



**Inteligencia Artificial. Revista Iberoamericana
de Inteligencia Artificial**

ISSN: 1137-3601

revista@aepia.org

Asociación Española para la Inteligencia
Artificial
España

Abascal, Julio; Moriyón, Roberto

Tendencias en Interacción Persona Computador

Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 6, núm. 16, verano, 2002,
pp. 9-24

Asociación Española para la Inteligencia Artificial

Valencia, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=92561602>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Tendencias en Interacción Persona Computador

Julio Abascal^{*} y Roberto Moriyón^{}**

^{*} Facultad de Informática. Universidad del País Vasco-Euskal Herriko Unibertsitatea
Manuel Lardizabal 1. Donostia-San Sebastián, 20018
julio@si.ehu.es

^{**} Escuela T. S. de Informática. Universidad Autónoma de Madrid
28049 Madrid
Roberto.Moriyon@uam.es

Resumen

La Interacción Persona-Computador es una disciplina relativamente reciente dentro de la Informática, que ha ido tomando cuerpo debido, fundamentalmente, a la necesidad de facilitar el uso de los computadores a una población más amplia -y de este modo alcanzar un mayor mercado- y, por otro lado, a la necesidad de técnicas de interacción sofisticadas para determinados tipos de aplicaciones. Para ello se ha nutrido de métodos y técnicas procedentes de otras materias hasta ser capaz de crear sus propias teorías. En este artículo se pretende hacer un repaso de la situación actual de la Interacción Persona-Computador, haciendo hincapié en su relación con la Inteligencia Artificial.

Palabras clave: Interacción Persona-Computador, herramientas de diseño de interfaces, usabilidad, interfaces inteligentes

1. Introducción

La Interacción Persona-Computador, IPC, es una disciplina relativamente joven dentro del ámbito de la Informática que ha experimentado un sorprendente desarrollo debido a diversos factores que trataremos de identificar. Históricamente, los primeros avances significativos en la capacidad de interacción entre usuario y computador se produjeron al nivel de los dispositivos de entrada de datos; posteriormente se produjeron avances importantes mediante la construcción de pequeñas interfaces gráficas que permitían una interacción más simple en aplicaciones concretas. Uno de los primeros tipos de interfaces de

usuario propiamente dichas consistió en programas de simulación que representaban en la pantalla del computador el comportamiento del instrumento que simulaban. Ya poco después aparecieron los primeros sistemas que pretendían simplificar el diálogo persona-máquina en aplicaciones genéricas.

La enorme difusión de los computadores personales y la necesidad de ampliar el mercado atrayendo a los posibles usuarios que rechazaban los computadores provistos de monótonas interfaces de comandos dieron un gran impulso a la búsqueda de interfaces gráficas de usuario (GUI) cada vez más sofisticadas y que permitieran una interacción más *natural*. Para ello se trató de utilizar metáforas procedentes de un contexto familiar al usuario que evitaran la necesidad

de aprender y utilizar conceptos puramente informáticos, que muchas veces venían forzados por la estructura interna de las aplicaciones.

En poco tiempo la IPC ha desarrollado un cuerpo de doctrina propio, relativamente diferenciado de otras materias de las que es deudora. Los conceptos utilizados en ella proceden -además de la Informática- de la Psicología, la Sociología, el Diseño Gráfico, etc. Particularmente, la Inteligencia Artificial ha sido y es un campo con el que el de las interfaces de usuario mantiene una continua interacción, como se verá más adelante.

Hoy en día la IPC está encontrando su lugar dentro de la Informática, tanto a nivel académico como industrial. La industria es consciente de que una interfaz más adecuada puede suponer la captación del mercado frente a competidores de similares características funcionales; la evolución de las simples interfaces de los teléfonos móviles es un buen ejemplo de ello. De hecho, junto a la Ingeniería del Software, la IPC constituye una de las ramas de la computación que cuenta con una mayor implicación por parte de la industria del software. En algunos de sus aspectos ligados a la producción de grandes programas interactivos, se pueden encontrar nexos claros entre la IPO y actividades que han tenido un gran auge desde el siglo pasado, como el diseño gráfico y el cine. Paralelamente a esto, la IPC mantiene un ritmo destacado de actividad a través de grupos organizados en torno a ella como *SigCHI*¹, grupo de interés en IPC de la ACM. La producción investigadora en este área se difunde a través de congresos especializados, entre los que destacan CHI, INTERACT² y UIST³, y de revistas como *Transactions on Computer Human Interaction*⁴ (TOCHI). También conviene destacar la existencia de congresos dedicados a aspectos de la IPO ligados a otras disciplinas, como la Inteligencia Artificial (*Intelligent User Interfaces*⁵, IUI).

En el ámbito docente, las propuestas de *curricula* más prestigiosas incluyen a la IPO en los estudios de Informática y muchos de los nuevos planes de estudios de la Ingeniería Informática incorporan, en mayor o menor medida, asignaturas de este área. En estas asignaturas se suele hacer énfasis en las herramientas y técnicas de programación de interfaces, en detrimento de los aspectos metodológicos y los relativos al usuario: técnicas y estilos de interacción, procedimientos de prueba y

evaluación, etc. La falta de tradición en la enseñanza de la IPO puede crear dificultades a la hora de decidir el contenido y la orientación de este tipo de asignaturas. Sin embargo existen materiales que pueden ayudar en este sentido, tales como propuestas para la creación de un plan de estudios completo en IPO, [Hewett96], una compilación actualizada de programas, centros que los imparten y asignaturas correspondientes a este área⁶ y publicaciones periódicas que incluyen trabajos sobre educación en IPO⁷.

En este artículo pretendemos describir algunos de los aspectos más relevantes de la IPC y su situación actual, haciendo hincapié en su relación con la Inteligencia Artificial. El artículo está organizado de la forma siguiente: la próxima sección se dedica a los principales paradigmas y técnicas de interacción; la sección siguiente se dedica a cuestiones relativas a la usabilidad de las aplicaciones interactivas; a continuación se introducen las principales arquitecturas software que se utilizan en las interfaces de usuario y se dedica una sección a las herramientas para la construcción de las mismas; la sección siguiente está dedicada a las interfaces inteligentes. Por último se hacen algunas consideraciones acerca de lo que serán las interfaces del mañana.

2. Paradigmas y técnicas de interacción

2.1 Estilos de interacción

La visión inicial de los computadores como instrumentos capaces de hacer operaciones arbitrarias sobre determinados datos y proporcionar un resultado -a la que correspondía en gran medida el sistema de entrada/salida de datos mediante tarjetas perforadas e impresora- se ha ido modificando a lo largo de los años con distintos estilos de interacción con el usuario cada vez más ricos y (dentro de las limitaciones que impone la tecnología disponible) más próximos a la forma en que interaccionamos los humanos. Sucesivamente, la interfaz basada en una línea para las instrucciones, la basada en menús y formularios y la basada en manipulación directa representan avances cada vez más sofisticados en esta dirección, que permiten que el flujo de información en ambos sentidos entre el usuario y el computador sea mayor y se realice con menos esfuerzo.

¹ <http://www.acm.org/sigchi/>

² http://www.csd.uu.se/ifip_tc13/

³ <http://www.acm.org/uist/>

⁴ <http://www.acm.org/tochi/>

⁵ <http://www.iuiconf.org/>

⁶ ACM SIGCHI Curriculum Development Group
<http://www.acm.org/sigchi/eduichi/>.

