SIGCHI Curricula for Human-Computer Interaction en 1992, que contiene una serie de recomendaciones para la realización de cursos de IPO.

Para poder cubrir todos los aspectos de la definición y los objetivos, la IPO ha de abarcar una gran cantidad de áreas diferentes, que incluyan distintos aspectos del ser humano y del ordenador: Informática (diseño e ingeniería de las interfases), psicología (teoría y aplicación de los procesos cognitivos y el análisis empírico del comportamiento de los usuarios), sociología y antropología (interacción entre tecnología, trabajo y organizaciones) y diseño industrial (productos interactivos).

Los temas que se escogieron en el curriculum de ACM se derivaron de la consideración de los aspectos interrelacionados de la interacción *Persona-ordenador*: (N) la naturaleza de la interacción, (U) Uso y contexto de los ordenadores, (H) características del ser humano, (C) Ordenadores y arquitectura de la interfase y (D) el proceso de desarrollo. También hay que tener en cuenta la (P) presentación de proyectos y la evaluación.

4 Contenidos del curso

El curso actualmente consta de 13 capítulos, el primer capítulo es una introducción a la interacción Persona-Ordenador, donde se introduce la disciplina, el concepto de interfase y su usabilidad.

El segundo y tercer capítulo trata de estudiar las dos entidades que intervienen en la interacción, el ordenador con el estudio de los dispositivos y el estudio del ser humano desde el punto de vista la interacción.

Los estilos de interacción es un capítulo fundamental donde presentamos una perspectiva de los paradigmas de interacción y estudios comparativos.

A continuación presentamos un estudio de metáforas y su uso y diseño. El soporte al usuario y las guías de estilo permiten completar la formación en modelos de interacción.

La internacionalización es un capítulo poco habitual en la bibliografía, pero que consideramos muy importante en un contexto globalizado de la interacción como el que vivimos.

El siguiente bloque corresponde al apartado de diseño, con un capítulo dedicado a la ingeniería de la interfase, otro dedicado a herramientas de desarrollo, diseño gráfico y estudio de evaluación de las interfases.. Finalmente se dedica un tema a las interfases de trabajo en grupo.

5 El curso virtual

La estructura actual del curso es un sitio web ubicado en el servidor del grupo Griho de la Universidad de Lleida (<http://griho.udl.es/ipo>) que contiene toda la documentación del curso y las transparencias correspondientes a cada capítulo

El diseño del curso sigue una estructura de tres partes tal como se ve en la figura.

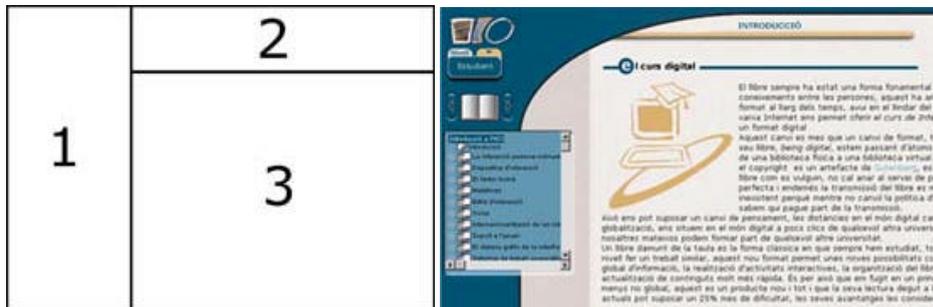


Figura 1. Diseño del libro digital

Una vez el estudiante accede al libro digital, se presenta una hoja dispuesta en tres marcos, a partir de ahora es la disposición de todas las hojas que conforman el curso, su esquema es el siguiente

El marco número 1 corresponde a la parte de navegación por el curso, en ella se encuentran tres diferentes niveles de navegación:

El marco número 2 es el destinado a informar al usuario en donde se encuentra en cada momento.

El marco número 3 es el destinado a la información del curso, en el se presentan todos los datos.

El estudiante tiene a su alcance toda la información del curso dispuesto en un conjunto de temas relacionados todos ellos con la Interacción Persona Ordenador.

El estudiante puede dirigirse a cualquiera de los temas que conforman el curso con la navegación del marco 1, esta navegación es un índice del curso donde se presentan todos los temas y los apartados en que quedan divididos.

6 El diseño del curso

Para la realización del curso hemos estado trabajando en la confección del mismo, un colectivo de profesores de la asociación con experiencia en diversas partes del temario. Psicología, diseño, trabajo en grupo, métodos formales, etc..

El curso permite la inclusión de documentación a través de la modificación del índice, un fichero de tipo texto, que es visualizado por un applet de java, los contenidos han sido discutidos a través de correo electrónico y migrados al formato html, por la facilidad que supone el uso de plantillas.

7 Ejemplos de aplicación. Universidad de Lleida

El curso ha sido aplicado en la docencia presencial de la asignatura de Introducción a la interacción Persona-Ordenador de la Universidad de Lleida.

Esta asignatura está dividida en 2 horas teóricas y dos horas de laboratorio.

Para las clases de teoría utilizamos un aula general para todos los alumnos, debido a no poder reducir el número de alumnos por grupo.

En estas clases utilizamos el ordenador portátil conectado a internet y el cañón proyector y un reproductor de vídeo.

Como base de la presentación se utilizan transparencias basadas en el curso digital. El esquema que seguimos es una presentación del tema, los objetivos a conseguir, una exposición de contenidos del tema y finalmente las conclusiones.

También presentamos el capítulo del libro digital y se visitan enlaces del libro que se consideren idóneos para la presentación.

Como complemento se realizan proyecciones de vídeo relativas a los temas que se presentan.

También se presenta una bibliografía y enlaces a sitios *web* visitables del tema que hemos expuesto.

Como soporte docente el alumno puede consultar el curso digital por internet o bajarlo a su ordenador para tenerlo off-line así como las transparencias.

También se dispone de una copia en papel, en previsión de los problemas que pueda suponer la lectura por pantalla.

8 El proyecto al completo

El libro digital es una parte de un proyecto más completo en el que pretendemos utilizar los medios disponibles en Internet para ofrecer una posibilidad de formación completa al estudiante mediante la posibilidad de utilizar el curso digital, buscar más información a través de los enlaces disponibles, visionar una clase, realizar ejercicios interactivos y autoevaluarse.

9 Conclusiones

La experiencia obtenida en la confección del temario la consideramos muy positiva en la ventaja que supone la creación de contenidos docentes por un grupo de profesores de

diferentes universidades, expertos en aspectos determinados del temario. El uso de Internet por tanto ha permitido su realización a través del uso del correo electrónico, listas de correo y transferencias de fichero. Un aspecto a considerar sería el desarrollo de una herramienta de trabajo en grupo que permitiera la actualización simultánea de materia.

La experiencia docente de uso del curso digital en los dos últimos cursos en la Universidad de Lleida ha sido muy positiva, sobre todo en el último curso por el diseño del curso digital en grupo.

No obstante es interesante comentar la necesidad todavía de imprimir el libro en papel por la dificultad de aprender toda la documentación directamente en la pantalla.

Han quedado opciones abiertas que no se han podido hacer por falta de tiempo como videconferencia de diferentes profesores para compartir la docencia con el profesor presencial.

Referencias

- [1] Gabriel Ferraté. La universidad en el cambio de siglo. Alianza Editorial. 1998
- [2] ACM/IEEE-CS joint Curriculum Task Force Computing Curricula de 1991
- [3] ACM SIGCHI Curricula for Human-Computer Interaction. ACM 1992

Sesión IV

Entornos de Interacción en la Educación

Diseño de libros electrónicos educativos

I. Aedo, P. Díaz†.

† Laboratorio DEI. Departamento de Informática.
Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés (Madrid)
Tlf: 91 624 94 90 Fax: 91 624 91 29 e-mail: {aedo@ia, pdp@inf}.uc3m.es

Resumen

En los últimos años han proliferado los libros electrónicos educativos. Para que dichos libros sean realmente útiles como herramienta de apoyo en el proceso de enseñanza/aprendizaje tienen que cumplir algunas características o requisitos que afectan, fundamentalmente, a la forma en que sus usuarios pueden interactuar con ellos. En este trabajo se presentan precisamente una serie de principios y requisitos que pueden tenerse en cuenta derivados de la experiencia de los autores en el desarrollo y evaluación de distintos libros electrónicos educativos.

Palabras clave: libro electrónico, diseño, interacción, sistemas educativos.

1. Introducción

El libro se ha considerado tradicionalmente como una ayuda indispensable en cualquier proceso de formación, puesto que en el que no sólo se encuentra información sino también un medio de comunicación. En los últimos años han comenzado a aparecer numerosos libros electrónicos educativos, aunque también han surgido alrededor de este concepto múltiples definiciones y taxonomías que pueden consultarse en [1]. A grosso modo, puede decirse que un libro electrónico es, simplemente, un libro que se presenta en formato electrónico. Como tal, el libro electrónico tendrá al menos las mismas características funcionales que el libro de papel y, además, podrá ofrecer servicios más complejos merced a la potencia y facilidades que le ofrece el medio electrónico [2].

El libro electrónico puede "escribirse" empleando diversas herramientas software (v.g., herramientas de autor o de edición de material multimedia), puede almacenarse en diferentes formatos (v.g., PDF, Open eBook, HTML o XML) y puede publicarse utilizando distintas plataformas (v.g., páginas web, CD-ROM o DVD), dependiendo del tipo de libro que se quiera producir y de los recursos de que dispongan sus usuarios. Así pues, ni la plataforma

de desarrollo, ni la de almacenamiento, ni la de distribución definen al libro electrónico. Otro aspecto muy polémico es si el libro electrónico debe obligatoriamente seguir la metáfora del libro tradicional. Mientras algunos autores están de acuerdo con esta afirmación [3, 4], bajo el supuesto de que si se mantiene en la pantalla un modelo de interacción parecido al del libro tradicional se reduce la carga cognitiva requerida para aprender a utilizar el libro electrónico, otros como Nielsen consideran que las metáforas pueden limitar a usuarios y diseñadores, poniendo en duda la familiaridad del usuario con el libro: "¿Por qué seguir imitando la interfaz de los libros cuando nuestros futuros usuarios pasan más tiempo haciendo 'zapping' o jugando con videoconsolas que leyendo?" [5].

En cualquier caso, para que el libro sea realmente útil como herramienta de apoyo en el proceso de enseñanza/aprendizaje tiene que cumplir algunas características o requisitos que afectan, fundamentalmente, a la forma en que sus usuarios pueden interactuar con él. Éste es el objetivo del presente trabajo, en el que se analizarán algunos aspectos a tener en cuenta en el diseño libros electrónicos educativos con el fin de mejorar su utilidad y su "usabilidad". Estos aspectos son producto de la experiencia de los autores de este trabajo en el desarrollo y evaluación de libros electrónicos educativos, entre los que cabe destacar los siguientes: CESAR, que es un libro que ayuda a aprender los lenguajes signado y escrito a niños con deficiencias auditivas [6]; CIPP, un libro para aprender Pascal como primer lenguaje de programación [7]; y Now-Meta, un libro destinado a ayudar a las mujeres con baja cualificación a adquirir los conocimientos propios del Graduado Escolar [8].

2. El libro electrónico educativo y el proceso de interacción

Un libro electrónico educativo deberá incluir información adecuada, completa, bien organizada y fácilmente localizable y accesible como ocurre en los libros de papel. Además, deberá dar soporte a actividades formativas, distintas de la lectura secuencial, que refuercen el proceso de aprendizaje, tales como ejercicios que inviten al estudiante a utilizar sus conocimientos en situaciones concretas.

Estos y otros servicios, tales como la comunicación entre estudiantes o con el profesor, pueden mejorarse haciendo uso de las facilidades que ofrece el entorno electrónico pero, para ello, deben diseñarse cuidadosamente y poniendo especial atención al proceso de interacción subyacente a cualquier entorno formativo. Dicho proceso de interacción, que se encuentra resumido en la figura 1, involucra, normalmente, a dos tipos de agentes capaces de tomar iniciativas, los profesores y los alumnos, que se comunican bien utilizando como mediador el material didáctico o de forma directa. El profesor prepara un material didáctico, imparte sus clases tratando de motivar al alumno, aclara dudas y lleva a cabo un seguimiento del progreso de sus alumnos para corregir, si es preciso, el ritmo seguido o el

método didáctico. El alumno, por su parte, participa en las clases siguiendo las explicaciones, realizando actividades y planteando dudas. Estas labores pueden ser parcial o totalmente soportadas por el medio electrónico.

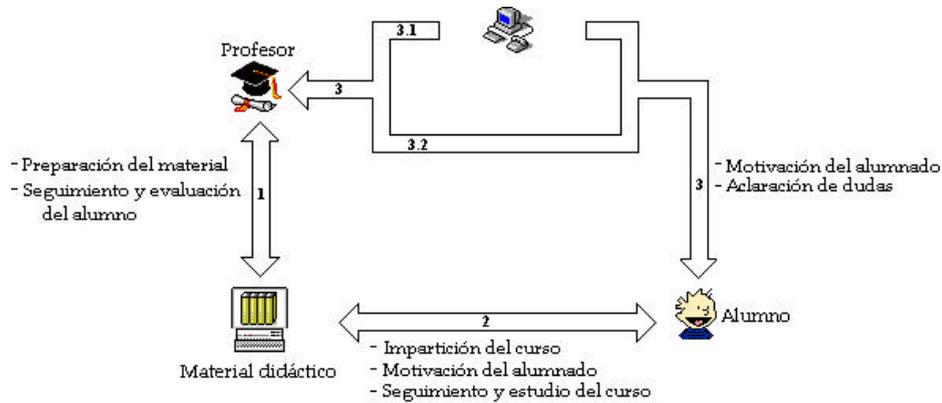


Figura 1: El proceso de interacción en un entorno de enseñanza/aprendizaje

En general, un libro electrónico educativo deberá al menos ofrecer el material didáctico en formato digital, de manera que tanto el profesor pueda crearlo como el alumno utilizarlo y, además, sería conveniente que se diera el soporte adecuado para seguir los progresos realizados por el alumno, tanto por parte del profesor como del propio estudiante. Además, el material didáctico, que ahora puede ser multimedia, hipertexto e interactivo, debe motivar al alumno, invitándole a abandonar actitudes pasivas y ofreciéndole refuerzos positivos. Estos tipos de interacción son los reflejados en las flechas 1 y 2 de la figura 1.

Finalmente, la comunicación directa entre alumno y profesor (representada en la flecha 3 de la figura 1) puede o no llevarse a cabo a través del medio electrónico (flechas 3.1 y 3.2, respectivamente) e, incluso, puede optarse por una solución mixta en la que se combinen medios tradicionales de comunicación con los soportados por el ordenador.

3. Diseño de libros electrónicos educativos

Como se mencionó en el apartado anterior, dos son los tipos de usuarios de los libros electrónicos educativos: el alumno, que utilizará el libro para adquirir un determinado conocimiento, y el profesor, que recurrirá a él como herramienta de apoyo a su método didáctico. Existen algunos aspectos que deben tenerse en cuenta para satisfacer las necesidades de ambos tipos de usuarios a través de un conjunto de mecanismos de interacción completos y fáciles de utilizar.

3.1 Requisitos desde el punto de vista del estudiante

El estudiante debe poder usar el libro para desarrollar actividades de aprendizaje de naturaleza diversa. Así, se pueden considerar actividades pasivas, dirigidas, explicatorias, activas, creativas o reactivas [9], como se resume en la Tabla 1. La inclusión de varios tipos de actividades incrementa la utilidad del sistema puesto que cada una de ellas tiene un objetivo didáctico diferente y, además, ofrece una mayor riqueza en el proceso de interacción con el libro electrónico que favorece la participación del estudiante.

| Tipo | Descripción | Ejemplo |
|---------------|--|--|
| Pasivas | Tratan al alumno como un sujeto pasivo que sólo recibe información | Presentación multimedia |
| Dirigidas | Muestran al alumno el camino a seguir. | Lectura secuencial |
| Explicatorias | Tienen carácter aclaratorio con respecto al tema tratado | Ejemplos |
| Activas | El alumno participa resolviendo una tarea | Ejercicios interactivos |
| Creativas | El alumno participa introduciendo nueva información en el sistema | Notas personales Preguntas abiertas |
| Reactivas | Provocan una reacción en el alumno | Navegación libre |

Tabla 1: Tipos de actividades formativas

Por ejemplo, en el caso de Now-Meta cada una de las lecciones que componen el curso combinan las presentaciones multimedia prácticamente pasivas (aunque son iniciadas y paradas por el usuario), con la navegación libre y secuencial, los ejemplos ilustrativos y una gran variedad de tipos de ejercicios interactivos. En este último punto, es interesante que no todos los ejercicios respondan a un mismo modo de interacción ya que se corre el riesgo de provocar el aburrimiento del alumno. Así, una de las principales características de Now-Meta es la gran variedad de tipos de ejercicios que ofrece a sus usuarios, que incluyen formas de interactuar tan diversas como son el arrastrar objetos por la ventana (ya sea para clasificarlos o para relacionarlos), dibujar, escribir o seleccionar objetos. CIPP ofrece, además, actividades creativas como son la posibilidad de añadir de notas personales en cada una de las páginas del libro o la inclusión de nuevos programas a la lista de los ya proporcionados en el libro.

El libro electrónico también puede tener en cuenta que cada alumno tiene unas motivaciones y condicionamientos que hacen que desarrolle su propio modo de aprender o estilo de aprendizaje [3, 9]. En este sentido, los sistemas de aprendizaje adaptativos pueden emplear diversas técnicas para ajustar los contenidos, el ritmo y orden de presentación de los contenidos y de las actividades de aprendizaje o las estrategias de resolución de ejercicios a las necesidades de cada estudiante. Además, el estilo de aprendizaje de cada alumno no es

fijo, sino que puede variar en función de diversas circunstancias tales como el estado de ánimo, el tipo de material utilizado o el punto en concreto en que se encuentra en el proceso de adquisición de conocimiento. Por ejemplo, en el caso del libro electrónico CESAR, cada ejercicio tiene una meta didáctica concreta aunque puede resolverse utilizando distintas estrategias. Para conseguir este objetivo, un ejercicio se divide en los objetos de información involucrados, definidos a través de una serie de atributos, y la forma en que debe resolverse que, a su vez, se estructura en tres procesos: enunciado, interacción y solución. De esta forma,, utilizando los mismos objetos de información se puede asignar a cada alumno aquella estrategia que más se ajuste a su estilo de aprendizaje [10].

Finalmente, el tipo de conocimiento involucrado y las características de los alumnos, puede ser necesario trabajar en grupo, en cuyo caso habrá que construir sistemas de aprendizaje colaborativo. En este caso, el libro deberá ofrecer soporte a la definición y gestión de grupos de trabajo, de roles dentro del grupo y de protocolos de comunicación.

3.2 Requisitos desde el punto de vista del profesor

Si se pretende integrar el libro electrónico en el aula, es preciso promover una actitud positiva por parte del profesor [11], de manera que éste contemple los nuevos medios como herramientas de ayuda y no como sustitutos. Así por ejemplo, CIPP fue evaluado durante un curso en el que se combinaba el empleo del libro electrónico con otros recursos didácticos tradicionales, tales como las clases magistrales, y se obtuvieron resultados bastante satisfactorios, tanto desde el punto de vista de los profesores como de los alumnos [12]. Los entornos de enseñanza/aprendizaje deben ser, en consecuencia, flexibles y promover la integración de diversas herramientas y técnicas educativas. Además, el libro electrónico como tal puede proporcionar mecanismos que permitan adaptar el material didáctico a los métodos y necesidades del profesor.

También pueden necesitarse mecanismos para seguir el progreso de los alumnos, ya sea de forma individual o como grupo. Así, el profesor debe tener medios para acceder a los datos sobre un alumno específico, a los de un grupo específico e incluso obtener estudios comparativos que relacionen los progresos de los alumnos de un mismo grupo. Estos mecanismos ayudan a detectar problemas en el material didáctico o en el ritmo de impartición del curso, de manera que si están disponibles desde el principio, el profesor puede adaptar dinámicamente el método didáctico a las exigencias de sus alumnos.

3.3 Requisitos de comunicación

En algunos casos deben incluirse herramientas que faciliten la comunicación directa entre alumnos y profesores. Dicha comunicación puede llevarse a cabo de distintas formas, dependiendo de cómo se produce en el tiempo y de cuántos agentes participan en ella.

Con respecto al tiempo, la comunicación puede ser sincrónica o asincrónica, dependiendo de la disponibilidad de los agentes involucrados así como del tipo de mensaje. Así, la resolución de dudas puede implementarse a través de medios sincrónicos, textuales (v.g., charlas) o multimedia (v.g., videoconferencia), con el fin de favorecer un diálogo más fluido aunque esto supone una cierta disponibilidad que no se requiere con medios como el correo electrónico. También se pueden utilizar mecanismos asincrónicos, tales como los tableros de discusión o de preguntas frecuentes, en los que los alumnos suelen plantear sus dudas con más libertad y, además, pueden encontrar respuestas a sus dudas antes de plantearlas.

También hay que tener en cuenta qué agentes participan, de que forma lo hacen y en qué número. Así, lo más habitual es incluir mecanismos para dar soporte a una comunicación bidireccional entre un profesor y uno o más alumnos, aunque también se podría pensar en ampliarlos para dar soporte a la comunicación entre dos o más alumnos, fomentando las charlas y foros de discusión.

3.4 El proceso de diseño del libro electrónico

Tras haber participado en el desarrollo y evaluación de algunos libros electrónicos, los autores de este trabajo han constatado que no existe un paradigma educativo universalmente válido. El tipo de conocimiento y las características del alumno influyen en la aproximación a seguir. Así por ejemplo, la navegación libre, tan defendida por muchos autores, puede derivar en pasividad e indecisión, pues no todos los alumnos quieren aprender por exploración sino que prefieren que el profesor les marque una secuencia. De hecho, este fue uno de los resultados más llamativos obtenidos de la experiencia de evaluación de CIPP [12]. Del mismo modo, tampoco puede afirmarse categóricamente que seguir la metáfora del libro tradicional asegure que la interfaz resultante va a ser más intuitiva o fácil de usar, ya sea porque está muy desvirtuada o porque el usuario no espera que pueda trasladar sus conocimientos sobre el funcionamiento del libro de papel al entorno electrónico, donde presupone que tiene que desarrollar unas habilidades específicas para aprender a utilizar la aplicación.

En consecuencia, si se quiere construir un libro electrónico educativo es imprescindible llevar a cabo un proceso de desarrollo centrado en el usuario. En este sentido, dos son los paradigmas más habituales a la hora de definir el ciclo de vida del proceso de desarrollo: el diseño iterativo [13] y el ciclo de vida en estrella [14]. Mientras que en el primer caso se llevan a cabo ciclos de análisis-diseño-prototipado-evaluación hasta alcanzar una aplicación

que se considera válida desde el punto de vista del usuario, en el segundo no se indica qué orden hay que seguir sino que se establece que la evaluación es el eje del desarrollo y puede llevarse a cabo en las distintas etapas del desarrollo. Así por ejemplo, durante la especificación de requisitos pueden emplearse técnicas de evaluación que permitan comprender mejor el problema a solucionar, el entorno en que se produce y el conocimiento que del mismo tienen sus usuarios, mientras que durante el diseño se probarán si las soluciones y mecanismos implementados son los adecuados.

En cualquier caso, la evaluación del sistema se considera una actividad básica en el desarrollo de un sistema interactivo, ya que es el único medio eficaz de comprobar si los mecanismos de interacción son realmente válidos, útiles y utilizables. Recomendaciones sobre cómo llevar a cabo la evaluación de libros electrónicos pueden encontrarse en [15].

Además, el equipo de desarrollo debe ser multidisciplinar en el que se cuente con informáticos, pedagogos con experiencia en educación asistida por ordenador, especialistas en interfaz de usuario y en el tratamiento de contenidos multimedia y usuarios.

4. Conclusiones

En este trabajo hemos presentado algunos principios de diseño que pueden tenerse en cuenta a la hora de desarrollar un libro electrónico educativo. Estos principios están orientados, fundamentalmente, a incrementar la utilidad y "usabilidad" de dichos libros, teniendo en cuenta cómo se lleva a cabo el proceso de interacción propio de un entorno educativo y que tratan de satisfacer las necesidades de los dos principales agentes de este proceso: el alumno y el profesor.

También se apunta como aspecto básico llevar a cabo un desarrollo centrado en el usuario, en el que se utilice cualquier método de evaluación (v.g., analítico, experto o experimental) durante cualquiera de las fases del ciclo de vida, de manera que se puedan entender mejor las necesidades del usuario, ya sean alumnos o profesores, y se plasmen en soluciones de diseño útiles y utilizables.

Referencias

- [1] P. Díaz, N. Catenazzi, I. Aedo, "De la Multimedia a la Hipermedia". Ed. RA-MA, Madrid, 1996.
- [2] P. Barker, Interactive Electronic Books. *Interactive Multimedia* 2(1), pp. 11-28, 1991.
- [3] L. R. Reynolds y S. J. Derose, Electronic books. *Byte*, 17 (6), Junio, pp. 263-268, 1992.
- [4] I. Benest, An alternative Approach to Hypertext. *Educational and Training Technology International*, 28 (4), Noviembre, pp. 341-346, 1991.
- [5] D. Gentner y J. Nielsen, "The Anti-Mac Interface". *Communications of the ACM*, 39(8), Agosto, 1996.

- [6] I. Aedo, I. P. Díaz, y N. Catenazzi, A model for hypermedia learning environments based on electronic books *Association for Learning Technology Journal (ALT-J)*, 5(3), pp. 4-21, 1997.
- [7] L. Sommaruga, N. Catenazzi, P. Díaz, I. Aedo, A. Berlanga, "Curso interactivo de programación en PASCAL". Ed. McGraw-Hill Interamericana, Madrid 1997.
- [8] I. Aedo, P. Díaz, F. Panetsos, M. Carmona, S. Ortega y E. Huete, "A hypermedia Tool for Teaching Primary School Concepts to Adults". *IFIP WG 3.3 Working Conference "Human Computer Interaction and Educational Tools"*. Sozopol (Bulgaria). Mayo, pp. 180-188, 1997.
- [9] L. Allinson y N. Hammond, *Learning Support Enviroments: Rationale and Evaluation*. *Computers & Education*, 15 (1-3), pp. 137-143, 1990.
- [10] P. Díaz, I. Aedo, N. Torra, P. Miranda y M. Martín, "Meeting the needs of teachers and students within the CESAR Training System". *British Journal of Educational Technology*. 29 (1), pp. 35-46, 1998.
- [11] A. Brown, *Processes to support the use of information technology to enhance learning*. *Computers & Education*, 22 (1/2), pp. 145-153, 1994.
- [12] I. Aedo, P. Díaz, C. Fernández, G. Muñoz y A. Berlanga, "Assessing the utility of an interactive electronic book for learning the Pascal programming language". Aceptado para publicación en *IEEE Transactions on Education*.
- [13] J. Rubin, "Handbook of usability testing". John Wiley & Sons. Inc. Nueva York, 1994.
- [14] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, "Human-Computer Interaction". The Open University. Addison Wesley, 1994.
- [15] N. Catenazzi, I. Aedo, P. Díaz. y L. Sommaruga, "The evaluation of electronic books: Guidelines from two practical experiences". *Journal of Educational Multimedia and Hypermedia*. 6(1), pp. 91-114, 1997.

Organización de objetos del sistema TANGOW: creación y seguimiento de cursos adaptativos a través de Internet

R. Carro, R. Moriyón, E. Pulido, P. Rodríguez

Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
Cantoblanco, 28049 Madrid

Tlf: 913.482.283 Fax: 913.482.235 e-mail: plar.rodriguez@ii.uam.es

Resumen

En este trabajo se presenta la organización de los objetos básicos que utiliza el sistema TANGOW, Task-based learNer Guidance On the Web, para la creación y seguimiento de cursos adaptativos a través de Internet. El conjunto de estos objetos, denominados tareas y reglas docentes, constituye la estructura adaptativa de cada curso y, junto con los contenidos definidos para las tareas docentes, permite su generación, en tiempo de ejecución, para cada estudiante. Las características que permiten adaptar el curso pueden estar relacionadas tanto con el perfil del estudiante, como con su interacción con el sistema durante el curso.

Palabras clave: Enseñanza Adaptativa, Multimedia, Internet, Herramientas

1 El sistema TANGOW

TANGOW, Task-based learNer Guidance On the Web, es un sistema para el desarrollo de cursos a través de Internet [1] [2], que permite la adaptación de estos cursos a factores que los diseñadores determinan que son significativos, y a las preferencias de los estudiantes. Entre los factores que los diseñadores de un curso pueden considerar que son significativos se encuentran características propias de cada estudiante, directamente relacionadas con su perfil, y efectos de sus interacciones con el sistema, por ejemplo, el porcentaje de ejercicios realizados correctamente en algún tema. Respecto a las preferencias de los estudiantes, en la actualidad el sistema permite que indiquen cuál es la estrategia de aprendizaje que desean utilizar durante el desarrollo del curso: teoría antes de prácticas o

viceversa. Teniendo en cuenta la estrategia preferida, el sistema adapta en lo posible el orden de presentación de contenidos teóricos y prácticos.

A continuación se describen los antecedentes y la forma en que se organizan los distintos tipos de objetos que describen los cursos de TANGOW, así como su relación con los diseñadores, que los crean, y los estudiantes, que los utilizan. En las páginas del sistema se pueden encontrar información más detallada [3].

2 Antecedentes

El trabajo descrito en este artículo está directamente relacionado con trabajos de investigación en las áreas de Interfaces de Usuario y Sistemas Tutores adaptativos.

Las técnicas basadas en restricciones, utilizadas en sistemas como Amulet, [4], están entre las primeras que permiten el desarrollo de interfaces adaptativas. En estos sistemas, algunos atributos de partes de la presentación se actualizan automáticamente en función de los valores de los atributos de otras partes de la misma. Para conseguir un grado mayor de adaptatividad, en el que los tipos de componentes presentes en las presentaciones, e incluso su presencia, dependen de determinados datos asociados a la aplicación, fue necesario el desarrollo de nuevas técnicas de modelización, [5]. Posteriormente, [6], se introdujeron técnicas semejantes en presentaciones en HTML, por lo tanto accesibles a través de la web.

Paralelamente, P. Brusilowski propuso un potente mecanismo para el desarrollo de sistemas de hipertexto sensibles al contexto, [7], que posteriormente utilizó para desarrollar sistemas tutores adaptativos. TANGOW representa un avance en la capacidad de adaptación de los cursos desarrollados al permitir la generación dinámica de lecciones y sesiones de prácticas de forma dependiente del desarrollo del proceso de aprendizaje y del perfil del alumno. Para ello utiliza un sistema de representación de tareas que representan tanto acciones que el usuario debe llevar a cabo como objetivos didácticos que se plantea el sistema tutor.

También dentro del ámbito de las Interfaces de Usuario, la modelización de tareas de usuario ha permitido en los últimos años añadir con relativa facilidad a las aplicaciones interactivas sistemas que hacen un seguimiento de las acciones del usuario, reaccionando dinámicamente en función de éstas. Estas técnicas han permitido el desarrollo de herramientas para la generación automática de sistemas de ayuda y tutores acerca del manejo de interfaces, [8].

3 Estructura y contenido de los cursos en TANGOW

Las posibilidades de adaptación del sistema se apoyan en los dos tipos de elementos que componen los cursos, Curso-S y Curso-C, que definen, respectivamente, la estructura y el contenido de cada curso.

2.1 La estructura: Curso-S

Son dos los tipos de objetos que constituyen el conjunto *Curso-S* para cada curso: las tareas y las reglas docentes. Las reglas describen la forma en que se debe desarrollar cada unidad a partir de la descomposición de la estructura del curso en tareas docentes. La función de cada una de estas tareas determina cuál es su tipo, pudiéndose definir tareas de teoría (explicativas), prácticas (de ejercicios) o de ejemplos. Cualquier tarea definida puede descomponerse en subtareas, cuyo modo de secuenciación se establece mediante reglas docentes. Mediante las reglas, el diseñador puede indicar si es conveniente que todas las subtareas se realicen estrictamente en el orden indicado, si se pueden realizar en cualquier orden, y también, si se deben realizar todas las subtareas o basta con que el estudiante realice una de ellas. El diseñador del curso puede establecer varias descomposiciones alternativas, estableciendo también bajo qué condiciones se activará cada una de las reglas. Estas condiciones de activación pueden estar asociadas tanto a los parámetros estáticos de cada estudiante reflejados en su perfil (edad, conocimientos previos), como a parámetros dinámicos relacionados con la forma en que éste sigue el curso (ejercicios realizados, tiempo empleado, etc.).

En el diseño de cada curso es necesario indicar cuál es la tarea principal, entendida como aquella tarea que permite que el estudiante inicie el curso. En principio, ésta es la única tarea que se puede asegurar realizarán todos los estudiantes, ya que su descomposición en subtareas, así como la descomposición de todas las demás subtareas, dependerá de cuáles sean las reglas que se activen en cada caso. La selección de la siguiente regla activable o, lo que es lo mismo, del siguiente conjunto de tareas que se propondrá al estudiante y de la forma en que se encadenarán, se realiza cada vez que se da por concluida una tarea compuesta. De esta forma es posible incluir entre las condiciones de activación los parámetros dinámicos que se obtienen de la interacción del estudiante con el sistema.

El efecto de la selección dinámica de las reglas docentes y, por lo tanto, de las tareas, es que el itinerario que sigue cada estudiante puede ser diferente, y se adaptará a sus

características personales en aquellos aspectos que hayan sido decididos por el diseñador del curso [4].

2.2 Los contenidos: Curso-C

El anterior conjunto *Curso-S* de tareas y reglas docentes tiene como principal objetivo definir el curso desde el punto de vista estructural, articulándolo de la forma más conveniente para cada estudiante. Sin embargo, no es necesario que la sucesión de las tareas docentes se haga evidente para los estudiantes. Desde su punto de vista, el curso se presenta como una sucesión coherente de páginas HTML con las que interactúan, visualizando los diferentes elementos hipermedia que las componen y respondiendo a las preguntas propuestas.

La generación de las páginas HTML que se presentan finalmente a los estudiantes también es dinámica, ya que no existe ninguna página como tal en el sistema. Cada página se genera en el momento que se va a presentar al estudiante a partir de los elementos hipermedia individuales que constituyen el conjunto de objetos *Curso-C*. La lista de los identificadores de los elementos que se utilizarán para cada página se obtiene de la definición de la tarea que se ejecuta en ese momento. Además, en aquellos casos en que la tarea actual se descomponga en subtareas cuya secuenciación pueda ser elegida por el estudiante, se incluyen en la página los enlaces necesarios para realizar cada una de las subtareas posibles. En todos los casos, la composición final de las páginas se ciñe a un modelo estético definido para el curso, y que permite que todas las páginas presenten un aspecto uniforme.

2.3 Los procesos de creación y ejecución de cursos

En la figura 1 se muestra la arquitectura de TANGOW desde el punto de vista de los objetos utilizados por el sistema. Los objetos *Curso-S*, tareas y reglas docentes, se definen a través de una aplicación que actúa como herramienta para los autores de los cursos. Esta herramienta permite que los diseñadores determinen el conjunto de tareas y reglas que forman el curso, definiendo para todas ellas los parámetros que las caracterizan [5]. En esta etapa de diseño, se ha utilizado como formalismo de representación de tareas y reglas el lenguaje HTML. La herramienta de autor permite verificar la coherencia del curso, detectando algunos errores frecuentes, tales como tareas no definidas y que son referenciadas en alguna regla, o tareas definidas y no utilizadas. Una vez el curso ha sido verificado, el conjunto *Curso-S* se compila, almacenándolo en una base de datos. Durante la

ejecución del curso, el sistema utiliza la versión compilada para generar los itinerarios para cada estudiante, aunque la versión HTML se mantiene por razones de mantenimiento y accesibilidad. El conjunto de los elementos hipermedia que se utilizan para generar los contenidos de cada página, *Curso-C*, es único, y puede ser modificado sin necesidad de actualizar la estructura del curso en aquellos casos en que se desee cambiar un texto, una imagen, etc. Si se desea eliminar, o añadir, alguna de las referencias a los contenidos asociados a una tarea, entonces sí sería necesario modificarla utilizando la herramienta del diseñador.

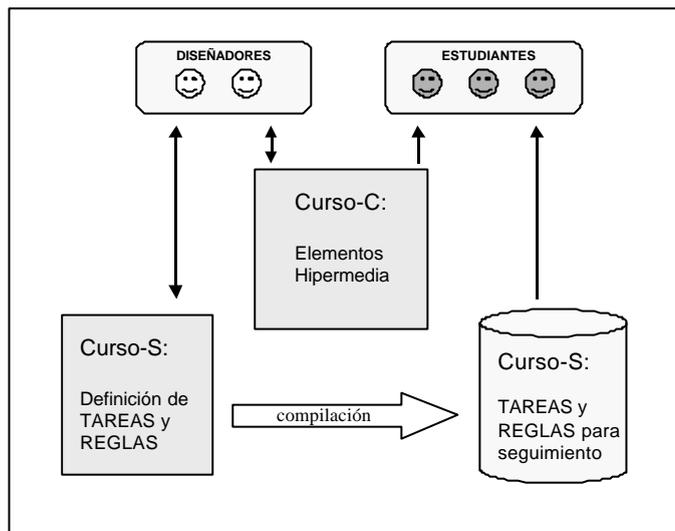


Figura 1. Objetos del sistema TANGOW

4 Conclusiones y agradecimientos

El sistema TANGOW permite la creación y seguimiento de cursos adaptativos a través de Internet. En TANGOW, la estructura y el contenido de cada curso se gestionan de forma independiente, facilitando los procedimientos de adaptación a cada estudiante, así como las acciones de mantenimiento de la información.

La estructura de cada curso, *Curso-S*, se compone de un conjunto de objetos del sistema, tareas y reglas docentes, mientras que los contenidos, *Curso-C*, están formados por los

elementos multimedia que se utilizan para generar dinámicamente las páginas HTML que se presentan a los estudiantes. Para el seguimiento de los cursos, la estructura de tareas y reglas se compila, almacenándola en una base de datos para mejorar los tiempos de acceso.

Este trabajo está financiado parcialmente por la Comisión Interministerial de Ciencia y Tecnología, CICYT, proyecto TEL97-0306.

5 Referencias

- [1] P. Brusilovsky, P. Miller, "Web-based Testing for Distance Education", Actas de la conferencia WebNet-99, AACE, pag. 149-154, 1999.
- [2] P. de Bra, L. Calvi. "AHA: A Generic Adaptive Hypermedia System", Second Workshop on Adaptive Hypertext and Hypermedia, Ninth ACM Conference on Hypertext and Hypermedia, pag. 5-11, 1998.
- [3] Páginas Web del TANGOW, "Task-based Adaptive learner Guidance On the Web", accesibles en: <http://www.ii.uam.es/esp/investigacion/tangow/present.html>
- [4] B. A. Myers et al, "The Amulet Environment: New Models for Effective User Interface Software Development", IEEE Transactions on Software Engineering, v. 23, nº 6, 1997, pp. 347-365.
- [5] P. Szekely, P. Sukaviriya, P. Castells, J. Muthukumarasamy and E. Salcher, "Declarative Interface Models for User Interface Construction: The Mastermind Approach", en "Engineering for Human-Computer Interaction", L. Bass and C. Unger (eds), Chapman & Hall, 1996, pp. 120-150.
- [6] F. Saiz, P. Szekely, D. Devang, "Customized Web-based Data Presentation", proc. World Conference on the WWW, Internet and Intranet (WebNet'98), Orlando (Florida), 1998, pp. 792-798.
- [7] Brusilovsky, P., Schwarz, E. & Weber, G. (1996). A tool for Developing Adaptive Electronic Textbooks on WWW. , Proceedings of WebNet-96 of the AACE, Boston, 1996.
- [8] F.García, J.Contreras, P.Rodríguez, R.Moriyón, "Help generation for task based applications with HATS", en "Engineering for Human-Computer Interaction", P. Dewan, ed., Kluwer, 1999, pp. 149-168.
- [9] R.M. Carro, E. Pulido, P. Rodríguez, "Dynamic generation of adaptive Internet-based courses", Journal of Network and Computer Applications, vol. 22, pag- 249-257, 1999.
- [10] R.M. Carro, E. Pulido, P. Rodríguez, "Designing Adaptive Web-based Courses with TANGOW", en Cumming, G., Okamoto, T., Gomez, L. Advanced Research in Computers and communications in Education (IOS Press), vol. 2, pag. 697-704, 1999.

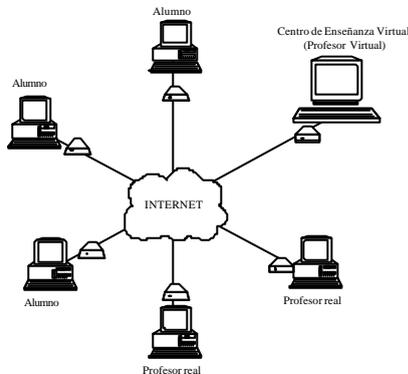
Nuevas tecnologías aplicadas a la educación mediante la interacción persona-ordenador: Proyecto experimental CIEV.

J. García Galera, F. Luis Cintrano, J.D. Caravaca
ETSI. Informática, Universidad de Granada
mail: jgarcia galera@ole.com

1. El proyecto CIEV

La llegada de la informática a los centros educativos españoles ha producido una nueva mentalidad en el profesorado y en las administraciones educativas cuyas consecuencias se hacen más patentes con la entrada del siglo XXI. De esta idea nace el proyecto CIEV (Centro Iberoamericano de Enseñanza Virtual).

El proyecto CIEV integra un servidor de Internet cuyo objetivo principal será la implementación de un *Aula Virtual gestionada por un agente inteligente que evalúe de forma personalizada a los alumnos matriculados en los distintos cursos* que se impartan a través de este centro de enseñanza virtual. Se desea crear un sistema experimental aplicado al ámbito educativo debido al gran impacto que están sufriendo las nuevas tecnologías educativas en la actualidad.



Para ello el sistema hace uso de un *generador de cursos automático* para que tanto el personal relacionado con la informática como aquellas personas menos introducidas en este

mundo puedan generar e implantar cursos en este sistema de forma rápida, sencilla e intuitiva.

Una de las ventajas más importantes de este sistema de docencia es que un conjunto de alumnos matriculados en el mismo curso podrán ver los mismos conceptos pero con la particularidad de que verán la explicación del mismo concepto explicada de maneras distintas. Para ello, el tutor inteligente gestionará internamente dos tipos de conceptos (abstractos y específicos). La idea básica es la siguiente: denominamos “*concepto abstracto*” a un concepto genérico que deberán aprender todos los alumnos matriculados en un curso (un ejemplo de concepto abstracto podría ser la “definición de informática”). A su vez cada concepto abstracto consta de uno o varios “*conceptos específicos*”. Los conceptos específicos contendrán la explicación del concepto abstracto (“definición de informática”), además de contener información adicional que indique la complejidad de la explicación. De esta manera, cuando un alumno entre en alguno de los cursos en los que se encuentre matriculado, el tutor virtual buscará dónde se quedó la última vez que entró, obtendrá el concepto abstracto que debe mostrar a continuación y, en base al seguimiento del alumno, el tutor virtual seleccionará el concepto específico asociado al concepto abstracto. Lo que visualizará finalmente el alumno será el concepto específico seleccionado por el tutor virtual (el nivel de complejidad del concepto específico habrá sido seleccionado por el agente inteligente en base al seguimiento del alumno y la heurística utilizada).