⁷ Columna mensual en el Boletín SIGCHI de ACM.

El avance de las interfaces de manipulación directa no solamente se ha basado en desarrollos tecnológicos, como pueden ser los sistemas de gestión de ventanas y los sistemas basados en eventos, sino que en él ha jugado un papel importante el desarrollo de múltiples ideas que han permitido adaptar los datos y contextos de trabajo de las aplicaciones interactivas a las limitaciones que presentan las pantallas gráficas actuales. En este aspecto juegan un papel muy importante desde el punto de vista de la producción de software las metáforas basadas en la representación gráfica de una gran diversidad de datos utilizados por aplicaciones mediante objetos más o menos familiares de la vida diaria. La diversidad de estas metáforas alcanza a ámbitos tan diversos como los escritorios, documentos, fotografías, electrodomésticos (como la televisión), discos musicales, pilas de cartas, juegos, películas, contenedores, árboles, mapas o ciudades. Este aspecto del desarrollo de aplicaciones interactivas, que desde una perspectiva tecnológica parece secundario, es de una importancia fundamental para las grandes empresas productoras de software interactivo, al igual que lo son aspectos muy diversos relacionados con la impresión que produce en el usuario el aspecto de las distintas pantallas que componen una aplicación, incluyendo su colorido, y la mayor o menor simplicidad de la ejecución de determinadas tareas en base a la distribución de las distintas componentes interactivas. Todo esto, junto con otros aspectos que tendremos ocasión de comentar en otros apartados de este trabajo, hacen que una aplicación interactiva de cierta envergadura necesite la participación de un equipo multidisciplinar, en el que pueden incluirse expertos en ingeniería del software, en aspectos audiovisuales e incluso en Psicología o Sociología.

Una dimensión totalmente diferente dentro de los estilos de interacción la constituye la idea de que el computador debe comportarse frente al usuario intentando anticipar la información que éste necesita, tanto en cuanto a los datos en sí que se le deben mostrar como muy especialmente en cuanto a la forma en que puede conseguir alcanzar sus objetivos. Se trata de la llamada interacción asistida, que con distintos grados de éxito y distintos tipos de tecnologías se viene poniendo a disposición de los usuarios en una gran diversidad de aplicaciones en los últimos años. Esta forma de interacción asistida utiliza con cierta frecuencia la metáfora del asistente personal o agente interactivo que colabora con el usuario en el mismo ambiente de trabajo permitiendo que el usuario, en lugar de dirigir la interacción, trabaje en un entorno cooperativo en el que él y los agentes o asistentes se comunican, controlan eventos y realizan tareas.

En la misma dirección, se han hecho propuestas, [Coutaz99], para conseguir que el computador siga las acciones del usuario, incluso las que tienen lugar en su ambiente de trabajo sin relación directa con el computador en sí, por ejemplo mediante una cámara de vídeo y tratamiento de imágenes, y utilice esa información tanto para tomar decisiones acerca de qué mostrarle como para filtrar la comunicación a través de la red con otros usuarios o para informar a éstos de la disponibilidad y posibilidad de interacción con el usuario inicial. Por último, la computación emocional, [Picard97], capaz de captar y transmitir sentimientos en su relación con los usuarios, se puede considerar como un paso adicional en esta dirección de acercar la relación entre usuarios y computadores a la existente entre personas. Las propuestas realizadas en este sentido se limitan de momento al ámbito investigador.

2.2 Paradigmas interactivos

Los estilos de interacción que acabamos de describir corresponden todos ellos a un mismo paradigma interactivo, el del computador de sobremesa, tal como lo concebimos actualmente. En los últimos años, sin embargo, han surgido varios otros paradigmas de interacción, [Lorés01], que plantean nuevas perspectivas para el futuro de la computación, especialmente en los aspectos más ligados a las actividades de las personas, en contraste con los más relacionados con su capacidad de realizar cálculos más rápidos o complicados.

Los nuevos paradigmas a los que nos referimos son la realidad virtual, la computación ubicua y la realidad aumentada. En todos ellos se han dado pasos adelante en entornos de investigación o de utilización con objetivos muy especializados.

La realidad virtual, [Burdea94], consiste en la utilización de dispositivos que hacen que el usuario tenga la misma sensación que si estuviera interaccionando realmente con un objeto físico determinado. Un sistema de emulación para la conducción que incluya un mecanismo que mueva el asiento del conductor haciéndole sentir impulsos semejantes a los que siente a causa de la inercia cuando conduce un automóvil se puede considerar como un sistema simple de realidad virtual. Otros sistemas más complejos permiten la manipulación de sus componentes con mayores grados de libertad. Las condiciones esenciales en un sistema de este tipo son la sensación de presencia física directa, la presentación de información en forma tridimensional y la manipulación de objetos virtuales en la misma forma en que se manipulan los objetos reales que representan.

La computación ubicua, [Mattern01], consiste en la creación de entornos integrados dentro de los cuales los usuarios tienen a su disposición la información y servicios que precisan en todo momento. Ello se consigue mediante la interacción con diferentes dispositivos y computadores en diferentes lugares y contextos, pero sin que el usuario tenga la necesidad de preocuparse por ello. En los últimos años se han dado importantes pasos en la dirección de la computación ubicua en entornos habituales de trabajo gracias a la gran variedad de dispositivos existentes y a los importantes avances en la comunicación sin hilos.

La realidad aumentada, [Barfield00], se centra en la integración entre el computador y el entorno físico que le rodea. A diferencia de la realidad virtual, que utiliza dispositivos especiales para interpretar las acciones del usuario, la realidad aumentada pretende que el computador potencie sus actividades proporcionándole en cada momento información no intrusiva acerca de los objetos reales con los que interacciona y mediante un seguimiento de las mismas. Las propuestas más avanzadas que hemos comentado en relación con el estilo de interacción precisan utilizar para su implementación mecanismos de realidad aumentada.

2.3 Tipos básicos de interfaces de usuario

Hasta aquí esta sección ha estado dedicada a la descripción de los aspectos globales que definen la forma en que se produce la interacción entre usuarios y computadores, concretamente los paradigmas y los estilos de interacción. A continuación describiremos algunos tipos de interfaces basadas en aspectos más específicos, que también son de gran importancia desde el punto de vista de la forma en que tiene lugar la interacción y de la forma en que se desarrollan las aplicaciones correspondientes. Conviene señalar aquí dos cuestiones: por una parte, en este apartado se podrá observar cierta heterogeneidad, pues en algunos casos los tipos de interfaces que describiremos estarán basados en una técnica de interacción, como puede ser el caso del hipermedia, mientras que en otros se tratará de un tipo específico de aplicación determinado por su aportación a la interacción de otra aplicación de la que depende, como ocurre con los sistemas de ayuda al usuario. Esto, junto con la compatibilidad mutua de varias de las técnicas que se describen, hace que los tipos básicos de interfaces que mostraremos a continuación no sean necesariamente excluyentes, en el sentido de que, por ejemplo, un sistema de ayuda puede estar basado en hipermedia. En segundo lugar, así como la selección de los estilos y paradigmas de interacción mostrados anteriormente se ha realizado en base a criterios admitidos universalmente, esta

subsección de tipos básicos de interfaces está basada en criterios más subjetivos, como son la relevancia que se les ha dado entre algunos sectores de la comunidad investigadora en los últimos años, la experiencia e intereses de los autores e incluso los objetivos de este artículo. En particular es importante tener en cuenta que en esta sección no pretendemos ser exhaustivos ni realizar una taxonomía de aplicaciones interactivas, algo que caería fuera de los propósitos de este trabajo.

Los tipos de interfaces que vamos a resaltar son los siguientes: las interfaces multimedia e hipermedia, las que utilizan restricciones, las que utilizan sistemas de tareas del usuario, los sistemas de trabajo colaborativo, las interfaces autoadaptables y los sistemas de ayuda al usuario. En casi todos ellos faremos comentarios generales acerca de sus características y posibilidades y posteriormente, en la sección 6, tendremos ocasión de ponerlos en relación con técnicas y trabajos recientes en el ámbito de la Inteligencia Artificial.

En primer lugar resaltaremos las interfaces hipermedia, que permiten al usuario acceder a la información almacenada en un documento formado por presentaciones interactivas conectadas entre sí de forma que el usuario tiene cierta libertad en el paso de unas presentaciones a otras. Entre las ventajas de este tipo de aplicaciones, [Lorés01], destacan su facilidad de uso, la posibilidad de distribuir las componentes de los documentos en distintos lugares y de acceder a ellas a través de la red y la posibilidad de representar información con distintos niveles de estructuración, y de forma muy especial asociaciones entre conceptos e información en general.

El segundo tipo de interfaz que destacaremos son las interfaces con sistemas de restricciones. Durante la última década se ha dedicado un esfuerzo considerable a la extensión del paradigma de las hojas de cálculo a las interfaces de usuario, [Myers90], permitiendo definir restricciones entre el contenido de determinadas partes de la interfaz, por las cuales los datos asociados a dichas partes (incluyendo los que definen su formato) dependen dinámicamente de otros datos. De esta forma se puede conseguir, por ejemplo, extraer o simplemente resaltar sublistas formadas por los elementos de una lista que verifican un criterio dado, que se actualizan automáticamente a medida que varían las condiciones de los elementos de la lista inicial. También se puede conseguir mediante este mecanismo que en la representación de una escena tridimensional un foco ilumine constantemente a un objeto que se mueve. Hoy día es frecuente entre las herramientas para la construcción interactiva de interfaces de usuario que incorporen la posibilidad de

definir restricciones entre sus componentes del tipo anterior. Sin embargo, tanto la definición de sistemas más potentes de restricciones como de mecanismos sencillos para su especificación son temas en los que se trabaja activamente.

Destacaremos a continuación los sistemas de trabajo en grupo o colaborativo (*Groupware*), [Churchill01]. Este tipo de sistemas permite que varias personas colaboren en su trabajo con el computador pudiendo realizarlo simultáneamente, compartir materiales y coordinarse entre sí. Los sistemas de trabajo en grupo permiten la colaboración de usuarios locales y/o remotos trabajando tanto síncrona como asíncronamente. Los sistemas de trabajo en grupo tienen su principal aplicación en contextos como el soporte a la toma colectiva de decisiones, los procesos de trabajo estructurado y el aprendizaje colaborativo.

Las interfaces que utilizan sistemas de tareas del usuario, [Diaper89], [Puerta96], [Hinrichs96], forman un tipo de interfaces de la mayor importancia. A partir de una descripción de las acciones que puede hacer el usuario se puede generar un modelo de aplicación que admite el sistema de tareas dado, así como se pueden desarrollar herramientas de valor añadido basadas en el seguimiento dinámico de las tareas. Además, la utilización de un sistema de tareas puede permitir la suspensión de una sesión y su reinicio posterior en el mismo u otro lugar, o la incorporación de nuevos usuarios a una sesión de colaboración síncrona, [Chung99], así como la simulación y repetición posterior del trabajo realizado durante una sesión, con posibles aplicaciones a algunos ámbitos específicos como la enseñanza, [García99], [Mora02].

Evidentemente, la diferencia entre una aplicación interactiva y una que no lo es radica en las acciones que el usuario puede hacer en los diferentes contextos, por lo que ya en una primera aproximación parece lógico que la determinación de estas acciones sea una parte esencial de la especificación de la aplicación en sí. Además, una descripción de una sesión de trabajo con una aplicación interactiva debería consistir en la descripción del entorno inicial de trabajo seguido de las acciones o tareas realizadas por el usuario, por lo que cualquier herramienta que pretenda prestar apoyo al usuario a cualquier nivel debería de resultar más fácil de construir si la aplicación dispone de un sistema que almacene una descripción de las tareas realizadas por el usuario. Además, suponiendo que el sistema de tareas se base en un protocolo estándar, se puede pensar en herramientas de valor añadido que se puedan utilizar

con aplicaciones interactivas arbitrarias que utilicen el modelo de tareas dado.

Las posibilidades reales de las ideas anteriores se pueden percibir mejor si pensamos en el proceso interactivo entre el usuario y una aplicación como un diálogo en el que las respuestas de la aplicación son automáticas en función de las intervenciones del usuario, por lo que a todos los efectos podemos pensar en la intervención de un único actor. De hecho, puede ocurrir que la intervención del usuario incluya expresiones de viva voz en las que parte de la información contextual se incorpora a través de la posición del ratón u otros mecanismos análogos. Una de las dificultades inherentes a este proceso de comunicación del usuario al computador es que el usuario, que no considera a éste como un interlocutor en todo el alcance de la palabra, no se preocupa de que sus intervenciones sean consistentes, sino que con frecuencia intercala acciones parciales con objetivos diferentes, por lo que el análisis del trabajo realizado para extraer información elaborada acerca del mismo exige una labor de desambiguación diferente de la habitual en la comprensión del lenguaje natural.

Otro tipo de interfaces que tiene gran importancia son las interfaces *adaptativas*. Hay dos factores principales que pueden hacer necesaria la utilización de interfaces que se adapten al contexto: la diversidad de dispositivos interactivos y la diversidad de tipos de usuario. Ambos factores son especialmente relevantes ante la multiplicidad de aplicaciones interactivas previstas para ser utilizadas a través de la web desde computadores con características arbitrarias, o incluso a través de teléfonos móviles, y por personas indeterminadas.

Las interfaces adaptables al usuario, [Malinowski93], [Browne90] normalmente incluyen un perfil del usuario basado en datos que se le piden directamente o extraídos a partir de su utilización previa del sistema. Evidentemente, este es un terreno que se presta a la utilización de técnicas de inteligencia artificial. De la importancia de este campo puede dar idea el hecho de que existe una revista, *User Modeling and User-Adapted Interaction: An International Journal*⁸, dedicada exclusivamente a él.

Similarmente, las interfaces adaptables al dispositivo, [Abrams99], parten de una descripción de las características esenciales del dispositivo de interacción utilizado. En este caso lo habitual es que estas características sean simples parámetros que

⁸ UMUAI: <http://umuai.informatik.uni-essen.de>.

utiliza el sistema para autoconfigurarse, aunque el proceso posterior de elaboración de la información a mostrar durante el funcionamiento de la aplicación y de las formas posibles de interacción con ésta en función de la configuración puede ser de suma complejidad, incluyendo en ocasiones técnicas de planificación.

También son objeto de estudio, aunque su utilización está mucho más restringida a campos concretos en los que es esencial la minimización de la información que se muestra al usuario, las interfaces en las que la propia estructura de la información mostrada o de las posibilidades de interacción dependen de los datos que maneja la aplicación, [Szekely92]. Por ejemplo, en un sistema para el control de una instalación que incluye distintas componentes, puede ser importante que se muestre información detallada sobre el estado de una componente únicamente cuando se pasa determinado umbral de emergencia.