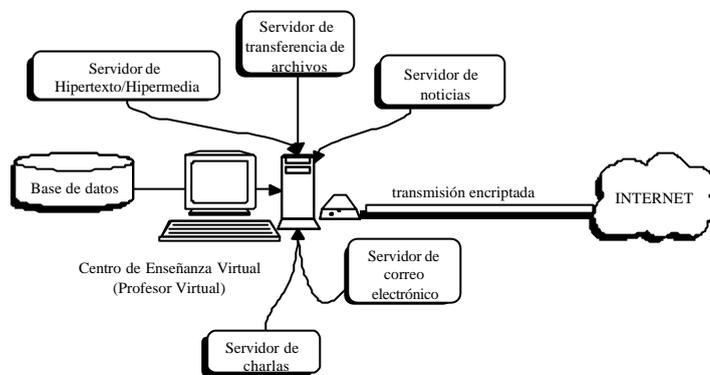
Además del tutor virtual que gestione los cursos implantados por el generador de cursos automático, este entorno virtual esta montado sobre páginas Web utilizando herramientas automáticas para la gestión de los cursos y búsqueda de información, de manera que el alumno pueda tener un ambiente de estudio adecuado, así como una fácil accesibilidad a todo tipo de material docente relacionado con la temática del curso en el que éste se encuentre matriculado.

El servidor de Internet debe apoyar a la docencia mediante la utilización de un *sistema de Noticias* donde el alumno podrá obtener información actualizada de la materia que este cursando, así como las últimas noticias relacionadas con alguna asignatura determinada. También debe proporcionar un sistema de transferencia de archivos para poder bajar de la red documentación y apuntes del curso en varios formatos. El sistema de transferencia de archivos nos permitirá la creación de *bibliotecas virtuales*, fonotecas, videotecas, etc... que complementen la formación del alumnado en la materia cursada.



El sistema inteligente que imparte cursos a través de Internet no es suficiente para conseguir las metas perseguidas (conseguir un sistema de apoyo a la educación que pueda ser implantado con seriedad en la sociedad) debido a que es necesario la interacción entre personas (entre alumnos y entre alumno-profesor real). Aunque esta interacción no será posible de manera presencial sí podrá realizarse mediante algunas herramientas proporcionadas por el CIEV que se comentan a continuación.

La aplicación que completará este centro educativo virtual será la implantación de un *sistema de Charlas para la gestión de tutorías telemáticas*. De esta manera, cada alumno que curse una asignatura en este Centro Virtual tendrá asignado un tutor (profesor real) con el cual podrá comunicarse en tiempo real a través de la red, para plantear dudas o problemas no resueltos, en las horas de tutorías preestablecidas. Para ello el alumno solamente tendrá que conectarse a Internet y mediante algún software determinado (cliente de Chat o IRC) acceder al foro que tenga el nombre del curso que esté realizando.



Otro aspecto importante es la *interfaz de usuario*, ya que el alumno al acceder al CIEV debe de encontrarse totalmente motivado y con ganas de aprender. Al conectarse debe encontrar en el sistema un centro de enseñanza real, donde exista una secretaría para obtener información de los cursos y poder matricularse, así como un buen sistema de ayuda. Es indispensable un buen diseño de la interfaz para que los alumnos no se pierdan dentro de la Web y puedan aprender la temática en la que se matriculen sin que la navegación a través de los distintos módulos del Aula Virtual sea un inconveniente. La interfaz debe ser intuitiva, fácil de manejar y con buenos sistemas de búsqueda para evitar confundir y perder al alumnado entre las páginas de hipertexto, lo que hace necesario un buen control de los hiperenlaces en el centro educativo.

Otro módulo interesante dentro de este proyecto es la gestión de estadísticas del servidor. Se denominan *estadísticas externas*, a todas aquellas que hacen referencia a las conexiones no referentes al acceso a los cursos del centro educativo. Estas estadísticas son de carácter público y podrán servir de referencia a aquellas personas que estén interesadas en realizar

algún curso y quieran conocer las visitas que el centro virtual tiene a diario desde Internet (accesos diarios al CIEV, accesos clasificados por dominios, accesos clasificados por países, máximos y mínimos accesos por horas, días de la semana, etc). Las *estadísticas internas* hacen referencia a toda la gestión de los cursos del CIEV y a las estadísticas personalizadas de los alumnos en los cursos en que éstos se encuentren matriculados. Este tipo de estadísticas son de carácter privado, de modo que a las estadísticas relacionadas con los cursos solo tendrán acceso el personal pertinente y a aquellas relacionadas con los alumnos solamente podrá acceder el alumno afectado. Dentro de las estadísticas de los cursos podemos mencionar: porcentaje de los cursos más/menos solicitados, porcentaje de los cursos con más/menos aprobados, porcentaje de los cursos con mayor/menor aceptación en base a las encuestas cumplimentadas por los alumnos, etc. Podemos incluir dentro de las estadísticas personalizadas de cada alumno: porcentaje de aciertos/fallos en la evaluación por temas, porcentaje de temas con mayor/menor puntuación obtenida, porcentaje de las calificaciones globales al superar el curso, etc.

Un aspecto interesante de la gestión de estadísticas internas es el estudio de viabilidad de los diversos cursos impartidos en el CIEV. Con estas estadísticas podremos saber qué cursos se deben seguir impartiendo, qué cursos se deben eliminar y cuáles se deberían implantar.

Para entender completamente el funcionamiento del CIEV será necesario ponernos en el papel de cada una de las personas que interactúan en el centro de enseñanza. Todo usuario que quiera interactuar, o que represente un papel importante en el centro deberá identificarse mediante una palabra identificadora de usuario (login) y una palabra secreta o de paso (password). El login de cada usuario será asignado por el administrador del sistema, con la posibilidad de que cada usuario pueda cambiar su password (por seguridad).

2. Participantes

Los usuarios que interactúan con Centro Iberoamericano de Enseñanza Virtual son los siguientes:

1) **Alumno:** Cuando un alumno se identifica en nuestro centro de formación las actividades que éste puede realizar son: darse de alta en más asignaturas o darse de baja en una o varias asignaturas así como ver sus estadísticas internas, es decir, ver las estadísticas generadas por el centro acerca de sus calificaciones en los diversos cursos en los que se encuentra matriculado, etc..

Además, como es de esperar, podrá navegar a lo largo de los cursos en los que se encuentre matriculado. Al finalizar cada tema, el alumno deberá superar un examen acerca de la materia impartida en dicho tema. Una condición indispensable para pasar al siguiente tema será superar el tema que se acaba de estudiar. Tras aprobar la evaluación sobre el tema

estudiado, el alumno podrá optar, bien por repasar algunos de los conceptos que no le quedaron suficientemente claros o bien podrá decidir continuar con el estudio del siguiente tema. Al finalizar un curso, el alumno podrá rellenar un cuestionario indicando su opinión acerca del curso realizado así como visualizar todos los datos estadísticos generados por el centro acerca de los resultados obtenidos por el alumno. Además de la realización de los cursos en los que se encuentren matriculados los alumnos del CIEV, éstos tendrán acceso libre a las bibliotecas virtuales del centro (biblioteca de documentos de texto, fonoteca, videoteca y biblioteca de fotografías digitales). Finalmente, para resolver cualquier duda que tenga el alumno, éste podrá recurrir a las tutorías telemáticas ofertadas por el Centro de Enseñanza Virtual.

2) **Profesor:** El profesor que genere un curso mediante el generador de cursos del CIEV podrá gestionar dicho curso (este profesor será el profesor responsable del curso). El profesor podrá crear el curso propiamente dicho, modificar algunos de los conceptos introducidos en el curso así como borrar un curso. Cuando un profesor crea un curso mediante el generador de cursos, deberá realizar una petición de publicación de dicho curso, mediante correo electrónico, al administrador del CIEV. De la misma forma, cuando un curso se encuentra publicado y el profesor responsable del mismo desea modificar alguno de los conceptos o gráficos de alguna de las secciones de dicho curso, deberá realizar la modificación y enviarla, junto con una petición de modificación al administrador del centro virtual, el cual será la persona con privilegios para realizar la modificación. Finalmente, si el área responsable de un curso desea eliminar dicho curso por la razón que sea, el profesor responsable deberá realizar una petición de eliminación de dicho curso al administrador del sistema. La eliminación de un curso requiere el cumplimiento de una serie de restricciones para que el administrador pueda borrar dicho curso. Cabe destacar que es necesario que no exista ningún alumno realizando el curso.

En el momento en el que un profesor genera un curso y éste es publicado por el administrador del sistema, dicho profesor pasa a ser el responsable de curso en cuestión. A partir de ese momento, el profesor responsable de un curso tendrá el poder de asignar ciertos privilegios a otros profesores reales que participen en dicho curso, para que estos puedan realizar cambios, en el caso de ser necesario, en el mencionado curso. A su vez, el profesor responsable de la asignatura podrá modificar y eliminar los privilegios asignados a los otros profesores.

Además de la gestión de este tipo de privilegios, el profesor responsable de un curso, o los profesores con los suficientes privilegios sobre el curso, podrán realizar un tipo especial de navegación sobre el curso para verificar que el curso está correctamente montado sin ningún tipo de errores. Este tipo de navegación no contendrá ningún tipo de requerimientos para pasar de una página Web a otra (sin exámenes, etc.). Finalmente, cada profesor que participe en un curso, deberá resolver todo tipo de problemas a los alumnos mediante

tutorías telemáticas. Los horarios de las tutorías serán asignados a los profesores al inicio del curso.

3) **Administrador:** El papel del administrador del sistema consiste en, como su nombre indica, administrar y gestionar todo el Centro de Enseñanza Virtual. Cuando una persona no identificada o cuando un alumno del CIEV desean matricularse de una nueva asignatura, tras superar la prueba inicial, es el administrador el que debe confirmar la matrícula para que ésta se lleve a cabo. Del mismo modo, cuando un profesor termina un curso, será el administrador el que lo publique, actualice las posibles modificaciones y elimine un curso, siempre tras la solicitud del profesor responsable. También el administrador será el encargado de actualizar las bibliotecas virtuales tras la correspondiente solicitud de los profesores (serán estos los que adjunten en su solicitud, vía correo electrónico, los archivos que se deseen publicar en las diferentes secciones de las bibliotecas virtuales). Otras tareas del administrador son mantener y supervisar el correcto funcionamiento de todas las secciones del Centro de Enseñanza Virtual (Chat para tutorías telemáticas, correos internos, problemas en las páginas del centro, etc..)

4) **Personal visitante:** El personal visitante o personal no identificado engloba a todas aquellas personas que visitan nuestro centro virtual pero no pueden hacer uso de las posibilidades que ofrecemos al no estar registrados en el. Una persona no identificada que selecciona un curso y supera la evaluación inicial, tras rellenar el formulario de inscripción, pasa a ser alumno (tras la aprobación del administrador). El personal no identificado tan solo puede navegar por la estructura externa del CIEV, así como acceder a las estadísticas externas del centro. Para poder hacer uso de la funcionalidad del centro (cursos, tutorías, acceso a bibliotecas virtuales, etc..) es necesario estar identificado por lo que es necesario estar matriculado de alguna asignatura.

Otro aspecto importante a tratar en el CIEV es el relacionado con los aspectos de *seguridad del sistema*, de manera que el personal no autorizado no pueda acceder a las ventajas ofrecidas por este Centro de Enseñanza Virtual. Además es necesario en ciertos casos que la transmisión de datos a través de Internet esté encriptada. Por ejemplo, cuando un alumno acceda a uno de los cursos en los que se encuentre matriculado, será necesario que toda la información transmitida entre la relación hombre-máquina (alumno-tutor virtual) esté encriptada para salvaguardar el derecho a la intimidad del personal matriculado en el centro.

3. Educación a distancia

Tras lo comentado en esta exposición sería conveniente realizar una reflexión acerca de lo que nos puede aportar un sistema de estas características. No podemos afirmar que todo son ventajas debido a que las Nuevas Tecnologías de la Información además de aportar grandes facilidades a la sociedad también pueden carecer de ciertos elementos que las herramientas tradicionales nos aportaban. Uno de los aspectos más importantes que los sistemas

informáticos no nos pueden proporcionar dentro del ámbito educativo es el "calor humano". Las relaciones humanas, alumno-profesor, alumno-alumno nunca podrán ser sustituidas por un sistema informático.

A continuación se expondrán, a grandes rasgos, una serie de *ventajas e inconvenientes* que las redes de ordenadores proporcionan al ámbito educativo.

1) Ventajas:

- *Rompe las barreras del espacio*: Gracias a las redes de ordenadores, y sobre todo a Internet, las distancias se ven reducidas a la mínima expresión. Las posibilidades que nos ofrece la "red de redes" rompe la problemática de la localización física de la persona (es posible estar conectado a un ordenador que se encuentre en la misma sala en la que nos encontramos con la misma facilidad que a un ordenador o sistema informático que se encuentre localizado a miles de kilómetros de distancia).
- *Flexibilidad temporal*: Los ordenadores nunca descansan. Con esto queremos expresar la posibilidad que ofrece nuestro sistema a trabajar ininterrumpidamente. De esta forma el alumno podrá acceder a los cursos cuando quiera y desde donde quiera (el profesor se adapta al horario del alumnado y no al revés como ocurre actualmente).
- *Expansión de la información*: Internet ya es considerado como un potente medio de información por lo que nuestros cursos podrán llegar a un mayor número de personal.

2) Desventajas:

- *Limitaciones técnicas*: Implantar el modelo procesual al completo como técnica de evaluación educativa en un sistema informático puede llegar a ser una tarea impracticable debido al indefinido número de factores a tener en cuenta a la hora de evaluar al alumnado.
- *Pérdida de relaciones sociales*: Como se comentó anteriormente, las relaciones humanas son un factor muy importante en la educación y la docencia. Aunque es posible mantener el contacto humano mediante las aplicaciones que nos ofrece Internet para realizar trabajos en grupo, preguntar dudas al profesor de la asignatura, etc, las relaciones interpersonales, cara a cara, se pierden mediante el uso de estos sistemas.

Una posible aplicación real del proyecto podrá ser el uso del mismo para el apoyo a la docencia en universidades, como por ejemplo las universidades a distancia. De esta manera, las personas que no se puedan permitir económicamente, o por razones de trabajo, estar residiendo en la ciudad donde se encuentre la Universidad o asistir a las clases, este sistema permitirá la formación del alumnado en las materias pertinentes en el momento que deseen (sin horarios y sin desplazamientos). Tras superar los cursos explicados por el Tutor Virtual, estas personas estarán preparadas para realizar un examen oficial (y presencial) de la asignatura impartida en la Universidad. Otra posibilidad sería impartir cursos

complementarios a las asignaturas, o incluso el Centro de Enseñanza Virtual se podría utilizar para la formación de los trabajadores en las empresas.

Bibliografía:

DE PABLO, E. y otros: *Diseño del Currículo en el Aula Una propuesta de autoformación*. Mare Nostrum, Col. Forum Didáctico, Madrid, 1992.

MARTINEZ BONAFE, J. y SALINAS FERNANDEZ, D. (1988): *Programación y evaluación de la enseñanza: problemas y sugerencias didácticas*. MESTRAL. Valencia.

DIAZ RODRIGUEZ, J.J. : *Guía para la elaboración de unidades didácticas en la enseñanza secundaria*. Edita: Junta de personal docente de centros no universitarios de la provincia de Jaén, Colabora: Junta de Andalucía. Jaén, 1993.

VACCA, J. : *Los secretos de la seguridad en Internet*. Edita: Anaya Multimedia, 1997.

TACKETT J. y GUNTER, D. : *Utilizando LINUX*. Edita: Prentice Hall, 1997.

Revista NOVATICA (marzo-abril 1998): *Las TIC en la Educación*.

ORTEGA, J.A. : *Comunicación Visual y tecnología educativa*. Edita: Grupo editorial universitario, 1999.

Descripción de una experiencia de evaluación de los Sistemas de Navegación en teleformación

Gayà, C.; Dolz, I.; Alcantud, F.

U.I. Acceso Departamento de Psicología Evolutiva y de la Educación
Universidad de Valencia Estudi General,
C. Artes Gráficas 13, 46010 Valencia, Tlf: 963.864.135 Fax: 963. 963.864.758
e-mail: carmen.gaya@uv.es, isabel.dolz@uv.es, francisco.alcantud@uv.es

Resumen

En esta ponencia, se presenta una experiencia de teleformación o formación a través de la red, llevada a cabo en la Unidad de Investigación Acceso con el objetivo de estudiar cual es el sistema de navegación más adecuado cuando se presentan unos contenidos concretos, a estudiar por un usuario en situación experimental y siendo presentados estos contenidos en cuatro formatos distintos, que exigen del estudiante la utilización de sistemas y formas de navegación diferentes.

Palabras clave: teleformación, hipertexto, mapas conceptuales, sistemas de navegación

1 Introducción

La práctica docente habitual se basa en la coincidencia espacio/temporal de los tres elementos que intervienen e interactúan en el proceso enseñanza/aprendizaje (profesor/contenidos/alumno) mediante el establecimiento de un sistema de comunicación con la intención de: a) transmitir información, b) que el alumno desarrolle actitudes, y c) que desarrolle destrezas anteriormente programadas, o cualquier combinación de estas tres acciones. Dándose como características de esta situación docente, tanto la intencionalidad como la coincidencia espacio/temporal, en la cual el alumno comunica sus nuevas destrezas o errores, estableciéndose de este modo un mecanismo de "feedback" consecuencia del cual, entre otros factores, se produce el aprendizaje.

En ocasiones, la diseminación geográfica de los alumnos o sus circunstancias y características personales o familiares, impiden un contacto temporal permanente con el profesor. Esta circunstancia es más frecuente en las tareas de formación no reglada, es decir en la formación no obligatoria, aquella que se desarrolla una vez adquiridas las destrezas instrumentales básicas. La teleformación o el teleaprendizaje, no es más que la enseñanza

tutelada o dirigida a distancia. Es decir, el alumno se encuentra con el material, con los contenidos previamente programados y diseñados por el profesor, pero éste no se encuentra físicamente delante de él. Este tipo de formación a distancia se ha beneficiado enormemente del desarrollo de las tecnologías de la información y la comunicación, hasta el punto que en muchos casos se plantea como un complemento y alternativa a la práctica docente habitual.

La popularización de la red Internet ha generado un creativo y enriquecedor punto de encuentro para todas estas herramientas de teleformación. Podríamos definir el "Web Based Instruction" como el término que englobaría el uso de la red telemática Internet y más específicamente el servicio de World Wide Web con fines educativos, con todas sus posibilidades multimedia ([1] Khan, B.H.(1997). En este sentido, el desarrollo de la red y en particular de la WWW y del lenguaje HTML como lenguaje universal han hecho aparecer nuevas posibilidades como los hipertextos, los hiperenlaces o los multimedia, cuyo valor educativo está siendo hoy en día objeto de estudio ([2] Rouet, J.F.(1990, 1998); [3] Rouet, Levonen, Dillon y Spiro (1996)). Con el fin de unificar criterios acudiremos a las definiciones propuestas por [4] Díaz, Catenazzi y Aedo (1998).

Hipertexto: Desde hace algún tiempo, se viene utilizando este término para referirse a los documentos no lineales o secuenciales. Un hipertexto, en definitiva, es un sistema de organización de la información sobre la base de bloques discretos de contenido llamados *nodos*, conectados a través de una serie de *enlaces* cuya selección provoca la inmediata recuperación de la información destino (Díaz, Catenazzi y Aedo (o.c.)pág.3). Rouet (1998,pág. 89) define el hipertexto como un conjunto de unidades de textos conectados a través de múltiples enlaces que forman una especie de red textual.

Nodo: Diferentes sistemas de hipertexto emplean terminología distinta para este concepto y le confieren además características diferentes (card, document, frame, article, statement, etc.). Un nodo es una cantidad discreta de información que tiene significado por sí misma y que se encuentra incardinada en un sistema hipertextual. Los nodos pueden clasificarse como: *marcos*, la información contenida se debe adaptar a un espacio fijo en pantalla; *ventanas*, cuando utilizan toda la pantalla para presentar su contenido informativo; *principales*, cuando se encuentran al inicio de la ramificación de nodos; y *secundarios*, cuando no lo están; también según su tipo de contenido se clasifican en *dinámicos*, *estáticos*, *simples*, *compuestos*, etc.

Enlace: Son uno de los elementos más importantes de los sistemas hipertextuales. Un enlace es una conexión entre dos nodos que proporciona una forma de seguir las referencias entre un *origen* y un *destino*. Cuando se activa un enlace el flujo de control se traslada al siguiente nodo dando lugar a una variedad de acciones como mostrar una animación, una referencia, una ilustración, un video, un índice, etc. Los enlaces se pueden establecer *entre nodos*, origen y destino diferente, o bien *entre posiciones*, dentro de un mismo nodo; *hiperenlaces* o *enlaces multipunto*, cuando su origen o destino está compuesto por más de

un nodo; *condicionado*, si sólo es activado cuando se cumple una condición predefinida; y *dinámico* cuando produce un destino calculado por una función. Una vez analizados los elementos que componen los hipertextos es necesario analizar las diferentes formas de acceso a la información.

La lectura de un hipertexto difiere significativamente de la lectura de un texto convencional, cuando el lector se enfrenta a un hipertexto debe leer repetidamente el texto del nodo, seleccionar uno de los nodos propuestos y tras su lectura volver al nodo principal, si procede; por tanto debe hacer evaluaciones y tomar decisiones de forma más o menos continua. El hipertexto, al tener una organización global no lineal, puede no ser fácilmente accesible para el lector; esta característica tiene aspectos positivos, así el lector experto puede tener la posibilidad de seleccionar sólo la información que le interese frente a los textos convencionales, cuyo orden de lectura está básicamente predefinido. En consecuencia, la eficacia de los hipertextos dependerá: de las características de los usuarios, el contexto de la tarea, el área de conocimiento, etc. (Rouet (1998); Fundesco (1998)).

En la Teoría de la Flexibilidad Cognitiva propuesta por Spiro, Feltovitch, Jakobson y Coulson (1991), uno de los pocos intentos de proporcionar una teoría general de aprendizaje basado en el uso de hipertextos, aunque su validez empírica sólo ha sido parcialmente demostrada; uno de los *problemas*, todavía por resolver, está relacionado con la capacidad de los estudiantes para construir sus propias secuencias de aprendizaje. Retomamos este problema planteando la hipótesis de que una navegación libre por el hipertexto puede introducir dificultades añadidas en los alumnos que, por sus conocimientos previos o características personales, no son capaces de dirigir su navegación hacia su meta de aprendizaje.

La comprensión y recuerdo de la información de un texto pueden mejorar incrementando la coherencia del mismo. [5] Dee-Lucas y Larquin (1995) después de varios experimentos utilizando textos lineales(convencionales), e hipertextos concluyen diciendo que *los hipertextos estructurados ayudaron a la comprensión* aunque no obtuvieron diferencias significativas en tareas más específicas como la síntesis, y concluyen recomendando el *uso de ayudas* como la *visión global del texto* o la *presentación explícita de los propósitos de la lectura* para eliminar los efectos derivados de los diferentes formatos de presentación de los textos. Nosotros mismos [6] (Alcantud, F. y Tormo, MJ (1998)) encontramos cómo el uso de hipertextos incrementaba la *motivación hacia la tarea* frente a textos lineales en formato electrónico.

2 Método

Numerosos estudios evidencian como los tipos de enlaces entre nodos incrementan la dificultad de su uso, es por ello que al planificar el sistema de organización de los diferentes nodos de un hipertexto o *diseñar la arquitectura del hipertexto*, denominación que

utilizamos como analogía con la construcción de edificios, con una finalidad instruccional, tal como muestra la *figura 1*, tuvimos en cuenta: 1.-La existencia de un nodo de información introductoria y organizadora de la tarea general(nivel 1); en este nodo se presentaba un esquema de todo el contenido y los objetivos , de forma que el aprendiz pudiese tomar la decisión de profundizar más o menos en el hipertexto. 2.- Los nodos de nivel 2 o 3 estaban concebidos como elementos independientes en el sistema y relacionados mediante nodos de conexión donde se mostraba al aprendiz la relación del enlace y su importancia. Dándose la posibilidad de un cuarto nivel que enlazase directamente con nodos o hipertextos remotos, no diseñados por nosotros.

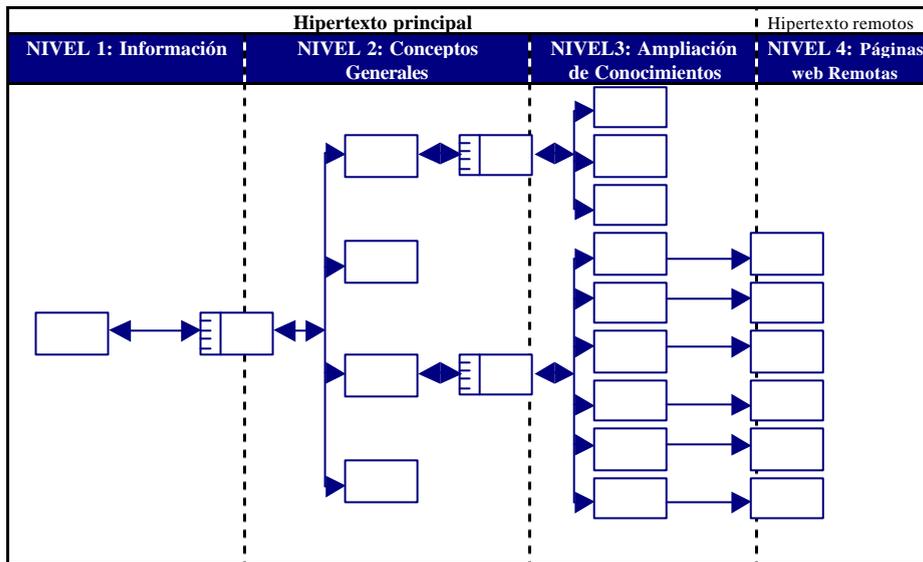


figura 1: Arquitectura de un hipertexto diseñado con finalidad instruccional.

Preparamos los diferentes formatos, con el fin de buscar un equilibrio en las tareas de aprendizaje individual mediante tareas de entrenamiento y evaluación, que sin caer en una programación rígida de corte conductista ni en el “laissez-faire”, permitiese al aprendiz adoptar un enfoque más exploratorio y autónomo. Se decidió incluir *mapas conceptuales*, por ser una de las herramientas de navegación y aprendizaje que está ganando más adeptos [7] (Kommers, Lanzing (1998)). Un mapa conceptual es una técnica gráfica, basada en la posibilidad de representar nuestro propio conocimiento en forma de redes en un espacio bi o tridimensional, como cualquier elemento gráfico permite su inclusión en un hipertexto, de forma que cada uno de sus elementos puede servir de enlace con un nodo de ampliación. El mapa conceptual intenta aprovechar al máximo nuestra percepción visual por las ventajas

que aporta la representación visual de la información al facilitar una visión de conjunto, el reconocimiento, la concisión de la representación y el establecimiento de relaciones entre los elementos del mapa, tal como vemos en la *figura 2*. El papel de los mapas conceptuales en el marco educativo es múltiple (Kommers, Lanzing (o.c.)); así: 1.- Es un método o herramienta de diseño que sirve como técnica de andamiaje estructural, antes y durante el desarrollo de los productos hipermedia. 2.- Mecanismo idóneo de navegación para estudiantes que necesitan cierta orientación mientras están explorando dominios muy extensos de información. 3.- Una técnica de representación de conocimiento para los alumnos cuando tratan de articular y sintetizar sus niveles de conocimiento reales en las distintas fases del proceso de aprendizaje. 4.- Una herramienta de evaluación del conocimiento que permite a los sujetos analizar su propio nivel de comprensión y detectar ideas y conceptos erróneos.

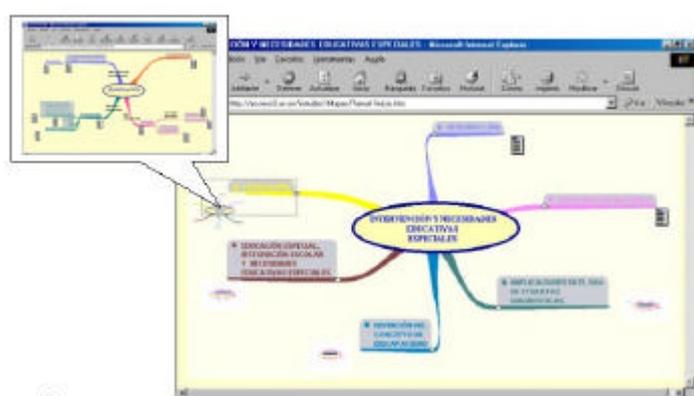


Figura 2: formato mapa conceptual.

La función del mapa conceptual es estimular un enfoque global dentro del camino escogido por el alumno para solucionar un problema, como mecanismo de navegación y representación puede auxiliar al alumno orientando su proceso de aprendizaje, articulando los conocimientos iniciales y finales, proporcionando la transferencia de conocimientos entre distintos temas o dominios y posibilitando la detección de falsos conceptos. El supuesto del que partimos es que el mapa conceptual puede integrar los procesos de *navegación* y *regulación cognitiva*, pueden proporcionar una visión gráfica de los diferentes nodos y enlaces del hipertexto, mostrando: 1.- Los nodos que ya han sido visitados; 2.- El camino de regreso a los puntos iniciales; 3.- El contenido disponible en cada nodo; y 4.- Las relaciones entre nodos, tal como vemos en la *figura 2*.

Partimos de un tema de instrucción en formato texto clásico en el cual se cuidó su estructura y coherencia para posteriormente prepararlo en los cuatro formatos diferentes. El contenido del primer nodo, común a todos los formatos, presentaba un índice-esquema del tema de

estudio junto con los objetivos que se pretendía alcanzase el alumno tras su estudio. El enlace de este primer nodo llevaba al estudiante al segundo nodo de información, específico para cada uno de los formatos, que se muestran a continuación:

Activas: texto convencional con información lineal, incluyendo una serie de *palabras activas* cuyo enlace llevaba a nodos o nuevas ventanas con información de ampliación de conocimientos, como vemos en la *figura 3*.



Figura 3 Formato Activas.

Mapas: Imagen gráfica del mapa conceptual del contenido del tema, las distintas zonas activas del mapa activaban los enlaces que llevaban a nodos ventana o bien a nuevos mapas con sus correspondientes zonas activas cuyo enlace llevaba a un nuevo nodo ventana con información de ampliación, como vemos en la *figura 2*.

Tablas: Nodo con marcos donde aparecía un *marco fijo* del esquema o tabla de contenidos, en la cual se actualizaba de forma constante la situación en la que se encontraba el alumno dentro de los contenidos del tema, y otro marco en el cual se podía acceder a nuevos nodos o ventanas con información de ampliación de los contenidos sin perder el origen en ningún momento, como vemos en la *figura 4*.

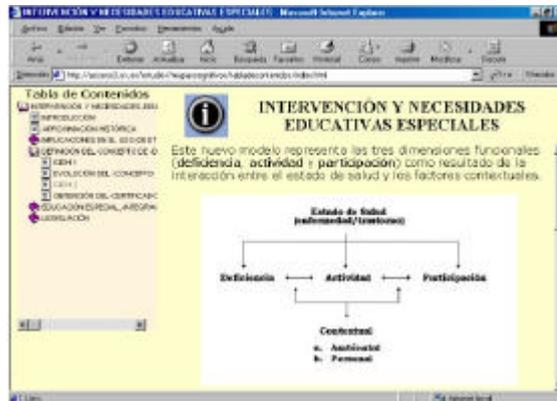


figura 4 Formato Tablas.

Índice: Este formato presentaba una variante en el primer nodo ya que el esquema del tema era activo, sus enlaces llevaban a nuevas ventanas que presentaban los diferentes apartados del tema incluyendo nuevas palabras activas cuyo enlace llevaba a tercer nivel, de modo que el estudiante siempre debía volver de forma inexcusable al primer nivel para continuar el estudio del tema, como vemos en la figura 5.

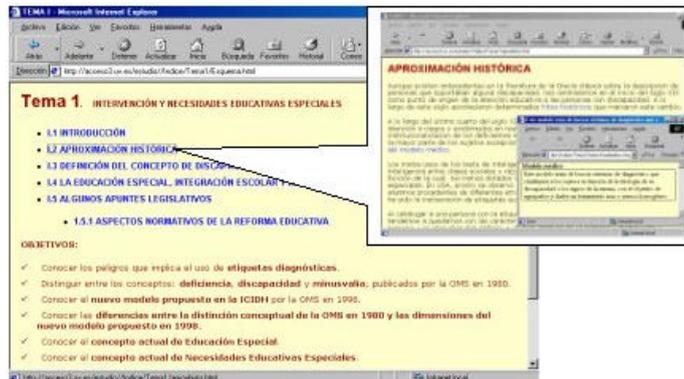


figura 5 Formato Índice

3 Conclusión

La utilización de los distintos sistemas de navegación que imponen los cuatro formatos, fue explorada por de un grupo de estudiantes, con distintos grados de conocimiento en el uso y

manejo de la red, esta experiencia se estudió en el Laboratorio de Usabilidad de la Unidad de Investigación Acceso de la Universidad de Valencia, pudiendo observar que en general que tras un breve periodo de práctica, aumentaba la eficacia en el manejo de los sistemas en general, y en concreto en el sistema de Mapas Conceptuales, mejorando su motivación y el nivel de implicación con la tarea así como la propia satisfacción, aunque inicialmente la mayor dificultad la presentaban aquellos sistemas que exigían de los usuarios la realización de diferentes acciones en cuanto a su manejo.

Referencias Bibliográficas

- [1] KHAN, B.H. (1997) Web-Based Instrucción. Educational Technology Publications Englewood Cliffs, New Jersey.
- [2] ROUET, J.F. (1998) Sistemas de hipertexto: de los modelos cognitivos a las aplicaciones educativas. En VISCARRO, C. & LEON, J.A. (Eds) NUEVAS TECNOLOGIAS PARA EL APRENDIZAJE. Madrid: Editorial Piramide.
ROUET, J.F. (1990) Interactive text processing bu inexperienced (hyper-) readers. En A. RIZK, N. STREITZ & J. ANDRE (Eds) HYPERTEXTS: CONCEPTS, SYSTEMS AND APLICATIONS (pag. 250-260) Cambridge UK: Cambridge University Press.
- [3] ROUET, J.F.; LEVONEN, A.D. ; DILLON, A. & SPIRO, R.J. (1996) HYPERTEXT AND COGNITION Lawrence Erlbaum Associates: New Jersey.
- [4] DÍAZ, P.; CATENAZZI, N. & AEDO, I. (1998) DE LA MULTIMEDIA A LA HIPERMEDIA. Madrid; Ra-Ma Editorial
- [5] DEE-LUCAS, D. & LARKIN, J.H. (1995) Learning from electronic texts: Effects of interactive overviews for information access. COGNITION & INSTRUCTION, 13(3) pag 431-468.
- [6] ALCANTUD, F. y TORMOS, M.J. (1998) La teleformación: Una experiencia en la Universitat de València Estudi General. En ALCANTUD, F. (ed) UNIVERSIDAD DIVERSIDAD. Universitat de Valencia Estudi General.
- [7] KOMMERS, P. & LANZING, J. (1998) Mapas conceptuales para el diseño de sistemas hipermedia, navegación por la web y autoevaluación. En VIZCARRO, C. Y LEON, J.A. (Ed) NUEVAS TECNOLOGIAS PARA EL APRENDIZAJE- Madrid: Editorial Piramide.

Colaboración en entornos de aprendizaje basados en casos reales. Aplicación en ambientes de diseño y simulación.

M.A. Redondo, C. Bravo, J. Bravo, M. Ortega

Grupo CHICO

Departamento de Informática.

Universidad de Castilla – La Mancha, Paseo de la Universidad s/n, Ciudad Real

Tlf: 926.295.300 Ext.3713 Fax: 926.295.391.

e-mail: {mredondo,cbravo,jbravo,mortega}@inf-cr.uclm.es

Resumen

En este trabajo se presenta un entorno de diseño y simulación que tiene como fin servir como herramienta de apoyo al aprendizaje a distancia en la disciplina de la automatización integral de viviendas y edificios (domótica). En el sistema se recurre a la utilización de técnicas como resolución de problemas mediante lenguajes de representación intermedios, colaboración y discusión de propuestas, tanto de forma síncrona como asíncrona. Todo el proceso de aprendizaje y colaboración, caracterizado por las soluciones obtenidas, el camino seguido para obtenerlas y la discusión del grupo, es sometido a análisis para hacer posible la extracción de conclusiones sobre el mecanismo de construcción colaborativa del conocimiento y los beneficios que la colaboración en la distancia pueden aportar en los procesos de aprendizaje.

Palabras clave: aprendizaje interactivo, entornos activos, aprendizaje a distancia, aprendizaje colaborativo, diseño y simulación.

1 Introducción

Los sistemas de aprendizaje basados en la resolución de proyectos reales [1] han demostrado ser eficientes para el aprendizaje a distancia, en los distintos niveles educativos. Se pone más énfasis en el aprendiz como agente activo en el proceso de adquisición conocimiento. El entorno SMISLE [2] constituye una referencia excelente para poner de relieve la afirmación anterior.

No obstante, un entorno de diseño y simulación por sí solo no ofrece un método de entrenamiento, debido a que el proceso de aprendizaje lo rige la persona que se entrena. Por otro lado, en el sistema no quedan registrados los avances que el alumno va realizando. Por lo tanto, es necesaria una monitorización o tutorización interactiva del aprendiz [3]. Este seguimiento hace posible que el sistema, automáticamente, pueda tomar decisiones sobre el trabajo que el alumno realiza, incluso que pueda aconsejar cual es el nivel instruccional del problema que mejor se adapta a sus conocimientos [4].

Es necesario tener presente la excesiva libertad que permiten los sistemas de simulación en el diseño del modelo a estudiar. Esto nos hace pensar que debemos añadir la posibilidad de que el alumno planifique sus acciones de diseño [3], [5], a modo de lenguaje de representación intermedio [6], [7]. De esta forma, es posible obtener mejores respuestas a los problemas planteados por el sistema.

El proceso se debe completar dando al alumno la posibilidad de simular el diseño planificado, para lo cual, deberá llevar a cabo el diseño en el lenguaje inherente al dominio a estudio. Así, el sistema puede contrastar la corrección del diseño con respecto a las acciones que previamente había especificado.

Tanto las acciones de planificación como las de diseño y simulación se pueden almacenar en una base de datos con la oportuna estructuración. De esta forma, el profesor o evaluador, con ayuda del sistema, puede reconstruir el razonamiento seguido por el alumno [8].

Las soluciones anteriores adolecen de una importante ausencia del concepto organizativo y social del aprendizaje, y por extensión de trabajo en grupo. Para solventar estas deficiencias nos apoyamos en el paradigma Computer Supported Collaborative Learning (CSCL), basado en una visión sociocultural de la cognición, para enriquecer los contextos interpersonales del aprendizaje. De este modo entendemos el aprendizaje como un proceso social y distribuido, donde se tiene en cuenta el diálogo de los participantes que se produce de forma cooperativa por los participantes. En este proceso, tanto el profesor como la tecnología, tienen un papel mediador de facilitación cognitiva y social.

2 Un caso de estudio en Domótica: DomoSim-TPC

Teniendo en cuenta las ideas apuntadas anteriormente, hemos desarrollado una herramienta denominada DomoSim-TPC [9] en la línea de lo que McGreal [10] definió como un Integrated Distributed Learning Environment (IDLE). En DomoSim-TPC los estudiantes y profesores que trabajan en tareas reales sobre un entorno asistido y colaborando mediante un modelo de conversación apropiado, guiados por conocimiento experto constituyen lo que se denomina un Rich Environment for Active Learning, REAL [11], donde apoyándose en teorías constructivistas como la colaboración, la autonomía personal, la generalidad, la flexibilidad, el compromiso activo, la relevancia personal y el pluralismo se da soporte a actividades de aprendizaje en grupo y a distancia [12], con las siguientes características:

- La herramienta utiliza como dominio de aplicación la Domótica, o Automatización Integral de Viviendas y Edificios, entendida como el conjunto de elementos que, instalados, interconectados y controlados automáticamente en una vivienda liberan al usuario de las acciones rutinarias de cada día y que proporcionan, al mismo tiempo, la optimización en el confort, el consumo energético, la seguridad y las comunicaciones.
- Los alumnos, desde el espacio de alumnado, pueden participar en actividades de resolución de problemas basados en escenarios reales.

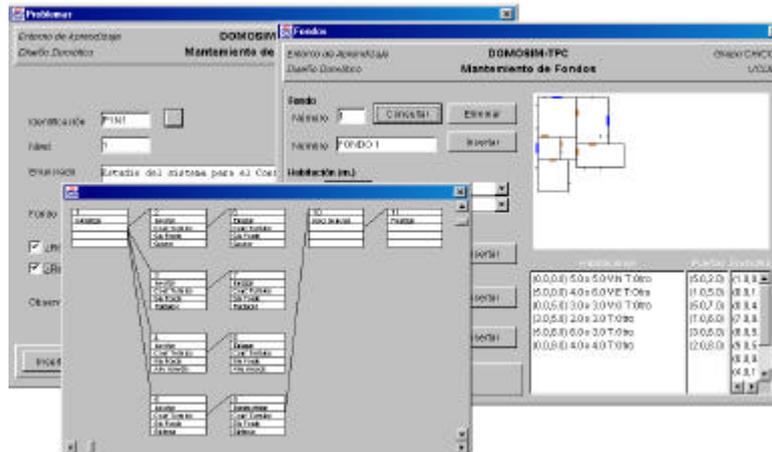


Figura 1. Mantenimiento de la memoria organizativa de problemas.

- Los profesores, desde el espacio de profesorado, gestionan y desarrollan problemas, en base a los cuales se plantean actividades a realizar individualmente o en grupo. Estos problemas se almacenan en una memoria organizativa clasificados según su nivel instruccional. La figura 1 muestra un problema planteado con su enunciado, su solución a modo de acciones genéricas de diseño representadas mediante un diagrama de Pert y la unidad de fondo (o plano en el dominio a estudio) que lo soporta.

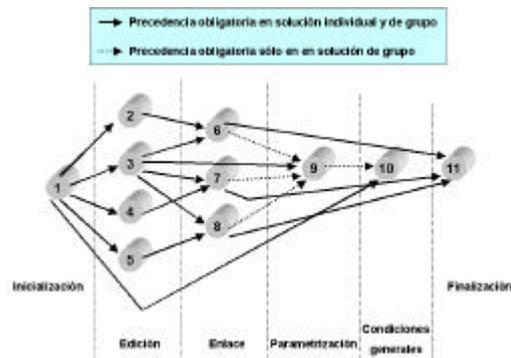


Figura 2. Grafo de precedencia de acciones de diseño para un problema de estudio del sistema control del confort térmico.

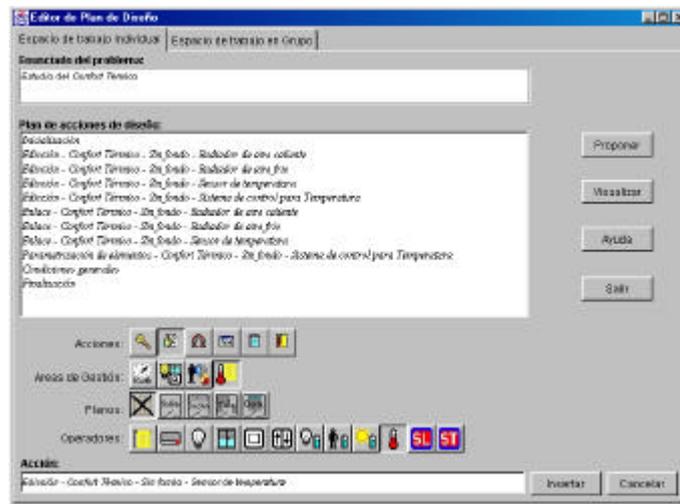


Figura 3. Espacio de trabajo individual: Editor de planes de diseño.