En general, las interfaces *adaptativas* se pueden ver como aplicaciones interactivas con un sistema de restricciones por el cual las propias componentes que se muestran al usuario y su contenido dependen del modelo de usuario, de dispositivo, etc. Este tipo de restricciones, que es diferente del descrito más arriba en esta misma sección, puesto que la propia estructura de la interfaz se puede modificar en función de las dependencias existentes y no solamente el contenido de sus componentes, se puede ver al igual que en el caso anterior como un conjunto de reglas que determinan los criterios a tener en cuenta para construir la información a mostrar al usuario y la interacción posible en cada momento. En un sistema de presentación dinámico en el que la presentación de los datos se hace en dos etapas, con una construcción previa de las componentes a mostrar seguida de una fase de presentación gráfica, las restricciones necesarias en un sistema *adaptativo* se pueden reducir a restricciones en el sentido utilizado anteriormente que se satisfacen durante la etapa de construcción de las componentes de la interfaz. En la práctica, sin embargo, la implementación de este procedimiento suele presentar problemas de eficiencia importantes, por lo que se recurre a distintas formas de compilación del conocimiento contenido en las reglas de adaptación.

Los sistemas de ayuda acerca de la utilización de aplicaciones interactivas constituyen un tipo de interfaz que merece ser considerado separadamente. En una línea de avance progresivo partimos de los sistemas basados en la línea de comandos propios de los sistemas operativos y de la utilización del hipertexto, un sistema consolidado pero que tiene importantes limitaciones ya que las explicaciones habitualmente se dan fuera del contexto real de

trabajo del usuario. Algunas aplicaciones ofimáticas de gran difusión han incluido sistemas tutores interactivos para el aprendizaje de la aplicación, basados en emulaciones parciales de la misma. Estos sistemas tienen un altísimo coste de desarrollo y mantenimiento y no siempre cubren las necesidades del usuario al proporcionar las explicaciones en un contexto que puede no ser apropiado. En la práctica, este tipo de tutores ha sido desecharo tras su utilización durante algún tiempo. La utilización de agentes basados en redes *bayesianas* e incluso con un módulo en lenguaje natural que flexibiliza la forma en que se pueden realizar las consultas ha sido un paso de gran importancia, que sin embargo ha tenido limitaciones en cuanto a su aceptación, por lo que en estos momentos se ha replanteado su uso en las principales aplicaciones ofimáticas. Sin embargo, hay evidencias recientes a través de presentaciones en distintos foros, [Horvitz02], de que este tipo de técnicas se sigue considerando de fundamental importancia de cara al futuro. Por último, los trabajos de investigación de los últimos años apuntan hacia la generación automática de ayuda a partir de modelos asociados a diferentes componentes de la interfaz, como el modelo de navegación, [Palanque93], un modelo de objetivos de la interacción, [Sukaviriya90], un modelo de las componentes interactivas, [Moriyón94], o un modelo de tareas, [García99]. En estos trabajos se han utilizado distintos tipos de técnicas de Inteligencia Artificial, como el razonamiento hacia atrás en el sistema de ayuda basado en un modelo de objetivos para la interacción hasta el razonamiento hacia adelante basado en un conjunto de reglas de deducción en el caso de un sistema basado en un modelo de las componentes interactivas o el análisis gramatical (*parsing*) en el caso de un sistema basado en un modelo de tareas del usuario.

También las interfaces para la *programación mediante demostración*, [Cypher 93], [CommACM-00], merecen una mención especial. En estos sistemas los programadores trabajan sobre una interfaz a la que se pretende añadir una determinada funcionalidad a partir de otras preexistentes. El usuario (programador) proporciona ejemplos acerca de cómo realizar la tarea que desea especificar. El sistema infiere la tarea que especifica el usuario a partir de sus ejemplos, sintetizando un programa que se puede reutilizar para automatizar la tarea demostrada, reduciendo por lo tanto la cantidad de trabajo rutinario que tiene que hacer el usuario en el proceso de demostración.

Los sistemas de grabación de *macros* y muchas herramientas para la construcción de interfaces, de las que hablaremos más adelante, son ejemplos simples de interfaces del tipo anterior que

normalmente funcionan con un solo ejemplo por parte del usuario y no realizan ninguna inferencia, limitándose a transcribir en código de forma literal las acciones ejecutadas por el usuario. Un sistema de programación mediante demostración auténtico, sin embargo, efectúa inferencias tales como considerar alguna forma natural de identificar los objetos sobre los que actúa el usuario aparte de su simple localización en términos de *pixels*, o bien deduciendo la existencia de un bucle con una condición de terminación que determina como parte del proceso deductivo.

3. Interfaces para el usuario

3.1 Diseño orientado a la usabilidad

Pese a la idea ampliamente extendida de que la incorporación a una aplicación de una interfaz gráfica interactiva es siempre una mejora para el usuario, la realidad demuestra que esto no es así necesariamente y que la forma en que está diseñada la interfaz puede tener un efecto muy importante sobre las posibilidades de interacción con la misma que, especialmente para usuarios experimentados que utilizan la aplicación como herramienta de trabajo, dan lugar a niveles muy diferentes de productividad. Un ejemplo muy claro de esta situación se da en la utilización de aplicaciones de diseño (CAD); en ellas los usuarios poco experimentados tienden a trabajar utilizando el ratón y los menús y barras de tareas, mientras que los usuarios experimentados especifican gran parte de los comandos que utilizan de forma textual, reduciendo la información que especifican mediante el ratón casi exclusivamente a la de carácter puramente multidimensional. Hoy día, el paso en una institución de las interfaces alfanuméricas tradicionales en aplicaciones de gestión a interfaces gráficas interactivas requiere un estudio a menudo complejo de las consecuencias que ello tiene en cuanto a necesidad de formación de los usuarios de la aplicación y cambios en su productividad, aparte de los aspectos económicos inherentes a los cambios necesarios en el equipamiento informático.

Uno de los conceptos acuñados dentro de la IPC que más éxito ha tenido es la *usabilidad*. La usabilidad está ligada a conceptos de productividad y satisfacción del usuario y va más allá de aquella idea inicial de *interfaz amigable* que se manejaba al comienzo de las interfaces gráficas de usuario. Para su aplicación ha sido necesario desarrollar metodologías de diseño de interfaces orientadas al usuario y, sobre todo, sistemas para evaluar la usabilidad de las interfaces a lo largo del ciclo de desarrollo.

A pesar de su difusión es difícil encontrar una definición aceptada universalmente de usabilidad. Jenny Preece, [Preece93]⁹, define Usabilidad como *una medida de la facilidad de aprendizaje y de uso de un sistema, de su seguridad, efectividad y eficiencia, y de la actitud de los usuarios hacia él, y la evaluación de la usabilidad como la valoración de la usabilidad de un sistema en términos de facilidad de aprendizaje y de uso, flexibilidad, seguridad, efectividad, eficiencia y de la actitud de los usuarios respecto del sistema*

Para superar la resistencia de la industria -que teme al aumento de los costos- a la evaluación de la usabilidad, Nielsen recomienda en [Nielsen93] lo que él denomina *discount usability engineering*, métodos basados en el uso de tiempo y recursos reducidos, a la vez que desarrolla en detalle el proceso de introducción de la usabilidad en el ciclo de vida de la ingeniería. Por otro lado, en [Nielsen94], se describen diversos métodos de evaluación de la usabilidad que tienen en común el uso de diferentes tipos de evaluadores para examinar los aspectos de usabilidad en la interfaz de usuario. Otros autores, como Rubin [Rubin94], se centran en métodos de análisis de la usabilidad que emplean como evaluadores solamente a representantes de la población objetivo (lo que excluye los métodos descritos por Nielsen, que se basan en la actuación de expertos o comités) incidiendo en la distinción entre la metodología clásica de evaluación de experimentos, frecuentemente empleada en las ciencias sociales, y los métodos de análisis de la usabilidad (menos formales)¹⁰. A partir de la revisión de estos métodos se puede concluir que ninguno de ellos es *el definitivo*. En cada caso hay que buscar aquél que mejor se adapte a los objetivos, las características técnicas y, sobre todo, a los plazos y las posibilidades económicas.

Últimamente se están haciendo interesantes avances en el desarrollo de sistemas de análisis automático de la usabilidad [Falguras01], sobre todo orientados al entorno de la web [Paternó97].

3.2 Diseño para todos

Los usuarios con necesidades especiales forman una importante fracción del total de las personas que

⁹ Ver también [Preece01].

¹⁰ Como contrapunto irónico, mencionemos que no todo el mundo está de acuerdo en que la evaluación detallada de la usabilidad mereza la pena: "I have little respect for testing and evaluation in interface research. My argument, perhaps arrogant, is that if you have to test something carefully to see the difference it makes, then it is not making enough of a difference in the first place" [Negroponte95].

utilizan sistemas informáticos. Una gran parte de ellos corresponde a las personas con discapacidades físicas, sensoriales o cognitivas¹¹. Estos usuarios se pueden beneficiar en gran medida del uso de equipamiento informático para realizar tareas que de otra manera les resultarían muy difíciles o imposibles. Este es el caso, por ejemplo, de los sistemas de comunicación interpersonal para personas que carecen de habla y tienen limitaciones motoras, o sistemas de control de entorno para personas con graves restricciones de movimiento.

Durante bastante tiempo, la accesibilidad al computador de este tipo de usuarios se resolvía mediante arreglos “artesanos” dirigidos a las necesidades concretas de una persona o de un restringido grupo de personas con las mismas características. La evolución de los dispositivos y de los programas hacia frecuentemente que estas soluciones resultaran obsoletas al poco tiempo. En este sentido, la IPC ha aportado métodos y técnicas de diseño de sistemas de interacción más generales que han permitido plantear soluciones duraderas y con un grupo de usuarios amplio. En este sentido resulta muy recomendable la utilización de las pautas de diseño¹² recopiladas por grupos de trabajo con experiencia en el desarrollo de interfaces no excluyentes [Abascal01].

En los últimos años se ha acuñado la expresión *diseño para todos* para denominar a una filosofía que trata de que se tengan en cuenta las necesidades de

¹¹ *La población de usuarios con discapacidad es difícil de medir, ya que en cada país se usa diferentes parámetros para definir la discapacidad. Sin embargo, todas las estadísticas coinciden en que es una importante parte de la población, cuya importancia relativa aumenta con el envejecimiento.*

¹² *La rápida expansión de Internet ha evidenciado la necesidad de garantizar a las personas con discapacidad el acceso también a las redes telemáticas, como un paso más en pro de la no discriminación. En este sentido, el trabajo de la Web Accessibility Initiative (WAI, <http://www.w3.org/WAI>) dependiente del Worldwide Web Consortium (W3C, <http://www.w3.org/>) ofrece un buen ejemplo de pautas de diseño para mejorar la accesibilidad. Esta iniciativa ha permitido no sólo dar solución a las barreras de accesibilidad existentes, sino también encarar la aparición de nuevas dificultades, más difíciles de resolver, mediante la participación en la discusión y propuesta de nuevos métodos, recomendaciones, estándares, etc. La Comunidad Europea también viene realizando esfuerzos para hacer que las personas con discapacidad no estén discriminadas a la hora de hacer uso de los servicios públicos que se ofrecen a través de Internet [CEC01]. El objetivo es que a más largo plazo también los sitios web privados sean completamente accesibles.*

todos los usuarios desde las primeras fases del diseño, lo que evitaría las modificaciones *ad hoc* para usuarios con necesidades especiales [Stephanidis01]. Aunque esta filosofía está dando resultados muy prometedores todavía cuenta con grandes dificultades para su implantación general. En primer lugar, requiere una mayor difusión, sobre todo entre las grandes compañías productoras de software. Además, existen casos especiales en los que el *diseño universal* no puede garantizar la accesibilidad. Para esos casos sigue siendo necesario el diseño de dispositivos especialmente adaptados a las características del usuario¹³.

4. Arquitecturas de interfaces de usuario

La peculiaridad inherente a las interfaces de usuario interactivas, que reciben eventos asíncronos emitidos por el usuario y reaccionan en función de los mismos, es la causa de que las aplicaciones con una interfaz de este tipo tengan una estructura basada en uno o más bucles de captura de eventos y en un mecanismo de asignación de acciones a cada tipo de evento, como es bien sabido. Los creadores del lenguaje *Smalltalk*, uno de los primeros y más puros lenguajes orientados a objetos, se plantearon la simplificación del diseño y el desarrollo de las interfaces de usuario estructurando las aplicaciones mediante la separación de su componentes gráfica y algorítmica. Con ello crearon la denominada arquitectura MVC (Modelo, Vista, Controlador), [Goldberg84], que es una de las más extendidas hoy día para aplicaciones interactivas. Esta arquitectura consta de tres componentes, correspondientes a las siglas que le dan nombre. La primera componente, el Modelo, incluye los datos propios de la aplicación a los que ésta accede y que modifica mediante sus algoritmos internos o mediante el proceso de interacción, así como los algoritmos citados. La segunda componente, la Vista, contiene los datos y algoritmos correspondientes a la interfaz en sí, necesarios para mostrar y actualizar la componente gráfica de la aplicación. La tercera componente, el Controlador, contiene la información y los mecanismos necesarios para asociar una acción a cada evento.

En la mayoría de las implementaciones basadas en programación orientada a objetos esta estructura está distribuida entre una multitud de objetos, cada uno de los cuales contiene una parte que enlaza con el

¹³ *La Tecnología de la Rehabilitación (Assistive Technology) es la encargada de diseñar este tipo de equipamiento. Cook y Hussey [Cook95] hacen una completa descripción de este tipo de dispositivos.*

modelo, otra con la correspondiente de la vista y una tercera que corresponde al control de dicha parte.

El aspecto más negativo de la arquitectura MVC es el hecho de agrupar los diálogos de las aplicaciones conjuntamente con su funcionalidad, lo que dificulta la *reusabilidad* de la aplicación o del diálogo como componentes separados.

La arquitectura PAC (Presentación, Abstracción, Control), [Coutaz95], pretende superar este y otros inconvenientes de la arquitectura MVC. En una primera aproximación, la Presentación incluye a la Vista de MVC, pero también a los diálogos, que dejan de estar incluidas en la Abstracción (el Modelo en MVC). Dado que los diálogos pueden en principio llevar a otras partes arbitrarias de la aplicación, esta definición se tiene que modificar permitiendo que el modelo arquitectónico PAC, basado en varios agentes o entidades cooperantes, sea recursivo: la Presentación de un agente PAC puede estar compuesta por otros agentes PAC, en cuyo caso el Control efectúa la sincronización entre la Presentación y la Abstracción correspondientes y se encarga de comunicarse con los controles de sus subagentes, administrando la relación entre todos ellos (por ejemplo, la historia y ayudas a la interacción).