La Figura 2, muestra un ejemplo de grafo de precedencias para un problema sencillo de diseño donde se relacionan las acciones de diseño, representadas en un lenguaje de representación intermedio, el plan de acciones de diseño [13], en este grafo hay dos tipos de relaciones de precedencia: las que se representan por una flecha en trazo discontinuo que es necesario respetar tanto en soluciones propuestas por los estudiantes a nivel individual como en la solución consensuada por parte del grupo; las precedencias representadas mediante flechas con trazo discontinuo solamente se exige su cumplimiento en las soluciones que suponen un consenso del grupo. De este modo se pretende establecer una separación entre conceptos que el aprendiz debe adquirir y dominar por si mismo y

conceptos cuya adquisición y revisión se pospone al momento en que se lleva a cabo el proceso colaborativo de discusión en grupo.

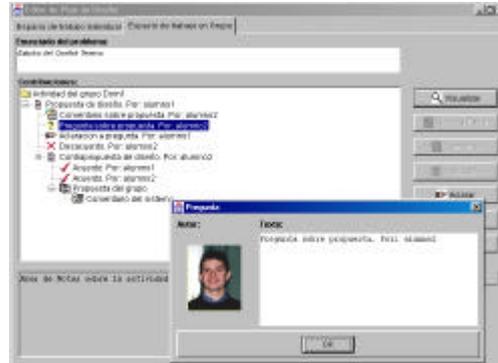


Figura 4. Espacio de discusión y organización de propuestas de planes de diseño.

- Los alumnos realizan una planificación de acciones de diseño, cuya solución final se obtiene fruto de la colaboración del grupo y enriquecida con los puntos de vista de cada uno de los miembros de dicho grupo. Para lo cual, disponen de un subespacio de trabajo individual (figura 3) donde haciendo uso de iconos que se asocian a acciones genéricas de diseño desarrollan y proponen su plan de diseño. En un segundo subespacio de trabajo (espacio de trabajo en grupo, figura 4) tiene lugar la discusión asíncrona y organización de las propuestas de cada uno de los participantes, para así, consensuar una solución de grupo, siguiendo un modelo de diálogo (figura 5) basado en el modelo IBIS que reprodujeron Kunz y Rittel [14]. Hay que destacar que las extensiones que hemos aplicado al modelo base enfocadas a constituir un sistema que no solo se encarga de soportar la propia discusión, sino que además debe analizar las contribuciones de los participantes, tomando decisiones al respecto (propuesta que nunca conducirá a solución válida, propuesta suficientemente válida, etc.). Dicho de otra forma, el trabajo en ambos subespacios es guiado por el sistema en base a conocimiento experto sobre el dominio, que el propio profesor puede incorporar cuando propone el problema (ver figuras 1 y 2), de este modo, se pretende ayudar y enriquecer el proceso de aprendizaje de los alumnos, ya que de otra forma, un entorno de diseño y simulación utilizado a distancia puede suponer el desconcierto de sus usuarios.



Figura 5. Modelo de diálogo empleado en es espacio de discusión asíncrono.

- Una vez realizada la planificación del diseño, el grupo realiza el propio diseño, ahora de forma síncrona. El sistema monitoriza y comprueba que el diseño se corresponde con el plan trazado con anterioridad. Para facilitar este proceso se cuenta con diversas herramientas de comunicación, tanto síncronas como asíncronas (irc o chat, e-mail, sistemas de votación, etc.), lo cual constituye un tercer espacio denominado espacio de coordinación, accesible tanto por profesores como por alumnos, pero donde la información se clasifica de acuerdo al concepto de grupo de trabajo. La figura 6 muestra el aspecto de la herramienta de correo electrónico que emplea el sistema, en nuestro caso apoyada en el empleo de tecnología de bases de datos remotas.



Figura 6. Subsistema de mensajería electrónica.

- El sistema puede simular el comportamiento del diseño realizado. La simulación se lleva a cabo de forma síncrona. Todos y cada uno de los alumnos pueden intervenir en la misma, generando eventos distribuidos que afectan al proceso de simulación como incidencias externas (Ver figura 7).
- El sistema ofrece al coordinador o profesor, dentro del espacio de profesorado, un análisis cuantitativo y cualitativo, fruto de la reconstrucción del razonamiento seguido por los alumnos y del diálogo practicado por los mismos a lo largo de la actividad. Estas intervenciones se almacenan en una base de datos estructurada al estilo de la utilizada por Scardamalia en sus trabajos [15].

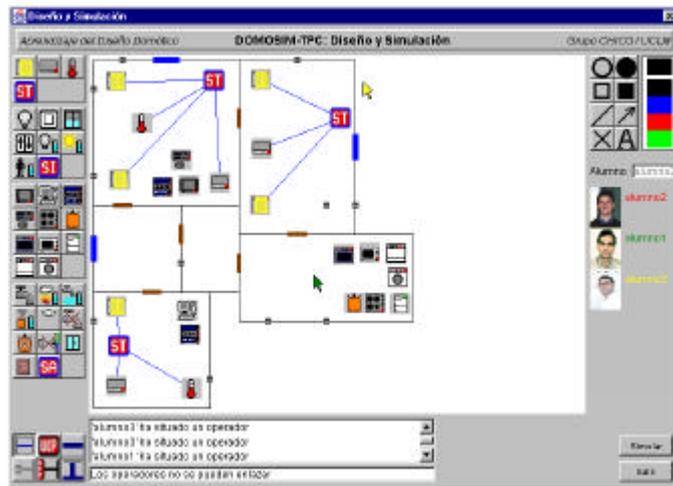


Figura 7. Aspecto del simulador durante una sesión de diseño.

- Todas y cada una de las características son accesibles a distancia, empleando para ello cualquier navegador Web con soporte para código Java, es decir, que incorpore una máquina virtual java (JVM).
- Como ya se puede intuir, todo el desarrollo se ha realizado en lenguaje Java, proporcionando un servidor Web donde se encuentran las páginas con formato HTML que soportan los Applets Java. Para recoger las intervenciones que tienen lugar durante las actividades, así como para toda la gestión derivada se emplea Java Databases Connection (JDBC) y SQL, delegando las funciones de control de concurrencia, seguridad, etc., al propio sistema de gestión de bases de datos que se opte por utilizar. En nuestro caso hemos empleado Microsoft Access sobre Windows NT y Oracle Server sobre Linux. En ambos casos se hace necesario recurrir a la utilización de Remote Method Invocation (RMI). En la figura 8 se puede observar la arquitectura funcional de DomoSim-TPC.

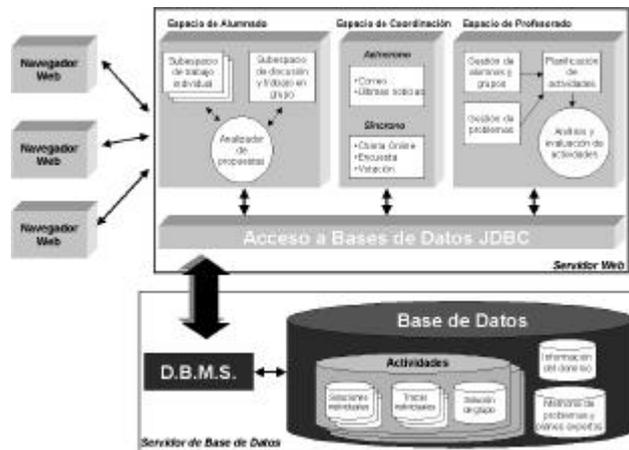


Figura 8. Arquitectura funcional de DomoSim-TPC.

DomoSim-TPC constituye un caso de estudio en el que se llevan a la práctica distintas teorías relacionadas con: tutores inteligentes, colaboración en tareas de grupo, aprendizaje por descubrimiento y a distancia, lenguajes de resolución intermedios y simulación con eventos distribuidos. Todo ello en el marco del aprendizaje a distancia (ver figura 9).



Figura 9. Conceptos teóricos abordados en DomoSim-TPC.

3 Conclusiones y trabajos futuros

DomoSim-TPC viene a ser un modelo de solución completa y altamente interactiva. No solo emplea herramientas de comunicación asíncrona sino que intenta abordar el problema de la comunicación síncrona como mecanismo de coordinación. Dispone de distintos espacios de trabajo para profesores y alumnos. Además el espacio de trabajo a nivel de alumno se divide en espacios de trabajo individual y de grupo, donde se dispone de operaciones muy diversas: planificación, edición, diseño, simulación, coordinación, etc.).

Algunas de los subsistemas de esta herramienta no son muy diferentes de otros entornos ya existentes. Por citar algunos, SimulNet [16] presenta un sistema de aprendizaje distribuido

a modo de laboratorio virtual, DEGREE [12] supone un prototipo para hacer posible que alumnos de tercer ciclo realicen en grupo y a distancia síntesis de documentos científicos; sistemas como CSILE [17], Collaboratory Notebook [18], C-CHENE [19] o Belvedere [20] dan soluciones parciales a problemas planteados, pero en general son extremadamente rígidos en la comunicación y en el tipo de tareas. En la mayoría de los casos, no aportan mecanismos de análisis del proceso de trabajo para favorecer la realimentación y mejora de los elementos del sistema y de los métodos de intervención pedagógica asociados con el aprendizaje.



Figura 10. Dispositivos que facilitarán la computación ubícua.

El sistema se ha planteado como una herramienta para educación a distancia, lo cual no exime su empleo en educación presencial y en entornos de red local, siempre con el fin de eliminar la definición plenamente espacial del término "distancia", recurriendo para ello a principios de la interacción persona-ordenador (Computer-Human Interaction, CHI). En este sentido, el próximo paso debe tratar la migración de la herramienta a entornos que hagan posible la computación ubícua, tales como los PalmTop (ver figura 10) y exploten al máximo los principios de la interacción persona-ordenador según este paradigma.

Con nuestro sistema pretendemos poder contar con datos reales sobre la adecuación de distintos modelos de interacción y discusión en grupo, aplicados a distintas fases en un caso de estudio sobre aprendizaje a distancia basado en la resolución de proyectos basados en escenarios reales. Actualmente se dispone de resultados del empleo de la herramienta a nivel individual, que pretendemos contrastar con datos procedentes de su empleo en ambientes de aprendizaje en grupo. Todos estos datos proceden de varios centros de Formación Profesional de la provincia de Ciudad Real (España). Una vez que dispongamos de esta información, nos abordaremos la aplicación de nuestros planteamientos a otros dominios, donde el diseño y la simulación estén justificados como herramienta de apoyo al aprendizaje empleando proyectos basados en escenarios reales.

Referencias

- [1] de Jong, T., van Joolingen, W., Pieters, J. & van der Hulst, Anja, (1993) "Why is discovery learning so difficult? And what can we do about it?". EARLI conference . Aix-en-Provence.
- [2] de Jong, T., van Joolinger W. & King, S., (1997) "The authoring environment SIMQUEST and the need for author support". In Supporting authors in the design of simulation based learning environments. Ton de Jong (Ed.), Servive Project.
- [3] Verdejo, M.F. (1992). "A Framework for Instructional Planning and Discourse Modelling in Intelligent Tutoring Systems". In Costa, E. (Eds.), New Directions for Intelligent Tutoring Systems, vol 91, pp 147-170. NATO ASI Series, Springer Verlag, Berlin.
- [4] Ortega, M., Bravo, J., Bravo, C., Muñoz, J.J. & Redondo, M.A., (1998), "Scaffolding and Planning Techniques in Distance Education: A case Study in Statistics". Proceeding of 4th International Conference on Technology Supported Learning. Online Educa. Berlin (Alemania).
- [5] Bravo, J., Ortega, M., & Verdejo, M.F., (1999), "Planning in Distance Simulation Environments". Full Paper in Communications and Networking in Education COMNED'99. Aulanko, Hämeenlinna, Finlandia. 13-18 de Junio.
- [6] Soloway, E., (1986). " Learning to Program = Learning to Construct Mechanisms and Explanations". Communications of the ACM.
- [7] Bonar, J.G. & Cunningham, R. (1988). "Intelligent Tutoring with Intermediate Representations" ITS-88 Montreal.
- [8] Boder, A., (1992), "The process of knowledge reification in human-human interaction", Journal of computer Assited Learning, Vol. 8, No 3, September, pp. 177-185.
- [9] Bravo, C., Bravo, J., Ortega, M., & Redondo, M.A., (1999), "A Simulation Distributed Cooperative Environment for the Domotic Design", En 4th International Workshop Proceedings on Computer Supported Cooperative Work in Design. Compiègne (Francia).
- [10] McGreal, R., (1998), "Integrated Distributed Learning Environments (IDLE's) on the Internet: A Survey", Educational Technology Review, N°. 9. Spring/Summer, pp. 25-31.
- [11] Grabinger, R.S. (1996), "Rich environments for active learning". In D.H. Jonassen, ed. *Handbook of Research for Educational Communications and Technologi*, 665692, New York: Simon & Schuster Macmillan.
- [12] Barros, B., (1999), "Aprendizaje Colaborativo en Enseñanza a Distancia: Entornos Genérico para Configurar, Realizar y Analizar Actividades en Grupo", Tesis Doctoral, Departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid.

- [13] Bravo, J., (1999) "*Estrategias para la planificación del diseño en entornos de simulación a distancia*". Tesis Doctoral, Escuela Superior de Ingenieros Industriales, Universidad Nacional de Educación a Distancia. Madrid.
- [14] Kunz, W & Rittel, H., (1970), "*Issues as Elements of Information Systems*". Working Paper #131, Institute of Urban and Regional Development, U. of California at Berkeley.
- [15] Scardamalia, M., & Bereiter, C., (1996) "*Student Communities for the advancement of Knowledge*". Communications of the ACM, 39(4).
- [16] Llamas, M., Anido, L., & Fernández, M.J., (1997) "*SimulNet: Virtual Tele-Laboratories Over the Internet*". In IFIP Working Group 3.3 & 3.6 Joint Working Conference. The Virtual Campus: Trends for Higher Education and Training. Madrid
- [17] Scardamalia, M., & Bereiter, C., (1991) "*Higher Levels of Agency for Children in Knowledge Building: A Challenge for the Design of New Knowledge Media*", The Journal of the Learning Sciences, Vol. 1, No. 1, pp. 37-68.
- [18] Edelson, D.C., & O'Neill, D.K., (1994), "*The CoVis Collaboratory Notebook: Supporting Collaborative Scientific Inquiry*", NECC'94.
- [19] Baker, M., & Lund, K., (1997), "*Promoting relective interactions in a CSCL environment*", Journal of Computer Assisted Learning, No. 3, Vol. 13, September, pp. 175-193.
- [20] Paulucci, M., Suthers, D., & Weiner, A., (1996), "*Automated Advice-giving Strategies for Scientific Inquiry*" en (Frasson, Gauthier, Ledgold, 1996), pp. 372-381.

Notas

Los miembros del Grupo CHICO constituyen el Comité Organizador del 2º Simposio Internacional de Informática Educativa a celebrar en Puertollano (España) durante los días 15 a 17 de Noviembre de 2000. <http://chico.inf-cr.uclm.es/siie2000>

Sesión V

Paradigmas de la Interacción y comunicación

Interacción Hombre-Máquina mediante Sistemas Automáticos de Diálogo

R. López-Cózar, A. J. Rubio, P. García, J. E. Díaz-Verdejo
Dpto. Electrónica y Tecnología de Computadores
Universidad de Granada, 18071 Granada, España
Tel.: +34-958-243271, FAX: +34-958-243230
E-mail: {ramon,rubio,pedro,jedv}@hal.ugr.es

Resumen

Los sistemas automáticos de diálogo constituyen una tecnología aparecida a finales de la década de los ochenta, impulsada principalmente por los proyectos DARPA SLS (*Spoken Language Systems*) en Estados Unidos, y SUNDIAL (*Speech UNDerstanding and DIALog*) en Europa. El objetivo principal de ambos proyectos es desarrollar sistemas informáticos capaces de proporcionar información acerca de viajes a los usuarios, usando la voz como medio de interacción. Actualmente, es posible encontrar en el mercado un amplio número de sistemas automáticos de diálogo, para una diversidad de tareas. Dichos sistemas funcionan en tiempo real, en estaciones de trabajo u ordenadores personales convencionales. En este artículo se realiza una breve introducción a algunas de las características principales de los sistemas automáticos de diálogo. Asimismo, a modo de ejemplo, se realiza una descripción de la estructura y funcionamiento de un sistema de diálogo automático desarrollado en nuestro laboratorio. Finalmente, se indican algunas líneas de investigación actuales en el campo y se presentan las conclusiones.

Palabras clave: sistemas de diálogo, interacción hombre-máquina, reconocimiento de voz, comprensión del lenguaje, control del diálogo, síntesis de voz.

1. Introducción

En trabajos recientes se puede encontrar un considerable número de sistemas automáticos de diálogo destinados a proporcionar diversos tipos de servicios a los usuarios [1], [2]. Estos sistemas utilizan principalmente las tecnologías de reconocimiento de voz, comprensión del lenguaje, control del diálogo, y síntesis (generación) de voz. Inicialmente, la interacción hombre-máquina mediante el lenguaje humano se llevaba a cabo haciendo uso de mensajes escritos. Sin embargo, los sistemas automáticos de dialogo deben tratar

diversos fenómenos no presentes en los mensajes escritos, como por ejemplo, diferencias en la voz de los usuarios, falso comienzo de las frases, ruido ambiental, cruce de conversaciones (*crossstalk*), palabras no incluidas en el vocabulario, etc.

Los sistemas de diálogo permiten realizar de forma automática determinadas tareas anteriormente reservadas a personas, con el consiguiente ahorro económico para las empresas. Además, dado que la interacción se realiza mediante el habla, es posible realizar dichas tareas de forma remota usando el teléfono, proporcionando un mayor grado de comodidad, flexibilidad y naturalidad a los usuarios. No obstante, los sistemas de diálogo deben enfrentarse a diversos problemas. Por una parte, deben tener un tiempo de respuesta reducido, aceptable por los usuarios. Sin embargo, para lograr tasas de error de reconocimiento suficientemente pequeñas, generalmente se necesitan tiempos de análisis altos. Por otra parte, los sistemas de diálogo deben averiguar lo que los usuarios dicen y, además, lo que realmente quieren. Por consiguiente, deben contar con altas tasas de reconocimiento, y en la medida de lo posible, deben ser robustos frente a los diversos problemas inherentes al lenguaje humano (uso de información contextual, existencia de ambigüedades, etc.). Todo ello implica emplear estrategias que permitan evitar o corregir los errores, y en cualquier caso, proseguir la conversación con los usuarios de una forma satisfactoria. El comportamiento de estos sistemas ha de ser coherente dentro de la conversación, por ejemplo, no deben volver a solicitar información que los usuarios hayan proporcionado previamente (quizás de forma inesperada), a no ser que realmente sea necesario. Finalmente, los sistemas de diálogo deben generar respuestas fácilmente comprensibles por las personas, para lo cual se deben emplear técnicas que permitan generar frases de una gran inteligibilidad y naturalidad. Por todos estos motivos, la gran mayoría de los sistemas de diálogo existentes actualmente están especializados, es decir, son diseñados para realizar una única tarea concreta, la cual suele estar relacionada con consultas y actualizaciones de bases de datos (de información turística, meteorológica, financiera, etc.).

Los corpora de grabaciones de diálogos, palabras y frases son esenciales para el desarrollo de estos sistemas. Por una parte, es necesario determinar el conjunto de palabras y expresiones lingüísticas relacionadas con el dominio de aplicación de cada sistema. Por otra parte, es necesario obtener los modelos acústicos independientes del locutor que utilizará el módulo de reconocimiento del sistema, así como estimar los modelos del lenguaje [3]. A fin de obtener diálogos y comprobar ideas durante las etapas iniciales de desarrollo, suele utilizarse la técnica *del Mago de Oz*, mediante la cual un experto humano sustituye al ordenador, y los usuarios hablan pensando que realmente lo hacen con una máquina.

2. Un ejemplo: El sistema SAPLEN

Hemos desarrollado un sistema automático de diálogo, denominado SAPLEN (Sistema Automático de Pedidos en LENGuaje Natural), cuya finalidad es atender telefónicamente a los clientes de restaurantes de comida rápida [4]. El sistema ha sido diseñado para proporcionar información telefónicamente a los usuarios, y para mantener un registro de productos encargados telefónicamente, los cuales podrían ser enviados posteriormente a domicilio.

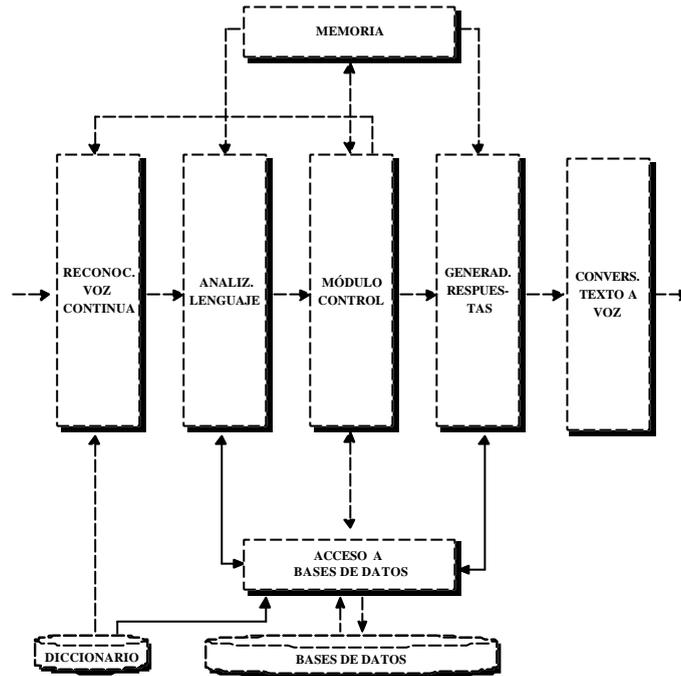


Figura 1. Diagrama modular del sistema SAPLEN

Desde un punto de vista global, el sistema se compone de cuatro módulos principales: interfaz de entrada, módulo de control, bases de datos e interfaz de salida.

2.1 Interfaz de entrada

La interfaz de entrada está constituida por un módulo de reconocimiento de voz continua y un analizador del lenguaje. El módulo de reconocimiento se usa también en otro sistema de diálogo desarrollado en nuestro laboratorio, llamado STACC, cuya finalidad es informar

telefónicamente acerca de las calificaciones obtenidas por los estudiantes de diversas titulaciones de la Universidad de Granada [5]. El tamaño del vocabulario del sistema SAPLEN es de 2.000 palabras aproximadamente, incluyendo nombres de productos de restaurante, números, nombres de calles, avenidas, plazas, etc. El lenguaje se modela mediante bigramáticas basadas en 53 clases de palabras [6]. El analizador del lenguaje detecta las palabras y expresiones clave proporcionadas por el módulo de reconocimiento, y utiliza reglas de análisis basadas en conceptos semánticos, las cuales tienden a ignorar las palabras no significativas desde un punto de vista semántico [7], [8]. Las principales ventajas de este tipo de análisis son dos. Por una parte, pueden evitarse ambigüedades sintácticas si las interpretaciones correspondientes no tienen significado semántico. Por otra parte, durante el análisis pueden ignorarse los detalles sintácticos que no afectan a la interpretación semántica. En total se utilizan 45 reglas de análisis, las cuales permiten resolver anáforas, elipsis, ambigüedades, y tautologías. Esta metodología de análisis proporciona una cierta robustez frente a errores de reconocimiento. En efecto, el analizador del lenguaje puede obtener la interpretación semántica correcta a pesar de que, en ocasiones, algunas palabras no hayan sido reconocidas correctamente (recuperación implícita) [9].

2.2 Módulo de control

El módulo de control recibe las interpretaciones semánticas construidas por el analizador del lenguaje y las almacena en el módulo de memoria mediante *frames*. Cada frame está compuesto por un conjunto de *slots* (huecos) que almacenan los datos relevantes de las frases. Una misma interacción del usuario puede generar uno o más frames.

El módulo de control implementa una estrategia de gestión del diálogo mixta [10], [11]. Dicha estrategia está basada en un conjunto de objetivos que el sistema debe tratar de lograr, y diversos subobjetivos que los usuarios pueden generar durante la conversación. Por ejemplo, un objetivo del sistema consiste en intentar vender productos del restaurante, otro consiste en obtener el número de teléfono del usuario actual, etc. Decimos que una determinada interacción de un usuario contiene *información parcial* si la interacción no proporciona suficiente información al sistema. Una interacción de este tipo da lugar a uno o más subobjetivos que el sistema debe tratar de satisfacer para completar la información parcial. Un mecanismo de unificación permite manejar las informaciones parciales proporcionadas por los usuarios, mediante la combinación de los frames correspondientes. Una vez logrados los subobjetivos, el sistema puede realizar alguna acción, como por ejemplo, registrar el pedido de algún producto, o responder a alguna pregunta del usuario. Tras lograr los subobjetivos generados por el usuario, el sistema vuelve a intentar satisfacer sus objetivos originales. Por consiguiente, el control del diálogo se lleva a cabo basándose en una estrategia de control mixta, considerando tanto los propios objetivos del sistema como los subobjetivos generados por los usuarios.

2.3 Bases de datos

Las bases de datos almacenan la información de las entidades relacionadas con el dominio de aplicación del sistema (productos del restaurante, direcciones de la ciudad, y clientes conocidos). En la base de datos de productos del restaurante se encuentra la información relacionada con algunos de los productos que comúnmente pueden ser solicitados en restaurantes de comida rápida. En la base de datos de direcciones se almacena la información relacionada con las direcciones de la ciudad conocidas por el sistema. La base de datos de clientes conocidos almacena la información de los clientes que previamente han realizado pedidos al sistema. Para cada cliente se almacena su número de teléfono y su dirección completa. El sistema pregunta el número de teléfono al usuario una vez que éste ha realizado algún pedido. En función del número de teléfono proporcionado el sistema determina si se trata de un cliente conocido. En caso de ser así, el sistema le informa de los datos de su dirección existentes en la base de datos y le solicita la confirmación de los mismos. Si el usuario no los confirma el sistema los solicita de nuevo, quedando actualizados en la base de datos al final de la conversación. En el diccionario se encuentran almacenadas las palabras relacionadas con el dominio de aplicación del sistema. Las palabras que el módulo de reconocimiento proporciona como salidas están contenidas en el diccionario. La información del diccionario es necesaria para el analizador del lenguaje, el cual construye los constituyentes semánticos teniendo en cuenta las clases léxicas a las que pertenecen las palabras reconocidas.

2.4 Interfaz de salida

La interfaz de salida del sistema consta de un módulo que genera las respuestas del sistema en modo texto, y un módulo que realiza la conversión de las mismas de texto a voz. El generador de respuestas utiliza 41 patrones para realizar la formación de los diversos tipos de frases. Asimismo, usa varias reglas para determinar el género y el número de las palabras, y el uso (o no) de pronombres en las frases, teniendo en cuenta la información existente en el contexto en cada momento. Cada patrón consiste en una serie de conceptos, expresiones y huecos. Durante la generación de las respuestas, el sistema expande los conceptos y las expresiones, y rellena los huecos con las palabras correspondientes. La generación de respuestas tiene una doble finalidad. Por una parte, proporcionar a los usuarios la información solicitada, y por otra, proporcionarles otros tipos de información, como por ejemplo, confirmaciones de comprensión de sus frases, mensajes de error, etc. Para realizar la transformación de texto a voz se utiliza la plataforma de síntesis de voz FESTIVAL, desarrollada por el Centro de Investigación de la Tecnología del Habla de la Universidad de Edimburgo (Reino Unido) [12].

2.5 Trabajo futuro

Son varias las líneas de trabajo en las cuales debemos seguir trabajando para mejorar el funcionamiento del sistema SAPLEN. En particular, debemos mejorar el módulo de reconocimiento de voz continua, a fin de que asigne valores de confianza a las palabras que proporciona como salidas. Actualmente, únicamente proporciona un valor de probabilidad de la frase que se está reconociendo en cada momento, el cual se utiliza para descartar las frases poco probables. Sin embargo, el módulo de control del sistema está diseñado para utilizar valores de confianza asignados a las palabras de las frases. En función de estos valores, el sistema puede decidir entre solicitar al usuario la repetición de una frase completa, o bien, la repetición únicamente de aquellas palabras que quizás no haya sido reconocidas de forma correcta.

3. Algunas líneas de investigación actuales en el campo de los sistemas automáticos de diálogo

Si bien los avances logrados en los últimos años son notables, existen aún diversas líneas de investigación abiertas, las cuales tienen como finalidad lograr una interacción más cómoda y natural, logrando así una mayor aceptación por parte de los usuarios. Por una parte, son necesarios avances en las tecnologías involucradas (reconocimiento de voz, comprensión del habla, control del diálogo y síntesis de voz) a fin de facilitar la integración e interacción entre las mismas. Por otra parte, existen problemas de portabilidad entre dominios o tareas. Dado que la gran mayoría de los sistemas de diálogo existentes hoy en día son dependientes del dominio para el cual han sido diseñados, si se cambia de dominio de aplicación los sistemas creados dejan de ser útiles, y hay que crear sistemas nuevos, o bien, adaptar en mayor o menor medida los existentes. Además, son necesarias mejoras en las técnicas de evaluación, a fin de que se tengan en cuenta las características y dificultades intrínsecas de las tareas. Finalmente, es necesario mejorar y aumentar el número de herramientas de desarrollo, las cuales permitan crear de forma automática, o casi automática, los modelos acústicos y del lenguaje, gramáticas, estructuras semánticas, y modelos del diálogo. Recientemente, han aparecido varias herramientas de este tipo, entre las cuales podemos mencionar la plataforma del Centro para el Estudio del Habla y la Comprensión del Lenguaje, CSLU (Oregón, Estados Unidos) [13].

4. Conclusiones

Los sistemas de diálogo constituyen una tecnología basada en el reconocimiento de voz, comprensión del lenguaje, control del diálogo y generación de voz. La voz del usuario constituye la entrada de un módulo de reconocimiento de voz continua independiente del locutor, el cual devuelve como salida una o varias frases posibles. Un módulo de análisis del lenguaje se encarga de buscar palabras, o secuencias de palabras, que tengan información relevante para la aplicación. Esta información se suele almacenar mediante estructuras

semánticas. Un módulo de control determina la próxima acción a realizar por el sistema, como por ejemplo, consultar o actualizar alguna base de datos. Finalmente, un módulo de generación de respuestas proporciona la salida del sistema, generalmente mediante un sintetizador de voz.

La tecnología de los sistemas automáticos de diálogo puede aumentar la facilidad, comodidad y flexibilidad a la hora de realizar de forma automática determinadas tareas, pues permite interactuar con los ordenadores mediante el lenguaje humano. El interés de empresas privadas y el incremento de fondos de investigación públicos están propiciando grandes avances tecnológicos, los cuales están generando la aparición de sistemas cada vez más perfeccionados. Además, existe un gran número de empresas que se están especializando en la tecnología de desarrollo de estos sistemas. Todo ello indica que el interés en este campo continuará durante varios años, dado que su mercado potencial es inmenso.

5. Referencias

- [1] Asoh H., Matsui T., Fry J., Asano F. Hayamizu S., "A Spoken Dialog System for a Mobile Office Robot", Eurospeech '99, pág. 1139-1142
- [2] Chao H., Xu P., Zhang X., Zhao S., Huang T., Xu B., "LODESTAR: A Mandarin Spoken Dialogue System for Travel Information Retrieval", Eurospeech '99, pág. 1159-1162
- [3] Ziegenhain U., Harengel S., Kaiser J., Wilhem R. "Creating Large Pronunciation Lexica for Speech Applications", First International Conference on Language Resources and Evaluation, pág. 1039-1043
- [4] R. López-Cózar, A. J. Rubio, P. García, J. C. Segura, "Uso de Valores de Confianza y Expectativas en el Sistema de Diálogo SAPLEN", Revista SEPLN, nº 24, diciembre 1999, pág. 37-41
- [5] Rubio A.J., García P., De la Torre A., Segura J.C., Díaz-Verdejo J.E., Benítez M.C., Sánchez V., Peinado A.M., López-Soler J.M., Pérez-Córdoba J.L., "STACC: An Automatic Service for Information Access Using Continuous Speech Recognition Through Telephone Line", Eurospeech '97, pág. 1779-1782 and Stochastic Finite State Automata", Eurospeech '99, pág. 2175-2178
- [6] Jelinek F., "Statistical Methods for Speech Recognition", MA: MIT Press, 1999
- [7] Noeth E., Boros M., Haas J., Warnke V., Gallwitz F., "A Hybrid Approach to Spoken Dialogue Understanding: Prosody, Statistics and Partial Parsing", Eurospeech '99, pág. 2019-2022
- [8] J. Allen, "Natural language Understanding", Benjamin/Cummings Publishing Company Inc. 1995

- [9] M. Danieli, E. Gerbino, "Metrics for evaluating dialogue strategies in a spoken language system", AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, 1995, pág. 34-37
- [10] Rosset S., Bennacef S., Lamel L., "Design Strategies for Spoken Language Dialog Systems", Eurospeech '99, pág. 1535-1538
- [11] Relaño Gil J., Tapias D., Villar J. M., Gancedo M. C., Hernández L. A., "Flexible Mixed-Initiative Dialogue for Telephone Services", Eurospeech '99, pág. 1179-1182
- [12] <http://www.cstr.ed.ac.uk/projects/festival/festival.html>
- [13] <http://cslu.cse.ogi.edu/toolkit>

Diseño de sistemas basados en voz para la interacción hombre-máquina

P.García, J. Díaz-Verdejo, J.C. Segura, R. López-Cózar, A. Rubio

Departamento de Electrónica y Tecnología de Computadores
Facultad de Ciencias. Campus Fuentenueva, S/N
18071 - Granada
Tf: 958-244011 Fax: 958-243230 e-mail: pedro@hal.ugr.es

Resumen

Es creciente en la actualidad la aplicación de las tecnologías del lenguaje en entornos de comunicación hombre-máquina. En esta línea se presentan aquí varios sistemas diseñados para proporcionar sendos servicios en entornos reducidos específicos. Todos ellos tratan de resolver los problemas asociados a la interacción vocal usuario-máquina de una forma intuitiva y natural al tiempo que se consiguen unas elevadas prestaciones.

Palabras clave: tecnologías del lenguaje, reconocimiento automático de voz, modelado del lenguaje, diálogo

1 Introducción

El lenguaje es la forma más natural de comunicación humana, por lo que la disponibilidad de nuevos servicios telemáticos se encuentra altamente influenciada por el desarrollo de las tecnologías del lenguaje. El diagrama de bloques general de un sistema hombre-máquina basado en voz consta de [5]:

- a) La entrada, que comprende: 1) un sistema de reconocimiento automático de voz (ASR, "automatic speech recognition"), 2) un módulo de procesado del lenguaje (NLP, "natural language processing"), y 3) un módulo de diálogo (DM, "dialogue module").
- b) El subsistema que genera la salida, compuesto de: 1) un módulo de generación de respuesta en modo texto (TG, "text generation") y 2) un bloque texto-a-voz (TTS, "text-to-speech").

No todos los módulos mencionados son estrictamente necesarios en un sistema real; por ejemplo, el módulo TTS puede sustituirse por un conjunto de frases pregrabadas. No es este el caso del bloque ASR, el cual, núcleo del sistema de interacción vocal, trata de decodificar la secuencia acústica pronunciada por un locutor de acuerdo con un modelo de lenguaje

aceptado para la tarea a abordar. El modelo de lenguaje o gramática es de importancia capital, ya que especifica las respuestas del locutor aceptadas por el sistema, definiendo por tanto la interacción usuario → máquina.

A medida que la gramática se hace más compleja con objeto de modelar mejor el comportamiento del usuario, el proceso de reconocimiento resulta más difícil; por ello, los sistemas reales deben llegar a un compromiso entre la naturalidad en la interacción y las prestaciones del servicio proporcionado. Siguiendo este criterio, existen dos aproximaciones al problema de la interacción hablada hombre-máquina. En la primera el diálogo se estructura en forma de árbol, en cuyo caso, si bien los resultados obtenidos son satisfactorios desde el punto de vista de las prestaciones del sistema, no ocurre lo mismo con la naturalidad del flujo de la interacción. Siguiendo esta técnica se presentan aquí dos sistemas desarrollados por nuestro grupo de investigación. La segunda aproximación al problema del diseño de sistemas de diálogo vocal hombre-máquina emplea un módulo de diálogo para posibilitar la adaptación del sistema a las respuestas del usuario. Así, en la actualidad estamos utilizando esta aproximación con objeto de diseñar e implementar un sistema hablado para proporcionar información de viajes por teléfono.

2 Sistemas de interacción hombre-máquina basados en voz

En los últimos años se han diseñado y desarrollado numerosos sistemas para ofrecer servicios telemáticos basados en voz. Algunos ejemplos de ellos son el proyecto DARPA ATIS [8] y varios proyectos europeos tales como REWARD, ARISE, ACCeSS, etc. En esta línea, el trabajo de investigación desarrollado por varios de los miembros del GiPSyC (Grupo de Investigación en Procesamiento de Señales y Comunicaciones) de la Universidad de Granada, ha posibilitado la implementación de un sistema de reconocimiento de voz continua (CSR, “continuous speech recognition”) en base a un conjunto de librerías [2][3]. A partir de ellas es posible el diseño de distintas aplicaciones sin más que especificar el flujo del servicio y el modelo del lenguaje de la tarea a abordar.

2.1 Sistema telefónico automático para la consulta de calificaciones

El sistema STACC (Sistema Automático de Consulta de Calificaciones) proporciona a los estudiantes sus calificaciones mediante una simple llamada telefónica [9]. Este sistema ha encontrado una buena aceptación entre el alumnado dada la flexibilidad y facilidad en la consecución de la referida información.

STACC comprende los siguientes módulos (Figura 1)8: a) un autómata de estados finitos que define y controla la interacción usuario-máquina, b) una interfaz vocal para adquirir y reproducir los mensajes de voz, y c) una base de datos con la información objeto del servicio. Dependiendo del estado indicado por el módulo de control de flujo, éste obtiene las gramáticas específicas a usar por el reconocedor en cada momento, genera la salida de audio sobre la línea telefónica e interacciona con el sistema de reconocimiento y la base de datos.

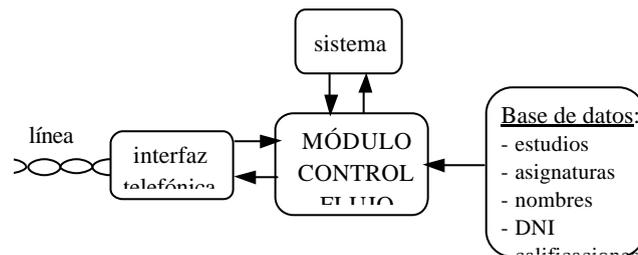


Figura 1. Diagrama de bloques de STACC.

2.2 Centralita telefónica automática basada en reconocimiento de voz con interfaz e-mail

Otro sistema diseñado por nuestro grupo de investigación es CABE (Centralita telefónica Automática Basada en voz con interfaz Email) [4]. El servicio telemático ofrecido en este caso integra las tecnologías del lenguaje con las de las comunicaciones, en concreto con el correo electrónico.

El sistema pregunta inicialmente el nombre de la persona con quien desea contactar el llamante y encamina la llamada hacia el teléfono correspondiente. Si la llamada no puede ser atendida, se da la opción al usuario del servicio de grabar un mensaje de voz, el cual se enviará a la dirección de correo del destinatario.

La aplicación PCI ("phone-computer interaction") aquí diseñada consta de los siguientes bloques (Figura 2):

- Interfaz vocal, que comprende la interfaz telefónica y el módulo CSR.

8 Web de STACC: <http://ceres.ugr.es/sacc>

- Interfaz e-mail. Una vez dirigida la llamada telefónica hacia el número de destino, si no es atendida dentro de un período de tiempo dado se posibilita al usuario del servicio la grabación de un mensaje de voz, el cual se enviará a
- la dirección de correo electrónico correspondiente al destinatario del servicio.
- Módulo de control. Es responsable de la interacción entre los dos módulos previos y controla el flujo general del servicio.

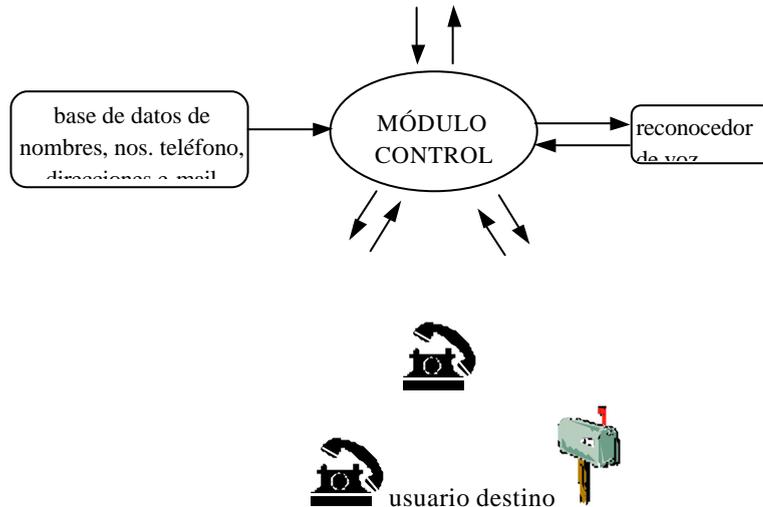


Figura 2. Representación conceptual del sistema CABE.

2.4 Sistema de diálogo para consulta remota de información de viajes

A pesar de la utilidad de los sistemas anteriores hemos de hacer notar que la interacción usuario-máquina implementada en ellos es muy rígida, no considerándose módulo de diálogo alguno. En la actualidad, en el seno del GiPSyC se está desarrollando un sistema basado en voz para la consulta de información acerca de viajes a través de teléfono. Desde un punto de vista técnico el principal objetivo de este proyecto es hacer más flexible la interacción usuario-máquina con objeto de que el servicio sea dirigido por el usuario en lugar de por el sistema como sucedía en las realizaciones anteriores.

El sistema consta así de dos módulos: una interfaz de voz y un módulo de diálogo (DM). El primero es idéntico al usado en STACC y CABE, mientras que el subsistema de diálogo simula el comportamiento de un trabajador de una empresa de viajes atendiendo el servicio. El bloque DM se basa en el usado por el sistema SAPLEN [6], y su función es doble: permite proporcionar una respuesta adecuada a cada una de las frases pronunciadas por el

usuario al tiempo que limita el contexto y define de forma dinámica el modelo de lenguaje a usar en cada momento por el subsistema de reconocimiento.

Adicionalmente al bloque DM resulta deseable la incorporación al sistema de un módulo TTS con objeto de mejorar la interacción usuario-máquina mediante la generación de voz sintética a partir de la respuesta texto proporcionada en cada momento por el módulo DM. Si bien existen varias técnicas de síntesis disponibles, en nuestro caso se está haciendo uso por el momento de la técnica multipulso con difonemas como unidades acústicas [10].

Dos consideraciones adicionales acerca del sistema son:

a) El modelo de lenguaje definido por el módulo DM en cada momento se basa en clases de palabras [7], de modo que la gramática es más general y rápida de usar de forma dinámica.

b) Se hace uso de una medida de confianza en el proceso de reconocimiento, de forma que cada palabra de la secuencia reconocida tiene asociado un número indicativo de la fiabilidad en su reconocimiento [1]. Estos valores de confianza se usan por el bloque DM para generar la respuesta adecuada hacia el usuario.

3 Prestaciones de los sistemas y conclusiones

Todos los sistemas descritos en la sección anterior se han implementado en una máquina Sun ELC, siendo las principales características de ellos las siguientes:

- Análisis de voz y parametrización: 20 MFCC por cada 10 ms de voz.
- Modelado acústico. Estando probada sobradamente la potencia del modelado HMM (“hidden Markov modelling”) para el reconocimiento de voz, en los sistemas aquí descritos se ha hecho uso de la variante semicontinua (SCHMM).
- Flujo del servicio: autómata de estados finitos preestablecido para los sistemas STACC y CABE y módulo de diálogo para el de viajes.
- Modelado del lenguaje: autómata de estados finitos obtenido automáticamente a partir del conjunto de frases aceptado en el servicio correspondiente (sistemas STACC y CABE) o bigramática de clases de palabras.
- La interacción sistema→usuario en el caso de los sistemas STACC y CABE se ha llevado a cabo mediante frases pregrabadas, mientras que el sistema de consulta remota de información de viajes se utiliza un módulo TTS.