Las distintas técnicas de interacción incluidas en la sección 2 dan lugar a aspectos específicos en la arquitectura de las aplicaciones correspondientes o incluso en casos como el de las aplicaciones para el trabajo en grupo, a arquitecturas específicas.

Programas de aplicación
Toolkit
Sistema de ventanas
Sistema operativo

Figura 1. Componentes del software de interfaz de usuario

Independientemente de las arquitecturas globales de las aplicaciones, es interesante mencionar la estructuración que suele tener el software que se utiliza en las aplicaciones interactivas, en las que habitualmente se reutilizan componentes que adoptan la forma de bibliotecas de funciones o *frameworks* de clases. Desde este punto de vista la clasificación más natural del software es desde la perspectiva de la abstracción funcional, de manera que las componentes de nivel superior hacen llamadas a las de nivel inferior. La figura 1 resume la clasificación por niveles de abstracción funcional de las componentes del software de interfaz propuesta por Myers, [Myers92]. El nivel inferior lo forman los

sistemas de ventanas que gestionan y controlan múltiples contextos separándolos en diferentes partes de la pantalla. Pueden formar parte del programa (Smalltalk), parte del sistema operativo (Macintosh) o ser un programa separado (X).

Por encima de ellos están los *toolkits*, bibliotecas de rutinas reutilizables que contienen *widgets* (objetos de interacción tales como menús, botones, barras de scroll, ...). Por último, los toolkits son utilizados por las aplicaciones interactivas.

5. Herramientas de diseño de interfaces

Según Myers, *una herramienta de diseño de interfaces de usuario es una aplicación que ayuda a los diseñadores de la interfaz de usuario o a los programadores a su diseño, implementación y prueba de forma interactiva* [Myers92]. Él mismo indica que, por término medio, aproximadamente el 50% del esfuerzo de programación de una aplicación tiene relación con la interfaz de usuario, seguramente debido al interés comercial que están adquiriendo los aspectos de interacción, lo que justifica el esfuerzo dedicado a producir mejores herramientas para el diseño de interfaces. La idea de simplificar el diseño, implementación y prueba de aplicaciones interactivas mediante métodos a su vez interactivos, evitando en lo posible la labor tediosa y en gran medida rutinaria de escritura de código es una de las más profundas y productivas en la corta historia de las interfaces de usuario.

El objetivo de las herramientas de soporte es facilitar el diseño de sistemas independientes de la información que fluye entre la aplicación y el usuario, y permitir la separación de la programación de la interfaz respecto de la aplicación. También permiten estas herramientas optimizar mediante técnicas específicas el diseño e implementación de la interfaz.

Los *toolkits*, a los que nos hemos referido en la sección anterior como una de las piezas básicas de la componente interactiva de las aplicaciones, pueden ser usados desde los programas para la realización de interfaces basadas en procedimientos u orientadas a objetos. Ofrecen un aspecto coherente y permiten la reutilización del código, pero pueden resultar difíciles de usar debido a lo engoroso de las llamadas a las bibliotecas y la complejidad de los manuales. Se trata pues de componentes software naturales desde el punto de vista estrictamente tecnológico, pero que presentan deficiencias a menudo desde el punto de vista de la programación global de la aplicación, ya que no tratan de describir

o mantener la interfaz de usuario como un todo y suelen imponer limitaciones al estilo de interacción.

En un nivel superior desde la perspectiva de la abstracción en el diseño se encuentran las herramientas de alto nivel, cuya necesidad se justifica por la dificultad de uso de los *toolkits*. Existen varios tipos, entre los que se encuentran las herramientas de construcción de prototipos, los constructores de interfaces y los sistemas de gestión de interfaces de usuario (UIMS) o entornos de desarrollo de interfaces de usuario (UIDE).

Las maquetas de pantalla y constructores de prototipos permiten la realización de prototipos rápidos que dan una idea del aspecto externo de la interfaz. Los más sencillos producen secuencias de pantallas controladas por el usuario (como una proyección de diapositivas), los más avanzados (MacroMind Director, Hypercard...) actúan como UIMS pero sin conexión con la aplicación.

Los constructores de interfaces (*interface builders*) son herramientas de configuración para hacer ventanas que producen el código para la aplicación. Permiten el diseño de menús, cajas de diálogo, controles, etc. Un caso particular son los constructores interactivos que usan técnicas de manipulación directa para especificar interfaces también de manipulación directa, conectando diferentes componentes con comportamiento interactivo predefinido (como por ejemplo *Sun DevGuide*).

Los sistemas de gestión de interfaces de usuario (UIMS, *User Interface Management Systems*) o los entornos de desarrollo de interfaces de usuario, son el nivel más alto en el *software* de interfaz de usuario. Suelen incluir un programa principal que permite el control externo de la aplicación. Poseen herramientas usadas en *tiempo de diseño*, en *tiempo de ejecución* y para prueba del resultado, lo que garantiza la independencia con respecto de la aplicación. El diseñador de la interfaz de usuario describe al UIMS la interfaz deseada utilizando un lenguaje de descripción de la interfaz de usuario (UIDL). El UIDL describe el comportamiento de la interfaz tal como será visto por el usuario, pero no su realización. De este modo, el diseñador escribe la especificación y el UIMS la ejecuta.

La tendencia general en la construcción de software interactivo es la de avanzar produciendo nuevos estilos de interacción, aumentando la independencia del diálogo, desarrollando nuevos tipos de herramientas de diseño y mejorando las existentes. En este sentido se han publicado interesantes experiencias en especificación por demostración

interactiva, generación automática de interfaces de usuario, definición de conceptos y herramientas para construir interfaces de realidad virtual, desarrollo de *software* para interfaces multiusuario y para interfaces destinadas a usuarios con necesidades especiales.

6. Las interfaces inteligentes

Como ya hemos comentado anteriormente, la interfaz permite la comunicación bidireccional entre el usuario y el computador. En esta comunicación el usuario juega un papel activo en dos aspectos diferentes: por una parte, en la comprensión de la información que le proporciona el computador y en segundo lugar, en el conocimiento de la forma en que puede enviar información a éste. Estos dos aspectos son más críticos a medida que aumenta la complejidad de las aplicaciones informáticas, lo que en muchos casos implica el manejo de modelos abstractos de datos y de complejas representaciones gráficas del sistema, reforzando el papel de la interfaz a nivel cognitivo. Consiguientemente, tiene cada vez más interés la mejora del proceso de intercambio de información mediante la utilización de módulos inteligentes, que jueguen un papel análogo al que juega la inteligencia en la comunicación entre personas. Esta mejora se puede realizar sobre todo en dos aspectos: por una parte, facilitando que la interpretación por parte del usuario de la información ofrecida por el sistema se adecue al modelo de tarea y a los objetivos previstos. Por otra parte, ayudando al usuario a aportar al sistema la información requerida dentro de las acciones que lleva a cabo. En ambos aspectos, la utilización de técnicas de Inteligencia Artificial puede ser de interés. En lo que sigue analizaremos distintas cuestiones relacionadas con la inclusión de módulos *inteligentes* en la interfaz de usuario, dejando de lado el problema diferente de la construcción de interfaces adecuadas para programas que tienen un comportamiento inteligente.¹⁴

De acuerdo con lo dicho anteriormente, podemos concebir una interfaz inteligente como un intermediario que permite salvar la distancia cognitiva entre la persona y el computador. Como la consecución de una comunicación efectiva entre

¹⁴ Don Norman planteó en IUI'02 la polémica al respecto de las interfaces inteligentes en la siguiente forma: “¿Necesitamos interfaces inteligentes? No lo creo: La inteligencia debería estar en la aplicación. La interfaz es la parte visible del sistema; con respecto a ella, los usuarios exigen estabilidad, predictibilidad y un sistema de imágenes coherente que puedan entender y por lo tanto aprender a manejar.”

ambos depende de que existan modelos cognitivos y de tareas compartidos (basados en suposiciones similares), cuando dichos modelos son lejanos, la interfaz inteligente realiza la labor de traducción de uno a otro. En este sentido, [Chignell90] define la interfaz como una *entidad inteligente entre dos o más agentes que interaccionan y que poseen información incompleta del conocimiento de los demás y/o de su forma de comunicación*. Desde el punto de vista arquitectónico, la inteligencia de una interfaz suele estar incluida en su módulo de control, en la forma de uno o varios submódulos que realizan tareas específicas.

La interfaz inteligente puede tomar diferentes formas. Por ejemplo, puede actuar como intermediario, comparable con la labor de un operador de una base de datos que atiende la consulta de un usuario final. El intermediario humano no sólo traduce, también guía y aconseja basándose en su experiencia en la estructura de la base de datos y en el comportamiento del usuario final. La interfaz inteligente también puede predecir las acciones del usuario en base a sus acciones previas, y darle sugerencias acerca de la mejor manera de llevarlas a cabo o, incluso, ofrecerle su realización automática.

En lo que respecta a las funciones y según Rissland [Chignell88], la interfaz inteligente debe ofrecer: *automatización de tareas rutinarias, acceso fácil a las herramientas, ayuda y documentación on-line en un estilo que anime a la experimentación, minimice los errores y no sea invasivo. Además, debe ser fácilmente entendible, susceptible de ser aprendida mediante modelos conceptuales y con transición suave del modo de operación de novato a experto.*

Independientemente de la atención más o menos puntual que prestan a estos temas congresos especializados en Interfaces de Usuario y en Inteligencia Artificial, la *International Conference on Intelligent User Interfaces*¹⁵, IUI, está dedicada exclusivamente a este tipo de interfaces y representa por lo tanto el principal foro en que se presentan los avances más importantes en este área.

La mayor parte de los aspectos considerados en los distintos tipos de interfaces de usuario descritos más arriba en este artículo son susceptibles de mejoras mediante módulos que aportan un comportamiento que se puede calificar de inteligente, como tendremos ocasión de comprobar a continuación. Además, entre las herramientas susceptibles de la utilización intensiva de técnicas de Inteligencia

Artificial se encuentran las dedicadas al diseño y la construcción interactivos de interfaces de usuario de distintos tipos.

En general, observamos en primer lugar que la posibilidad de incorporar un comportamiento inteligente a las interfaces adaptables al usuario es evidente, ya que la capacidad de adaptación depende en gran medida del desarrollo de un modelo suficientemente rico del usuario. A priori se podría pensar incluso en la posibilidad de definir un protocolo universal extensible de modelado de usuarios desde el punto de vista de su capacidad de manejo de aplicaciones interactivas (tanto en términos genéricos como específicos), pero no hay un modelo que haya demostrado su utilidad en este sentido, siendo en general cada aplicación responsable de acumular el tipo de información relevante para permitir su propia adaptación.

Por otra parte, las interfaces que incorporan restricciones contienen desde el punto de vista conceptual sistemas de reglas que permiten la actualización automática de una parte de la interfaz en función de las acciones llevadas a cabo por la aplicación. Además, las restricciones pueden ser multidireccionales y un sistema arbitrario de restricciones puede conllevar circularidades que hagan imposible su aplicación sin una planificación previa de las acciones a llevar a cabo en cada caso, [VanderZanden96].

La inclusión de un modelo de las tareas que se pueden llevar a cabo en una aplicación interactiva es otro aspecto que de por sí, por cuanto conlleva el desarrollo de una base de conocimiento relativo a dichas tareas, tiene aspectos muy ligados a la Inteligencia Artificial. En este terreno sí existen modelos genéricos que se pueden especificar adaptándolos a cada interfaz, [García99]. Sin embargo, los aspectos que realmente involucran una mayor profundidad desde el punto de vista de la utilización de técnicas de Inteligencia Artificial en relación con las interfaces que contienen modelos de tareas son aquellos relacionados con la explotación de la información de que dispone el módulo de tareas acerca de las acciones realizadas por el usuario. En este sentido son especialmente relevantes cuestiones como la construcción automática del modelo de tareas en función de las acciones de los usuarios (en ocasiones incluso resulta de utilidad que distintos usuarios tengan asociados modelos de tareas diferentes, que por lo tanto pasan a formar parte del perfil de usuario), [Maulsby97], lo que constituye un problema típico de aprendizaje automático, la anticipación de las acciones pendientes de realizar por parte de éste y la detección de dificultades en el

¹⁵ IUI (<http://www.iuiconf.org/>) está organizada por ACM SIGART y SIGCHI. Las actas las publica la ACM (<http://www.acm.org/catalog/proceedings/iui.html>).

trabajo del usuario y la propuesta de soluciones a éstas.

Por otra parte, los sistemas de ayuda forman un tipo de interfaces en las que tradicionalmente han incorporado algún tipo de inteligencia, como ya se ha puesto de manifiesto en la sección 2. En esta sección tendremos ocasión de describir otras contribuciones a este tipo de interfaces basadas en técnicas de Inteligencia Artificial.

En cuanto a las aplicaciones colaborativas, entre las principales dificultades que suelen encontrar los usuarios de las mismas está la de tener en cuenta las posibilidades y limitaciones que les impone su rol en una sesión de trabajo. La combinación de varias de las técnicas descritas en los párrafos anteriores, adaptadas convenientemente al entorno colaborativo, pueden ser de gran utilidad para evitar este tipo de problemas.

Dada la heterogeneidad de las técnicas que se pueden aplicar no se puede describir el proceso de construcción de una interfaz inteligente de una forma que sea válida para todos los casos. Sin embargo, pensando en los componentes más representativos y habituales, el proceso de construcción de una interfaz características puede incluir como etapas el análisis de tareas, el diseño de un modelo de usuario en base a la selección de una muestra representativa de usuarios, el desarrollo de un modelo de interfaz común basado en la integración de los modelos de usuario y de tarea y, por último, el diseño conceptual: a partir del modelo común de interfaz, hay que especificar las operaciones realizables dentro del modelo y las operaciones ejecutables en la realización de la tarea. Todo ello orientado a la construcción de una representación cognitiva adecuada de las tareas y de cómo las lleva a cabo el usuario. Estas etapas se realizan dentro del diseño del módulo de control de la aplicación, mientras que los módulos correspondientes al modelo y a la vista de la misma se diseñan de forma similar a como se hace para una aplicación interactiva estándar¹⁶.

Los avances más recientes en el terreno de las interfaces inteligentes hacen referencia a varios de los tipos de interfaces que hemos mencionado anteriormente. Haremos a continuación una breve descripción de algunos de ellos.