Tras algún tiempo en funcionamiento activo, STACC ha demostrado ser un servicio de gran utilidad y buenos resultados [9]. En este sentido, hemos de indicar que si bien la evaluación de los sistemas hablados en general resulta una tarea difícil dado que se deben tener en

consideración diversos aspectos, desde el punto de vista del usuario del servicio existen dos de especial relevancia

- Robustez en el reconocimiento de voz. El usuario tolera una interacción hombre-máquina rígida, pero no un porcentaje de reconocimiento inferior a aproximadamente el 90%.
- Naturalidad de la respuesta vocal proporcionada por el sistema, siendo preferible la utilización de frases pregrabadas que la consideración de un módulo de síntesis de voz de baja calidad aunque éste permita una mayor flexibilidad en la interacción.

Referencias

- [1] S. Cox, R. Rose: "Confidence Measures for the Switchboard Database". Proc. on ICCASP, vol. 1, pp. 511-514. Mayo, 1996.
- [2] J. Díaz: "Reconocimiento de Voz Continua Mediante una Aproximación Híbrida Basada en SLHMM". Tesis doctoral. Noviembre, 1995.
- [3] P. García: "Reconocimiento de Voz Continua Basada en Técnicas MVQHMM". Tesis doctoral. Febrero, 1996.
- [4] P. García, J.C. Segura, J. Díaz-Verdejo, A. Rubio, M.C. Benítez, A. Peinado, J.M. López-Soler, J.L. Pérez, V. Sánchez, A. de la Torre, R. López-Cózar, "Speech Recognition and the New Technologies in Communication: A Continuous Speech Recognition-based Switchboard with an Answering Module for E-mailing Voice Messages". LREC-97, vol. 1, pp. 1263-66. Septiembre, 1997.
- [5] C.H. Lee: "Stochastic Modeling in Spoken Dialogue System Design". Speech Communication, pp. 311-322. Diciembre, 1994.
- [6] R. López-Cózar, Pedro García, J. Díaz, Antonio J. Rubio: "A Voice Activated Dialog System for Fast-Food Restaurant Applications". EUROSPEECH-97, vol. 4, pp. 1783-86. Septiembre, 1997.
- [7] H. Ney, U. Essen, R. Kneser: "On Structuring Dependences in Stochastic Language Modelling". Computer Speech and Language. Vol. 8, pp. 1-38, 1994.
- [8] R. Pieraccini, E. Levin: "A Learning Approach to Natural Language Understanding". Chapter in Speech Recognition and Coding. New Advances and Trends. NATO ASI Series-F, vol. 17, pp. 139-156. 1995.
- [9] A.J. Rubio, P. García, A. de la Torre, J.C. Segura, J. Díaz-Verdejo, M. C. Benítez, V. Sánchez, A.M. Peinado, J.M. López-Soler, J.L. Pérez-Córdoba: "STACC: an automatic

- service for information access using continuous speech recognition through telephone line". Proceedings of EUROSPEECH-97, vol. 4, pp. 1779-82, Septiembre. 1997.
- [10] J. C. Segura, A. Rubio, V. Sánchez, J. L. Pérez, A. M. Peinado, J. M. López, P. García, J. Díaz, M. C. Benítez: "Conversor Texto-Voz Para el Castellano". SEPLN, boletín no. 14. Marzo, 1994.

Interacción a través del Lenguaje Natural

L. A. Ureña, M. García, J. R. Balsas
M.C. Díaz, A. Montejo, F. Martínez
Departamento de Informática
Universidad de Jaén, Av. Madrid 35, Jaén
Tlf: 953.212445 Fax: 953.212420 e-mail: laurena@ujaen.es

Resumen

El objetivo del PLN es facilitar la interrelación persona-ordenador de manera que posibilite una comunicación más natural, así como mucho más fluida y menos rígida que los lenguajes formales. En este trabajo se describen brevemente las tareas de acceso avanzado a la información mediante lenguaje natural, así como su relación con la interacción persona-ordenador. La recuperación de información, la categorización de texto, la resolución de la ambigüedad léxica y el encaminamiento de texto en la clasificación son tareas del PLN que facilitan la comunicación persona-ordenador y pueden ser resueltas aplicando un mismo método. Como ejemplo, describimos someramente la creación de un sistema de ayuda con consultas en lenguaje natural, aplicable automáticamente a sistemas de información cuyas fuentes de información de ayuda sean textuales, independientemente de su presentación.

Palabras clave: Procesamiento de Lenguaje Natural (PLN), Interacción persona-ordenador (IPO), Lenguaje Natural (LN), Recuperación de Información (RI), Categorización de texto (CT), desambiguación del significado de palabras (WSD).

1 Introducción

El lenguaje natural es uno de los aspectos fundamentales del comportamiento humano. "Vale más una imagen que mil palabras" [1]. Sin embargo, la imagen se dirige a los sentidos mientras que la palabra al entendimiento. Las representaciones iconográficas mediante interfaces gráficas tienen sus campos de aplicación propios y naturales. Pero no eliminan, ni mucho menos, la necesidad de comunicación en lenguaje natural entre las personas y las máquinas. La tecnología multimedia permite que la interacción con el ordenador pueda contener texto, figuras, imagen, voz y sonidos, pero el lenguaje natural tanto hablado como escrito, es más que nunca un vehículo indispensable de la información [2]. Así podemos hablar de interfaces multimodales.

Los métodos existentes actualmente de comunicación con las computadoras son prácticamente gráficos, cuando queremos ordenar algo, lo más normal es que lo hagamos con una combinación adecuada de movimientos y pulsaciones de ratón. Esto es así por la facilidad de las computadoras para manejar gráficos y es la forma de trabajar con un sistema informático, pero conducido por éste. Proporcionará lo que se le pide, si lo tiene, a través de lo que ofrezca mediante un despliegue descendente de sus posibilidades, mostradas mediante iconos.

Sería bueno abrir nuevas formas de diálogo entre los ordenadores y el ser humano. Las ventajas de contar con una interfaz en LN radican en la flexibilidad y accesibilidad que le ofrecen al usuario no experto u ocasional a un sistema informático, liberándolo del aprendizaje. Actualmente hay una tendencia a extender la IPO en LN con una gran diversidad de propósitos, así como a promover la automatización del tratamiento de la información en LN existente en textos.

Las interfaces persona-ordenador pueden presentar limitaciones en la libertad de uso del LN; pero, si están limitadas a un dominio específico, el usuario no debe sentir esas limitaciones. Sin embargo, a pesar de la ambigüedad del lenguaje natural para representar el significado de las oraciones, la desambiguación del significado de las palabras está representando un papel importante, de manera que hace de este lenguaje un lenguaje más preciso, independientemente del dominio. Se han desarrollado desambiguadores [3] de ámbito general para mejorar las tareas de clasificación automática de documentos. Gracias a éstos se podrá resolver la ambigüedad en la misma interacción haciendo más precisa ésta.

2 Aplicaciones del procesamiento del lenguaje natural

La información textual hoy por hoy es la predominante, gracias sobre todo a Internet, así como también a los medios de almacenamiento masivo de datos, por lo que la cantidad de información a la que una persona puede tener acceso crece exponencialmente. Así por ejemplo, aproximadamente el 90% del total de la información que manejan las empresas y organizaciones es texto [4]. Podemos encontrar texto en documentos, circulares, manuales, informes, correos electrónicos, faxes y también en páginas web. Sólo para este último medio, hay estimaciones [5] de que la cantidad de texto disponible es del orden de un terabyte. En este escenario se entiende el creciente interés por los sistemas de acceso a la información y, en general, por los sistemas de clasificación automática de documentos. Estos sistemas realizan diferentes operaciones de clasificación, basándose en el análisis del contenido del texto de los documentos que procesan.

Dentro de este tipo de sistemas podemos incluir los sistemas de recuperación de información o de texto [6] que seleccionan, en grandes colecciones de textos, aquellos textos o documentos que son adecuados a una necesidad del usuario [7]. El sistema

selecciona los documentos que considera relevantes a dichas necesidades, expresadas normalmente en forma de consulta. La relevancia se decide en función de la frecuencia de aparición en los documentos de la base de datos de los términos utilizados por el usuario en la consulta.

La figura 1 muestra la salida típica de un sistema RI ampliamente utilizado en Internet como es *Infoseek*. En el ejemplo, aparecen los resultados a la consulta "human computer interaction" sobre el Web. Como vemos el sistema ha seleccionado una serie de documentos y, además, los ha clasificado asignándoles un factor de relevancia con respecto a la consulta. La interfaz del sistema presenta una lista de documentos ordenada por este factor y, para cada uno, su título y las primeras palabras del mismo, además de información adicional (tamaño, fecha de creación y lugar donde se encuentra).

Asimismo, son de especial interés los sistemas de categorización de textos que consisten en la asignación de documentos, o partes de documentos a uno o más grupos de un conjunto de ellos dado. Otro ejemplo de los sistemas de clasificación lo constituyen los de encaminamiento de texto, que pueden incluirse en sistemas de gestión de correo electrónico. Estos sistemas pueden determinar, a partir del análisis de contenido del texto de un mensaje, la dirección más adecuada a la que debe enviarse. Toda una serie de operaciones sobre documentos, como las anteriores, resultan de interés, al poder ser realizadas por los sistemas de clasificación de texto. Por otra parte, a pesar de que el lenguaje natural es ambiguo para representar el significado de las oraciones, la tarea de desambiguación del significado de las palabras está representando un papel importante, como ayuda a las citadas tareas, de manera que hace de este lenguaje un lenguaje más preciso, independientemente del dominio. Por otra parte, en cuanto a la construcción de estos sistemas, existe un conjunto de elementos comunes que permiten la implementación de las diversas aplicaciones de clasificación de documentos. Históricamente, estos elementos fundamentales fueron definidos para la implementación de un tipo concreto de sistemas de clasificación: los sistemas de recuperación de información o de texto, el tipo de sistemas de clasificación que primero se desarrolló. En concreto, tres cuestiones fundamentales que deben resolverse en todo sistema de clasificación son: el *método de representación, de análisis y cálculo de la similitud*, es decir, la forma en que serán representados internamente los documentos a partir del análisis de su contenido, determinando un valor de similitud entre los elementos representados.

Search results
eb directory topics
* Human-computer interaction

Web search results 13,157 matches [Next >](#) | [Hide summaries](#) | [Sort by date](#) | [Ungroup results](#)

- 1. Human-Computer Interaction Resources on the Net**
Jobs, events, links, organizations, and much more.
Relevance: 95% Date: 8 Jul 1998, Size 6.4K, <http://www.ida.liu.se/labs/aslab/groups/um/hc/>
[Find similar pages](#) | [Translate this page](#)
- 2. University of Maryland, Human-Computer Interaction Lab**
=> Summaries Visualization Education Digital Libraries Design Process Physical Devices Publications Online Tach Reports Video Reports Books...
Relevance: 91% Date: 23 Jun 1999, Size 44.6K, <http://www.cs.umd.edu/projects/hcii/>
[Find similar pages](#) | [More results from www.cs.umd.edu](#) | [Translate this page](#)
- 3. Human-Computer Interaction Institute**
Relevance: 90% Date: 20 May 1999, Size 2.7K, <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/hcii/www/hcii-home.html>
[Find similar pages](#) | [More results from www.cs.cmu.edu](#) | [Translate this page](#)
- 4. Usable Web**
Usable Web is a collection of resources and links on user interface issues, and usable design specific to the World Wide Web.
Relevance: 89% Date: 19 Sep 1999, Size 8.2K, <http://usableweb.com/>

Figura 1. Salida del sistema Infoseek

Además de estos conceptos suelen aparecer otros dos ligados a los anteriores. Así, se habla de método de indexación cuando incluye la representación de documentos y consultas y el análisis de sus contenidos. Mientras que por proceso de indexación se entiende el proceso de construcción de la representación interna de un conjunto de documentos y consultas.

3 Sistemas de ayuda con consulta en lenguaje natural

Como muestra de aplicación de este método, proponemos la resolución de un problema típico del PLN: la recuperación de información. Como hemos mencionado, la complejidad de los sistemas informáticos actuales motiva la creación de ayudas, a su vez más complejas. La IPO tiene aquí un extenso campo de estudio, ya que no sólo intervienen las formas gráficas sino la inteligencia artificial si se incorporan interfaces en lenguaje natural. Dado un sistema de ayuda textual, podemos considerar cada elemento de ayuda como un documento independiente. Se pretende encontrar, ante una consulta en lenguaje natural, el elemento más indicado, todo ello procediendo de una forma automática y transparente para el usuario. La aplicación del método se muestra a continuación.

El Modelo de Espacio Vectorial (MEV), descrito por Salton [7], soluciona este problema satisfactoriamente, ya que se trata de resolver un problema clásico de la recuperación de información como es la cuestión de encontrar cierta información a partir de una consulta dada en lenguaje natural. El Modelo del Espacio Vectorial fue originalmente desarrollado para la Recuperación de Información, pero provee un soporte muy adecuado para realizar otras tareas como WSD o TC. También, el modelo está avalado por muchas experiencias en recuperación de texto [6, 7]. De hecho, el MEV es un entorno muy adecuado para expresar nuestro enfoque ya que permite fácilmente obtener una medida de similitud entre documentos.

Dado que la comunicación se hará en lenguaje natural, debemos crear un espacio vectorial de dimensión n , siendo ésta el número de palabras distintas en todos los textos de la ayuda. Para aplicar el MEV en este problema debemos representar cada elemento de la ayuda como un vector. La respuesta a cualquier consulta será un texto acordado de antemano compuesto por un número indeterminado de palabras. Muchas de ellas no tendrán ningún significado semántico en la respuesta, como pueden ser los artículos, las preposiciones y aquellas palabras que, aún teniendo significado, son tan comunes que no discriminan, es decir, que aparecen en prácticamente cualquier documento, por lo que pueden ser eliminadas del cálculo. Sin embargo, con el resto de palabras, podemos formar un vector con el peso, calculado como en [3], de cada palabra como componente de cada dimensión.

Este modelo representa la estructura del sistema de ayuda y permite resolver fiablemente cuestiones descritas en lenguaje natural. Cada pregunta que se quiera hacer a nuestro sistema debe ser transformada de la misma forma que el modelo. De esta forma la consulta será representada como un vector, dependiendo de sus palabras, y podremos calcular su similitud con cada uno de los vectores del modelo, encontrando fácilmente el elemento de ayuda necesario que será el de similitud mayor.

4 Conclusiones

Hemos presentado una breve descripción de algunas tareas de PLN que pueden ayudar a mejorar la interacción entre la persona y el ordenador. Asimismo, la ingente cantidad de recursos lingüísticos (WordNet [8], EuroWordNet [9], etc.) que se han compilado en los últimos años, así como las herramientas desarrolladas para gestionar grandes cantidades de información han supuesto un cambio radical en las tendencias del PLN en los últimos años potenciando los sistemas a gran escala. La aplicación de estas herramientas sin duda mejorarán la interacción, tanto escrita como hablada.

Está claro que el LN debe integrarse con otras formas de comunicación (interfaces gráficas, sistemas hipermedias, etc.). En muchas ocasiones el LN sigue siendo la forma más expresiva y natural de interactuar con un sistema informático o de suministrar información. Por tanto, el nuevo problema que se plantea es combinar adecuadamente las formas de

interacción, integrando la tecnología y los usuarios. Esto será un factor fundamental del éxito.

5 Referencias

- [1] A. Vaquero y M. Buenaga, "Aplicaciones de las Bases de Datos Léxicas en la Clasificación Automática de Textos", Curso de la XVIII Escuela de Verano de Informática, 1996.
- [2] Verdejo, M.F. y Gonzalo J. "Del procesamiento del lenguaje natural a la ingeniería lingüística: ¿dónde nos encontramos?". Novática nº 138, 1998.
- [3] L.A. Ureña, M. Buenaga, M. García y J.M. Gómez, "Integrating and evaluating WSD in the adaptation of a lexical database in text categorization task", Proceedings of the First Workshop on Text, Speech, Dialogue, 1998.
- [4] Oracle, "Managing Text with Oracle8TM ConText Cartridge", An Oracle Technical White Paper, 1997.
- [5] Baeza-Yates, R. y B. Ribeiro-Neto, "Modern Information Retrieval". ACM Press Books, Nueva York, Página 367, 1999.
- [6] Lewis, D., "Representation and learning in information retrieval". Ph.D. Thesis, Department of Computer and Information Science, University of Massachusetts, 1992.
- [7] Salton, G. y M.J. McGill. "Introduction to Modern Information Retrieval". McGraw-Hill, New York, 1983.
- [8] Miller G., "WordNet: lexical database". Communications of the ACM Vol 38, No. 11, 1995.
- [9] P. Vossen y otros, "The Restructured Core WordNet in EuroWordNet: Subset1", Deliverable D014, D015, WP3, WP4. EuroWordNet, LE2-4003, 1998.

Una Técnica de Especificación Espacial para Diseño de Sistemas Interactivos

M.L. Rodríguez, M. Gea
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada, Av. Andalucía 38, Granada
Tlf: 958.242.811 Fax: 958.243.179 e-mail: <mlra, mgea>@ugr.es

Resumen

Hoy en día, la mayoría de los sistemas interactivos gestionan la información en un entorno gráfico utilizando como estilo de interacción el paradigma de manipulación directa. Este estilo de interacción se caracteriza por una interfaz de usuario amigable que ofrece un modelo visual del mundo, compuesto de objetos y acciones. Los objetos poseen una representación gráfica visible por el usuario, que representa su interfaz y suministra información sobre su estado (seleccionable, activo, etc). Además, los objetos tienen capacidad manipulativa (moverse, soltarse, etc) asociada a acciones que el usuario realiza mediante los dispositivos de entrada (ratón, teclado, etc).

Palabras clave: Técnicas formales aplicadas a sistemas interactivos, Diseño de Sistemas Interactivos, Técnicas de Interacción, Especificación de relaciones espaciales

1 Una Técnica de Especificación Espacial para Diseño de Sistemas Interactivos

Todas las características de estos sistemas facilitan la gestión de la información por parte de los usuarios, sin embargo, complican su diseño, programación e implantación. Ante esta situación, las técnicas de especificación formal [1,2] pueden ayudar en la descripción del sistema, centrándose en los aspectos relevantes que definen el modelo de interacción.

En la actualidad, los sistemas de manipulación directa [3] existentes se basan en mecanismos simples de acciones de arrastrar y soltar sobre los objetos (*drag&drop*). Los lenguajes de programación y entornos actuales permiten crear aplicaciones nuevas basadas en este paradigma, sin embargo, estas aplicaciones presentan limitaciones debido a que disponen de un conjunto de eventos muy restrictivo. Por lo tanto, es necesario extender el concepto de entornos interactivos espaciales para diseñar comportamientos más complejos basados en relaciones espaciales (a la derecha, en medio de, etc.).

Al extender las características del estilo de manipulación directa mediante el uso de técnicas de especificación espacial para describir relaciones complejas entre objetos [4,5,6], necesitamos considerar los objetos (componentes del sistema con capacidad de manipulación) y las relaciones que definen los grados de libertad que suministran los objetos para su manipulación.

A continuación, nos centraremos en describir algunas de las características básicas de los objetos y relaciones, obteniendo como consecuencia una caracterización de un sistema de manipulación directa y un modelo de especificación para estos sistemas.

1.1 Estudio de los objetos

Los objetos constituyen el soporte de la aplicación ya que son los elementos motivo de la interacción, tienen naturaleza propia y las siguientes características:

- **Representación física:** Todo objeto tiene asociada una representación gráfica que es visible por el usuario, constituye la interfaz del objeto y suministra información sobre su estado.
- **Ubicación espacial:** Cualquier objeto ocupa una posición en el espacio.
- **Estado:** Todo objeto de la aplicación se encuentra en cada momento en un estado posible dentro de un conjunto de estados definidos de antemano.
- **Acciones:** Cada objeto posee una funcionalidad dentro del contexto de la aplicación que está ligada a las acciones que el usuario realiza sobre los objetos, manipulando directamente su representación gráfica. Realizar una acción sobre un objeto supone un cambio de estado del objeto.

El sistema va a estar formado por un conjunto de objetos **O**, cada uno de ellos con una identidad, un posible conjunto de estados y capacidad de procesamiento (acciones). Además, también consideramos los dispositivos de entrada objetos del dominio de la aplicación y los denominamos **objetos de control (D)**.

Uno de los ejemplos más extendido de un sistema de manipulación directa es un sistema de escritorio. En este caso la pantalla del ordenador es análoga al escritorio, los objetos del escritorio se representan por iconos y el dispositivo ratón nos permite manipular los iconos para realizar las acciones necesarias. Vamos a considerar un sistema de escritorio muy sencillo, compuesto por una impresora y archivos que se pueden imprimir. Por lo tanto, el conjunto **O** estará formado por un objeto impresora con dos estados posibles (imprimiendo y no imprimiendo) y una acción asociada (abrirla para ver los archivos que se están imprimiendo), y objetos archivos con dos estados posibles (seleccionado y no

seleccionado) y las acciones de moverse y seleccionarse. Además, necesitamos un objeto de control, un ratón, para manipular los objetos. El ratón tiene tres estados posibles (arrastrándose, pulsado y soltado) y tres acciones asociadas (moverlo, pulsar el botón y soltar el botón).

Como todo objeto con capacidad de manipulación tiene asociada una representación gráfica que nos permite manipularlo, para cada uno de los elementos del conjunto **O** debemos tener una única representación gráfica asociada que llamamos **icono**.

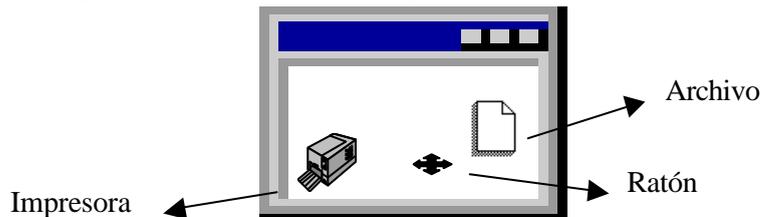


Figura 1. Ejemplo de sistema de escritorio

No existirán iconos que no estén asociados a algún objeto, y para cada estado de un objeto se considera un único icono. Por ejemplo, el icono que representa la impresora en estado imprimiendo es distinto del icono que la representa cuando no está activa.

Figura 2. Iconos que representan los dos estados posibles de la impresora



Para realizar una acción de un objeto hay que manipular su icono asociado, a este proceso lo denominamos gesto. Por ejemplo, mover un archivo supone un movimiento del ratón para colocarlo sobre el archivo, presionar el botón del ratón, moverlo a la nueva posición y soltar el botón. Todas estas actividades de interacción sólo representan un cambio de localización del icono del archivo. Puede existir más de un gesto para realizar una acción de un objeto.

De todas las características que hemos establecido podemos concluir una serie de propiedades sobre los sistemas de manipulación directa. Un sistema de manipulación directa es **coherente** si cualquier cambio realizado en la interfaz se aplica inmediatamente al objeto correspondiente. Además, un sistema de manipulación directa es **transparente** si cualquier acción sobre un objeto es observable desde la interfaz de usuario. Por último, un sistema de manipulación directa es **completamente accesible** si existe al menos un gesto asociado para cada una de las acciones definidas sobre los objetos.

1.2 Estudio de las relaciones

Cuando se realiza una acción de manipulación (gesto) se establecen relaciones entre los objetos de la aplicación (estar encima de, a la derecha de, etc), e incluso entre un objeto y sus atributos (al seleccionar un objeto que éste cambie de color). La mayoría de estas relaciones vienen dadas por la dimensión espacial del dominio de los objetos en base a su ubicación (en el plano) y extensión (área que ocupan).

En el ejemplo de sistema de escritorio un gesto para seleccionar un archivo supone una **relación gesto** (relaciones entre iconos y objetos de control que proporcionan información sobre la realización de un gesto) del objeto de control ratón (arrastrar el ratón), una **relación topológica** (relación entre iconos obtenidas por su localización espacial) entre el ratón y el archivo seleccionado (colocar el ratón encima del archivo), y una **relación propiedad** (relaciones que comprueban propiedades que se dan entre objetos del sistema) del archivo para comprobar que se ha seleccionado (por ejemplo, ha cambiado de color).

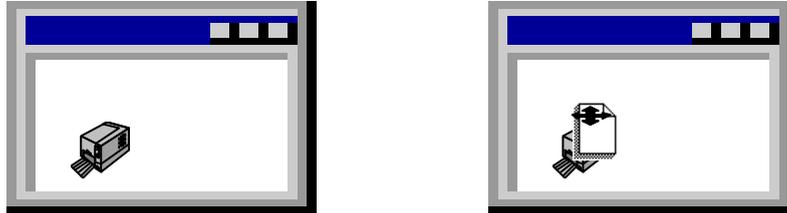
2 Especificación espacial de sistemas interactivos

En un sistema de manipulación directa el usuario accede a la funcionalidad del sistema a través de los gestos. Estos permiten al usuario manipular los objetos e interactuar con el sistema, y son la única forma que tiene el usuario para manipular el sistema. Cuando se produce un gesto en el sistema se establecen relaciones entre los objetos del dominio de la aplicación y un objeto de control. Además, también se dan relaciones nuevas entre los objetos del sistema, las cuales definen el comportamiento del mismo.

En la especificación de un sistema de manipulación directa tenemos que definir las relaciones (topológicas, gesto y propiedad) que se establecen entre los objetos del sistema como consecuencia de los procesos de manipulación del usuario. El cumplimiento de esas relaciones supondrá comportamientos o acciones definidas en el sistema. Y por último, debemos especificar la conexión entre las relaciones y el comportamiento del sistema. Es decir, el comportamiento del sistema (conjunto de acciones del sistema) será disparado por las relaciones que se producen entre los objetos del sistema, definiendo la semántica del mismo.

Siguiendo con el ejemplo del sistema de escritorio las relaciones que se tienen que satisfacer para implicar la acción del sistema imprimir un archivo son:

Figura 3. Impresión de un archivo



Imprimir un archivo

$R_{imprimir}(\text{archivo}) ::= \exists a \in \text{dom}(\text{archivos}), \exists i \in \text{dom}(\text{impresoras}), / \hat{A}_{mover}(a) \wedge (a \sim i)$

Para imprimir un archivo, primero se mueve el archivo (relación gesto \hat{A}_{mover} del objeto archivo), y se posiciona encima de la impresora (relación topológica $(a \sim i)$). Y la regla que nos describe la acción de imprimir un archivo que se produce en el sistema cuando se satisface un conjunto de relaciones entre los objetos de dicho sistema, es:

$R_{imprimir}(\text{archivo}) \text{ } \hat{P} \text{ } \text{Imprimir un archivo}$

3 Conclusiones y trabajos futuros

En este trabajo hemos expuesto las características de un sistema de manipulación directa, y descrito un modelo teórico alternativo para especificar los sistemas de manipulación directa en base a los objetos y las relaciones que se establecen entre ellos. A partir del modelo teórico podemos estudiar algunas propiedades de los sistemas de manipulación directa como coherencia, transparencia y accesibilidad.

Este modelo nos permite identificar el conjunto de objetos de la aplicación, y considera la manipulación del usuario sobre ellos como mecanismos de interacción propios de los objetos. Todas estas características nos proporcionan un nivel de abstracción mayor que el obtenido con otros métodos de especificación como UAN [7] y Diagramas de transición de estados [8]. UAN utiliza conceptos similares para describir las tareas pero considera un conjunto de relaciones topológicas muy restringido.

Actualmente hemos desarrollado un prototipo que nos permite trasladar una especificación basada en relaciones espaciales a código Tcl/Tk [9]. Los trabajos futuros se centrarán en automatizar la implementación de especificaciones de sistemas descritos con este modelo mediante el desarrollo de las herramientas apropiadas. Además, se estudiarán en más

profundidad las relaciones en su dimensión espacial y temporal, y su extensión a tres dimensiones, de modo que permita especificar mediante relaciones topológicas las interacciones de un sistema de realidad virtual.

Referencias

- [1] Harrison, M.; Thimbleby, H.; (eds): "Formal Methods in Human-Computer Interaction". Cambridge University Press, 1990.
- [2] Palanque, P.; Paterno, F.; (eds): "Formal Methods in Human-Computer Interaction". Springer, 1997.
- [3] Shneiderman, B.: "Designing the User Interface". Addison-Wesley Publishing Company. Second Edition, 1992.
- [4] Gea, M.; Rodríguez, M.L.; Gutiérrez, F.L.: "Técnicas de Especificación Espacial de Sistemas Interactivos". Actas del IX Congreso Español de Informática Gráfica CEIG'99. Jaén, 1999.
- [5] Rodríguez, M.L.; Gea, M.; Gutiérrez, F.L.: "Towards Spatial Specification of Interactive System". EUROGRAPHICS'99. State the art Report. ISSN: 1017-4656. B. Falcidieno, J. Rossignac (Eds) Milán, 1999.
- [6] Rodríguez, M.L.; Garvía, E.; Gea, M.: "A Spatial Specification Technique for Interactive System". Design, Specification and Verification of Interactive System (DSVIS'2000). Pendiente de publicación.
- [7] Hartson, H.R.; Mayo, K.: "A framework for precise, reusable task abstraction". Design, Specification and Verification of Interactive System (DSVIS'96). Springer Verlag EG, 1996.
- [8] Foley, V.D.: "Computer and Graphics, 2ª ed". Addison Wesley, 1990.
- [9] Kamal, H.M.: "Desarrollo de una librería de Manipulación Directa de Objetos". Proyecto fin de Carrera. Dpt. Lenguajes y Sistemas Informáticos. Universidad de Granada, 1998.

Mas allá de la manipulación directa. Hacia nuevos paradigmas de interacción

Jesús Lorés, Montse Sendín, Carles Aguiló

Departamento de Informática e Ingeniería Industrial.

Universidad de Lleida, C/ Jaume II 69, Lleida

Tlf: 973.702.700 Fax: 973.702.702

e-mail: {jesus, msendin}@eup.udl.es

Resumen

En los proyectos de investigación llevados a cabo en los últimos años hemos estado desarrollando sistemas interactivos aplicados en una proporción importante al patrimonio cultural y natural, utilizando estilos de interacción de manipulación directa, tiempo real y realidad virtual. Podemos citar, entre otros, el proyecto Studium (reconstrucción de la Universidad medieval de Lleida), Vilars (sistema interactivo de un yacimiento de la Edad de Hierro aplicado a la gestión del registro arqueológico), o el caballero y el ajedrez que propone un viaje en el tiempo y en el espacio alrededor de unas piezas de ajedrez del siglo X. En este artículo presentamos una exploración de nuevos paradigmas de interacción: interacción asistida, computación ubicua y realidad aumentada en sistemas multimodales aplicados a proyectos de patrimonio natural y cultural. Estos son, concretamente, el sistema de información multimedia del macizo del Montsec, situado en el prepirineo de Lleida, y el esbozo de un entorno interactivo aplicado al futuro museo diocesano de Lleida.

En el primer caso se propone el uso de un agente personal con una interacción multimodal como complemento a la manipulación directa para conseguir una interacción más sencilla, y en el segundo la integración de la realidad aumentada junto con asistentes personales para la interacción con objetos museísticos enmarcados en el futuro museo diocesano y comarcal de Lleida.

Palabras clave: sistema multimodal, manipulación directa, interacción asistida, agente personal, computación ubicua, realidad aumentada, meta-museo.

1 Introducción

La Interacción Persona-Ordenador lleva cerca de dos décadas sin experimentar ningún cambio profundo en su fundamentación. La consecución de un cambio surge de la necesidad de disponer de alternativas de interacción más generales e intuitivas.

El estilo de interacción basado en apuntar, clicar y teclear (paradigma WIMP o manipulación directa), aunque constituye un estilo apropiado para numerosos campos, tenderá a ser complementado, o incluso substituído por otros paradigmas en un futuro próximo. Esto es así, sobretodo, si se pretende extender aún más el uso del computador e integrarlo en la casi totalidad de las actividades de la vida cotidiana en las que intervengan usuarios diversos y no necesariamente especializados en el uso de computadores.

Las técnicas de interacción deben adaptarse y especializarse para facilitar esas actividades, sin que ello suponga un cambio de actitud considerable para aquel usuario que no esté familiarizado con el uso del ordenador.

La esencia de los nuevos paradigmas radica en trasladar la computación, hasta ahora focalizada en el ordenador, al mundo real, de forma que se pueda capturar y extraer información de cualquier movimiento o acción del usuario en su actividad normal para tener conocimiento constante de las situaciones por las que éste pasa, a fin de poder asistirlo si es preciso.

La zona interactiva deja de estar limitada a la pantalla. A partir de ahora hablaremos de "espacio interactivo", en el que los objetos de la vida cotidiana dejan de ser entes inanimados que permanecen pasivos y estáticos para convertirse en sistemas activos que reaccionan y participan en la escena, pudiendo resultar incluso útiles y eficientes, puesto que son conscientes de su entorno. Ahora bien, para conseguirlo, es de vital importancia el diseño de una interfaz adecuada entre el humano y esta nueva generación de máquinas.

Se empieza a concebir un mundo donde la frontera entre objetos animados (en el sentido de interacción) e inanimados cada vez se difumina más, y los objetos que nos rodean se comportan como útiles asistentes.

Por otro lado, la interacción deja de ser una actividad explícita, que tenga que ser ejercitada expresamente, para convertirse en una actividad que, sin indicación expresa, se produce de forma implícita.

Esta nueva concepción de la interacción hace necesario considerar y explotar las capacidades humanas naturales en el espacio de interacción; particularmente, la comunicación, el conocimiento y las habilidades de percepción. Éstas deben entrar en combinación con las capacidades de razonamiento y percepción artificial. Esto requiere una integración de la tecnología a varios niveles (reconocimiento de voz y sonido, visión por computador, animación gráfica, percepción basada en el tacto, aprendizaje, etc). En definitiva, lo que se pretende es dotar al ordenador de las capacidades de la percepción humana para que pueda ser consciente de todo lo que acontece.

A este despliegamiento de medios tecnológicos se le denomina *realidad aumentada*, y conviene remarcar que su propósito es el de crear interfaces más intuitivas y naturales.

El resto del documento está dividido en cuatro secciones más. En la sección 2 se presentan los distintos estilos de interacción. Partiendo de la manipulación directa, sobradamente conocida, y que constituye el estilo de partida a mejorar o suplantar, se definen las distintas alternativas (interacción asistida, computación ubicua y realidad aumentada), proporcionando así una perspectiva de la evolución que los entornos interactivos van sufriendo. La combinación de todas estas alternativas en un mismo sistema da lugar a lo que se denomina como interacción multimodal.

A continuación, en la sección 3, se presentan dos ejemplos de aplicación: el proyecto Montsec, que recurre a una interacción asistida, y el museo diocesano comarcal donde, a través de un sistema multimodal, se pretende virtualizar la visita al mismo.

Para concluir, se presenta una sección de conclusiones y, por último, la sección de referencias bibliográficas.

2 Recorrido por los distintos paradigmas de interacción

2.1 Manipulación directa

Las interfases de manipulación directa presentan una representación visual de objetos físicos o conceptuales y permiten al usuario ejecutar acciones sobre ellos para cambiar su estado, el cual aparece reflejado en la interfase. Existe una relación uno a uno entre las acciones explícitamente invocadas por el usuario y tales cambios [1].

En este estilo de interacción el usuario debe explicitar todas las tareas y controlar todos los eventos, lo cual le supone mucho esfuerzo. Esta forma de trabajo debe evolucionar, pues presenta importantes limitaciones. Se requieren nuevas metodologías con objeto de facilitar la interacción del usuario con la aplicación. Si se insiste en el mantenimiento de la correspondencia uno a uno entre las acciones y las capacidades de la interfase llegará un momento en que no se podrán añadir más funcionalidades a los programas.

2.2 Interacción asistida

La *interacción asistida* utiliza la metáfora del *asistente personal* o *agente* que colabora con el usuario en el mismo ámbito de trabajo, obteniendo de esta forma un entorno *cooperativo* en el que ambos se comunican, controlan eventos y realizan tareas en paralelo, esto es, autónomamente, sin que toda la responsabilidad de dirigir la interacción recaiga sobre el usuario [2].

Además, el agente tiene siempre asociadas unas cualidades y una personalidad y, en ocasiones, también una visualización que le permite hacerse visible en la aplicación. Cuando es así, suele aparecer con una representación antropomórfica que utiliza varias expresiones para mostrar diversas actitudes i/o estados por los que pasa. Consideramos interesante proporcionar una apariencia física acorde con su personalidad. Por lo tanto, la caracterización a otorgar al agente debe estar minuciosamente estudiada para que se corresponda con su carácter y sus potencialidades [2].

En este tipo de interacción los objetos de la interfase de manipulación directa pueden verse afectados, aunque no necesariamente siguiendo una relación uno a uno con respecto a las acciones del usuario. El agente captura y analiza las entradas del usuario durante un largo periodo de tiempo antes de decidir la realización de una acción determinada. Asimismo, una sola entrada del usuario puede provocar una serie de acciones por parte del agente. Se trata, por lo tanto, de un programa autónomo que trabaja en un segundo plano como complemento a los mecanismos de manipulación directa típicos de un aplicación interactiva multimedia [3]. La idea de utilizar agentes en la interfase fue introducida por visionarios como Negroponte [4] y Alan Kay [5].

2.3 Computación Ubícua

En 1991 Mark Weiser (Xerox PARC) publicó un artículo sobre su visión acerca de la Computación Ubícua [6], ilustrando un nuevo paradigma de computación.

La idea de la *computación ubicua* es que la capacidad de información está en todas partes (omnipresencia de la computación), en forma de pequeños dispositivos muy diversos (*appliances of information*) que permiten interacciones de poca dificultad, conectados en red a servidores de información. El diseño y localización de estos dispositivos debe ser ideado especialmente a la tarea objeto de interacción.

En este estilo de interacción el ordenador queda delegado a un segundo plano, intentando que resulte invisible al usuario, quedando resuelta así la computación de forma transparente al usuario.

2.4 Realidad Aumentada

La *realidad aumentada* es un estilo de interacción que trata también de reducir las interacciones con el ordenador utilizando la información del entorno como una entrada implícita. Esto se consigue acoplando información digital a los objetos físicos y entornos de la vida cotidiana. Se dice que de esta forma la interacción del usuario con el ordenador queda “aumentada” con la información digital que estos objetos proporcionan [7].

Por ejemplo, la situación del usuario será automáticamente reconocida utilizando un amplio conjunto de métodos de reconocimiento (tiempo, posición y reconocimiento de objetos utilizando la visión por ordenador o dispositivos de detección personal). Por lo tanto, el centro de atención del usuario deja de ser el ordenador para pasar a ser el mundo real.

En un sistema de realidad aumentada el computador proporciona información adicional que incrementa o “aumenta” el mundo real, en lugar de reemplazarlo por un entorno completamente virtual (realidad inmersiva).

Se pretende que sea el propio mundo físico el que se encargue de interactuar con el usuario, de forma que la interacción se resuelva sin que el usuario altere su actividad normal, ni realice ninguna operación explícita sobre el computador.

En síntesis, de lo que se trata es de descubrir cómo fusionar el mundo real con el mundo computacional, haciendo uso de todos los medios tecnológicos disponibles.

No obstante, otro aspecto importante es la disposición de los objetos interactivos. Éstos no deben estar localizados únicamente en un primer plano. Así como subconscientemente la gente está constantemente recibiendo información de la periferia sin que se le preste atención explícita, de la misma manera en un sistema aumentado la periferia debe ser otro factor integrante que tenga la capacidad de interactuar con el usuario si es preciso y éste sea consciente de ella, constituyendo otra fuente de realimentación de información. Éste es, por tanto, otro aspecto que requiere atención especial.

Ishii y Ullmer presentan en [7] sistemas prototipo para explorar el uso de objetos físicos como un medio para manipular bits (información digital) en el centro de atención del usuario. Son el metaDESK y el transBOARD. Por otro lado, presenta también otro sistema prototipo centrado en la periferia de la percepción humana: el ambientROOM, en el que entra en juego información ambiental diversa (luz ambiental, sombras, sonido y flujo del aire, entre otros).

3 Ejemplos de aplicación

3.1 El Proyecto MONTSEC

Como ya ha sido apuntado anteriormente, la introducción del agente en una aplicación multimedia interactiva supone una evolución en las posibilidades de interacción del usuario con la aplicación que permitirá reducir el esfuerzo necesario para obtener la información en la que se está interesado. Estamos trabajando en la integración del agente personal en el proyecto Montsec.

El Montsec es una cadena montañosa que se encuentra ubicada en el prepirineo de Lleida. La riqueza geológica, histórica y paleontológica, así como el paisaje, la flora y la fauna del Montsec y alrededores, hacen de éste un verdadero laboratorio natural muy apreciado por los especialistas, pero desgraciadamente poco conocido por el público en general. Se pretende dar a conocer este patrimonio natural alrededor del mundo gracias al contenido, la interactividad y las posibilidades de una aplicación multimedia. La aplicación ofrecerá información estructurada del Montsec y alrededores, desde hace 200 millones de años hasta nuestros días, con el fin de guiar al usuario, permitiendo su navegación a través de hiperenlaces.

Se pretende, por tanto, obtener una interacción más sencilla. El agente, que dispondrá de una base de conocimientos con toda la información referente a la aplicación, será un experto idóneo para intervenir en este tipo de aplicación. Responderá a preguntas directas del usuario (como, por ejemplo, proporcionar una ruta de interés), mostrará sus estados de ánimo y se comunicará con la aplicación para mostrar la información que el usuario requiera, convenientemente clasificada según el interés del usuario. Respecto a la comunicación entre el usuario y el agente se prevee poder utilizar el reconocimiento de voz.

La arquitectura que proponemos ha sido diseñada para que los agentes puedan recibir tareas en forma de “retos” explícitamente descritos y realicen su actuación a partir de un amplio conocimiento sobre el mundo en el que habitan.

Su diseño responde a una arquitectura *cooperante*, con el fin de proveer los mecanismos que se consideran necesarios para la comunicación correcta entre la aplicación y el agente.

En la figura 1 se muestra un esquema del sistema global donde aparecen reflejadas todas las conexiones existentes con respecto a los módulos más importantes.

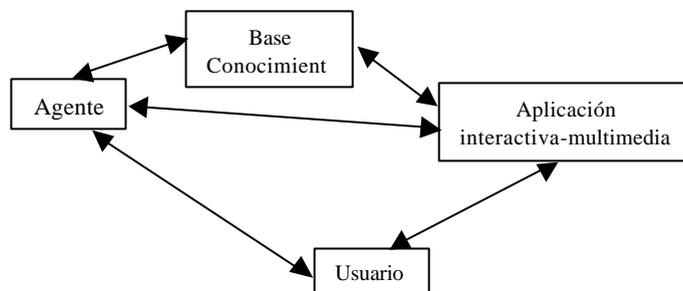


Figura 1. Sistema con interacción asistida integrada

Como puede verse, existe una comunicación tanto entre el agente y el usuario como entre el agente y la aplicación. El primer tipo de comunicación consiste en una comunicación síncrona y directa entre agente y usuario.

La comunicación desde el agente a la aplicación permite al agente modificar el entorno principal de la aplicación, en función de las decisiones que haya tomado tras un análisis previo de las entradas del usuario. El otro sentido de comunicación permite al agente conocer de forma exhaustiva las acciones que realiza el usuario en la aplicación y que le sirven para llevar a cabo la toma de decisiones y la evaluación del comportamiento del usuario. Se trata de señales que se reciben asincrónicamente y de forma transparente al usuario (en *background*). De esta forma se consigue que todas las acciones que realiza el usuario sobre el entorno sean directa o indirectamente conocidas y procesadas por el agente.

El conocimiento sobre el entorno está almacenado en una *base de conocimientos*, la cual constituye la pieza central de nuestra arquitectura. Tiene como objetivo prioritario facilitar la información necesaria al agente referente a la aplicación. Su contenido se soporta a través de unas estructuras básicas (predicados, frames y metarreglas), y de una serie de relaciones entre estas estructuras, las cuales aportan las capacidades necesarias para obtener una información razonada.

Para poder explotar el conocimiento almacenado se utiliza un *motor de inferencia* que dispone de diferentes mecanismos para inferir resultados. El primero de ellos es el mecanismo de comparación entre las palabras de entrada del usuario y las que se disponen en la base de conocimientos. A continuación se aplica un mecanismo de filtrado *compensatorio* basado en la relación existente entre los frames obtenidos y los rasgos característicos del agente. Recibe este nombre porque, en lugar de aplicar un filtrado restrictivo con respecto al carácter del agente, se realiza una ponderación teniendo en cuenta también el peso del frame obtenido después del primer filtrado. Por último se aplica un mecanismo de filtrado deductivo a partir de las reglas que son satisfechas por algunos de los frames obtenidos en el proceso anterior.

Conviene destacar también que resulta imprescindible almacenar un histórico referente a un episodio de la vida del agente, con objeto de ajustar sus acciones a las circunstancias vividas y evitar repeticiones en la información facilitada al usuario.

3.2 El museo virtual diocesano y comarcal de Lleida

El museo es una institución dedicada a preservar e interpretar la evidencia primaria y tangible de la humanidad y su entorno. En el museo, el objeto, en muchos casos desplazado en tiempo, lugar y circunstancias de su contexto original, tiene la oportunidad de

comunicarse directamente con el visitante de una manera inviable a través de otros medios [8].

Por otro lado, entendemos por modelo virtual una representación de algunos (no necesariamente todos) los elementos de una entidad abstracta o concreta. El propósito del modelo es permitir que podamos entender la estructura o la conducta de la entidad que representa, permitiéndonos "experimentar" con esa entidad, modificándola de acuerdo con nuestros propósitos.

Un *museo virtual* constituye un modelo virtual donde la entidad a representar es el conjunto de objetos museísticos más relevantes de un museo determinado, junto con toda la información subyacente a ellos.

En el concepto de museo virtual el centro de atención deja de estar en los objetos para pasar a estar en la información vinculada a ellos. Esta información que, por supuesto, es interactiva, podrá ser tanto presencial como a distancia.

Proponemos el diseño de un entorno multimodal aumentado en el diseño del futuro museo diocesano de Lleida en el que la contemplación de un objeto por parte del usuario se vea asistida por un agente personal. Además, pretendemos explotar todas las posibilidades multimedia, relacionando objetos dinámicos que son transformados interactivamente por los usuarios por medio de texto, imágenes, sonido, bases de datos, sistemas expertos, etc, los cuales permiten ir más allá de las apariencias. En una realidad aumentada podemos preguntar directamente a un objeto, a un edificio, o a un paisaje.

En esta línea de actuación el trabajo de Brogni [9] nos presenta la visualización de objetos 3D en un entorno ubícuo y que cambia en función de la atención que le presta el usuario. Este sistema es sensible a la posición y orientación del usuario. La detección física y cognitiva de los usuarios constituye una de las claves de este tipo de interacción.

Por otro lado, se pretende automatizar la creación de visitas personalizadas al museo, esto es, adaptadas a cada visitante. Esto implica la reconstrucción de la relación lógica de cada visualización, que podemos considerar como una representación del conocimiento de los expertos a los intereses del visitante.

Esta idea, propuesta por Kadobayashi en [10] se conoce con el término de *Meta-Museo*. De esta forma, el museo dejará de ser el lugar donde contemplamos con respeto y admiración las evidencias de un pasado presentado de manera estática, para convertirse en un lugar donde estudiamos y aprendemos cómo es nuestro mundo, y de qué manera nuestra sociedad ha llegado a ser lo que es.

4 Conclusión

Actualmente los avances tecnológicos sufren avances constantes y considerables. Prueba de ello es que hoy en día resulta posible trazar el movimiento de objetos móviles, la identificación de personas a través de la voz y aspecto facial en tiempo real, etc, y todo ello con unos recursos computacionales modestos.

Los nuevos paradigmas de interacción extenderán la accesibilidad a la computación a usuarios de diversas edades, niveles de habilidad, estilos cognitivos, lenguajes y capacidades sensoriales y motoras, puesto que ofrece un estilo de interacción expresivo, transparente, eficiente, robusto y, sobre todo, flexible.

5 Referencias

- [1] B. Schneiderman, "Designing the User Interface". Addison and Wesley. Capítulos 3,4,5, 1997.
- [2] P. Maes, "Agentes that reduce work and information overload", Communications of the ACM Vol 37. Nº 7, july 1994.
- [3] H. Lieberman, "Autonomous interface agent", CHI'97, Atlanta, 1997.
- [4] N. Negroponte, "The architecture machine. Towards a more human environment", Mit Press, 1970.
- [5] A. Kay, "User Interface: A personal view". B. Laurel ed.. The Art of Human-Computer Interface design. Addison and Wesley. Reading Mass, 1990.
- [6] M. Weiser, "The computer for the twenty-first century", Scientific American, pg. 94-104, September 1991.
- [7] H. Ishii, B. Ullmer, "Tangible bits: Towards seamless interfaces between people, bits and atoms", CHI, 1997.
- [8] A. Morrison, "Cognitive Applications", UK Museums Association conference, 1997.
- [9] A. Brogni, "An interactive system for the presentation of a virtual egyptian flute in a real museum", Virtual reality in archaeology. Archaeopress, april 2000.
- [10] R. Kadobatahi, E. Neeter, K. Mase, R. Nakatsu, "VisTA: An Interactive Visualisation Tool for Archaeological Data", Archaeology in the age of the Internet.CAA. Edited by L.Dingwall, S.Exon, V. Gaffney, S. Laflin, M. Van Leusen. Oxford: British Archaeological Reports (Int. Series, S750), 1997.

Sesión VI

Aplicaciones y Desarrollo

Diseño Interactivo de Programas de Control Industrial Basado en Técnicas de Inteligencia Artificial

L. Castillo, J. Fernández Olivares, A. González.

Departamento de Ciencias de la Computación e I.A..
Universidad de Granada, Av. Andalucía 38, Granada
Tlf: 958.244019 Fax: 958.243317
e-mail: {L.Castillo,faro,A.Gonzalez}@decsai.ugr.es

Resumen

El diseño de programas de control industrial es una tarea difícil que normalmente se lleva a cabo por ingenieros humanos especializados. En los últimos años esta tarea está siendo automatizada gracias a la utilización de técnicas de planificación de inteligencia artificial, pero la gran cantidad de conocimiento involucrada provoca problemas de explosión combinatoria difíciles de superar. Este trabajo propone una versión interactiva de MACHINE, un planificador especializado en esta tarea, que interactúa con un ingeniero humano para evitar estos problemas de explosión combinatoria a la vez que se obtienen resultados de más calidad.

Palabras clave: Agentes inteligentes, diseño de sistemas interactivos, planificación, inteligencia artificial.

1 Introducción

Los programas de control industrial son programas de ordenador que coordinan el funcionamiento de las plantas industriales automatizadas con el objetivo de fabricar un producto a partir de sus materias primas. El diseño de programas de control industrial es una tarea que normalmente se lleva a cabo de forma manual por ingenieros de control humanos [0]. En los últimos años esta tarea está siendo automatizada gracias a la utilización de técnicas de planificación de I.A. obteniendo importantes mejoras en cuanto a tiempo de desarrollo y calidad de los programas de control diseñados de forma automática por un ordenador [0,0,0].

La planificación en I.A. se basa en técnicas de búsqueda por ordenador que permitan construir una secuencia de acciones cuya ejecución sea capaz de transformar una situación inicial en una situación objetivo (Figura 1). A los programas de ordenador capaces de planificar se les llama planificadores.

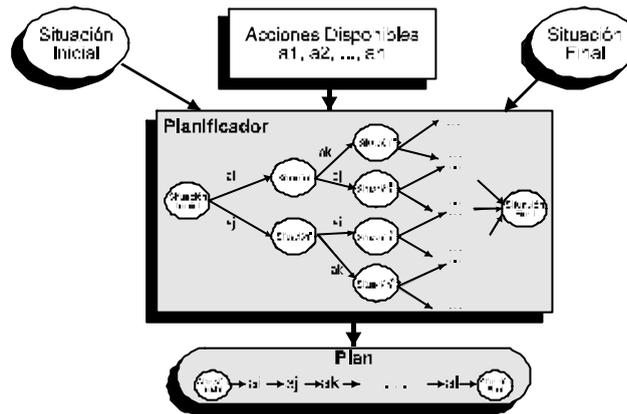


Figura 1

Este planteamiento genérico de un problema de planificación, desde el punto de vista de la I.A., es compartido por muchas tareas que se realizan de forma manual por los humanos, como por ejemplo la organización de un viaje, el diseño de un juguete o la escritura de un programa de ordenador, lo que hace que estas técnicas sean una herramienta adecuada para su automatización mediante un ordenador. En particular, en el caso del diseño de un programa de control industrial, la situación inicial consiste en una descripción de las materias primas, el objetivo es una descripción del producto fabricado y las acciones que permitan pasar de una situación a otra son las instrucciones del programa de control.

No obstante, a pesar de la potencia de cálculo y la capacidad de memoria cada vez mayores de los ordenadores actuales, el funcionamiento completamente automático de un programa planificador puede llegar a ser una tarea imposible debido a que la cantidad de conocimiento que deben manejar estos programas produciría un problema de búsqueda con un número exponencial de posibilidades que daría lugar al agotamiento de la memoria o a un tiempo de procesamiento irrealmente largo. Por esta razón, desde el campo de la planificación de I.A. se está tendiendo, cada vez más, a la construcción de lo que se conoce como planificadores con iniciativa mixta, es decir, planificadores que interactúan con un humano a la hora de resolver un problema.

2 Planificación con iniciativa mixta

La planificación con iniciativa mixta es un área interdisciplinar en la que se solapan áreas de estudio de informática, ingeniería, ciencias cognitivas y factores humanos. Su objetivo es la definición de un proceso de resolución de problemas, como por ejemplo, el diseño de un programa de control industrial, en el que un humano y un programa planificador interactúan en una especie de simbiosis durante la resolución del problema de forma que aproveche al máximo las tareas para las que son mejores cada una de las partes: mientras que la mayor responsabilidad del humano recae en resolver los aspectos más abstractos, la mayor responsabilidad del ordenador es la de resolver los aspectos con más nivel de detalle del programa de control. Es más, el papel desempeñado por cada uno de ellos no está determinado a priori, sino que se determina de forma dinámica conforme se resuelve un problema.

En la literatura sobre planificación en I.A. las aproximaciones basadas en técnicas con iniciativa mixta se centran en problemas en los que la total automatización de su resolución es inaceptable, bien por la imposibilidad de enfrentar la explosión exponencial de posibilidades [0], bien porque su resolución íntegramente por un humano es inadecuada debido a la exploración rutinaria de una gran cantidad de detalles (como por ejemplo el diseño de interfaces de usuario [0] o la planificación de operaciones de logística [0]) o porque se encuentra en el mismo borde de la capacidad humana (como por ejemplo la planificación de operaciones de despliegue militar [0,0]).

El espectro en el que pueden aparecer estas aproximaciones de iniciativa mixta varía entre sus interpretaciones más radicales, desde planificadores completamente autónomos hasta planificadores que incomodan a un humano con un número excesivo de consultas sobre decisiones aparentemente difíciles. Quizás la interpretación más adecuada sea una posición intermedia en la que el planificador selectivamente escoge las decisiones más críticas o difíciles para plantear su solución a un humano y aceptar las críticas o los consejos que éste pueda darle. Este es el enfoque que se ha escogido para este trabajo en el que se presenta una versión con iniciativa mixta de MACHINE, un planificador especialmente concebido para diseñar de forma autónoma programas de control industrial [0] que se ha demostrado ser muy útil en la solución de problemas de tamaño real [0,0] pero que requiere de la colaboración de un ingeniero de control humano para enfrentar problemas industriales complejos.

Las cuestiones a las que debe dar respuesta, no sólo este trabajo, sino cualquier trabajo que enfoque desde una perspectiva formal la construcción de un planificador con iniciativa mixta son las siguientes:

- *Control de la iniciativa.* Es necesario definir cuando recae la responsabilidad de la resolución en el operador humano y cuando en el planificador.
- *Contexto de comunicación.* Es necesario definir un medio de comunicación tal que el diálogo entre el operador humano y el planificador sin ambigüedades y que permita intercambiar la información necesaria para que haya una colaboración eficiente entre ambos, es decir, el operador humano y el planificador deben tener la misma visión del problema que se está resolviendo, lo que se podría denominar el contexto de la comunicación.
- *Información incompleta.* A menudo la comunicación entre ambas partes puede estar afectada de incertidumbre en alguno de sus parámetros, sobre todo en lo que respecta a la información transmitida por el operador humano. Es necesario que el medio de comunicación sea capaz de representar información incompleta.

3 Planificación con iniciativa mixta en MACHINE

MACHINE es un planificador autónomo capaz de diseñar programas de control para plantas industriales de tamaño real [0,0]. Generalmente, éste es un proceso en el que interviene conocimiento de ingeniería de control, química, física y electrónica. Aunque MACHINE puede representar este conocimiento como parte de su funcionamiento habitual, en el diseño de programas de control para plantas industriales complejas esto da lugar a una explosión combinatoria de posibilidades que desembocan en el agotamiento de la memoria principal⁹. Para poder resolver estos problemas se está diseñando una nueva arquitectura del programa de forma que permita la interacción con un ingeniero de control especializado.

La idea es representar sólo el conocimiento más importante sobre el problema para no provocar una explosión combinatoria de posibilidades y permitir al programa solicitar este conocimiento a un ingeniero de control sólo cuando sea necesario pero también permitir al ingeniero supervisar el funcionamiento del planificador y ser interrogado por él cuando juzgue que el proceso de resolución se esté desarrollando incorrectamente. La nueva arquitectura del planificador se muestra en la Figura 2, donde los nuevos módulos encargados de la interacción aparecen sombreados.

⁹ Ejecución en un Pentium II con 128 MB de memoria principal con sistema operativo Linux.

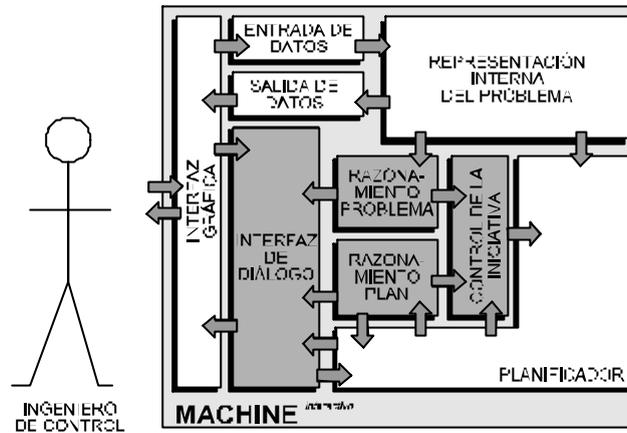


Figura 2

El funcionamiento normal de MACHINE, completamente autónomo, sólo afecta a los módulos que no están sombreados. Dado un problema en una planta industrial, éste se traduce a una representación interna basada en la lógica de predicados [0]. A partir de esta representación, MACHINE elabora un programa de control industrial y lo muestra a un ingeniero en un formalismo familiar para él [0,0], listo para ser utilizado. Los nuevos módulos que se añaden para poder interactuar con el ingeniero en el diseño del programa de control se describen a continuación.

3.1 Control de la iniciativa

Una de las cuestiones más importantes que tiene que resolver un planificador con iniciativa mixta es la definición de una política de interacción dinámica durante la resolución de un problema. Para ello, en MACHINE se utiliza un módulo de control de la iniciativa consistente en un sistema basado en reglas que identifican situaciones conflictivas durante la resolución de problemas. Por defecto, MACHINE actúa de forma autónoma, pero cuando se dispara alguna de estas reglas de control de la iniciativa significa que se ha producido una situación para cuya resolución MACHINE no dispone de conocimiento suficiente o que implica una decisión crítica y que el siguiente paso debería ser supervisado por un ingeniero.

Este es un mecanismo basado en el conocimiento para el control de la iniciativa durante el diseño de un programa de control industrial que permite adaptarse de forma dinámica a las dificultades que se pueden encontrar durante este proceso. Estas dificultades se han

dividido en cuatro categorías distintas que delimitan la responsabilidad del planificador y establecen las áreas de responsabilidad humana. Estas cuatro categorías son las siguientes.

Control de la estrategia de la búsqueda

Es un conjunto de reglas multivariable que analizan la topología del proceso de búsqueda y permiten detectar estancamientos, es decir, situaciones que se podrían calificar como que el planificador “se ha perdido”. Para ello, este conjunto de reglas se basa en una serie de variables que son continuamente actualizadas durante el proceso de búsqueda como son su anchura (ver Figura 1), la profundidad máxima, la profundidad media, su varianza o el factor de profundización (profundidad máxima entre la anchura de la búsqueda).

Normalmente el planificador no explora sólo una posibilidad, sino que mantiene varias posibilidades abiertas y va explorando sucesivamente la que considere mejor de ellas gracias a una función de evaluación que le permite calcular una estimación de la bondad de cada posibilidad. Esta función de evaluación es la siguiente:

$$f(p) = ((1-W) g(p) + W h(p)) s(p)^Z$$

donde p es un posible programa de control parcialmente construido, $g(p)$ es una función que mide el coste acumulado hasta llegar al punto en que se encuentra p , $h(p)$ es una estimación del coste necesario para terminar p , $s(p)$ es una medida de simplicidad de p , $W \in [0,1]$ es un peso que pondera la importancia relativa de h con respecto a g , y $Z \in [0,1]$ un peso que pondera la importancia relativa de $s(p)$. Pues bien, uno de estos estancamientos consiste en que esta función de evaluación no es capaz de discriminar entre un conjunto de posibles programas de control de forma que va completando un poco cada uno, sin decidirse a completar del todo ninguno de ellos. Esto produce un fuerte incremento de la anchura de la búsqueda con respecto a su profundidad máxima, que apenas se incrementa. Es decir, el planificador se encuentra perdido entre este conjunto de posibilidades y no encuentra una solución.

Esta situación se podría detectar comprobando si el factor de profundización es inferior a un umbral determinado, en cuyo caso se dispararía una regla de control que informaría al planificador de esta circunstancia. Una regla como ésta podría ser como la mostrada en la Figura 3 (en una notación basada en el lenguaje de programación LISP).

```
(WHEN (< FACT-PROF UMBRAL) (STOP :WHY '(PLANNER STALLED)))
```

Figura 3

Control del estado de la planta

Esta categoría consiste en un conjunto de reglas que detectan situaciones anómalas que se podrían producir en una planta industrial debidas a la ejecución del programa de control que se está intentando construir. Por ejemplo la mezcla de dos sustancias químicas que normalmente no se mezclan, un calentamiento excesivo de alguna sustancia, etc (Figura 4).

```
(WHEN (AND (CONTAINS ?X ACID) (CONTAINS ?X WATER)) (STOP :WHY  
'(SUSPICIOUS MIXING)))  
(WHEN (AND (> (TEMPERATURE ?X) 100) (= ?X WATER)) (STOP :WHY  
'(SUSPICIOUS HEATING)))
```

Figura 4

Para poder detectar estas situaciones las reglas de control del estado de la planta se basan en los módulos de razonamiento sobre planes y razonamiento sobre el problema que permiten simular el funcionamiento de un programa de control y deducir todas las consecuencias que tendría su ejecución en una planta industrial.

Control de fuentes de incertidumbre

Esta categoría consiste en un conjunto de reglas cuyo fin es detectar situaciones de incertidumbre en un programa de control y atajarlas desde el principio para evitar una explosión combinatoria motivada por esta incertidumbre. Normalmente son situaciones en las que el planificador tiene que utilizar la información proporcionada por sensores.

Un sensor es un aparato que se encuentra en algún lugar de una planta industrial y que proporciona información sobre el estado de ciertas variables como temperatura, nivel (de una sustancia en un tanque de almacenamiento), conductividad, etc. Cuando el planificador construye un programa de control debe tener en cuenta la información de estos sensores, pero ésta no estará disponible hasta que se ejecute el programa. Por esta razón, el planificador debe prever todos los valores relevantes de un sensor e incluirlos en el programa de control. Dado que el rango de valores relevante de entre los que proporciona un sensor depende del problema que se esté resolviendo y esto es algo que debe determinar un ingeniero de control, existen reglas especializadas en detectar esta situación (Figura 5).

```
(WHEN (AND (TEMPERATURE WATER ?X) (GIVEN-BY SENSOR ?X))  
(STOP :WHY '(SENSOR RANGE REQUIRED)))
```

Figura 5

Figura 7

Para cumplir esta tarea de comunicación la interfaz de diálogo permite visualizar gráficamente la estructura de un programa de control (Figura 7) así como realizar consultas sobre su estructura, las razones por las que una determinada instrucción se ha incluido en el programa o las dependencias existentes entre ellas, lo que podría denominarse la estructura causal del programa [0,0]. Algunas de estas órdenes de consulta se muestran en la Tabla 1.

| <i>Orden</i> | <i>Función</i> |
|------------------------|--|
| (WHY action) | Muestra la justificación de la inclusión de una instrucción. |
| (SUPPORTED-BY action) | Instrucciones de las que depende una instrucción. |
| (IS-SUPPORT-OF action) | Instrucciones que dependen de una instrucción. |
| (WHY-BINDING variable) | Justificación de la instanciación de una variable. |
| (DOMAIN variable) | Consulta el universo de discurso de una variable |
| (CONSTRAINTS variable) | Consulta las restricciones que hay definidas sobre una variable. |

Tabla 1

Además de estas órdenes de consulta destinadas a hacer comprender a un ingeniero las razones del diseño de un programa de control, la interfaz de diálogo proporciona órdenes específicas para cada una de las categorías del módulo de control de la iniciativa.

Control de la estrategia de la búsqueda

Cuando el planificador se encuentra perdido, el ingeniero puede alterar la estrategia de búsqueda modificando los pesos W o Z de la función de evaluación en la que se basa el proceso de búsqueda (haciendo la búsqueda más ancha o más profunda) o bien escoger directamente alguna de las posibilidades entre las que se encuentra perdido (Tabla 2).

| <i>Orden</i> | <i>Función</i> |
|----------------------------------|--|
| (SETQ W value) (SETQ Z value) | Modifica los pesos W o Z |
| (SHOW CHOICES) | Muestra todas las posibilidades entre las que se encuentra perdido |
| (FOLLOW choice) | Selecciona una posibilidad directamente |
| (CONTINUE) | No intervención en el proceso de búsqueda |

Tabla 2

Control del estado de la planta

La interfaz de diálogo tiene acceso directo a los módulos de razonamiento sobre el problema y sobre el plan, de forma que permite mostrar al ingeniero la situación conflictiva que se haya presentado y consultar (Tabla 1) las razones por las que se hubiese producido.

Control de fuentes de incertidumbre

Una vez que el planificador detecta que se va a usar un sensor pide al ingeniero que especifique cuál es el rango de valores relevante que debe tener en cuenta (Tabla 3).

| <i>Orden</i> | <i>Función</i> |
|---|---|
| (PARTITION-DOMAIN variable interv ₁ , interv ₂ , ...) | Define el rango de valores relevantes para una variable |

Tabla 3

Además de la información proveniente de sensores, el ingeniero tiene acceso a todas las variables presentes en un programa de control así como a sus restricciones asociadas (Tabla 1). Algunas de estas variables están totalmente instanciadas y otras no, representando conocimiento incompleto. Mediante el lenguaje estructurado de la interfaz de diálogo el ingeniero también puede forzar la instanciación de estas variables o revisar instanciaciones ya realizadas por el planificador (Tabla 4).

| <i>Orden</i> | <i>Función</i> |
|-----------------------|------------------------------------|
| (BIND variable value) | Instanciación directa de variables |

Tabla 4

Control de decisiones críticas

La interfaz de diálogo también permite al ingeniero especificar directamente la forma de resolver un problema crítico (Tabla 5).

| <i>Orden</i> | <i>Función</i> |
|--|---|
| (SOLVE requirement action) | Especifica cómo resolver un subproblema |
| (ORDER action ₁ action ₂) | Ordena dos instrucciones entre sí |

Tabla 5

Intervención libre del ingeniero

Además de los puntos de intervención de un ingeniero en cualquiera de las categorías del módulo de control de la iniciativa, el ingeniero puede observar el funcionamiento autónomo de MACHINE y conocer cómo se está desarrollando la búsqueda de forma que, aunque no se haya disparado ninguna regla de control de la iniciativa, puede decidir intervenir por su propia cuenta interrumpiendo el proceso autónomo e iniciando un proceso de diálogo.

En definitiva, aunque esta versión interactiva de MACHINE se encarga de forma autónoma de diseñar los aspectos del diseño de un programa de control industrial que son más rutinarios o que involucran el manejo de gran cantidad de detalles como el chequeo de interacciones entre las acciones o la búsqueda de posibles alternativas, las decisiones más conflictivas e incluso los estancamientos en el proceso de búsqueda son supervisados por un ingeniero humano, el cual tiene la posibilidad de participar en este proceso de diseño para resolver, precisamente, estas decisiones conflictivas.

4 Conclusiones

Esta versión interactiva de MACHINE está en fase de implementación y aún no pueden darse resultados de su funcionamiento aunque se espera que mejore los resultados previamente obtenidos en cuanto a calidad y tiempo de procesamiento. No obstante es una arquitectura que da respuesta a las cuestiones más importantes en un planificador con iniciativa mixta como son el control de la iniciativa, el establecimiento de un contexto de comunicación sin ambigüedades y la gestión de información incierta.

Referencias

- [1] J. F. Allen, J. Hendler, and A. Tate. *Readings in Planning*. Morgan-Kaufmann, 1990.
- [2] St. Amant. Navigation and Planning in a mixed initiative user interface. En *National conference on artificial intelligence, AAAI97*, p. 64-69, 1997.
- [3] R. Aylett, J. Soutter, G. Petley, and P. Chung. AI planning in a chemical plant domain. In *European conference on artificial intelligence*, 1998.
- [4] L. Castillo, J. Fdez-Olivares, and A. González. Automatic generation of control sequences for manufacturing systems based on nonlinear planning techniques. *Artificial Intelligence in Engineering*, 4(1):15--30, 2000.
- [5] L. Castillo, J. Fdez-Olivares, and A. González. Intelligent planning of Graft charts. To appear in *Robotics and Computer Integrated Manufacturing*, 2000.
- [6] L. Castillo, J. Fdez-Olivares, and A. González. A three-level knowledge based system for the generation of live and safe petri nets for manufacturing systems. To appear in *Journal of Intelligent Manufacturing*, 2000.

- [7] G. Ferguson, J. Allen and B. Miller. TRAINS-95: Towards a mixed initiative planning assistant. En *Artificial Intelligence Planning Systems*, 1996.
- [8] M. R. Genesereth and N. J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann, 1987.
- [9] W. A. Gruver and J. C. Boudreaux. *Intelligent manufacturing: programming environments for CIM*. Springer-Verlag, London, 1993.
- [10] I. Klein, P. Jonsson, and C. Backstrom. Efficient planning for a miniature assembly line. *Artificial Intelligence in Engineering*, 13(1):69--81, 1998.
- [11] S. Mitchell. A hybrid architecture for real-time mixed-initiative planning and control. En *Innovative applications of artificial intelligence*, p. 1032-1037, 1997.
- [12] S. Polyak, A. Tate. Rationale in planning: causality, dependencies and decisions. *Knowledge Engineering* (1998), vol. 13, n. 3, p. 247-262.
- [13] M. Veloso, A. Mulvehill and M. Cox. Rationale-supported mixed-initiative case-based planning. En *Innovative applications of artificial intelligence*, p. 1072-1077, 1997.
- [14] D. Wilkins and M. desJardins, A call for knowledge-based planning. Technical report, SRI International, 1999.

Requisitos para una interfaz visual de un SGBDOO construido sobre un sistema integral OO

Adolfo Hernández , Ana Belén Martínez, Juan Manuel Cueva y Darío Álvarez

Departamento de Informática, Área de Lenguajes y Sistemas
Universidad de Oviedo, Calvo Sotelo s/n 33007, Oviedo
Tlf: 985.10.33.94 Fax: 985.10.33.54 e-mail: belen@lsi.uniovi.es

Resumen

En esta comunicación se describen las características deseables así como las funcionalidades básicas exigibles a la interfaz de un sistema de gestión de bases de datos orientadas a objetos construido sobre un sistema integral totalmente orientado a objetos que está siendo desarrollado en el Laboratorio de Tecnologías Orientadas a Objetos de la Universidad de Oviedo.

Palabras clave : interfaz OO, sistema integral, SGBDOO

1 Introducción

El objetivo de esta exposición es mostrar el trabajo que se está realizando en la búsqueda de una interfaz para un sistema de gestión de bases de datos OO construido sobre un sistema integral que está siendo desarrollado por un grupo de profesores y alumnos de la Universidad de Oviedo.

Sistema Integral Oviedo3

Oviedo3 es un proyecto de investigación que intenta construir un sistema integral orientado a objetos en el que todos los componentes: sistema operativo, bases de datos, compiladores, interfaces de usuario, etc. comparten el mismo paradigma de orientación a objetos [1], eliminando así la desadaptación de impedancias y el problema de interoperabilidad entre modelos de objetos, que se producen al no adoptar el paradigma de OO todos los componentes del sistema de una forma integral. En la base de este sistema se encuentra una máquina abstracta que proporciona el modelo de objetos que será empleado por el resto de componentes del sistema [2].

Sistema de Gestión de Bases de Datos para Oviedo3

Sobre este sistema integral, compartiendo el mismo modelo de objetos, y aprovechando las características que le ofrece el sistema operativo: persistencia, distribución, seguridad y

conurrencia se está construyendo un sistema de gestión de bases de datos, BDOviedo3 [3]. Su desarrollo se ha estructurado en tres módulos estrechamente vinculados:

- *Motor*. Este módulo tiene dos funciones principales: procesamiento de consultas y gestión del almacenamiento.
- *Lenguajes*. En este módulo se especifican los lenguajes que permiten la definición, manipulación y consulta de la base de datos. Se ha optado por emplear el estándar propuesto por ODMG [4].
- *Herramientas Visuales*. En este módulo se pretende construir una interfaz de usuario que facilite la interacción con la base de datos.

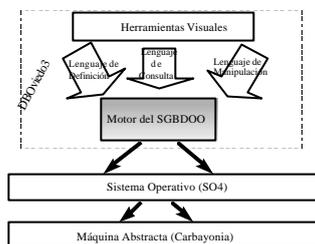


Figura 1. Arquitectura de BDOviedo3

Actualmente existe ya un primer prototipo en el que se incluyen los dos primeros módulos: motor y lenguajes [5]. Es por tanto necesaria la inclusión en dicho prototipo de una interfaz de usuario OO que facilite la usabilidad del sistema. En los apartados siguientes se enumeran las características deseables para la interfaz del SGBDOO anteriormente descrito, así como las funcionalidades básicas exigibles para el mismo.

2 Características deseables para la interfaz de un SGBDOO

Actualmente existen una gran cantidad de bases de datos OO, pero no existe uniformidad en cuanto a las características que presentan sus interfaces. A continuación se enumeran una serie de características que se consideran esenciales para la interfaz de usuario de cualquier SGBDOO [6] y en especial para la del gestor de Oviedo3:

- *Transparencia y realimentación*. Una interfaz es transparente cuando el usuario sabe en todo momento lo que ocurre en el sistema (realimentación).
- *Concisión y calidad en la representación*. La concisión se conseguirá en la presentación de opciones y de información al usuario, y en la cantidad de datos de entrada necesarios para que el usuario pueda expresar sus necesidades.

- *Adaptabilidad y tutorialidad.* La interfaz debe poder adaptarse a las preferencias del usuario, y éste tiene que poder preguntar al sistema sobre el contexto en que se encuentra.
- *Coherencia e integridad.* Coherencia definida de forma que idénticas acciones producen idénticos resultados en cualquier contexto, e integridad referida a los mecanismos de confirmación y a las rutinas de manejo de errores.
- *Complejidad funcional.* Una interfaz gráfica debe proporcionar el acceso a todas las funciones disponibles en un SGBD.
- *Generación automática de vistas.* El modelo de datos permite la representación de los objetos independientemente de la aplicación.
- *Soporte de diferentes niveles de abstracción.* La interfaz tiene que permitir que el usuario elija el nivel de detalle en la visualización de la información.
- *Acceso a los datos a través del SGBD.* Para dicho acceso es necesario un módulo de la interfaz que se responsabilice del envío y recepción de la información del SGBD. Ajustes mínimos en este módulo permitirán la portabilidad a otros SGBDs.
- *Independencia en las acciones.* Una interfaz de usuario debe garantizar que cualquier acción realizada produzca un resultado completo y predecible.
- *Integración y representación de Bases de Datos.* Las interfaces de usuario tienen que permitir la visualización y manipulación simultánea de los esquemas y los datos de diferentes Bases de Datos.

3 Funcionalidad Básica de la Interfaz

La completitud funcional mencionada anteriormente implica que la interfaz permita el acceso a toda la funcionalidad del SGBD. No obstante, en este resumen nos centraremos en tres funcionalidades que se consideran las básicas e imprescindibles para la interfaz de este SGBD.

3.1 Generador/Visualizador del esquema de la base de datos

El esquema de una base de datos de objetos es la estructura de atributos, definición de métodos, jerarquía y relaciones entre clases. El estándar ODMG [4] establece un lenguaje de definición de datos (ODL) que es independiente del lenguaje que se utilice para la escritura de los métodos (Java, C++, Smalltalk).

La definición de las clases de las que consta el esquema de una BD clásicamente se realiza utilizando un programa de edición de texto, normalmente sin formato, en el que el diseñador-programador escribe la especificación de una clase en un cierto lenguaje de programación, admitido por el gestor de bases de datos que se esté utilizando, o en ODL. Sin embargo, este entorno se considera muy poco amigable. Una solución a este problema pasa por la exigencia al sistema de las siguientes propiedades.

Editor dirigido por sintaxis

El entorno ha de permitir escribir - editar el código de la definición de las clases que componen la BD, y chequear la corrección de este código. Para ello se considera muy apropiado un editor de texto dirigido por sintaxis, en el que además de la utilización de los colores para distinguir unos elementos sintácticos de otros, hay otra función valorable consistente en mostrar un pequeño cuadro que acompaña al cursor según se escribe el nombre de un objeto o una clase. En este cuadro se muestran los métodos disponibles ordenados alfabéticamente, permitiendo la selección de uno de ellos, y evitando así errores de tipografía. Además de los nombres de los identificadores de los métodos, se pueden mostrar automáticamente los tipos de los parámetros de la invocación.

Definición visual de las clases

Es necesaria la existencia de un módulo de definición de las clases totalmente visual alternativo al editor de texto, y con el que se pudiese conmutar, de forma que una clase descrita visualmente tuviese su descripción textual en un correcto ODL, y viceversa. Algunas herramientas permiten construir gráficamente las clases, especificando sus atributos, las relaciones entre ellas, ya sean jerárquicas o de composición y la enunciación de los métodos con sus parámetros. En el caso de [7] proporciona además dos vistas del esquema de la BD (árbol jerárquico de clases y diagrama de clases [Figura 2]). Sin embargo, la aplicación es en realidad una herramienta de modelado de clases, pero no ofrece ninguna posibilidad para traducir el modelo a las clases que utilizará el SGBDOO, ni siquiera para escribir los métodos completos en el lenguaje elegido, sino que esa es una labor que debe hacerse desde el tradicional editor de texto.

3.2 Navegación-Inspección

Es deseable que el entorno para nuestro SGBDOO permita la inspección de la información contenida en la base de datos sin necesidad de acudir a la implementación de un programa o a la formulación de una consulta en el lenguaje OQL. Esta inspección o navegación (*browsing*) es una tarea realizada frecuentemente y como tal debe ser facilitada.

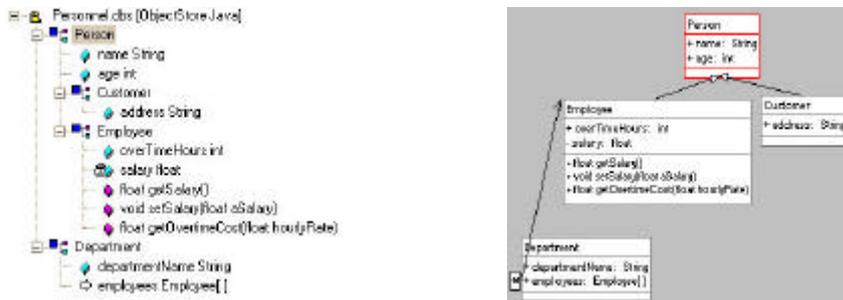


Figura2. Espacio de trabajo del ObjectStore Database Designer.

Inspección Sincronizada

Aparte de una visión general de los objetos almacenados en la base de datos, que la incorporan ya la mayoría de los sistemas existentes (ej. Poet [8] en forma tabular), a nuestro sistema se le exige la capacidad de realizar una inspección sincronizada como la propuesta en [9]. Los atributos de un objeto que no son tipos primitivos (referencias a otros objetos) pueden ser visualizados en una nueva ventana (se crea un enlace de sincronización). Este proceso puede realizarse recursivamente, con lo que se genera un árbol de sincronización. Al recorrer los objetos que son raíz de un cierto árbol de sincronización, en las ventanas que se encontraban abiertas se muestran de forma automática los sub-objetos correspondientes.

3.3 Entorno de consultas

El lenguaje de consulta empleado para el SGBDOO de Oviedo3 es el OQL (Object Query Language) propuesto por el estándar ODMG 2.0. Este lenguaje tiene un estilo similar al SQL-92 en cuanto a formulación de las consultas *select-from-where*, con los conceptos propios del manejo de objetos, como objetos compuestos, identidad de objetos, expresiones de camino, polimorfismo, invocación de métodos, y ligadura tardía o dinámica.

La mayoría de los SGBDOO no se preocupan en exceso de la creación de una interfaz adecuada para el lenguaje de consulta, que facilite la realización de las mismas y minimice además el número de errores. Así, algunos entornos como GOODIES [10] y OdeView[9] se limitan a seleccionar los objetos a visualizar mediante la formulación de predicados de selección según los valores de alguno de los atributos de los objetos. Otros como POET 6.0 permiten la escritura de consultas en lenguaje OQL guardándolas y recuperándolas como un fichero de texto.

El entorno de consultas de Oviedo3 permitirá la formulación de las consultas de una forma visual (además de textual). El modelo seguido para ello está basado en los trabajos mostrados en [11], y propone la representación visual de un árbol en el que cada nodo

representa una clase del esquema de la base de datos. Para formular la consulta se establecen valores para algunos de los atributos simples de la clase, y los atributos complejos pueden expandirse creando un nuevo nodo del árbol. El resultado de la consulta se genera desde las hojas hacia la raíz, de forma que el valor de un nodo se eleva al nodo predecesor para calcular su valor y así llegar hasta el nodo origen. El resultado de la consulta es el valor de la raíz del árbol.

4 Conclusiones

Actualmente existen una gran cantidad de SGBDOO pero no existe uniformidad en cuanto a las características suministradas por sus interfaces visuales. En esta comunicación se describen las características deseables para la interfaz de un SGBDOO construido sobre un sistema integral OO, así como la funcionalidad básica que debe proporcionar.

Referencias

- [1] J.M. Cueva et al. "Oviedo3: Acercando las Tecnologías Orientadas a Objetos al Hardware". Actas de I las Jornadas de Ingeniería del Software, Sevilla 1996.
- [2] D. Álvarez et al. "An Object-Oriented Abstract Machine as the Substrate for an Object-Oriented Operating System". 11th ECOOP, Jyväskylä 1997.
- [3] A. B. Martínez et al. "BDOViedo3: Un SGBDOO sobre una Máquina Abstracta Persistente". Actas de las III Jornadas de Investigación y Docencia en Bases de Datos, Valencia 1998.
- [4] R. Cattell et al. "The Object Database Standard: ODMG 2.0". Morgan Kaufmann, 1997.
- [5] F. Ortín et al. "An Implicit Persistence System on an OO Database Engine using Reflection". 14th Brazilian Symposium on Databases, Brazil 1999.
- [6] J. Lopes de Oliveira. "On the Development of User Interface Systems for Object-Oriented Databases". ACM pag. 237-239, 1994.
- [7] ObjectStore Database Designer. www.object-store.net/objectstore/po.html, 2000.
- [8] Poet Object Server Suite 6.0 User Guide. <http://www.poet.com/>, 2000.
- [9] R. Agrawal et al. "OdeView: The Graphical Interface to Ode". ACM pag 34-43, 1990.
- [10] J. Lopes de Oliveira et al. "Browsing and Querying in Object-Oriented Databases". ACM pag. 364-373, 1993.
- [11] L. Fegaras. "VOODOO: A Visual Object-Oriented Database Language for ODMG OQL". 13th ECOOP, Lisboa 1999.

Interfaz de Usuario en el Desarrollo de un Simulador de Conducción

M. Sánchez ⁽¹⁾, P. Valero ⁽²⁾, I. Pareja ⁽³⁾

⁽¹⁾ Instituto de Tráfico y Seguridad Vial (INTRAS).
Universitat de València, Hugo de Moncada, 4 bajo, Valencia
Tlf: 963383980. Fax: 963383981. e-mail: Mar.Sanchez@uv.es

⁽²⁾ Instituto de Tráfico y Seguridad Vial (INTRAS).
Universitat de València, Hugo de Moncada, 4 bajo, Valencia
Tlf: 963383980. Fax: 963383981. e-mail: valerop@uv.es

⁽³⁾ Instituto de Tráfico y Seguridad Vial (INTRAS).
Universitat de València, Hugo de Moncada, 4 bajo, Valencia
Tlf: 963383980. Fax: 963383981. e-mail: Ignacio.Pareja@uv.es

Resumen

El proyecto EVICA desarrollado por el Instituto de Tráfico y Seguridad Vial de la Universitat de Valencia (INTRAS) y el Instituto MAPFRE de Seguridad Vial, representa la aplicación de un simulador de conducción para la evaluación de conductores. Este proyecto presenta unas características especiales que centran una especial atención en el usuario y la interacción de éste con el sistema de conducción simulada.

Palabras clave: simulador de conducción, construcción de escenarios, evaluación de conductores.

1 EVICA: Sistema Informático de Evaluación Interactiva de Conductores por medio de Simulación

En el proyecto EVICA se plantea la utilización de la tecnología del simulador para evaluar y asesorar conductores. Esta evaluación se centra fundamentalmente en dos niveles, el de control o manejo del vehículo y el táctico o de toma de decisiones, (Parkes, 1991). De la ejecución del conductor en el simulador se obtienen una serie de medidas a partir de las

cuales se puede ofrecer información personalizada sobre su modo de conducir que ayude a mejorar su seguridad y la de los demás usuarios de la vía.

El simulador EVICA se compone de los siguientes elementos:

- Un Renault Twingo, sin motor, pero que conserva todos sus mandos completos: volante, cambio de marcha, indicadores de dirección, etc. Estos mandos se encuentran sensorizados, de tal modo que al ser accionados envían señales a un ordenador .
- Un ordenador PC que envía las señales del vehículo a un ordenador central Silicon Graphics modelo ONYX 2 Infinity Reality.
- Un ordenador Silicon Graphics que recibe las señales del ordenador PC y construye una simulación gráfica de vehículos desplazándose en un escenario virtual de carreteras. Este ordenador construye las imágenes en tiempo real de tal modo que las acciones del conductor se ven reflejadas en representaciones gráficas correspondientes. Este ordenador envía salidas gráficas a tres proyectores situados en el techo de la sala de proyección.

Las imágenes se proyectan en una pantalla dividida en tres partes que ofrece un campo de visión de 120 grados y que está situada frente al vehículo simulado.

2 Factores críticos en el diseño de un sistema de simulación para evaluación y asesoramiento de conductores

La utilización de EVICA como instrumento de evaluación de conductores ha requerido la construcción de una serie de situaciones por las que los conductores pasan, tomar una serie de medidas en cada una de esas situaciones y por último elaborar un informe automatizado en el que aparecen los resultados y recomendaciones a los usuarios sobre la base de su actuación.

Desde el punto de vista del usuario hay una serie de factores importantes, (Schiff, Arnone, Cross, 1994) que pueden afectar la credibilidad y fiabilidad del sistema y por tanto, van a determinar si el objetivo de utilizar el simulador de conducción como una herramienta de evaluación es viable o no. En la presente comunicación nos vamos a centrar en uno de estos factores, que bajo nuestro punto de vista es uno de los más críticos, el realismo de las situaciones de conducción.

2.1 Escenarios de Conducción

Desde el principio nos planteamos que un elemento clave del sistema EVICA era lograr que el sujeto evaluado aceptara como reales los sucesos que le ocurrían a lo largo de la conducción por los escenarios. Este elemento es de suma importancia porque el objetivo del sistema consiste en proporcionar una serie de indicaciones a los usuarios acerca de posibles malos hábitos en su estilo de conducción. El sistema también tendría un papel corrector, ya que el sujeto recibiría recomendaciones o sugerencias acerca de cómo corregir sus deficiencias.

Este tipo de información, no obstante, resulta a menudo difícil de aceptar por los individuos, y es posible que les llevara a rechazar la información que les sea proporcionada. Para hacerlo, un mecanismo psicológico muy sencillo que éstos pueden aplicar consiste en desvalorizar las situaciones presentadas en el simulador por considerarlas como artificiales o basadas en una mera “trampa” que en la vida real nunca se producirían. De este modo, los conductores afectados podrían rechazar las recomendaciones por considerar que están fundadas en situaciones inverosímiles.

Para evitar este efecto se decidió utilizar accidentes reales como base para la construcción de escenarios. Algunos de los inconvenientes que encontramos son los siguientes:

- a) Muchos de los accidentes reales tienen aspecto de haberse producido por causas fortuitas o excepcionales. Si los conductores tuvieran problemas en una situación, los atribuirían a esa excepcionalidad y por tanto no aceptarían los diagnósticos que pudiéramos hacerles. Un elemento importante es que los vehículos que produjeron el accidente realizaron comportamientos que infringían el reglamento de conducción (no señalizar, pasar la línea continua). Nuestros conductores no aceptarían este tipo de escenarios como forma de evaluar su comportamiento en la carretera.
- b) Debido a su peligrosidad, los sujetos podrían verdaderamente tener accidentes. Este suceso sería verdaderamente inapropiado pues podría generar una sensación de inseguridad a los conductores acerca de sus capacidades.
- c) Los sujetos podrían percibir este tipo de incidentes como unos obstáculos que pueden aparecer en cada momento y por tanto cambiar su forma habitual de conducir para así evitar cualquier posible riesgo. También, es necesario evitar la sensación de que el sistema es un videojuego, basado en una serie de obstáculos a sortear.

Puesto que muchos de los accidentes que analizamos producían escenarios poco adecuados para nuestros propósitos, optamos por realizar una serie de modificaciones sobre ellos. A estos cambios los denominamos “moderar la realidad” y, en general, consistieron en aprovechar aquellos elementos que siguieran unas normas de aceptabilidad. Estas son:

- a) Las situaciones no deben parecer producto de causas excepcionales. Por ejemplo, si un accidente ocurre como consecuencia de haber un obstáculo en la vía, la elección del mismo o su posición en la calzada es crucial para que la situación no aparezca como inverosímil.
- b) Los vehículos situados en el escenario no pueden transgredir el reglamento de conducción (en exceso). Aunque en muchos escenarios de accidentes que analizamos nos encontramos con vehículos que transgredían mucho la norma y por ello se produjo el accidente, pronto comprendimos que este tipo de incidentes no serían aceptados por los sujetos evaluados. Una transgresión leve probablemente es más fácilmente aceptable.
- c) No debería haber muchos incidentes. La conducción no debería convertirse en una carrera de obstáculos.

McGehee (1996) realiza las siguientes recomendaciones sobre la forma de construir escenarios de simulación de conducción desde el punto de vista de los factores humanos, y que nosotros incorporamos a las que se han mencionado anteriormente.

- a) Deben diseñarse escenarios que puedan servir para evaluar a la mayor parte de la población. No es conveniente que los escenarios sean sólo apropiados para condiciones o poblaciones muy específicas.
- b) Deberían utilizarse situaciones de conducción que sean importantes e interesantes. Esas situaciones además deberían evitar que los sujetos modificaran su estilo normal de conducción para, por ejemplo, intentar dar una “buena impresión”.
- c) Hay que evitar la sensación de que puede haber un peligro inesperado acechando a cada momento al sujeto evaluado. Esto llevaría a que el conductor modificara su estilo de conducción habitual, haciéndole que condujera, por ejemplo, mucho más lentamente de lo normal para evitar accidentes.
- d) Deberían corresponder a situaciones de conducción normal, realistas. Es conveniente evitar que los sujetos tengan accidentes en el simulador, sobre todo al final del experimento, puesto que les puede generar una sensación de inseguridad.
- e) Es necesario realizar una sincronización perfecta. Este es uno de los aspectos más difíciles de la creación de escenarios de conducción porque si los acontecimientos programados no se ponen en marcha en el momento adecuado, la comparación entre sujetos puede resultar difícil. De este modo, es necesario que todos los sujetos experimenten cada escenario de la misma manera o los resultados obtenidos no serán comparables entre ellos. No se debe dejar nada al azar en el sentido que un sujeto decida por ejemplo cambiar de carril y “estropee” la sincronización del escenario siguiente.

f) Hay que realizar experiencias pilotos con sujetos que no conozcan el recorrido para así examinar si llevan a cabo comportamientos inesperados o impredecibles que trastocuen el espíritu de las situaciones desarrolladas.

2.2 Especificación de un Escenario en el Simulador EVICA

En un simulador de conducción típico, los sujetos pueden conducir por su recorrido con casi total libertad. En ese circuito hay otros vehículos que responden de manera preprogramada a nuestras acciones. El comportamiento de estos vehículos, así como el de otros elementos, puede ser más o menos determinístico dependiendo de los objetivos para los que el simulador haya sido construido. La utilización de un simulador de conducción como herramienta de evaluación requiere un comportamiento determinístico de los vehículos, de tal manera que dos sujetos diferentes conduciendo por el escenario deberían encontrarse ante situaciones de tráfico semejantes a pesar de disponer de libertad de acción. Por ejemplo, si un vehículo debe encontrarse con nuestro conductor en un determinado punto, pero nuestro conductor ha conducido muy lentamente es posible que ambos no se encuentren. Por ello, será necesario tomar medidas que eviten este extremo tal y como modificar la velocidad del vehículo automático o introducir semáforos u otros. De este modo, dado nuestro interés en utilizar el simulador de conducción como una herramienta de diagnóstico necesitamos sincronizar el resto de elementos implicados en la simulación para así obtener situaciones semejantes o equivalentes sobre las que los sujetos puedan ser evaluados. Como un ejemplo de las estrategias utilizadas podemos ver la imagen correspondiente a un escenario denominado de “pisacolas” en la figura 1. En este escenario, el conductor observa por el espejo retrovisor un vehículo que se acerca por detrás demasiado, con la consiguiente impresión de riesgo de accidente. A partir de ese momento, el vehículo programado actúa de manera automática sin permitir que haya cambios en la distancia de seguridad entre ambos vehículos aunque el conductor intente acelerar para lograr despegarse de él. Esto ocurre independientemente de la velocidad de partida del conductor.

Por tanto, la sincronización de las situaciones debe permitir que éstas sirvan para todas las poblaciones, sin que los conductores tengan que modificar su forma de conducir habitual. La situación es frecuente durante la conducción, por tanto las personas que conducen en el simulador no tienen sensación de extrañeza o de que es poco habitual. La situación transgrede la norma, puesto que el vehículo programado no guarda la distancia de seguridad obligatoria, pero no la transgrede en exceso. No es una situación de “trampa o de peligro inesperado”, y puede ser fácilmente aceptada como una situación real.



Figura 1: Imagen de un escenario del simulador EVICA escenario

Referencias

- [1] Parkes, A. M. (1991). Data Capture Techniques for RTI Usability Evaluation. In *Advanced Telematics in Road Transport. Vol. II. Proceeding of Drive Conference*, (pp. 1440-1456) Brussels. Amsterdam, Elsevier.
- [2] Schiff, W., Arnone, W., y Cross, S. (1994). Driving assessment with computer-video scenarios: More is sometimes better. *Behavior Research Methods, Instruments & Computers*, 26(2), 192-194.
- [3] McGehee, D. V. (1996). Designing Driving Simulation Scenarios: A Human Factors Perspective. In *Workshop on Scenario and Traffic Generation for Driving Simulations*: Iowa.

Un Sistema de Etiquetado para el Resaltado de Texto en Entornos Web

M. Villarroel, P. de la Fuente

Departamento de Informática
Universidad de Valladolid

Edif. de Tec. Inf. y Tel., Carretera del cementerio s/n, 47011 Valladolid
Tlf: 983.423.710. Fax: 983.423.671. e-mail: {miguelv, pfuente}@infor.uva.es

Resumen

Internet se ha constituido en una importante fuente de información, debido en parte, a la cantidad y la diversidad de la documentación accesible a través de ella. A pesar de la eventual disponibilidad de esta documentación, varios estudios han mostrado que los usuarios aún obtienen copias en papel de los contenidos encontrados, dada la posibilidad de incrementar la información del documento impreso, ya sea mediante anotaciones y/o mediante el marcado de texto para resaltar ciertas partes del texto.

Este trabajo presenta una herramienta para la lectura de documentos, la cual permite el marcado de texto en documentos digitales recuperados del Web, con la perspectiva de admitir la realización de notas relacionadas a una parte específica del documento. Esto permitirá ampliar las posibilidades de interacción de los usuarios con los contenidos de una biblioteca digital y el Web en general. Además se propone el aprovechamiento de esta plataforma para la identificación interactiva de áreas de interés, utilizando los fragmentos marcados como información de entrada para el indexado del documento digital, esto ante la perspectiva de la interacción de varios usuarios con un mismo documento, a través de la herramienta. Finalmente, se presentan las posibilidades que ofrece la aplicación, en la perspectiva de usarla como elemento principal de un entorno cooperativo de recuperación de información.

Palabras clave: Hipertexto, Documento Digital, Recuperación de Información, Cooperación, Interacción.

1 Introducción

A continuación se expone brevemente la motivación de este estudio y los trabajos consultados en referencia al mismo.

1.1 Motivación

Internet se ha constituido en una importante fuente de información, esto debido en parte a la cantidad y la diversidad de documentación que está a disposición de los usuarios de la red. También contribuyen a este hecho, la significativa cantidad de iniciativas en procura de implementar bibliotecas digitales de distinta temática y contenido [1] [2] [3]. Se observa también la presencia de autores que utilizan Internet como un medio de publicar sus ideas, plasmadas en documentos de formato electrónico (por ejemplo, [4]). La mejora en los sistemas de navegación y visualización de datos han hecho más accesible la documentación contenida en los distintos repositorios de información disponibles a través de Internet [5].

Sin embargo, es significativo notar que algunas de las costumbres de los usuarios, en lo referente a actividades de lectura y revisión de documentación escrita, no han cambiado. Distintos estudios muestran que, aunque el usuario tiene a su disposición ejemplares digitales de distintos documentos, aún prefiere imprimir en papel éstos contenidos, para su revisión y lectura. El comportamiento descrito se atribuye a la flexibilidad de uso del papel, como herramienta de almacenamiento de información [6] [7] [8].

Actividades tales como el subrayado o el marcado de texto, con el fin de resaltar un fragmento o un conjunto de palabras, se realizan de una manera inmediata y sencilla sobre una copia en papel. La posibilidad de realizar apuntes y notas sobre un documento impreso, es un factor que sumado al anterior, determina que los usuarios prefieran aún realizar la lectura de texto sobre este tipo de medios, puesto que los sistemas de navegación y visualización más comunes no brindan la posibilidad de realizar tareas de marcado o adición de texto sobre el documento que se examina. Por otra parte la capacidad de referencia múltiple que permite un documento digital a través de enlaces a otros documentos, es una clara ventaja del formato digital sobre el papel como medio de organización y almacenamiento de información.

Este trabajo presenta una herramienta de marcado de texto, la cual permite al usuario el resaltado de fragmentos de un documento, admitiendo la posibilidad de incorporar notas en relación a trozos específicos del texto, incrementando de esta forma la capacidad interactiva del usuario con documentos en formato digital, y manteniendo las ventajas que conlleva la manipulación de información en este formato.

El trabajo presenta además el trabajo futuro a realizar con relación a esta herramienta, en especial el referido al papel de ésta como un entorno cooperativo de recuperación de información.

1.2 Antecedentes

La necesidad de organizar y relacionar la información, ya fue reconocida por Vannevar Bush en 1945, en su definición del "Memex", estableciendo además la necesidad de construir mecanismos para la realización de notas o apuntes adicionales sobre la información almacenada [9]. Actualmente aplicaciones tales como Microsoft™ Word™, permiten el marcado de texto con el fin de diferenciar un fragmento específico del mismo [10], otros formatos de representación también contemplan etiquetas para el resaltado de texto, por ejemplo, el formato RTF [11].

Trabajos anteriores que muestran la importancia de ofrecer al usuario herramientas para el resaltado de texto se aprecian en [6] [7]. Uno de los recientes trabajos en esta línea es el presentado por Schilit, Golovchinsky, y Price en [12] [13], en ellos los autores describen Xlibris. Este es un sistema que permite la realización de notas y trazos en formato libre sobre un texto, sin embargo, este sistema no contempla la interacción en un entorno Web. De acuerdo a las comunicaciones anteriormente citadas, el sistema no posee la estructura de plataforma abierta, lo que haría difícil el trabajo del usuario al usar este sistema en el Web. Una aportación importante en el trabajo de este equipo, es la discusión del sistema como herramienta de recuperación de información, en el cual se presentan resultados que contribuyen a respaldar el desarrollo de este trabajo.

2 Sistema de Etiquetado

A continuación se presenta la descripción del sistema de etiquetado para el resaltado de texto, inicialmente se ofrece una descripción de la herramienta, para luego exponer brevemente, el papel de ésta en una plataforma cooperativa de recuperación de información.

2.1 La herramienta

Actualmente el desarrollo de la herramienta se encuentra en la fase de desarrollo y pruebas del prototipo experimental. En esta versión el programa debe ser activado en forma local por el usuario, ejecutándose como una aplicación independiente del navegador que el usuario tenga instalado. De esta forma se hace explícita la diferencia funcional entre el programa y el navegador, ya que la navegación por el Web no es el objetivo primario de la aplicación, sino más bien la revisión detenida de un documento electrónico, en este caso un documento en formato HTML.

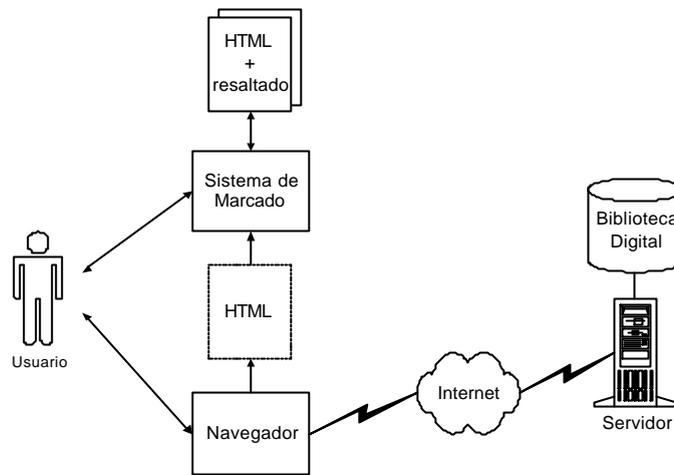


Figura 1. Modelo de operación del sistema

Como puede apreciarse en la figura 1, un documento recuperado a través del Web es almacenado localmente por el usuario, el sistema de marcado manipula el fichero local. Las etiquetas producto del marcado de texto son almacenadas en la forma de referencias al documento original en un fichero asociado, manteniéndose sin cambios el fichero original, preservando de esta forma su contenido y estructura. Debe notarse que esta estrategia incluso hace innecesario el almacenamiento local del texto original, lo cual además permite evitar consideraciones acerca del derecho del usuario a almacenar copias locales del documento accedido a través de Internet. Aunque la versión final del sistema contempla el almacenamiento de toda la información, producto de la interacción, en el servidor remoto (eventualmente una Librería Digital), es importante considerar que el usuario debe tener la posibilidad de almacenar de manera local estos datos.

Como se mencionó anteriormente, esta versión de la herramienta permite la visualización de documentos en formato HTML, los hiper-enlaces del documento HTML son presentados por el programa. En el caso que el usuario seleccione algún hiper-enlace, el documento referido es abierto en una ventana del navegador, de esta manera se preservan también las características de multi-referencia del documento original y la funcionalidad de navegación, características en los ficheros de hipertexto.

En la figura 2, se puede apreciar la ventana de presentación del prototipo, en ella se visualiza un texto en el que se han resaltado dos fragmentos de texto, como puede observarse por los iconos visibles, la herramienta solo ofrece opciones de marcado de texto y manipulación de ficheros (solo lectura en el fichero original y lectura/escritura en el fichero de etiquetas).

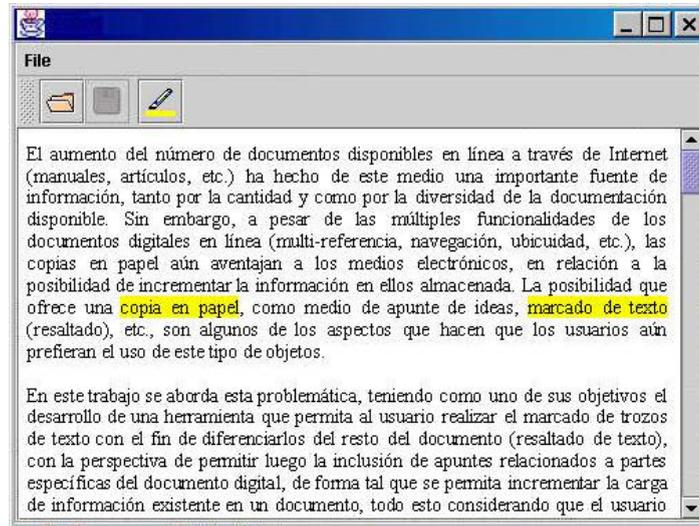


Figura 2. Ventana de presentación del prototipo con texto marcado

Se tiene prevista la adición de búsqueda de cadenas en el fichero visualizado y la adición de notas y apuntes en referencia a partes específicas del texto. Las anotaciones introducidas contemplan la inclusión de nuevos hiper-enlaces; de esta forma, se da al usuario la posibilidad de crear nuevas relaciones entre distintos documentos.

2.2 Un sistema cooperativo de recuperación de información

La potencialidad de aplicación de una herramienta computacional, inicialmente desarrollada para un solo usuario, puede incrementarse (en ciertos casos), si es utilizada sobre una plataforma cooperativa [14] [15]; en el caso de la herramienta descrita en el anterior apartado, este es el principal objetivo de su concepción. Si bien los trabajos citados en la sección 1 sostienen la necesidad de una aplicación de este tipo, no es menos importante investigar su potencialidad cooperativa [16] [17], dado el entorno en el que se tiene prevista su operación: el Web.

Producto de este análisis, y las consideraciones realizadas en los trabajos que anteceden a esta propuesta, ha sido planteado el diseño y desarrollo de una plataforma cooperativa de recuperación de información [18], teniendo como base el sistema de resaltado de texto descrito anteriormente. Los fragmentos de texto resaltados pueden ser usados como entrada, tanto para la elaboración de consultas en un sistema de recuperación de

información [13], como para el indexado del documento original por las frases y palabras marcadas, en este caso, por varios usuarios. Es factible asociar los términos marcados en el documento a niveles de relevancia, que están en función de la frecuencia con que cada uno de ellos ha sido señalado. Estos niveles de relevancia pueden ser determinados tanto a nivel de uno o de varios usuarios. En adición, las notas y apuntes asociados al documento también constituyen otra fuente de información susceptible de ser analizada.

Estas consideraciones permitirán definir funciones de interacción adicionales y complementarias al proceso de lectura y revisión de documentos digitales, las cuales se constituyen en la base del sistema cooperativo de recuperación de información descrito. De ésta forma, las funcionalidades de operación e interacción, que el sistema ofrece en un entorno de usuario individual, son incrementadas mediante su aplicación en un entorno cooperativo.

3 Trabajo Futuro

La experiencia obtenida, en el transcurso de este trabajo, ha dado lugar a la consolidación de algunos de los objetivos propuestos y a la revisión y replanteo de otros. Entre las actividades a realizar se contemplan las siguientes:

- Experimentación del prototipo con grupos de control, comparación y análisis de los resultados. Análisis de facilidad de uso del prototipo y del modelo del sistema.
- Programación de un módulo para la realización automática de consultas, basadas en fragmentos resaltados del documento visualizado.
- Desarrollo de la plataforma con sistema de almacenamiento basado en repositorios remotos. Comparación del nivel de rendimiento entre las versiones local y remota del sistema. Análisis de tiempo de respuesta, y su efecto sobre el nivel de facilidad de uso del sistema.
- Implementación del sistema en el entorno digital de la Biblioteca Histórica Santa Cruz, de la Universidad de Valladolid.

El aspecto del factor cooperativo de la plataforma, su aplicación y efectos, será un componente a tener en cuenta en las actividades anteriormente mencionadas.

4 Conclusiones

La impresión de documentos digitales es todavía una actividad bastante común en los usuarios, debido a la posibilidad de incrementar la información contenida en este tipo de medios mediante anotaciones y marcado de fragmentos de texto. El desarrollo de programas que permitan realizar estas actividades permite incrementar las funcionalidades de la visualización y revisión de documentos digitales. Los fragmentos de texto resaltados, pueden ser usados para la elaboración de consultas en un sistema de recuperación de información, así como para el indexado de estos documentos.

El uso de la aplicación en una plataforma cooperativa permitiría aprovechar el resultado del trabajo realizado por varios usuarios para determinar la relevancia del documento con relación a los términos resaltados en él por los usuarios, los lectores en este caso.

El sistema planteado propone mantener la estructura y contenido de los documentos visualizados, preservando así la información original en ellos contenida.

El sistema puede ser usado para comparar las actividades de lectura y revisión de documentos entre varios usuarios. También se puede usar para comparar la actividad realizada por el mismo usuario en diferentes sesiones.

Referencias

- [1] H. Gladney, F. Mintzer, F. Schiattarella, J. Bescós, and M. Treu, "Digital access to antiquities," *Communications of the ACM*, vol. 41, pp. 49–57, Apr 1998.
- [2] I. Witten, C. Nevill-Manning, R. McNab, and S. Cunningham, "A public library based on full-text retrieval," *Communications of the ACM*, vol. 41, pp. 71–75, Apr 1998.
- [3] P. Denning and B. Rous, "The ACM electronic publishing plan," *Communications of the ACM*, vol. 38, pp. 97–109, Apr 1995.
- [4] J. Nielsen, "useit.com: Jakob Nielsen's site (usable information technology)." <http://www.useit.com/>, Apr 2000. Visitado el 25 de Abril de 2000.
- [5] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, Massachusetts: Addison Wesley Publishing Company, 3 ed., 1998.
- [6] A. Adler, A. Gujar, B. L. Harrison, K. O'hara, and A. Sellen, "A diary study of work-related reading: Design implications for digital reading devices," in *CHI'98 Proceedings of the Conference on Human Factors in Computing Systems*, (Los Angeles, CA), pp. 241–248, Apr 1998.
- [7] K. O'Hara and A. Sellen, "A comparison of reading paper and on-line documents," in *CHI-97 Conference Proceedings on Human factors in Computing Systems*, (Atlanta, GA USA), pp. 335 – 342, Mar 1997.

- [8] V. Coleman, "Hardcopy to hypertext: putting a technical manual online," in CSD'91 Proceedings of the ACM ninth annual international conference on systems documentation, (Chicago, IL USA), pp. 67–72, Oct 1991.
- [9] V. Bush, "As we may think," *Atlantic Monthly*, vol. 176, no. 1, pp. 641–649, 1945. Also in [19].
- [10] D. Gookin, *Word for Windows 95 for Dummies*. IDG Books Worldwide, first ed., Sep 1995.
- [11] Microsoft Technical Support, "Rich Text Format (RTF v1.5) Specification and Sample RTF Reader Program." <http://night.primat.wisc.edu:80/software/RTF/>, Apr 2000. Visitado el 14 de Abril de 2000.
- [12] B. Schilit, G. Golovchinsky, and M. Price, "Beyond paper: Supporting active reading with free form digital ink annotations," in CHI'98 Proceedings of the Conference on Human Factors in Computing Systems, (Los Angeles, CA), pp. 249–256, Apr 1998.
- [13] G. Golovchinsky, M. Price, and B. Schilit, "From reading to retrieval: freeform ink annotations as queries," in SIGIR '99, Proceedings of the 22nd annual ACM SIGIR conference on Research and development in information retrieval, (Berkeley, CA USA), pp. 19–25, Aug 1999.
- [14] A. Chabert, E. Grossman, L. Jackson, S. Pietowicz, and C. Seguin, "Java object-sharing in Habanero," *Communications of the ACM*, vol. 41, pp. 69–76, June 1998.
- [15] C. Ellis, S. Gibbs, and G. Rein, "Groupware. Some issues and experiences," *Communications of the ACM*, vol. 34, pp. 9–28, January 1991.
- [16] J. Carroll, M. Rosson, G. Chin, and J. Koenemann, "Requirements development: stages of opportunity for collaborative needs discovery," in Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, pp. 55–64, 1997.
- [17] L. Terveen and W. Hill, "Evaluating emergent collaboration on the web," in Proceedings of the ACM 1998 conference on Computer Supported Cooperative Work, (Seattle, WA USA), pp. 355 – 362, Nov 1998.
- [18] J. Grudin, "Collaborative information retrieval (CIR)." CHI 2000 Extended Abstracts, Conference on Human Factors in Computing Systems, Apr 2000.
- [19] E. Edmonds, *The Separable User Interface*. London, England: Academic Press, 1992.

Tirsus IV, Navegación Multidimensional en Aplicaciones Hipermedia sobre Acontecimientos Históricos

(1) Ana María García Fernández, (2) B. Martín González Rodríguez
(3) José Manuel Redondo López, (4) Antonio J. Sánchez Padial

Grupo de investigación en HCI, Laboratorio de Tecnologías Orientadas a Objetos,
Universidad de Oviedo, Campus de Llamaquique, Departamento de Informática,
C/ Calvo Sotelo s/n 33007 OVIEDO.

e-mail:

(1)i9417208@petra.euitio.uniovi.es, (2) martin@lsi.uniovi.es
(3)i1637886@petra.euitio.uniovi.es, (4)tonisan8@hotmail.com

Resumen:

Tirsus IV es un proyecto que explora las posibilidades de la navegación multidimensional aplicada a acontecimientos históricos como puede ser una guerra. Mediante la navegación multidimensional y una construcción de la base de conocimiento orientada al usuario, se busca llegar a una visión global y al mismo tiempo amena de un acontecimiento histórico.

Palabras clave: OOHDM, Navegación Multidimensional, Atención, Motivación

1 Introducción

Existen acontecimientos que se caracterizan por estar formados por distintos eventos que suceden en paralelo, acontecimientos históricos como las revoluciones, descubrimientos científicos y las guerras presentan estas características. La navegación a través de acontecimientos históricos como puede ser una guerra se puede plantear de una forma alternativa a la clásica narración secuencial usando una narración multidimensional, que aporta una visión global de los hechos.

Otra de las razones por la que es interesante investigar la narrativa multidimensional es por su naturaleza dinámica en contraposición con la naturaleza estática de la narrativa secuencial. Una narrativa estática llega a agotar la memoria humana a corto plazo, que se limita a como mucho nueve elementos, provocando que la atención y la motivación se diluyan. En cambio con una narración dinámica logramos ofrecer una posible asociación de ideas, partiendo de los elementos más pequeños de la información a los más grandes, para reducir la saturación de la memoria a corto plazo, usando un mecanismo de asociación de ideas del tipo del empleado por Newell en su máquina cognitiva [1].

Pero además de una menor carga en la memoria a corto plazo podemos lograr dar una posible solución al problema que supone describir un acontecimiento histórico como la guerra de una forma amena. El objetivo es dar una visión global de los acontecimientos distinta a la mera enumeración secuencial que satura la memoria a corto plazo.

Para probar la eficacia de estas hipótesis, se encuentra en fase de elaboración un proyecto de aplicación de la navegación multidimensional a un acontecimiento histórico. El proyecto denominado *Tirsus IV* tiene como objetivo construir una aplicación hipermedia sobre la *Guerra Civil Española en Asturias*, usando para ello la metodología OOHDM [2] por ser la más adecuada para ofrecer diferentes modelos de navegación a partir de una misma base de conocimiento [3] en función de los distintos modelos usuario previstos, que veremos a continuación.

2 El Modelo de Usuario

Tirsus IV va dirigida preferentemente a tres modelos de usuarios:

1. Niños en los primeros cursos de ESO ya que se incluye el tema Guerra Civil en el plan de estudios adaptado de la ESO-Asturias. La aplicación se usará como punto de partida para introducir el tema y material de referencia fundamental.
2. Jóvenes con edades comprendidas entre los 15 y los 17 años, que han oído hablar de los acontecimientos descritos por la aplicación en los libros escolares de historia, utilizando esta aplicación como complemento.
3. Adultos que han vivido directamente la guerra o bien conocen los acontecimientos a través de los relatos de quienes la vivieron.

En función de estos modelos de usuario se estructuraremos la información de la Base de Conocimiento [4].

3 Base de Conocimiento

El primer paso ha sido recopilar la información, clasificándola por fecha y lugar para su posterior organización desde el punto de vista temporal y espacial.

Es de destacar las inconsistencias con respecto al tiempo detectadas, pues hay períodos de la guerra donde apenas hay datos, como por ejemplo la primavera de 1936. Una solución puede consistir en ofrecer una visión global de la guerra, así, si en un determinado lugar no sucede nada durante meses va a pasar desapercibido, pues se concentra la atención en lo que pasa en otros puntos.

Se encuentran inconsistencias en los datos sobre los enfrentamientos, pues hay batallas donde sólo hay los datos proporcionados por un bando, información que resulta poco fiable. Pero es interesante no ocultar al usuario esta información ni su procedencia, pues el caos y la falta de fiabilidad de las informaciones es un rasgo que caracteriza a las guerras.

4 La Navegación Multidimensional

Tradicionalmente la base de conocimiento se presenta en forma secuencial como una serie de datos, fechas, nombres y lugares. Veamos por ejemplo el siguiente extracto, adaptado de un libro sobre la historia de la guerra civil en Asturias [5], sobre los acontecimientos ocurridos durante unos días del mes de agosto de 1936:

1. Día 1, en Gijón se atacan los cuarteles nacionales del Coto y Simancas.
2. Día 8, la columna gallega de la costa lucha en las cercanías de Luarca. En Oviedo los milicianos inician el cerco a la ciudad.
3. Día 15, los milicianos asaltan y ocupan el cuartel del Coto.
4. Día 19, una columna gallega por el interior llega a Leitariegos.

Como podemos ver, en la guerra tenemos desplazamientos temporales, desde el inicio hasta el final de la misma, pero también desplazamientos espaciales a los distintos puntos geográficos donde se desarrolla. Palabras del extracto anterior como Gijón, Leitariegos, etc. nos sugieren movimiento espacial, una forma de organizar la información basada en su localización geográfica.

Una organización temporal o espacial plantea el problema de la secuencialidad en la presentación de la información, por lo que se busca un modo de reflejar los acontecimientos

en varias dimensiones, navegando a través del tiempo y navegando a través del espacio. Aquí es donde surge la idea de usar navegación multidimensional.

La misma información anterior se puede reestructurar en sucesivas líneas de desplazamiento temporal y espacial. (ver Figura 1).

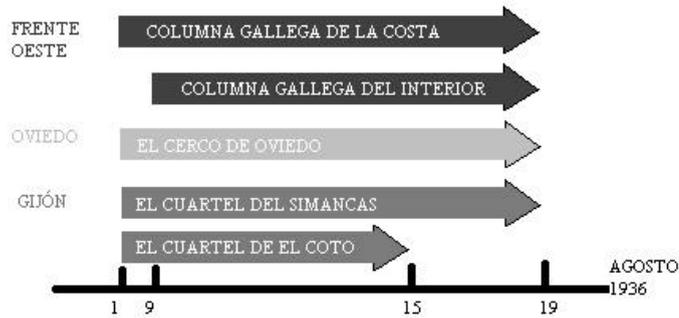


Figura 1. Ejemplo de desarrollo de acontecimientos en el tiempo y en el espacio.

La navegación por los escenarios de batalla se ha planteado en tres planos, estructurando la información en diferentes niveles de granularidad.

1. **Plano temporal.** Para navegar cronológicamente y en función del tipo de usuario, creamos unas metáforas de navegación a diferentes niveles:
 - I. **Nivel General:** Lo forman las distintas etapas de la guerra, como pueden ser el alzamiento o las primeras campañas, para las cuales se emplea una metáfora relacionada con las estaciones del año (usando nieve, flores, hojas o sol para indicar el paso del tiempo).
 - II. **Nivel Medio:** Un calendario, pues en muchos casos llegamos a ver la información de una batalla por días, usando el calendario como localizador.
 - III. **Nivel Detallado:** Un reloj. Hay casos concretos donde es posible conocer lo que ha pasado hora a hora, como en el caso del asalto al cuartel del Simancas en Gijón.
2. **Plano espacial.** Permite una navegación a través de los lugares donde se desarrollaron los acontecimientos. Para ello se crean metáforas de navegación espacial a diferentes niveles:

- I. **El Mapa de Asturias.** A la vez que refleja la situación de la región podemos hacer *zooms* a zonas de la región o a ciudades.
 - II. **Un sector del mapa.** En determinados casos se puede navegar por zonas del mapa como la sierra del Cuera, Gijón, etc.
 - III. **Una ciudad.** A través de un plano de la misma podemos navegar viendo acontecimientos que se desarrollaron en ciudades, como en Oviedo o en Gijón.
 - IV. **Una calle.** Se pueden ver los movimientos que se producen en una calle, por ejemplo en las calles adyacentes al cuartel del Simancas antes de su asalto.
 - V. **Un edificio.** Se puede llegar a saber lo que ocurrió dentro de un edificio, como en el caso del cuartel Simancas.
 - VI. **Una habitación.** Se puede llegar a conocer los datos sobre lo que paso en una zona de un cuartel.
3. Un plano estratégico con datos de cada una de las unidades enfrentadas y de los planes de ataque.

Gracias a la navegación multidimensional estas metáforas de navegación espacial y temporal pueden utilizarse al mismo tiempo y a diferentes niveles. Por ejemplo, se puede pasar de un sector de 100 km. de largo, como es el frente de León, donde no sucede nada durante meses, hasta una habitación y una hora concreta, como en el patio del cuartel del Simancas donde se produce un violento combate.

5 Atención y Motivación

Desde el momento que iniciamos la base de conocimiento estamos buscando elementos para mantener la atención del usuario. En *Tirsus IV* uno de los objetivos prioritarios es intentar en todo momento guardar una relación estrecha entre los hechos históricos del pasado y la realidad del presente en la que se mueve el usuario.

Se puede comparar la situación en período de guerra de los escenarios de las batallas con su situación actual. Un ejemplo es el cuartel del Coto de Gijón, escenario de un violento combate, hoy dedicado a fines sociales y su patio convertido en centro comercial. De hecho la relación con los escenarios de la guerra puede ser directa, pues se puede entrar en el citado centro comercial pensando que nos hallamos en el patio del cuartel donde tuvo lugar

un violento combate. Otro ejemplo de cómo se puede establecer esa comparativa es el campo de fútbol del Real Oviedo donde estuvo situada una trinchera, en el que podemos ver los destrozos que sufrió y los cambios experimentados en su estructura desde entonces (ver figura 2).

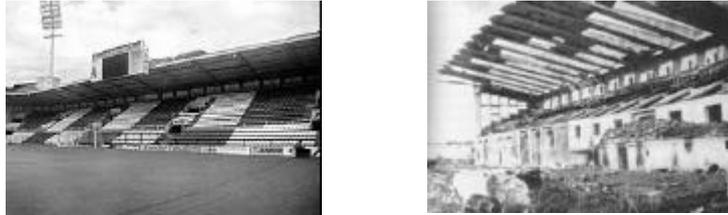


Figura 2. Estadio de fútbol de Oviedo hoy (izquierda) y su estado tras la guerra (derecha).

También se establecen comparativas de el estado de las ciudades actuales con su estado en período de guerra, comparar sus planos, sus calles, su número de habitantes, sus lugares más conocidos.

Estos aspectos hacen que una aplicación relacionada con eventos lejanos en el tiempo pueda vincularse con la realidad actual y con sitios que pueden ser conocidos por el usuario.

6 Conclusiones

En aplicaciones usadas para describir acontecimientos históricos encontramos dos problemas fundamentales: lograr transmitir una idea global de los acontecimientos y encontrar el modo ideal de transmitir estos datos de forma amena.

El uso de una navegación multidimensional comunica los datos históricos de manera más eficiente y permite extraer una idea general del suceso, mucho más cercana a la realidad que la aportada por los datos crudos proporcionados por los informes y análisis históricos publicados en otros formatos.

Referencias:

- [1] Newel, "Unified Theories of Cognition", Harvard University Press, 1990. ISBN 0-674-92101-1
- [2] Daniel Schwabe, Gustavo Rossi, Simone D. J. Barbosa, "Abstraction, Composition and Lay-Out Definition Mechanism in OOHDM",
<http://www.cs.tufts.edu/~isabel/schwabe/MainPage.html>
- [3] Daniel Schwabe, Gustavo Rossi, Simone D.J. Barbosa, "Systemic Hypermedia Applications Design with OOHDM", Hypertext 96, ACM.
<http://wwwx.cs.unc.edu/~barman/HT96/P52/section1.html>
- [4] Jesús Lorés Vidal, Juan Manuel Cueva Lovelle, Ana Belén Martínez Prieto, "Conceptos Básicos de Comunicación Hombre-Máquina", Cuadernos Didácticos Universidad de Oviedo nº 83
- [5] Varios Autores, "Historia de Asturias", Tomos IX y X. Editorial Silverio Cañada, 1978. ISBN 84-7286049-3

Pósters

“Information Walkthrough”

Visualización 3D de Sistemas de Información

Iván Fernández Lobo
Departamento de Informática.
Universidad de Oviedo, C/ Calvo Sotelo s/n, 33007 Oviedo (Asturias)
Tlf: 985.103.387 Email: lobo@acm.org

Resumen

A mediados de 1998 mi interés por la informática gráfica (en especial por los gráficos 3D interactivos) comienza a derivar hacia el campo de las interfaces de usuario. Para ser sincero, siempre me ha atraído la idea de trabajar para la industria del entretenimiento y debo confesar que por aquel entonces ya había empezado a aburrirme de los mecanismos de interacción al uso. Ratón, ventanas e iconos suponen un modelo de interfaz demasiado pobre para quienes han llegado (y continúan llegando) a la informática de la mano de las consolas de videojuegos e Internet. Mi sobrina, por ejemplo, se pasa el día transfiriendo información desde la consola de sus amigos para descargar esos dichosos “Pokémon”. ¿ Por qué yo no puedo hacer lo mismo divirtiéndome como ella ? ¿ Acaso no nos habían prometido algo así para el año 2000 ?

En mi modesta opinión, el siguiente paso en interfaces de usuario (o al menos el que dará un verdadero salto cualitativo) será la incorporación de tecnologías 3D al manejo y visualización de los espacios de información. Por supuesto, este avance hará la informática aún más adictiva pero, si la necesidad de divertirse trabajando aún no les convence, véanlo desde el punto de vista de la productividad. Les aseguro que los beneficios son aún mayores.

Palabras clave : Interfaces Post WIMP, Gráficos 3D, Visualización

1 Un paseo por los Sistemas de Información

Los problemas de espacio y de ocultación de información de los interfaces WIMP han sido discutidos hasta la saciedad por expertos de todo el mundo [1]. Pasar la representación visual de los datos al espacio tridimensional soluciona parcialmente estos dos problemas y añade, en el peor de los casos, una dimensión más de información. Conceptualmente, podrían obtenerse interesantes y variadas informaciones acerca de un sistema con tan solo mirarlo desde distintos puntos de vista. Las tres dimensiones posibilitan literalmente esta

libertad de observación. Podemos realizar cualquiera de las transformaciones clásicas del punto de vista, e incluso nuevos mecanismos de generación de vistas y transformaciones pueden ser explorados. Además, permite mostrar más objetos y expresar más propiedades en la misma cantidad de espacio visual que con cualquier método de visualización 2D.

La visualización 3D también ayuda a la identificación de nuevas metáforas. El modelado e interacción con objetos en el espacio ofrece muchas más posibilidades, y éstas van mucho más allá de las opciones planas basadas en ventanas, iconos o etiquetas de texto. Una serie de formas bien seleccionadas puede ayudar al usuario a entender y recordar estructuras y objetos complejos.

2 El diseño de “Information Walkthrough”

2.1. Evolución de una idea.

Hace varios meses comencé a trabajar en el diseño de “Information Walkthrough”, una alternativa 3D a los entornos de ventanas al uso. El problema de “inventar el futuro” es que enseguida te excedes en tus pretensiones. Intentas ser tan innovador que terminas dando un salto demasiado brusco y, por eso, las primeras ideas suelen resultar tan espectaculares como ineficientes. La siguiente etapa en el proceso creativo es radicalmente opuesta y lo complicado entonces es alejarse de las concepciones más conservadoras. IW ha sufrido este patrón evolutivo “en sus propias carnes”. En pocas semanas pasamos de mundos virtuales totalmente irrealizables, a metáforas que nos recordaban en exceso a los interfaces WIMP.

2.2. El Modelo Conceptual

El prototipo actual es “moderadamente” revolucionario. Se trata de un entorno virtual en el que se organizan un número ilimitado de habitaciones 3D. Cuatro de estas habitaciones se dedican a la gestión de datos, herramientas, tareas y usuarios, mientras que el resto son espacios de trabajo (workspaces) adaptados a las necesidades de cada una de las tareas. Esta división esta basada en la siguiente asunción :

$$\begin{aligned} < \text{Tarea} > = < \text{Más Tareas} > < \text{Tarea} > | (\text{Datos} + \text{Herramienta})_{\text{Usuario (s)}} \\ < \text{Más Tareas} > = < \text{Más Tareas} > < \text{Tarea} > | < \text{Vacío} > \end{aligned}$$

Es decir, cada tarea puede implicar a varias subtareas (tarea compuesta) o bien a unos datos, una herramienta y un usuario (tarea simple). Excepcionalmente, una tarea simple puede tener varios usuarios, pero sólo uno de ellos estará trabajando. El resto son

simplemente observadores. En cualquier momento, un usuario podría ceder su puesto a uno de los observadores. Los usuarios deciden en que workspace trabajan (o cual visualizan) en cada momento, pero siempre tienen acceso visual a las cuatro habitaciones principales. La siguiente figura ilustra esta idea.

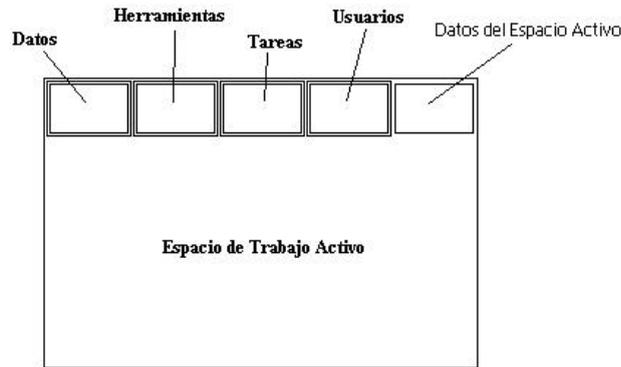


Figura 1. Configuración espacial de IW

2.3. La gestión del espacio de información.

Para las tareas de administración (habitaciones 1, 2, 3 y 4) hemos utilizado “Árboles Cono”. Estos árboles, propuestos por J.D. Mackinlay y Robertson en Xerox PARC [2], son “jerarquías desplegadas uniformemente en las tres dimensiones”. El elemento padre de la jerarquía se sitúa en el vértice del cono con todos sus hijos distribuidos uniformemente alrededor de él en un nivel inferior. Cada nueva capa de elementos se situará, a su vez, por debajo de la anterior, formando un cono con su elemento padre. Cuando un nodo es seleccionado, el cono rota hasta que este y cada nodo que forma su “path” aparecen resaltados en primer plano. La rotación de cada subestructura se realiza en paralelo, siguiendo el camino más corto, y de forma animada para que el usuario pueda ver y comprender perfectamente la transformación. La jerarquía es presentada en 3D para maximizar el uso de espacio en pantalla y permitir la visualización de la totalidad de la estructura. Los árboles cono ofrecen una especie de “Vista de Ojo de Pez” de la información en la que no es necesario definir una función de interés. La distancia a un nodo determina ahora el valor de esta función.

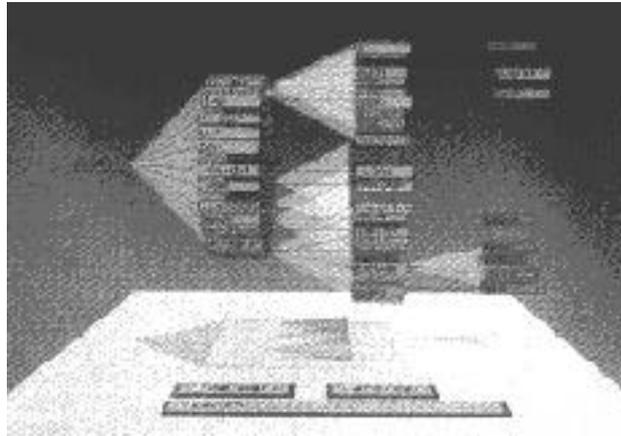


Figura 2. Ejemplo de Árbol Cono horizontal.

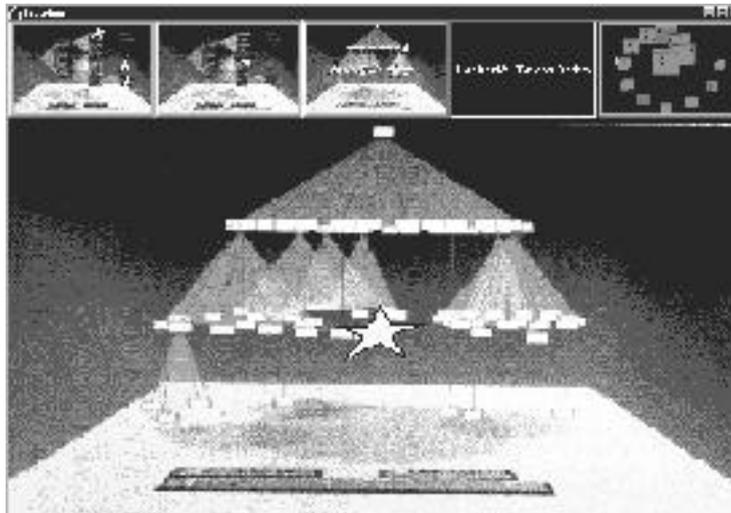


Figura 3. Vista previa de IW.

Para destacar la estrecha relación entre las distintas entidades del sistema, nuestro interfaz también modifica la iluminación y disposición de los cuatro árboles conos en función del foco de atención del usuario. Por ejemplo, cuando se selecciona una tarea, los ficheros que se modifican en ella, la herramienta que se utiliza y los usuarios implicados, son resaltados

en el resto de los árboles. De esta manera el interfaz se adapta en tiempo real a las necesidades del usuario.

El problema de este tipo de estructuras (árboles cono) es que sólo facilita información estructural del sistema. En SGI FSN [3], por ejemplo, la representación visual de un fichero también daba pistas sobre su tamaño y edad.

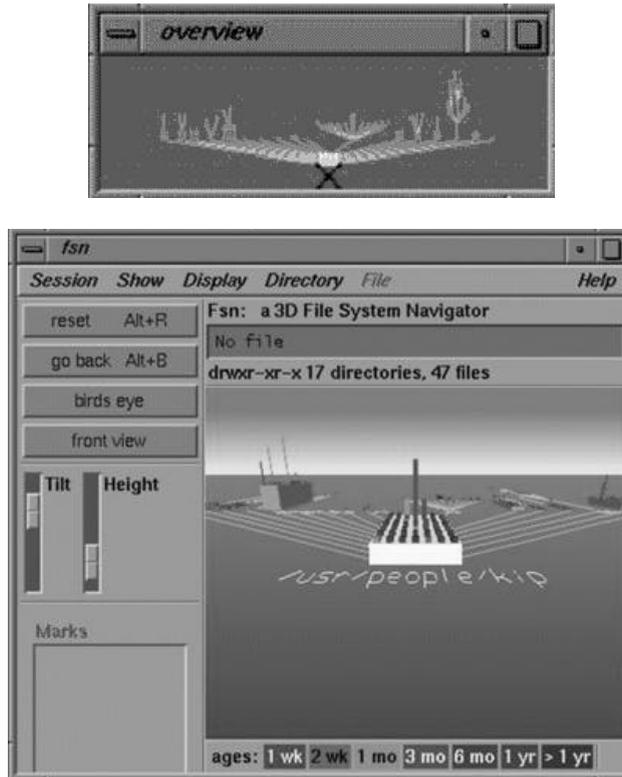


Figura 4. SGI FSN : Vistas de un sistema de ficheros UNIX.

Para integrar este tipo de información en nuestro modelo de interacción hemos creado el concepto de árbol cono contraído. A petición del usuario, un árbol cono debe ser capaz de recogerse sobre un único nivel, escalando su representación visual proporcionalmente al valor de sus atributos. Esto permitirá una comparación visual inmediata de las propiedades de los objetos de información. Un simple vistazo al árbol contraído nos permite identificar el elemento más antiguo o de más ocupación del sistema.

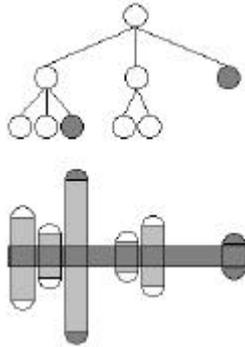


Figura 5. Árbol Cono contraído.

(Elementos de mayor y menor valor seleccionados)

2.4. Las aplicaciones.

Hasta ahora sólo nos hemos detenido en detalles relacionados con la gestión del sistema de información, e incluso comenzamos a intuir cuales son las mejoras que IW propone a este nivel. Sin embargo, el éxito de un sistema informático suele medirse por la funcionalidad ofrecida por sus herramientas. Los primeros sistemas WIMP fueron diseñados como aplicación para el trabajo de oficina y, como herencia directa de este objetivo, el interfaz de sistema actual aun sigue una metáfora de escritorio.

Con el paso del tiempo, las aplicaciones de los sistemas informáticos han ido cambiando. El PC ya no es patrimonio exclusivo de la oficina, y su aplicación resulta vital en trabajos tan diversos como la arquitectura, el control de la producción, la seguridad vial, etc. En este sentido, el diseño de IW también suponen un avance importante. Cada habitación constituye un entorno de trabajo amplio e independiente que ofrece gran potencia tanto a usuarios como a desarrolladores de aplicaciones. Además, al permitir organizar las tareas de forma jerárquica, el control y seguimiento de éstas resulta más sencillo y cercano a los planes de trabajo del “mundo real”.

3 Nuevos retos para el futuro

Si bien la inclusión de gráficos tridimensionales es un factor a tener muy en cuenta, quizás uno de los retos más interesantes pase por dotar a esos gráficos de “inteligencia” a varios niveles. Por una parte, los gráficos deben ser inteligentes a nivel de diseño. Al tener gestionar entornos mayores y más complejos, la organización espacial de los distintos

elementos del interfaz cobra una importancia decisiva. De nada servirán espacios de trabajo ilimitados y con gran funcionalidad si los objetos no son fácilmente alcanzables por el usuario.

Un segundo nivel de inteligencia, consistiría en dotar a los objetos de comportamiento, tanto reactivo como de más alto nivel. Animaciones, iluminación y sonido que ilustren lo que esta sucediendo en el sistema, agentes que muestren comportamientos similares al humano, etc. pueden ser de gran ayuda para conseguir mayor potencia en la interacción. De este modo, como si de un juego de estrategia se tratara, los objetos podrían colaborar "activamente" en las tareas del usuario.

En definitiva, los próximos años prometen nuevas y emocionantes sorpresas en el mundo de la Interacción. Puede que no tengan mucho que ver con las aquí presentadas pero, lo que es seguro, es que aun tardaremos en experimentarlas fuera de los laboratorios. Así que, de momento, siéntense cómodamente en sus sillones, y continúen soñando con un futuro mejor.

Referencias

- [1] Henderson, D.A., Card, S.K. "Rooms : The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface". *ACM Transactions on Graphics*, vol 5, no. 3, pag. 211-243, Julio 1993.
- [2] Robertson, G.G., Card, S.K., Mackinlay, J.D., "Information Visualization using 3D Interactive Animation". *Communications of the ACM*, vol. 36, no. 4, pag. 57-71, Abril 1993.
- [3] Tesler, J., Strasnick, S. "FSN : 3D Information Landscapes". Man pages entry for an unsupported but publicly released system from Silicon Graphics, Inc. Mountain View, Calif. Abril 1992.

Cinemia Astur: Herramienta Generadora de Títulos Hipermedia de Interfaz Flexible

José Manuel Redondo López ⁽¹⁾,
Antonio J. Sánchez Padial ⁽²⁾, Ana María García Fernández ⁽³⁾,
B. Martín González Rodríguez ⁽⁴⁾

Grupo de Investigación en HCI, Laboratorio de Tecnologías Orientadas a Objetos,
Campus de Llamaquique, Departamento de Informática
c/ Calvo Sotelo, s/n. 33007 OVIEDO
e-mail: i1637886@petra.euitio.uniovi.es, tonisan8@hotmail.com
(3) i9417208@petra.euitio.uniovi.es, (4) martin@lsi.uniovi.es

Resumen

Este documento explora las reglas de diseño de interfaces de una aplicación *Hipermedia*, exponiendo las decisiones que se han tomado para los mismos y las razones que nos han inclinado a hacerlo de ese modo. También se menciona una posible forma de cómo hacer que sean las aplicaciones las que estudien a su usuario, y se adapten a el, y no al revés, como es habitual en el mundo de la informática actual.

Palabras clave: Componente, Plantilla, Cognición, Hipermedia, Activador.

1 Producto desarrollado

CineMedia Astur es una aplicación desarrollada íntegramente en Java capaz de realizar aplicaciones *Hipermedia*, usando para ello componentes *JavaBeans* de procedencia cualquiera, permitiendo al usuario la manipulación gráfica de los mismos a través de sus propiedades visibles mediante controles. Los componentes pueden desplazarse en todo momento por cualquier parte del área de trabajo de la aplicación, mediante técnicas de *Drag and Drop*, y podrán interactuar con otros componentes.

La aplicación trabaja con proyectos, siendo cada uno de ellos configurable no solo en contenido sino también en la interfaz a través de una fuente externa (como un fichero – plantilla), pudiendo hacer que dicha aplicación sea completamente diferente en función del proyecto que se cargue sin tocar una sola línea de código.

La metáfora seleccionada para realizar esta aplicación es el modelo de gestores de tarjetas (o ventanas), usada por herramientas para *Hipermedia* tales como *ToolBook*, *HyperCard*, *HyperStudio* y *SuperCard* y que normalmente se considera la metáfora más sencilla de asimilar. Los productos multimedia creados bajo esta filosofía están basados en uno o varios conjuntos de tarjetas o ventanas, en cada una de las cuales se pueden colocar componentes multimedia a discreción [1]. Cada tarjeta podrá estar conectada con una o varias tarjetas a través de enlaces especiales llamados destinos o transiciones, disparados mediante la gestión de eventos generados dentro de un nodo.

2 Aspectos generales sobre la interacción persona – ordenador

En el desarrollo de interfaces de *CHM* queríamos evitar principalmente dos tipos de riesgos:

1. La saturación de información, que se produce cuando, por ejemplo, intentamos englobar en un mismo bloque conceptual muchas opciones o una cantidad excesiva de botones, consiguiendo ralentizar y dificultar el trabajo con la aplicación. Se sabe que la memoria a corto plazo (*MCP*) humana es capaz de recordar entre 5 y 9 elementos de información puestos juntos en una misma entidad o grupo conceptual [2], pero no conviene llegar al límite puesto que nuestra intención es hacer accesible nuestra aplicación a todo el mundo, y no solo a los más capacitados. Nos interesa producir en el usuario un efecto de '*Chunking*' (consistente en que el usuario, a partir de pequeños elementos de información, forma estructuras más complejas), tal y como define Newell [3].
2. La saturación de visualización consiste en el exceso de 'espectacularidad' de un interfaz de usuario. Cuando se diseña una interfaz se puede caer en la tentación de resaltar elementos del mismo con colores vistosos y llamativos, con sonidos (más o menos estridentes) y con otros elementos. Esto puede causar que el usuario reduzca la productividad al centrarse en estos aspectos, no siendo esto un objetivo deseable en ningún producto [4].

Diseño de componentes gráficos

Los menús que se implementaron en la aplicación se definirán en ficheros – plantilla externos, al igual que otros componentes como barras de herramientas, etc. El método

tradicional de implementar las llamadas a las opciones en el código, a través de la captura de eventos, hace muy difícil el mantenimiento. En estas plantillas, además de otros elementos como iconos, texto, etc. (típicos de cualquier componente) e incluso la operación a ejecutar, se especificará lo que se llamó *Activador*, que es una clase Java que a partir del estado actual de la aplicación, que el se encargará de analizar, determinará el aspecto (textos, iconos, etc.) adecuado de las opciones de menú asociadas al mismo, así como su disponibilidad. Los activadores se lanzarán cuando el usuario pulse en el menú asociado, tal y como se muestra en la figura:

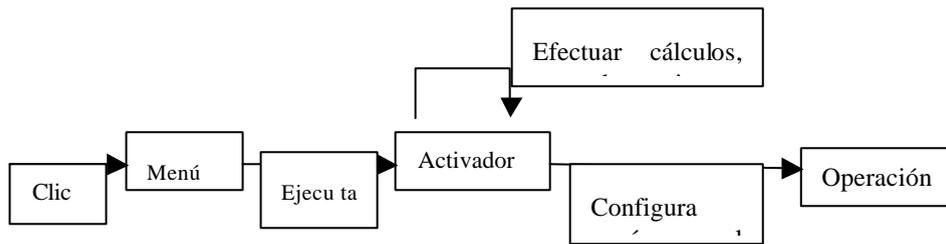


Figura 1. Operaciones de un Activador en la aplicación

Este diseño permitiría realizar estudios para analizar el comportamiento del usuario actual (en la parte de registros, controles, etc.), como por ejemplo el estudio de cuanto tiempo deambula con el ratón dentro del menú, cuentas veces se usa la opción x, etc. Esto se podría programar implementando un contador de pulsaciones o un temporizador de presencia del ratón dentro del activador, registrándonos al evento que deseemos para hacer el estudio.

De este modo podemos analizar al usuario y obtener un perfil de sus acciones con la aplicación, lo que permitiría incluso programar una aplicación inteligente, que modificase su interfaz automáticamente para ponerse en consonancia con el perfil de un determinado usuario, estableciendo previamente una serie de patrones [5]. Esta es una técnica similar a la empleada por *Ergolight* [6]. Esto último podría realizarse perfectamente en nuestra aplicación, puesto que cada menú (u otro componente) se lee de una plantilla contenida en un fichero externo, el cual puede modificarse y releerse por el programa, no teniendo que tocar el código del mismo (una de las razones por las cuales en otras aplicaciones no se podrían hacer operaciones similares), adaptándose en tiempo real a cualquier nuevo requisito.

Las plantillas están hechas con un lenguaje genérico y sencillo de especificaciones, en las que se indicarán nombre, icono, mnemónico, acelerador (si aplicable) y operación a ejecutar (a través de la localización de la clase Java que contiene la operación), todo ello traducido al idioma correspondiente a través de *Locales* y *ResourceBundles* de Java, además del

mencionado *Activador*. Dejamos así que todos los elementos visuales de uno de estos componentes pueda ser configurado externamente. El sistema de plantillas externas para los componentes permite pues, diseñar una aplicación a medida del usuario, bien analizándolo automáticamente o bien dejando que este sea el que elija que es lo que desea ver en pantalla, independizando esta parte (tan sujeta a modificaciones en la mayoría de los casos) de la aplicación, facilitando el mantenimiento de la misma.

Se detallan a continuación una serie de puntos que se siguieron en el desarrollo de la aplicación:

1. Se evitó el exceso de iconos y de combinaciones de acceso rápido (demasiadas de estas las convierten en combinaciones de acceso lento, el usuario es incapaz de recordar ninguna). El exceso de iconos en un menú hace que el usuario tarde más en ver la opción que esta buscando, puesto que se va a fijar primero en los iconos que en el texto [4], a lo que contribuye que estén colocados normalmente a la izquierda, lugar al que inconscientemente miraremos primero, según el sentido de lectura occidental [7].
2. Todas las opciones tienen su mnemónico único dentro del menú asociado, para que el usuario no dependa del ratón todo el tiempo, favoreciendo la *Comunicación Multimodal* en su forma más primitiva.
3. Nunca se colocaron más de 4 opciones juntas sin que haya un separador de por medio [4] [7]. Las opciones entre separadores están relacionadas de algún modo, de forma que puedan identificarse áreas o grupos de ellas. Como puede existir la posibilidad de opciones aisladas completamente distintas de las demás, estas se colocan las últimas en el menú, en un grupo aparte. Si existiesen más de 4 opciones relacionadas con el mismo tema, se procuró separarlas en submenús.
4. Los menús demasiado largos pierden al usuario. No se implementan menús con excesivas opciones ni con numerosos grupos de las mismas, para que los usuarios no tengan que estar buscando entre todas ellas.

4 Conclusiones

Podemos concluir que el diseño de nuestra aplicación está preparado para admitir un subsistema de seguimiento al usuario (o usuarios) que la usen, abriendo así una puerta al estudio de los mismos y que hemos conseguido independizar la interfaz de la aplicación de las funciones de las misma totalmente, siendo además dicho interfaz totalmente configurable en tiempo real.

Referencias

- [1] González Rodríguez, Bernardo Martín; 1999; "ANTS: An Automatic Navegability Testing Tool for Hipermedia in Proceedings of the Eurographics Workshop" in Milano, Italy, September 7-8. SpringerWienNewYork.
- [2] Miller, George A;1976; "Perception and Motivation". Jon Wikey and Sons, Nueva York.
- [3] Newel; 1990; "Unified Theories of Cognition". Harvard University Press. ISBN 0-674-92101-1
- [4] Faraday, Pete; 1999; "Visually critiquing Web pages in Proceedings of the Eurographics Workshop" in Milano, Italy, September 7-8. SpringerWienNewYork.
- [5] González Rodríguez, Bernardo Martín – Cueva Lovelle, Juan Manuel;1999. "Un HDP para JEDI – LEIA". Editorial SERVITEC, Oviedo.
- [6] Ergolight Usability Software GUI and test for Windows; 1999; <http://www.ergolight-sw.com>
- [7] Gordon, Druce; "Designing Covers, Mac Format". Future Publishing Company.

Tirsus II, Aplicación de Hipermedia para la Enseñanza de la Historia

Antonio J. Sánchez Padial⁽¹⁾, Ana María García Fernández⁽²⁾,
B. Martín González Rodríguez⁽³⁾, José Manuel Redondo López⁽⁴⁾

Grupo de Investigación en HCI, Laboratorio de Tecnologías Orientadas a Objetos,
Universidad de Oviedo, Campus de Llamaquique, Departamento de Informática
c/ Calvo Sotelo, s/n. 33007 OVIEDO

e-mail: ⁽¹⁾ tonisan8@hotmail.com, ⁽²⁾ i9417208@petra.euitio.uniovi.es,
⁽³⁾ martin@lsi.uniovi.es, ⁽⁴⁾ i1637886@petra.euitio.uniovi.es

Resumen

En este documento se describen las decisiones de diseño de navegación e interfaz de una aplicación multimedia cuyo objetivo es la enseñanza y su base de conocimiento se caracteriza por su falta de homogeneidad.

Palabras clave: modelo de usuario, modelo navegacional, multimedia, OOHD, pedagogía

1 Introducción

Tirsus es una línea de investigación del grupo HCI (*Human Computer Interaction*) desarrollado a partir de Proyectos de Fin de Carrera de la EUITIO centrados sobre la pedagogía y tomando como tema la historia de Asturias.

Tirsus II es la primera aplicación hipermedia cuyo contenido es la Historia Antigua de Asturias, aproximadamente del 1000 a.C. al 500 d.C. La base de conocimiento destaca por la falta de información; existen diversas teorías históricas, a menudo enfrentadas. Se evitó comparar estas teorías para no confundir a un usuario lego, siendo la solución adoptada la de combinar los datos.

2 Proceso de diseño

El objetivo de la aplicación es servir de apoyo informativo a las salas del Museo Arqueológico de Asturias. Los usuarios potenciales son variados, pero el proyecto ha sido enfocado a estudiantes (de primaria o secundaria) debido al carácter *básico* de la información que ofrece. También hacia el visitante ocasional del museo, poco conocedor de la base de conocimiento.

La metodología empleada en el diseño es OOHDM (*Object Oriented Hipermedia Design Method*) [1, 2]. Esta metodología diferencia varias fases sucesivas e iterativas en el proceso de desarrollo, de forma que éste es modular y con mayor capacidad de reusabilidad.

2.1 Diseño conceptual

El diseño conceptual es la fase en que se modela la base de conocimiento de la forma más neutral posible [1, 2]. Tanto el director del Museo Arqueológico, como algunos estudiantes de Historia colaboraron corrigiendo la información recopilada. En primer lugar se escribió un guión (idea cercana al diseño navegacional) para después clasificarlo en forma de diseño conceptual. Se ve cómo el diseño es iterativo. En el diseño conceptual se crea un modelo orientado a objetos formado, principalmente, por clases conceptuales y relaciones [1, 2].

2.2 Diseño navegacional

Esta es la fase en que diseñamos la aplicación teniendo en cuenta los usuarios a los que va dirigida y los objetivos de la misma. Consta de dos partes:

1. Creación del modelo de clases navegacionales. En él se crean los nodos y sus enlaces, como una vista (en el sentido de las bases de datos) del diseño conceptual [1, 2].
2. Diseño del modelo de contextos de navegación. Los contextos de navegación son los modos en los que el usuario puede acceder a la información [1, 2]. El usuario puede acceder a la información clasificada cronológicamente (guerras, romanización, ...) o bien, puede acceder a la misma clasificada por temas (economía, arte, ...). Los temas concretos dependían de la información disponible (del diseño conceptual). Además se plantea al usuario la posibilidad de comparar nodos fuera del contexto.

El diseño navegacional se modificó para solucionar los problemas en el diseño de la interfaz. El problema principal es que la pantalla era demasiado pequeña para cada nodo. Las soluciones fueron la reestructuración de los nodos; lo que modificó el diseño navegacional, y la creación de diferentes niveles de información.

2.3 Diseño de la interfaz abstracta

Una vez efectuado podemos dividir el diseño de la interfaz en dos fases:

1. La primera fase consiste en buscar una metáfora visual. Gracias a las metáforas el usuario desarrolla más rápidamente los mecanismos de uso de la aplicación; aplicando modelos mentales que ya posee [3]. Se plantearon diversas metáforas para compararlas, fueron las siguientes:

- La primera metáfora planteada fue la del libro. En ella, cada nodo representa la página de un libro. Un triángulo en una esquina de la pantalla representa la posibilidad de pasar una página. Se descartó porque no era bastante predecible ni clara.
- La segunda metáfora fue la de la máquina del tiempo, muy apropiada para niños. Pero el viaje en el tiempo no era factible, pues en la información existen lagunas de cientos de años.
- Finalmente se apostó por la metáfora del *proyector*. La navegación se realiza mediante un *mando a distancia*, concepto familiar a la mayor parte de los usuarios, ver Figura 1-a. Los usuarios pueden solicitar información adicional mediante símbolos fuera de la metáfora del *proyector*, ver Figura 1-b; o conseguir navegación adicional con hipertexto. Las primeras pruebas con un pequeño grupo de usuarios lo valoraron positivamente.



Figura 1. Mando a distancia y elemento auxiliar

2. La segunda parte consiste en el diseño de cada nodo de la aplicación en concreto. Los problemas encontrados (y sus soluciones) en esta parte se comentan a continuación. Hacer notar que la reestructuración afecta al diseño navegacional, mientras que los niveles de información sólo afectan al diseño de la interfaz.

3 Estructuración de los nodos

A menudo la pantalla resulta demasiado pequeña para la información que hay que meter en cada nodo [4]. Este problema apareció ya en las primeras pruebas de la interfaz. Como medida se reestructuraron los nodos, de forma que las diversas secciones que aparecían en los nodos originales se transformaron en nuevos nodos. De cada bloque vertical de la Figura 2-a se extrajo un nodo como el de la Figura 2-b. Al ser esto insuficiente, se repitió el proceso: de cada apartado de la Figura 2-b (mapa, dolmen, ...) se extrajo un nodo como el de la Figura 2-c.



Figura 2. En los sucesivos diseños se simplificó cada nodo.

4 Niveles de Información

Una vez reestructurados los nodos el problema persistía. No se podía combinar toda la información que se deseaba ofrecer con un diseño estético y de impacto visual. Este problema que aparece en [4], se resolvió distribuyendo la información de cada nodo en tres niveles:

1. El primer nivel corresponde a la información visual, compuesta por gráficos, animaciones explicativas y textos cortos. Este nivel es el de más rápida asimilación y se corresponde con la Figura 2-c, 3-a.
2. En el segundo nivel de información está la voz en *off*. Este segundo nivel en *off* no sólo complementa el primero; sino que lo repite de forma que se puedan salvar posibles dificultades del usuario.
3. El tercer nivel sólo se ofrece bajo demanda del usuario[4]. Éste nivel usa el canal de texto escrito que es el que más atención requiere para ser captado, pero ofrece la información más completa (Figura 3-b.)

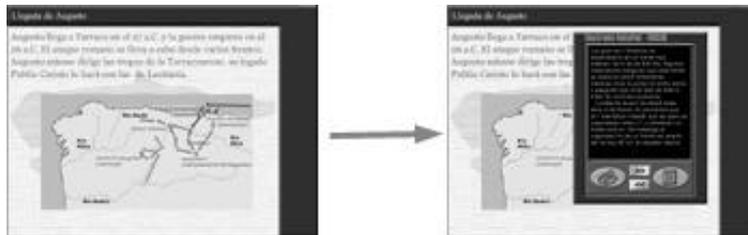


Figura 3. Nivel de información visual y nivel de información bajo demanda

5 Aprendizaje

Esta aplicación usa dos técnicas fundamentales de enseñanza:

1. Por un lado, el método de comparación: en cada nodo, al usuario se le da la opción de viajar a otro nodo que es el “antagonista” de éste, ver Figura 4.

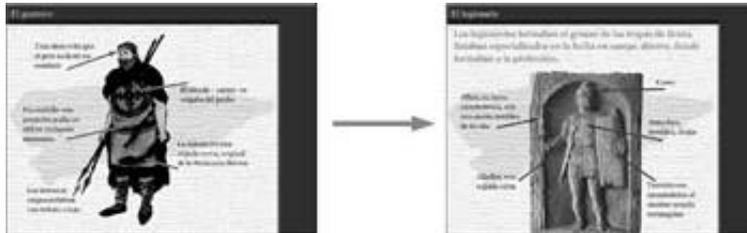


Figura 4. Comparamos al guerrero astur con un legionario romano

2. Además, al final de cada apartado, se ponen en práctica técnicas de logopedia proporcionando juegos sencillos y puzzles en los que se mide la destreza del usuario y su capacidad de síntesis de los conocimientos representados a lo largo de la unidad didáctica.

6 Conclusiones

Actualmente la aplicación está en la fase de pruebas. En la fase de diseño hemos visto cómo algunos problemas de diseño de la interfaz requieren cambios en el diseño navegacional, mientras otros fueron resueltos sin modificar éste. Al ser estos problemas (y sus soluciones) comunes en el campo de hipermedia, hace pensar que una investigación sistemática de los patrones de diseño sería de interés para sistematizar los procesos de desarrollo [4,5].

Agradecimientos

Adentrarse en un tema como la historia no es fácil, menos si el período es oscuro y aún menos proviniendo de una disciplina tan lejana (y no sólo en el tiempo). Por ello quiero agradecer su colaboración, abriendo siempre sus puertas y revisando los guiones a d.Enrique García Tessier, exdirector del Museo Arqueológico de Asturias. También a Elena Cambor, a Jorge Coto y a los demás de la lista de correo de historia de Asturias que también corrigieron mucho. Por último a los miembros de la Asociación de Amigos del Museo que han colaborado en las pruebas de usabilidad del programa.

Referencias

- [1] Daniel Schwabe, Gustavo Rossi, Simone D.J. Barbosa, "Abstraction, Composition and Lay-Out Definition Mechanism in OOHDM",
<http://www.cs.tufts.edu/~isabel/schwabe/MainPage.html>
- [2] Daniel Schwabe, Gustavo Rossi, Simone D.J. Barbosa, "Systematic Hypemedia Applications Design with OOHDM", Hypertext 96, ACM.
<http://wwwx.cs.unc.edu/~barman/HT96/P52/section1.html>
- [3] Jesús Lorés Vidal, Juan Manuel Cueva Lovelle, Ana Belén Martínez Prieto, "Conceptos Básicos de Comunicación Hombre-Máquina", Cuadernos Didácticos de la Universidad de Oviedo nº 83
- [4] Daniel Schwabe, Gustavo Rossi, Simone D.J. Barbosa, "Usign Design Patterns in Educational Multimedia application", ED-MEDIA'98
- [5] Gustavo Rossi, Daniel Schwabe, Alejandra Garrido, "Design Reuse in Hypermedia Applications Development", Hypertext 97, ACM.

Creación de una herramienta multimedia para la distribución de actas en un congreso. Adaptación de las técnicas de comunicación a los soportes multimedia

A.Gil†, J. Muñoz‡.

† Departamento de Informática y Automática.
Universidad de Salamanca, Plaza de la Merced S/N, 37008, Salamanca
Tlf: 923. 294.400. ext. 1302 Fax: +34-923-294514 e-mail: abg@gugu.usal.es

‡ Departamento de Ingeniería Mecánica
Universidad de Salamanca
Avda. Fernando Ballesteros 2, 37700 – Béjar (Salamanca)
Tlf: 923 402.416. Fax: 923 401.189. e-mail: juanramr@teleline.es

Resumen

Se exponen algunas experiencias en la elaboración de documentos multimedia con soporte en CD-ROM que hemos realizado. Las posibilidades que estamos descubriendo y la transición a una comunicación más eficaz, más centrado en el usuario que este nuevo medio nos está facilitando. También se exponen ciertas técnicas de interacción en el soporte electrónico para una transmisión más eficaz y amena de la información.

Palabras clave: CD-ROM, interactividad, técnicas de comunicación.

1. Introducción:

El avance de *las tecnologías multimedia* ha supuesto la posibilidad de aplicar en un ámbito tecnológico el desarrollo de nuevas técnicas de comunicación. Las actuales tendencias comunicativas aparecen como un proceso dinámico, que se construye a través de la interacción del usuario con la propia información.

No hay duda en que las nuevas tecnologías de la información y la comunicación están cambiando la vida cotidiana y por ello, también van a transformar los soportes, como señala Castell(1997, 55-56): “...al final del siglo XX, vivimos uno de esos raros intervalos de la historia. Un nuevo intervalo caracterizado por la transformación de nuestra cultura

material por obra de un nuevo paradigma tecnológico organizado en torno a las tecnologías de la información”.

El cambio acelerado que caracteriza nuestra sociedad implica necesariamente el desarrollo de sistemas de comunicación que permanentemente respondan a las cambiantes exigencias del sistema productivo y a los retos de la nueva sociedad.

Nos encontramos entonces ante una auténtica segunda "imprensa", imprescindible para la vida social y cultural en la medida en que los nuevos soportes de la información, y a la postre de la cultura, están imponiéndose por **su mayor rendimiento económico**. Se está convirtiendo en un elemento cotidiano en la vida de cualquier persona, y en un valor añadido para cualquier empresa, desaprovechar las oportunidades que ofrece la incorporación de estas tecnologías, sería como decir que la imprenta fue un invento innecesario para la propagación de la información y la cultura en todo el mundo.

2. Objetivos del proyecto

La propuesta surge como una aportación más en la organización del congreso "Las nuevas técnicas de la comunicación Científica", con celebración en la Universidad de Salamanca en Mayo del 2000. El proyecto consistiría en ofrecer las actas de los ponentes en formato electrónico a todos los asistentes al congreso junto con el resto del material. Se sustituye así al usual y voluminoso libro en papel, que hasta entonces se venía dando en todos los congresos.

Se impusieron una serie de características : facilidad de ampliación, capacidad de conversión según las necesidades del programa, su acomodación a varias funciones, aceptación de cualquier formato de texto... y una importante consideración técnica, el que fuera ejecutable desde cualquier tipo de ordenador convencional aunque para la plataforma PC. Todas estas cualidades son factibles con el no espacio y muchas posibilidades de los nuevos medios.

Optamos por configurar un CD con la herramienta de autor Director 8 de la empresa Macromedia, que integraría toda la serie de gráficos, texto, sonido... previamente elaborado con su software específico.

Se valoraron distintas opciones de tratar el texto, que era al fin la base fundamental del trabajo y la parte más complicada por su volumen: eran cinco mesas con una media de ocho ponentes cada una y cada artículo contenía alrededor de veinte páginas. Cada ponente manda su artículo en diferente formato: *.txt, con distintos procesadores de textos, *.pdf, *.ps, etc. De modo que se optó por tratarlo a partir del volcado de todos a PDF. En función del formato finalmente elegido y su tratamiento, se realizó toda la programación y la estructura del trabajo expuesto.

En cualquier discurso se promueve el estimular la reacción por parte del oyente/usuario y enriquecer la transmisión. Esto se consigue en el medio empleado mediante la interactividad que se le exige al usuario en el entorno de visualización[1]. Se permite además, la presentación de información de forma secuencial y sin la intervención por parte del usuario pero incluso en este caso hay diferentes métodos para mantener la atención del usuario. Estos métodos son gráficos en movimiento, control del tiempo de exposición de cada una de las pantallas de información y efectos gráficos de cambio de pantalla. El *objetivo* era conseguir un entorno amigable en el que usuario, de forma sencilla accediera a una información, de tipo principalmente textual.

Es importante en este punto, considerar el *tipo de usuario* al que nos dirigimos, son personas de 23 a 58 años con una formación universitaria sólida, y familiarizados con el mundo de la información y la tecnología, la mayoría de ellos procedentes de áreas de biblioteconomía que acuden a un congreso sobre su campo de trabajo. Esto es un punto principal a la hora de desarrollar, tanto el diseño gráfico como la interactividad. Deberá de ser un formato claro y consistente, sin excesivas animaciones ni efectos especiales, ya que quien vaya a visualizar la información estará ya suficientemente motivado y sin embargo habrá que mantener la intensidad de la comunicación con ciertas técnicas afines al usuario descrito: debe de ser novedoso pero inspirar confianza.

Se incluye además la posibilidad de obtener las ponencias en formato PDF para ser impresas directamente y sin ningún tipo de entorno multimedia. Se tiene en cuenta con esta opción a todos aquellos usuarios que prefiriendo acceder simplemente a un documento de texto, evitan el tener que navegar entre sonidos y gráficos de botones para acceder a la información.

3. Diseño de estructura de datos

Es importante señalar en este punto cómo el CD es parte de un conjunto de productos del congreso junto con la cartelería, los folletos informativos en papel y las escarapelas. Todo ello comparte un diseño gráfico común que el CD mantiene, reforzando los objetivos de comunicación globalizadora y de confianza que se persiguen.

El *trabajo interactivo o CD* se divide en un conjunto de escenas que agrupamos en los siguientes bloques:

Pantallas de presentación: animación de entrada, cuando el usuario introduce el CD en el ordenador y a modo de anuncio publicitario se presenta el título del congreso, la dirección del mismo y las empresas e instituciones colaboradoras con un tratamiento de sus logotipos, todo ello acompañado de una música de fondo. Ver Fig.1



Fig.1 : Imagen de la animación de presentación.

Menú principal: se trata del primer nivel para el usuario. Ofrece el acceso a varias escenas (información general, conferencia inaugural, comunicaciones, Salamanca y salir). Introduce la estructura estándar que se consolidará en el resto de las pantallas. Con el título en la viñeta superior izquierda, los organismos promotores del congreso en el área superior y el diseño gráfico común. Ver Fig.2

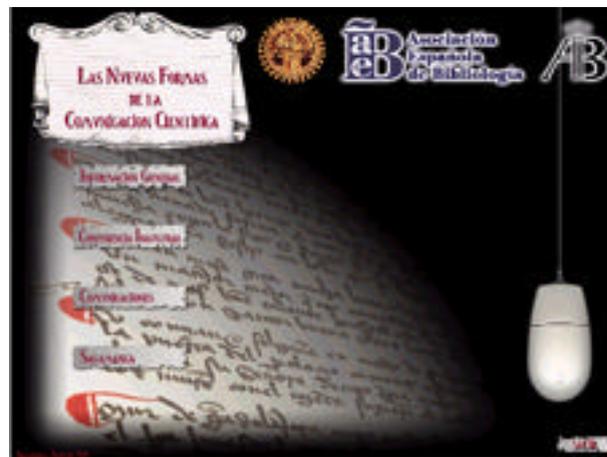


Fig.3 : Pantalla del menú principal

Información general: pantalla donde se detallan todos los puntos del Congreso, lugar, fecha, temas, organización... en forma de Scroll. Información que se añade de modo que complete todo lo referente al congreso en el soporte empleado, el CD , pero es importante hacer notar que en los trámites de anuncio y publicidad del congreso estuvo alojada convenientemente en una página Web. Ver Fig.3

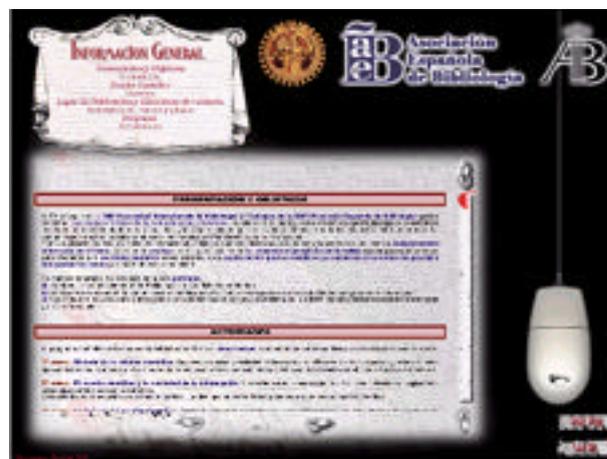


Fig. 3: Pantalla de información general

Conferencia inaugural: pantalla donde aparece el texto de la conferencia inaugural al congreso.

Comunicaciones: Pantalla de acceso a la presentación de las cinco mesas existentes en el congreso donde se da información de los temas tratados en cada una y se da acceso a cada uno de los participantes que presentan las ponencias en ellas. Ver Fig.4



Fig. 4: Pantalla relativa a ponencias de la mesa 3

Ponencias: Montada mediante una pequeña base de datos a la que se accede mediante la pantalla de comunicaciones. Posibilita la lectura del acta elegida en la mesa a la que se optó.

Salamanca: Pantalla de acceso a un álbum de fotos de la ciudad que aloja el congreso. De modo que el usuario elige en la zona del mapa el lugar que quiere ver en fotografías y navega a través de ellas por sus edificios y sus calles. Proporciona la inclusión de este apartado una faceta visual muy agradable y externa al propio objetivo del trabajo, pero que estos nuevos soportes el aligerar el contenido tan textual del proyecto. Ver Fig.5



Fig. 5: Pantalla de visualización fotográfica de Salamanca

El CD es además autoejecutable, de modo que el usuario con tan sólo introducirlo en la bandeja de su CD ya comenzará a visualizar la aplicación.

4. Implementación de interactividad

Todas estas estructuras descritas en el apartado anterior están apoyadas en una sencilla **navegación**, mediante una técnica muy sencilla para el usuario, consistente en el pulsado de simples botones.

El flujo a través de las estructuras es lineal de modo que se accede a cada nivel de modo directo pulsando los citados botones y con la opción de pulsando en el botón “volver” se retrocede un nivel en la jerarquía. En la misma idea de accesibilidad y flexibilidad el usuario puede en cualquier momento abandonar el CD pulsando el botón “salir”, que de forma instantánea, le devuelve al escritorio de su ordenador.

Se incluye además una *ayuda visual* a la utilización de toda esta información, bien con la simulación mediante eventos del comportamiento real de un botón (incluyendo estímulos sonoros) o bien reforzado, en las ocasiones en que se estima necesario, mediante cursores que cambian de forma informando que si se hace clic con el ratón en aquella zona “algo ocurre”. Todo esto constituye lo que pretende ser una guía intuitiva y directa a la consulta y manejo de toda la información almacenada en la aplicación.

Llegados a este punto en la navegación, es importante destacar como para mantener la consistencia y la claridad de las cosas, los botones aparecen siempre con el

mismo aspecto y acción de una pantalla a otra, incluso botones con la misma función mantienen su posición en el área de pantalla.

El fijar una estructura sólida con un flujo entre escenas directo y sencillo, *evita el riesgo de perderse* en toda esta cantidad de información. Apoyando a este objetivo, se estandarizan los elementos importantes de la interfaz .

Tan importante como evitar la confusión en un trabajo de estas características es el *mantener el interés del usuario*[2]. Para ello, se hace necesario presentar variaciones de estilo entre los distintos niveles, dentro de la ya citada estandarización, de modo que den al usuario no sólo idea de exploración o progreso sino además cierta distracción y sentido de sorpresa. Reforzamos esta idea con efectos gráficos y transiciones que suponen un seguimiento de la información, y con un cursor animado que mantiene la atención del usuario en los puntos de espera entre los bloques.

5. Conclusiones y vías futuras

Con el diseño de este CD se ha conseguido construir una herramienta de soporte a la distribución de comunicaciones o actas de un congreso. La posibilidad de trabajar con datos multimedia abre una nueva oferta a la inclusión de materiales en cualquier acta, de modo que bien podría incluirse un video explicativo, unas fotos incluso sonidos, atendiendo a la variedad de los campos de investigación e incluyendo diversidad en los materiales de apoyo a cualquier exposición.

Toda esta información adicional, no supone problemas a la hora de ser utilizada desde cualquier ordenador. En concreto su principal utilidad es la creación y visualización de presentaciones que permiten a los usuarios una gran interactividad con las mismas. Se ha conseguido un entorno fácil de utilizar pero sin que esto disminuyera las capacidades de la aplicación.

Por otra parte, gracias a la programación orientada a objetos que se ha realizado, queda lo suficientemente abierto para que en un futuro puedan ampliarse sus posibilidades con la incorporación de nuevas capacidades multimedia(audio y vídeo) sin demasiados cambios.

Todas estas posibilidades de ampliación, hacen que atendiendo a las demandas propias de cada campo se vayan incluyendo más opciones y principalmente un nuevo módulo de búsquedas en la base de datos de las ponencias, lo que facilitaría la consulta y aumentaría la utilidad de la herramienta.

Otro aspecto importante a la hora de incluir este formato como opción a distribuir cualquier tipo de información de gran volumen es el *ahorro económico* de éstos soportes frente a los usuales en papel. Suponen una disminución en gastos de imprenta, almacenamiento y transporte. Un CD es algo que se puede mandar por correo sin ninguna complicación, con lo que favorece la distribución de la información y con las nuevas herramientas, copiar la información a otro CD resulta una opción cómoda, rápida y eficaz en la difusión de cualquier resultado de las características que nos ocupa. Con la ventaja añadida del ahorro directo en papel, producto cada vez más escaso. Aunque siempre se debe proporcionar, como es el caso la posibilidad de imprimir aquellos artículos que al usuario le parezcan de interés a través del propio CD.

A nadie escapa la importancia creciente que están cobrando las nuevas tecnologías de la información y la comunicación, y cómo afecta al campo educativo e investigador y de difusión en todos los órdenes[3]. Si queremos avanzar con nuestro tiempo, será necesario manejar y servirse de estas nuevas herramientas de trabajo y adaptar las técnicas comunicativas a este nuevo medio.

6. Bibliografía

- [1] EL TIGI, M. Y BRANCH R.M. "Designing for Interaction, Learner Control, and Feedback During Web-Based Learning". Educ. Technol. ,1997.
- [2] JASÓN ROBERTS, PHIL GROSS. "Director 7, La guía oficial de Director 7 Shockwave Internet Studio", Anaya multimedia, Madrid, 2000.
- [3] A. GIL, J TEIRA, "Aplicación de las "Nuevas Tecnologías" al estudio de los textos. Incidencia de la creación de herramientas informáticas en la metodología de la investigación textual." Actas del XV Coloquio de la AIB. "Las nuevas formas de la comunicación científica". Salamanca, España: Mayo 9-11 2000. CD-Rom.

Interacción con mundos VRML utilizando Java

C. Romero¹, S. Ventura², C. Castro³

Departamento de Informática y Análisis Numérico.
Universidad de Córdoba, Campus de Rabanales, 14071 Córdoba.
Tlf: 957.211020 Fax: 957.101051.
e-mail: ¹cromero@uco.es, ²sventura@uco.es, ³cdcastro@uco.es

Resumen

En este artículo se describen las distintas formas con las que podemos interactuar con un mundo virtual desarrollado con VRML (Virtual Reality Modelation Language) y más concretamente utilizando el lenguaje Java. Además se ven las posibilidades que nos ofrece esta interacción para construir entornos multiusuarios, interactivos y distribuidos.

Palabras clave: Entornos Virtuales, VRML, Java, Interacción.

1 Introducción

El lenguaje VRML (Virtual Reality Modelation Language) [1] es un estándar ISO (ISO/IEC 14772-1:1997) para la transferencia de datos 3D a través de Internet. Con él se pueden construir mundos virtuales y publicarlos a nivel mundial con independencia de la plataforma y del sistema operativo. Lo único que se necesita para poder visualizar un mundo VRML es un navegador específico (Java3D Browser, etc.) o instalar un plugin (Cosmo Player, Cortona, Blaxun Contact, etc.) en nuestro navegador WEB preferido. Su amplia utilización está abriendo un nuevo rango de posibilidades en la creación de contenidos 3D para Internet (museos, ciudades, catálogos, simuladores, etc.). El estándar actual VRML 2.0, también denominado VRML97 o Moving Worlds [2] tiene una serie de mejoras con respecto a la primera versión VRML 1.0, con son: El lenguaje de descripción de escenas tridimensionales ha sido ampliado significativamente, posibilitando efectos de fondo, sonidos tridimensionales, niebla, etc. Posibilidad de interacción con el usuario mediante la definición de una serie de sensores de posición, de contacto, de colisión, etc. La información registrada por estos sensores es enviada a los diferentes objetos que componen el mundo virtual y, en función de los valores recibidos, cada objeto virtual actuará en consecuencia. Posibilidad de especificar comportamientos para los objetos, ya sea usando el propio lenguaje VRML o mediante scripts en lenguajes externos (Java script, Java, Visual Basic, etc.), los cuales no están limitados por la especificación. De estas características nos vamos a centrar en la última, ya que nos va a permitir ampliar la interactividad, rendimiento y

comunicación de nuestros mundos. En concreto, elegimos el lenguaje Java por su capacidades computacionales y de red en la creación de entornos interactivos y distribuidos.

2 Interactividad básica con VRML

El lenguaje VRML 2.0 posee dos partes fundamentales [1]: el lenguaje de descripción de la escena y el mecanismo para crear comportamiento. El modelo de ejecución de VRML permite pasar propiedades entre varios objetos de la escena, de forma que se pueden crear comportamientos. Las animaciones básicas se van a realizar capturando eventos, encaminándolos y produciendo una interpolación en un objeto.

- **Eventos.** Cuando un nodo quiere pasar información a otro nodo, debe crear un evento. Un evento consta de un dato y un tiempo. El tiempo indica cuando fue originado y permite al Navegador mantener una secuencia correcta.
- **Encaminamiento.** VRML no dispone de un mecanismo de llamada a funciones para pasar información entre nodos. En su lugar se debe hacer una conexión explícita entre dos campos (fields) de los dos nodos. Existen cuatro tipos de accesos a campos: field (ningun acceso externo), eventIn (escritura), eventOut (lectura) y exposedField (lectura y escritura). Para conectar dos campos, se deben de nombrar los dos nodos a conectar con la sentencia DEF. Para encaminar los campos se utiliza la sentencia ROUTE de la siguiente forma: ROUTE fromNode.eventOutField TO toNode.eventInField.
- **Sensores.** Una vez que se tiene un conjunto de nodos conectados por una serie de rutas, se debe comenzar a generar la secuencia de eventos. Para ello se utilizan los sensores, que son un nodo que puede generar nuevos eventos. Hay que añadir un sensor a un objeto. Un sensor no es un nodo de grupo. Existen tres tipos de sensores: **Sensores de contacto** dentro de los que están: TouchSensors: captura la entrada del usuario dentro de un área. El punto de toque se da en coordenadas 3D. Anchors: asocia una URL con un objeto. Drag sensors: traslada la entrada del usuario en un camino predefinido. Collision: detecta el choque del usuario con los objetos del mundo. **Sensores de Visibilidad** que son: VisibilitySensor: genera un evento cuando es visible o invisible desde el punto de vista actual. ProximitySensor: detecta cuando el usuario está dentro del área de un objeto. **Sensores de Tiempo** donde sólo está TimeSensor: Detecta el tiempo y da a la escena un sentido del tiempo que pasa emitiendo eventos. El tiempo en VRML se basa en segundos.

3 El nodo Script de VRML

Los nodos Script van a permitir realizar interacciones y comportamientos más complejos. Para ello, proporcionan un interfaz con otros lenguajes existentes. Para definir un comportamiento, primero hay que definir el campo (field) al que va a acceder el código Script y entonces se escribe dicho código utilizando cualquier lenguaje que lo permita: Java, JavaScript o VRMLScript. Para que un nodo Script tenga entradas y/o salidas se deben de declarar un conjunto de campos eventos que indiquen el tipo de acceso, el tipo de campo, el nombre del campo y el valor.

4 Script JavaScript

JavaScript [1] es un lenguaje Script que se suele utilizar dentro de las páginas Web. Se pueden escribir comportamientos con JavaScript de dos formas: **Interno** escribiendo el código JavaScript dentro del fichero VRML y **Externo** escribiendo el código JavaScript en un fichero externo .js. Para cada EventIn, se debe de crear una función con el mismo nombre, que debe de recibir dos argumentos: los datos y el tiempo. Leer de los campos es tan simple como utilizar una variable con el mismo nombre. Escribir un eventOut es tan simple como escribir en una variable con el mismo nombre que en el fichero VRML.

5 Script Java

Los Script Java son Script que están escritos en lenguaje Java. La ventaja de utilizar Java es que además de poder ejecutar los comportamientos más rápidos, se puede tener total funcionalidad de red, multihilos e interfaz de usuario proporcionado por los distintos APIs del lenguaje. La principal diferencia entre escribir comportamientos en Java y en JavaScript es el manejo de eventIn. El modelo de eventos es una copia casi exacta del manejador de eventos de AWT. En VRML existen dos métodos para eventIn: processEvent (que procesa sólo un evento cada vez) y processEvents (que maneja múltiples eventos). Los Script Java se escriben en ficheros externos .java que extienden a la clase Script. Java proporciona tres paquetes básicos para Script: vrml: que contiene las clases básicas. vrml.node: que contiene la definición de los nodos y vrml.field: que contiene la definición de los campos. Los Scripts tienen dos categorías de campos: campos constantes o de sólo lectura para pasar información al Script en forma de eventos y la versión general para eventOuts y dentro del Script que permite hacer lo que se desee. El nombre de la clase no tiene por que ser el mismo que el nodo DEF del fichero VRML. Los métodos que se escriben dentro de la clase Script son: el método initialize y el método de manejo de eventos. El fichero java se compila y se coloca en el mismo directorio del fichero VRML o en otro distinto pero luego hay que hacer referencia a su URL.

6 External Authoring Interface (EAI)

EIA [3] es un API para escribir código Java que está en un Applet separado y que permite directamente comunicarse con el Navegador. Hay algunas cosas que no están permitidas hacerlas desde un Script, las más importantes es con respecto a AWT. El Interfaz Externo permite realizar todas estas cosas y muchas más, ya que al utilizar un Applet separado en la misma página, se puede poner el interfaz gráfico que se desee y reducir la interacción con el Navegador a lo mínimo. La única similitud entre EAI y Script es que ambos implementan el mismo interfaz del Navegador y pueden pasar eventos entre la escena. Los Scripts funcionan dentro del Navegador VRML y sólo se pueden comunicar con un código externo Java mediante API de red. EAI por el contrario es parte de un Applet y puede utilizar todas los APIs de Java y comunicarse con el Navegador u otra aplicación Java o Applet, aunque con las restricciones de seguridad de los Applets. Pero se puede solventar haciendo Applets Seguros o Firmando los Applets. Ambos entornos (Script y EAI) utilizan partes distintas del paquete vrml. Los Scripts utilizan los paquetes vrml, vrml.node y vrml.field. Y los Applets el paquete vrml.external. Y no se pueden mezclar. Recibir eventIns es también diferente. Los Scripts tiene un método definido para todos los eventos y EAI debe registrar una clase para cada evento. Enviar valores a otros nodos es muy parecido en ambos. EAI dispone de más métodos para acceder a la escena, aunque tiene menos conocimiento a priori del entorno (para ello utiliza métodos extra para comenzar). EIA esta diseñado para servir de interfaz de un control de un Navegador VRML desde fuera. Por lo tanto sólo tenemos que escribir un Applet y añadirle el código de control y comunicación con VRML que proporciona EAI.

- **El fichero HTML.** Debemos de escribir una página HTML donde se incluya el Applet y el fichero VMRL. El Applet se inserta igual que cualquier Applet con la etiqueta <APPLET > pero con la etiqueta MAYSCRIPT que indica que nos podemos comunicar con código JavaScript de la página. El mundo VRML se inserta en la página con la etiqueta <EMBED >.
- **El Applet.** El Applet tiene el código típico de un Applet con el nuevo código de comunicación con la escena. Lo primero que tenemos que hacer es obtener una instancia del Navegador utilizando la clase Browser. Entonces ya podemos comenzar a obtener información de la escena VRML utilizando el resto de métodos de la clase Browser.
- **Obtener referencia a nodos.** Antes de poder leer o modificar propiedades de la escena, necesitamos tener una referencia al nodo del que queremos leer. Para ello utilizamos el método getNode que devuelve un puntero a un nodo declarado con DEF en el fichero VRML. Por ejemplo, para obtener una referencia a un nodo DEF denominado colorcubo.
- **Leer valores de la escena.** Se puede consultar a un nodo por un valor de un campo (field) en particular preguntado por la referencia del eventOut en particular utilizando los métodos getEventIn(String) y getEventOut(String). Sólo se pueden leer valores de campos eventOut y exposedField de nodos de la escena. Además debemos de realizar

un casting del valor que devuelve al tipo apropiado antes de leer el valor. Por ejemplo. Si tenemos un nodo Transform de nombre miTexto y queremos leer el campo translation.

- Una vez que tenemos la referencia a un campo (field) podemos leer valores de él utilizando los métodos getValue apropiados definidos para cada tipo de campo individual.
- **Recibir eventos.** Si necesitamos leer constantemente valores en tiempo real, necesitamos construirnos el equivalente a un ROUTE, ya que no hay forma de especificar en un ROUTE una fuente externa. Para crear un clase que maneje eventos necesitamos hacer 4 cosas: 1.Crear un clase que implemente el interfaz VrmlEventListener. 2.Recuperar el nodo y entonces el eventOut que se desea y asociarle un escuchador utilizando addVrmlEventListener. 3.Escribir el método eventOutChanged donde se procesa el evento y 4.Obtener el valor del evento con el método getData().
- **Escribir valores de vuelta en la escena.** La modificación del contenido de la escena desde un Applet se realiza de la misma forma que desde un Script. Sólo hay que generar un evento con la información y enviarlo al nodo destino. Para ello los pasos son: 1.Obtener una referencia al nodo al que se desea mandar el evento con el método getNode o directamente si el nodo se ha creado con createVrmlFromX. 2.Recuperar el campo eventIn deseado utilizando el método getEventIn. Se debe realizar un casting del valor devuelto al tipo apropiado y 3.Enviar los valores al campo utilizando el método SetValue para el tipo.

7 Conclusiones

En este artículo se ha repasado las distintas formas de interactuar con un mundo VRML, especialmente utilizando el lenguaje Java y más concretamente EAI, ya que es la opción que nos permite muchas más posibilidades. De forma que utilizando WWW, Java y VRML disponemos de una plataforma para desarrollar espacios virtuales interactivos, multiusuarios y distribuidos. Para ello sólo tenemos que escribir un Applet que implemente EAI y se encargue de controlar nuestro mundo. Al ser un Applet se puede comunicar con otras aplicaciones Java locales (firmando el applet) o con un servidor Java (mediante socket, rmi o corba). En la actualidad existen distintos grupos de trabajo [4] que estudian distintas formas de ampliar el estándar VRML para permitir mayor interactividad, mejorar su rendimiento, permitir entornos multiusuario, distribuidos, etc. Algunos de estos grupos son: dis-java-vrml, enterprise, vrml-eai, vrml-java3d, living-worlds, nlp-anim, kbinput, etc.

Referencias

- [1] Andrea L. Ames, David R. Nadeau, John L. Moreland; Wiley. "VRML 2.0 Sourcebook". October 1996.
- [2] Bernie Roehl, Justin Couch, Cindy Reed-Ballreich, Tim Rohaly and Geoff Brown. "Late Night VRML 2.0 with Java". Ziff-Davis Press. 1997.
- [3] Rodger Lea, Kouichi Matsuda, Ken Miyashita. "Java for 3D and VRML worlds". Diciembre 1996.
- [4] The VRML Consortium, www.vrml.org.

Virtual-PRISMAKER: Juegos de ordenador, educación e interfaces.¹⁰

Francisco Montero[‡], Víctor López[‡], María Lozano[†], Antonio Fernández[†], Pascual González^{†‡}

[†]Departamento de Informática

[‡]Instituto de Desarrollo Regional.(Secc. Tecnología de la Información)

Universidad de Castilla-La Mancha, 02071-ALBACETE

Tlf: 967599200 Fax: 967599224

[fmontero,vlopez,mlozano,caballer,pgonzalez]@info-ab.uclm.es

Resumen

Las grandes expectativas que ofrecen las nuevas tecnologías de la información están motivando numerosos proyectos dentro del ámbito educativo que pretenden demostrar las capacidades de nuevos materiales, herramientas o métodos de enseñanza. Dentro de este ámbito se encuadra este proyecto, cuyo objetivo principal es demostrar las potencialidades de los juegos para mejorar el proceso de aprendizaje. Podemos resaltar dos grandes objetivos: ver el potencial educativo de un juego comercial PRISMAKER [1] y comparar dicho potencial con el que se alcanza con un juego de ordenador que lo simula de manera realista y al que hemos denominado “Virtual-PRISMAKER” (VP). En este trabajo vamos a realizar una breve revisión de las investigaciones más interesantes realizadas hasta el momento en el ámbito del desarrollo de software para niños. Tras ello vamos a presentar las características y objetivos básicos del proyecto VP, así como la descripción del prototipo inicialmente construido.

Palabras clave: Interfaces de Usuario 3D, Interfaces para Niños, Diseño de Juegos Educativos, Diseño Participativo.

1 Introducción

Los avances tecnológicos están provocando cambios continuos en el modo de vida no solo de los adultos, sino también en el de los niños. El modo en que nuestros hijos aprenden, juegan y se comunican está cambiando, incluso podríamos decir que el cambio en los niños

¹⁰ Este trabajo está financiado por el proyecto CICYT-FEDER 1FD97-1017.

es más rápido y acusado que en los adultos. La carencia de patrones predefinidos, el interés por experimentar con nuevos artefactos y explorar nuevos entornos, hace que los niños estén muy abiertos a nuevas ideas, ideas que en nuestro entorno se esconden tras las nuevas tecnologías.

Las virtudes potenciales de estos nuevos entornos hacen que sea especialmente recomendado su uso en aquellas facetas de la enseñanza que tradicionalmente se han considerado pesadas y poco atractivas. Para alcanzar los objetivos establecidos, el proceso educativo necesita aumentar la motivación, es decir, que el alumno se interese por lo que se le enseña.

Las nuevas tecnologías de la información ofrecen nuevas herramientas que pueden modificar completamente el modo en que se concibe actualmente la enseñanza. Pero el diseño de estas herramientas tiene una especial complejidad dado el tipo de usuarios al que va dirigido, usuarios con menor capacidad expresiva y que están adquiriendo algunas de las habilidades de los adultos. En este tipo de entornos, junto a la dificultad de extraer los requerimientos funcionales, tal vez el mayor problema estribe en el diseño de la interfaz de usuario. Los niños son mucho más exigentes que los adultos a la hora de aceptar una determinada interfaz. Dentro de los criterios de lo que se suele denominar "usabilidad", el atractivo de la interfaz y su facilidad de manejo son tal vez los parámetros más relevantes. Las peculiaridades existentes en el diseño de interfaces de usuario para niños, ha provocado que existan grupos dentro de organizaciones como SIGCHI (CHIKids) cuyo interés se centra en el estudio de la problemática de este tipo aplicaciones.

2 Software para niños: Juegos de ordenador, educación e interfaces

La creación de software para niños va generalmente unida al binomio: juegos de ordenador (entretenimiento) e interfaces visuales (interfaces muy atractivos y con grandes aportaciones gráficas). Estas connotaciones lúdicas y el atractivo de sus interfaces hacen de este tipo de herramientas especialmente interesantes para el ámbito educativo. La motivación juega un papel muy relevante en el proceso de aprendizaje. Aunque existe numerosos estudios sobre la importancia de la motivación, hay un "escaso conocimiento científico...sobre los factores que afectan a la motivación, disfrute y satisfacción" [2]. En los últimos tiempos han aparecido trabajos que sugieren que los juegos de ordenador aumentan la motivación de los niños en el aprendizaje en las escuelas [2], [3]. Algunos llegan más lejos, e indican [4] que el juego es el equivalente del trabajo de los adultos. A través del juego, los niños crecen, ejercitan habilidades físicas, descubren cómo es el mundo, desarrollan las habilidades cognitivas y aprenden a interactuar con los demás.

Pero la utilización de juegos de ordenador aporta nuevas características que favorecen aún más el proceso de aprendizaje. Como indican Sedighian et al. [5] los juegos de ordenador aportan: comprensión con significado, establecimiento de metas, sentido de éxito, asociación a través del placer, atracción, estímulos sensoriales, etc. Como vemos, varias de estas aportaciones se consiguen gracias al atractivo que los juegos de ordenador, con sus mundos 3D y sus interfaces visuales, provocan en los niños.

Otro factor importante en el diseño de software para niños es la metodología de desarrollo utilizada y la participación del usuario en dicho proceso. En este ámbito podemos encontrar desde las visiones más pesimistas en cuanto a la participación de los usuarios en el diseño, hasta llegar a '*user-centred*' o '*participatory design*'.

Dentro del ámbito del software para niños, recientemente, han aparecido nuevos enfoques: '*informant design*' [6] o '*cooperative inquiry*' [7]. *Informant design* entiende el proceso de diseño como un trabajo interdisciplinar en el que tienen cabida niños, profesores, especialistas en educación y diseñadores de gráficos y software. Cada uno de estos colectivos participan en el proyecto de manera diferente, facilitando un tipo de información. Así el niño podrá informar de qué cosas son las que le mantiene motivado, el profesor podrá contribuir informando sobre la dificultad que han observado en los niños para aprender con los materiales tradicionales y cómo puede incorporarse las tecnologías de la información en esos casos. De forma similar los especialistas en educación podrán facilitar qué clase de aprendizaje puede ser utilizado con mayor efectividad en el diseño de entornos de aprendizaje interactivo. A partir de dicha información los diseñadores de software serán los encargados de buscar la mejor forma para recrear dichos entornos interactivos. Por otra parte, el *cooperative inquiry* busca una mayor integración de los niños en proceso de diseño, pretendiendo formar un equipo de diseño intergeneracional. En este caso se parte de la premisa de que los niños pueden y deben asumir el papel que otros usuarios juegan en el diseño de sistemas para adultos, ya que ellos pueden contribuir de "igual a igual" en el proceso de diseño, gracias a su experiencia como usuarios de tecnología en la actualidad y en el futuro.

3 Virtual-PRISMAKER.

El proyecto en el que se gesta el sistema Virtual-PRISMAKER surge de la colaboración de la industria-universidad-escuela. En él hay implicados desde empresarios que aportan la versión física del juego (PRISMAKER [1], ver figura 1), pasando por la universidad, donde colaboran psicólogos, pedagogos, informáticos, etc, hasta por supuesto llegar a la escuela que será nuestro banco de pruebas y que participará en el propio proceso de diseño de la interfaz, dentro de un proceso típico de *cooperative inquiry*.



Figura 1: PRISMAKER system. Descripción de las características del juego físico a simular mediante el Virtual-PRISMAKER.

Como vemos en la figura 1, PRISMAKER es un juego de construcción que dispone de un número muy limitado de tipos de piezas. A su vez la posibilidad de asociarle a una pieza un logotipo permite dotarla de significado y, por tanto, amplía las posibilidades educativas del juego, haciéndolo aplicable a la enseñanza de matemáticas, lecto-escritura, secuenciación de historias, etc.

3.1 Características de implementación de Virtual-PRISMAKER

Como ya hemos indicado, el objetivo básico de VP es plantear un juego de ordenador que simule lo más perfectamente posible el juego físico y a su vez estudiar las nuevas posibilidades que este juego de ordenador ofrece dentro del proceso educativo. La idea que hemos planteado originalmente es proporcionar al niño un entorno de trabajo lo más parecido posible al que se encuentra en la vida real. Como puede verse en la figura 2.a, se ha planteado una habitación (la habitación de un castillo) donde se encuentran dos elementos básicos: una estantería y una mesa. La estantería se utiliza para guardar las construcciones que el niño va realizando. Con ello se pretende eliminar la idea de archivo o fichero como objeto de almacenaje y se sustituye por un elemento usual en el entorno de los niños, como es una estantería donde se guardan los juguetes. Por otra parte, la mesa es el sitio de juego, en ella hay tres zonas claramente diferenciadas. A la izquierda se encuentran las fichas que pueden utilizarse para el juego. En la parte central se ha dejado un gran espacio para jugar. Por último, a la derecha se encuentra un libro, el cual contiene información sobre algunas de las actividades preestablecidas que el niño puede realizar. Como vemos si se selecciona el libro, éste se abre y muestran las actividades, las cuales tienen siempre asociado un dibujo y

un texto. La asociación del dibujo es fundamental en nuestro caso pues algunos de los juegos pueden ser realizados por niños que todavía no saben leer y, por tanto, ellos seleccionan en base al dibujo y el texto se lee automáticamente.



Figura 2: a) Escena de la sala de juego. b) Ejemplo del libro de actividades.

Para la construcción y desarrollo del sistema se está utilizando Java y el API Java 3D [8]. Este entorno de desarrollo nos permite crear mundos 3D e interactuar con ellos. El API contiene un conjunto de constructores de alto nivel para crear y manipular geometrías en 3D, crear escenas, animaciones y aplicaciones con objetos interactivos.

Aunque Java 3D no fue diseñado explícitamente para cubrir las expectativas de los desarrolladores de juegos, éste suministra las suficientes prestaciones para soportar muchas de estas aplicaciones. Se puede pensar que Java 3D puede tener penalización en cuanto a eficiencia, sin embargo otros factores como la portabilidad o el costo de desarrollo deben tenerse en cuenta frente a las prestaciones absolutas. A su vez este entorno ofrece, respecto a otros como OpenGL o DirectX, un conjunto de objetos ordenados en base a una serie de jerarquías basadas en la herencia. Este API ofrece unas 100 clases contenidas dentro del paquete javax.media.j3d, con las pueden crearse y gestionarse primitivas 3D, sonidos, luces, sombras y comportamientos.

Finalmente en esta primera fase, junto a la definición del primer prototipo de interfaz, se está abordando la definición del formato de representación de los modelos construidos por el niño, a la vez que se están resolviendo los problemas derivados del manejo de los objetos en este entorno 3D (posibilidades de interacción entre los diferentes bloques que conforman el juego físico, comportamiento y detección de colisiones, etc.)

4 Conclusiones

En este trabajo hemos presentado un primer prototipo del sistema Virtual-PRISMAKER, el cual pretende ofrecer un juego de ordenador que simule y aporte las mismas posibilidades que el juego educativo real. Para ello, se ha definido una interfaz 3D atractiva, que permita a los niños realizar construcciones virtuales y realizar otro tipo de juegos dentro del ámbito educativo. A su vez, en el diseño de la interfaz se ha pretendido prescindir de conceptos informáticos tales como tareas, fichero (abrir, guardar), etc. y han sido sustituidos por símbolos y objetos cotidianos como libro, estantería, etc., facilitando así la asimilación de dichos conceptos.

Referencias

- [1] <http://www.prismaker.com/>
- [2] Norman, D.A. Things that make us smart: defining human attributes in the age of the machine. NY: Addison-Wesley, 1993.
- [3] Klawe, M. "The educational potential of electronic games and the E-GEMS project". En T. Ottam, I. Tomek (Eds). Proc of ED-MEDIA 94: World Conference on Educational Multimedia and Hypermedia. Charlottesville, VA:AACE. 1994.
- [4] Garvey, C. Play. Cambridge, M.A. Harvard University Press.
- [5] Sedighian, K., Sedighian, A. S. "Can Educational Computer Games Help Educators Learn About the Psychology of Learning Mathematics in Children?". 18th Annual Meeting of the International Group for the Psychology of Mathematics Education, Florida, USA.1996.
- [6] Scaife, M. et al. "Designing For or Designing With? Informant Design for Interactive Learning Environments". CHI97 Electronic Publications. 1997
- [7] Druin,A. "Cooperative Inquiry Developing New Technologies for Children with Children". Proc. of CHI'99, ACM Press. 1999.
- [8] Java 3D API Specification. Versión 1.2 beta 1 Diciembre 1999.

Diseño y desarrollo de una aplicación informática para la gestión de laboratorios

M. Francisco, P. Vega, F. J. Blanco

Departamento de Informática y Automática.
Facultad de Ciencias. Universidad de Salamanca
Plza. de la Merced s/n. 37008 Salamanca
e-mail: mfs@gugu.usal.es

Resumen

En este trabajo se exponen las bases para el desarrollo de un sistema integrado para la gestión de laboratorios, cuyas funciones principales sean almacenar, integrar y gestionar de manera adecuada toda la información generada en un laboratorio, dotándola de vías de salida hacia los clientes (empresas, departamentos, etc). Además, se presenta la aplicación informática BDGL desarrollada en particular para la gestión de laboratorios de análisis de aguas, y más en concreto para los laboratorios del Centro de Investigación y Desarrollo Tecnológico del Agua (CIDTA) de la Universidad de Salamanca.

Palabras clave: Sistema Informático de Gestión de Laboratorios, Bases de datos relacionales, Interfaz gráfica de usuario, normativas e informes.

1 Introducción

Tradicionalmente, toda la información generada en un laboratorio referente a análisis de muestras se gestiona de forma manual sin ayuda de ninguna aplicación informática específica. Los datos se guardan en simples documentos y los informes solicitados por empresas u otras instituciones se redactan manualmente para cada solicitud. En la actualidad, la demanda de trabajo ha aumentado considerablemente debido al incremento en el número de solicitudes y por lo tanto de muestras a analizar. Además, se necesita tener un rápido acceso a toda la información existente. Debido a todo ello, parece conveniente utilizar alguna aplicación informática capaz de realizar estas tareas de forma sencilla y eficiente.

2 Requisitos generales de la aplicación

Un laboratorio analítico puede definirse como una unidad de producción de información de distintos tipos. Los Sistemas Informáticos de Gestión de Laboratorios, como la aplicación informática de la que es objeto este artículo, son la solución a las necesidades de generación, gestión y manipulación de dicha información.

De forma general, el sistema a desarrollar debe cumplir requisitos comunes a cualquier aplicación informática dedicada a la gestión y almacenamiento de datos: cómoda interfaz gráfica, elaboración de informes, sistema de búsquedas sencillo que me permita un rápido acceso a la información y mantenimiento de una base de datos relacional. Pero además de lo anterior, la normativa europea EN 45001 establece unos "principios de calidad" que afectan al funcionamiento del laboratorio y por extensión a los sistemas informáticos de gestión de laboratorios. Principalmente, esto se refleja en las características de los informes producidos (formato e información contenida) y en algunos aspectos de la estructura de la aplicación.



Figura 1. Panel de control principal de BDGL

3 Diseño de las estructuras de datos

Una vez conocido el problema a resolver, se deben diseñar las estructuras de datos que manejará la aplicación. A continuación se define toda la información que puede contener un sistema de gestión de laboratorios, y que BDGL en concreto utiliza de forma eficiente.

3.1. Información básica

Según se ha determinado a partir de entrevistas realizadas con el personal de un laboratorio, nuestro sistema debe ser capaz de almacenar y trabajar con la siguiente información:

1. **Muestras** que llegan al laboratorio y que deben ser analizadas.
2. **Parámetros** que se pueden determinar en los análisis de las muestras o ensayos.
3. **Métodos de Ensayo**, que son los distintos procedimientos técnicos especificados para la realización de un ensayo.
4. **Normativas** vigentes, que registrarán los niveles permitidos, máximos, mínimos, etc. para los diferentes parámetros.
5. **Solicitudes** de análisis que llegan al laboratorio incluyendo lotes de muestras con sus parámetros a analizar.
6. **Resultados** de los análisis realizados a las muestras.
7. **Informes** y documentos que presenten de forma clara la información existente.

Los datos anteriores se refieren a un laboratorio en general. Como es natural, laboratorios de diferentes ámbitos de trabajo presentan características específicas diferentes, de tal manera que debemos tenerlas en cuenta si queremos desarrollar una aplicación óptima para uno de ellos. En el caso particular de la aplicación BDGL para la gestión de un laboratorio de análisis de aguas, las muestras serán muestras de agua, los parámetros serán el pH o la alcalinidad y las normativas serán la RTS para abastecimiento y control de las aguas potables o la normativa de aguas residuales.

3.2. Información adicional

Además de la información anterior, que se puede considerar imprescindible, BDGL trabaja con tipos estándar de solicitudes y con parejas parámetros-métodos de ensayo permitidas. Debido a que muchas veces llegan al laboratorio solicitudes similares, existen tipos de solicitudes en los cuales aparecen ya prefijadas las siguientes características: normativa implicada y parámetros a analizar junto con sus métodos de ensayo. De esta forma el usuario sólo tiene que seleccionar el tipo de solicitud adecuado evitando tener que introducir repetidamente los parámetros requeridos con sus métodos de ensayo así como la normativa aplicada para solicitudes similares.

Por otro lado, BDGL asocia los parámetros con métodos de ensayo concretos. En el caso de análisis de aguas, no todos los métodos de ensayo pueden ser utilizados para analizar todos los parámetros. Debido a ello, la aplicación guarda información sobre qué métodos son aplicables y cuáles no lo son para cada parámetro concreto

3.3 Modelo relacional estándar

Todos los objetos o entidades que maneja la aplicación y que de forma intuitiva se han presentado en los párrafos anteriores, se deben estructurar en las distintas tablas relacionales. Cada entidad, junto con sus atributos asociados, tiene una tabla en la que se almacenan los datos concretos. Algunos ejemplos de atributos son el color, olor, volumen o fecha de muestreo para las muestras; y para las solicitudes su identificador, normativa aplicada, fecha de recepción o datos del cliente (Nombre, dirección, empresa, etc.)

La información que aparece por separado en las distintas tablas se combina entre sí mediante distintas relaciones. De esta forma, BDGL organiza de forma eficiente los datos que posee buscando la información recursivamente cuando hacemos búsquedas o le pedimos que imprima informes. Tablas y relaciones constituyen el modelo relacional estándar, que en el caso de BDGL se ha desarrollado con Microsoft Access como programa gestor de bases de datos.

4 Estructura general del sistema

A continuación se describe la estructura general de un sistema de gestión de laboratorios. Inicialmente se pueden distinguir cuatro partes más o menos diferenciadas, tal y como se indica en la figura 2.

La primera de ellas se dedica a lo que denominamos “Gestión de los recursos”, en la cual introducimos la información de base con la que se va a trabajar posteriormente. Se introducen todos los parámetros analizables posibles, métodos de ensayo existentes, normativas vigentes, etc., todo ello con sus características asociadas.

La segunda parte es la que me facilita la introducción de datos referente a las solicitudes concretas que llegan al laboratorio solicitando análisis de muestras. Aquí se introducen muestras, parámetros a analizar y resultados de los ensayos, todo ello también con sus características asociadas.

La tercera parte consiste en un sencillo sistema de búsquedas que me permite acceder de forma rápida a cierta información del sistema como son las muestras o las solicitudes conociendo alguna de sus características. Por último, la parte correspondiente a los informes se ocupa de gestionar los distintos tipos de informes, permitiendo al usuario elegir el tipo deseado dentro de los disponibles en la aplicación.

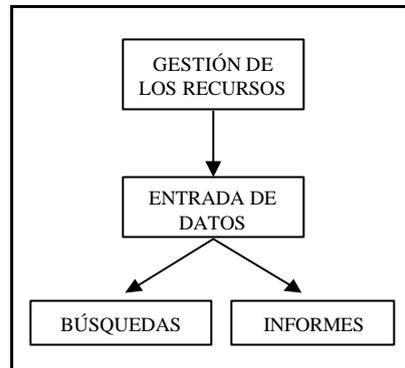


Figura 2. Estructura general de la aplicación

5 Características de BDGL

Como ejemplo concreto de desarrollo de un sistema integrado de gestión de laboratorios, presentamos la aplicación BDGL (Base de Datos para la Gestión de Laboratorios) diseñada específicamente para utilizarse en laboratorios de análisis de aguas. Dicha aplicación posee un carácter muy amplio y un alto grado de adaptabilidad, pudiéndose reorientar su utilización hacia laboratorios de otros campos como Centrales Nucleares, Papeleras, Industria Farmacéutica, etc.

La manera de navegar a través de BDGL es muy sencilla mediante los numerosos formularios que permiten al usuario seleccionar la opción deseada. Además, se incorpora un sistema de ayuda personalizado en cada uno de los botones indicando la acción que realizan si se pulsan.

5.1 Gestión de los recursos

Al iniciar BDGL, lo primero que aparece es el formulario de la figura 1, en el cual selecciono una de las opciones disponibles. Para empezar a utilizar BDGL, lo primero que debo hacer es seleccionar la opción "Gestión de los recursos" para definir los datos generales (parámetros, métodos de ensayo, normativas, etc.) con los que posteriormente se va a trabajar. De esta manera toda esa información estará disponible a la hora de gestionar las solicitudes concretas que lleguen al laboratorio y realizar los informes correspondientes. Por ejemplo, en el caso de los parámetros, debemos llenar la lista de la figura 3 con todos los parámetros analizables junto con sus unidades físicas o químicas. De forma similar tenemos listas para los métodos de ensayo disponibles, parejas parámetro-método de ensayo válidas, normativas disponibles y tipos de solicitudes definidos.

En el caso de las normativas, es importante resaltar que se dispone de una opción en la cual puedo definir el número y el nombre de los niveles que controla una normativa concreta. BDGL me permite trabajar con normativas que controlen hasta cinco niveles. Por ejemplo, la normativa de aguas potables me fija una “CMA” (Concentración máxima admisible) y un “Nivel guía” (ver figura 4). En el formulario de la figura 5 aparecen los valores numéricos correspondientes para cada nivel y para cada uno de los parámetros analizados.

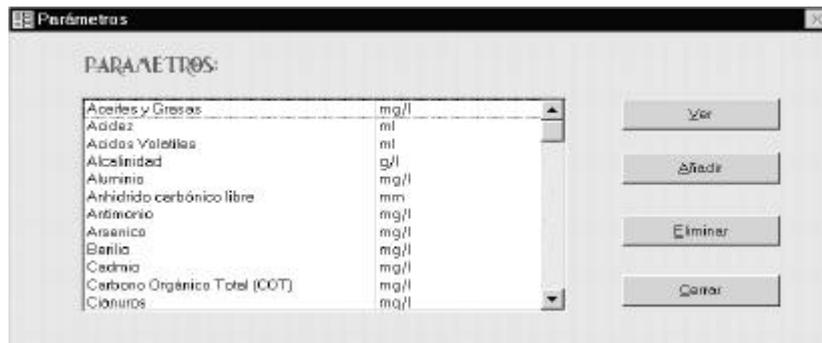


Figura 3. Parámetros analizables

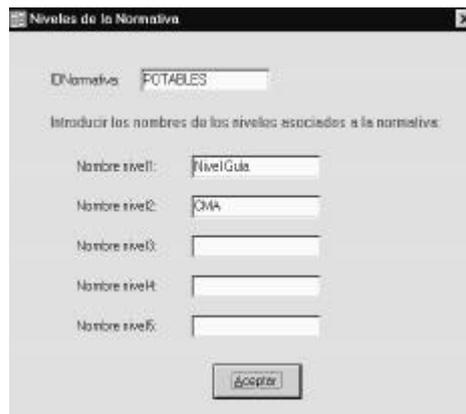


Figura 4: Nombres de los niveles asociados a una normativa

5.2 Entrada de datos

La segunda opción que aparece en el panel de control principal se denomina “Entrada de datos”. Desde un formulario como el de la figura 6 se introducen los datos correspondientes a las solicitudes de análisis que llegan al laboratorio procedentes de empresas, investigadores, etc. Parte de la potencia de BDGL radica en las relaciones existentes entre

toda la información, de forma que cuando introduzco los parámetros a analizar con sus métodos de ensayo, la normativa aplicable, o el tipo de solicitud, estoy basándome en los datos que BDGL ya posee. Por otro lado, si se selecciona un tipo de solicitud concreto, los parámetros a analizar junto con los métodos de ensayo a aplicar y la normativa que regula la solicitud quedan ya fijados, evitando al usuario el trabajo de introducirlos cada vez que llegue al laboratorio una solicitud de ese tipo.

Cada solicitud viene con unas muestras para analizar, cuyos datos también se introducen en la aplicación, y por último se introducen los resultados numéricos de los análisis una vez que éstos se hayan llevado a cabo.

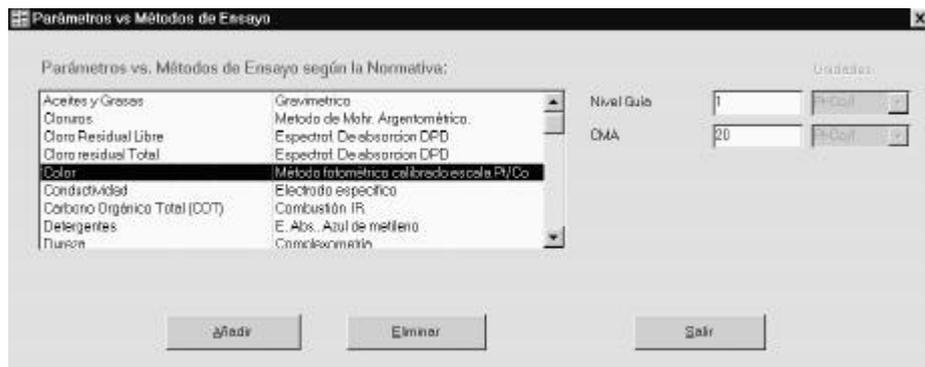


Figura 5. Parámetros y niveles permitidos según una normativa

5.3. Búsquedas

La siguiente opción a la que se accede desde el panel de control principal es la de "Búsquedas". BDGL me permite buscar muestras o solicitudes concretas de forma rápida y sencilla, fijando uno o varios criterios de búsqueda.

Figura 6. Formulario para la introducción de los datos de una nueva solicitud

5.4 Informes

Por último, BDGL me produce distintos informes de resultados imprescindibles para el laboratorio y los clientes. Existe la posibilidad de imprimir informes para una muestra concreta o informes globales para todas las muestras de una solicitud. Una de las características de todos los informes que genera BDGL es el hecho de que junto a cada parámetro que se analiza aparece el nivel permitido por la normativa que regula esa solicitud y que está ya indicada en sus características generales. De esta forma el analista o el cliente puede comparar ambos valores de forma cómoda. Además, cabe destacar que todos los informes se ajustan a lo dictado por la normativa EN 45001 en materia de laboratorios de ensayo.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo económico del Centro de Investigación y Desarrollo Tecnológico del Agua de la Universidad de Salamanca en el marco del acuerdo de colaboración con el Departamento de Informática y Automática de la Universidad de Salamanca.

Bibliografía

- [1] Manual de usuario de BDGL. Juan Carlos García Prieto (1999)
- [2] Programación Actual, números 1,2,3,4. Editorial Prensa Técnica, 1997
- [3] Criterios generales de Acreditación. Competencia Técnica de los laboratorios de ensayo. ENAC, 1997 (Entidad nacional de acreditación)

- [4] Guía para la acreditación de laboratorios que realizan ensayos químicos (1994)
- [5] Ingeniería del software. Un enfoque práctico. ROGER S. PRESSMAN Mc. Graw Hill 1998
- [6] Ingeniería del software. IRENE LUQUE, MIGUEL ÁNGEL GÓMEZ-NIETO. Servicio de publicaciones Universidad de Córdoba, 1999.
- [7] Guía práctica para usuarios de Access 2.0. Ediciones Anaya Multimedia, 1995.
- [8] Guía de Visual Basic 5.0. Ediciones Anaya Multimedia, 1997.
- [9] Informe sobre Labix: Solución de CPI en el mundo LIMS. Central de Procesos Informáticos, S.A.