En el ámbito de los agentes interactivos destaca en primer lugar el trabajo de Stocky y Cassell, [Stocky02], acerca de agentes que combinan gesto y habla para transmitir información, junto con técnicas de realidad aumentada. También merece mención el trabajo de Sidner y Dzikovska acerca de agentes anfitriones robóticos, [Sidner02], que describe los requisitos de un agente de estas características junto con criterios de evaluación. Ambos trabajos están basados en sendas experiencias de implementación de agentes de los tipos indicados.

Dentro del campo de los sistemas de ayuda, [Küpper02] describe un algoritmo de planificación para el asesoramiento a usuarios en la consecución de objetivos con una interfaz, proporcionándoles los conocimientos que les faltan para ello. También en el mismo campo, pero en una dirección diferente, Masthoff y Gupta, [Masthoff02], describen un dispositivo consistente en una placa sensible al tacto con control por voz, capaz de dar información sobre las cuestiones que el usuario le pide acerca de un dominio específico, y que incluye explicaciones sobre su propio funcionamiento

Aparte de los sistemas capaces de interaccionar mediante el habla que acabamos de mencionar merece la pena destacar también [Gottlieb02], una propuesta de principios de diseño para la creación de programas interactivos capaces de mantener un diálogo con el usuario dentro de la interacción de éste con una página web.

En el terreno de los sistemas *adaptativos* destaca el trabajo de Brusilovsky y Cooper, [Brusilovsky02], que describe un sistema *adaptativo hipermédia* para la ayuda al mantenimiento de equipos complejos que utiliza modelos del dominio, las tareas y el usuario.

En cuanto a los sistemas basados en tareas, Franklin, Budzik y Hammond, [Franklin02], describen las ideas generales basadas en el análisis de las tareas del usuario, utilizadas en una interfaz basada en el habla para Microsoft PowerPoint que cambia automáticamente las diapositivas, una herramienta de búsqueda en la web que utiliza como contexto de la búsqueda el documento que el usuario está viendo y una interfaz para la enseñanza en el aula controlada directamente por el profesor por medio de sus indicaciones a lo largo de la presentación. En el mismo campo, en relación con las herramientas para la construcción de interfaces de usuario, también es de destacar el trabajo de Eisenstein y Rich, [Eisenstein02], que muestra cómo la especificación de un modelo de tareas interactivas permite generar simultáneamente una interfaz para la realización de las mismas y un agente colaborativo para ellas.

¹⁶ Para más información sobre diseño de las interfaces inteligentes en general: [Helander90], [Helander97], [Maybury98], [Sullivan91], así como las actas de IUI; respecto a las interfaces multimedia inteligentes: [Kaplan97], [Maybury93].

Por último, dentro de la computación ubicua y extensiva Fails y Olsen, [Fails02], describen un sistema para la interacción ubicua que no requiere que los usuarios lleven ningún dispositivo físico, utilizando para ello una combinación de cámara y procesador que siguen a los usuarios a la vez que protegen su intimidad, haciendo que cualquier superficie visible se convierta en un *widget* que utiliza los gestos de la mano para interactuar.

A parte de los trabajos que hemos comentado relacionados con los distintos tipos de interacción introducidos en secciones anteriores, la investigación más reciente en interfaces inteligentes incluye trabajos en muchos otros campos, como la edición inteligente de documentos, los entornos colaborativos para la computación científica y la interacción inteligente para la clasificación del correo electrónico.

7. Nuevos dispositivos de interacción

Últimamente se anuncian grandes cambios en la forma en la que el usuario se relacionará en el futuro con los computadores. Veamos algunos de ellos.

Algunos expertos predicen que el computador, al menos tal como lo conocemos hoy, desaparecerá en un periodo de tiempo no muy largo¹⁷. De acuerdo con estas predicciones, el computador estará integrado en otros dispositivos y no seremos conscientes de su existencia más que por las funciones que ofrezca.

La computación ubicua parecen tener una filosofía opuesta a la de la realidad virtual, que trata de *introducir* a la persona dentro del computador. En este caso, los computadores se integran en la vida de las personas. El objetivo de tratar de romper con el paradigma del computador de escritorio es común a los trabajos sobre computación móvil, ubicua y vestible (*wearable*). Todos ellos pretenden que los servicios que provea el computador se desplacen con el usuario y le permitan aprovechar los constantes cambios del contexto en que son usados. Esto puede dar lugar a entornos activos en los que estos

computadores interactúan entre sí y con el usuario de manera inteligente y no *invasiva*, como por ejemplo en las casas cableadas [Dutta-Roy99], creando así lo que se denomina la *inteligencia ambiental*.

Respecto de los dispositivos de interacción, e intentando vislumbrar la posible relación entre lo que serán las interfaces de usuario en el computador ubicuo del futuro y lo que son las actuales interfaces gráficas, se puede predecir que una porción significativa de la interacción de las personas con los sistemas de información y computación que les atiendan será a través del lenguaje natural. Otros procesos interactivos que requieran el intercambio de información más estructurada (datos numéricos, tablas de datos, etc.) tendrán que recurrir a dispositivos especiales de muñeca, agendas, etc. En este tipo de dispositivos es de esperar que el tipo de interacción posible sea bastante semejante al actual, compatibilizando la utilización del lenguaje natural con un mecanismo apuntador (que posiblemente terminará siendo el dedo) para simplificar la indicación de referencias de tipo geométrico o espacial.

Los computadores portátiles darán un paso adelante y se integrarán entre lo que el usuario lleva *encima*. Se prevén diferentes sistemas de ayuda personal y profesional (agendas, sistemas para navegación en carretera o en ciudad, ayuda para la realización de tareas, acceso a telefonía e Internet, etc.) en lo que se conoce como *wearable computation*. Independientemente de la función de este tipo de dispositivos, está claro que los computadores que *se llevan puestos*, entre otras características propias, necesitan un sistema especial de interacción con el usuario. Desde luego, la entrada se realizará fundamentalmente por voz, aunque es necesario encontrar medios de salvaguardar la intimidad cuando se usen en sitios públicos. Por ello no se pueden descartar otros métodos de entrada como pequeños teclados de tipo telefónico o el seguimiento del movimiento de los ojos para hacer las veces de ratón. La salida también puede producirse como voz sintética, posiblemente dirigida al oído del usuario mediante auriculares, para no molestar a los demás y garantizar la confidencialidad. Por motivos de seguridad, sería necesario que los auriculares transmitieran también la información sonora que el usuario recibe del exterior (palabras y sonidos) para que el uso del computador no lo aisle del entorno. También son previsibles salidas gráficas sobre gafas de cristal líquido que, al igual que los auriculares, permitan percibir simultáneamente la información procedente del entorno y la que sale del computador. Por supuesto, el *hardware* debe incluir otros dispositivos, tales como videocámaras digitales, y

¹⁷ En palabras de Mark Weiser, el creador del concepto de computador *ubicuo*: Durante treinta años le diseño de interfaces se ha enfocado a la máquina "dramática". Su más alto ideal es hacer al computador tan excitante, tan maravilloso, tan interesante que nunca queramos estar sin él. Hay un camino menos trillado que yo llamo el "invisible"; su más alto ideal es hacer al computador tan integrado, tan natural que lo usemos sin pensar incluso en él. (También he denominado a este concepto "computación ubicua" y he situado su origen en el postmodernismo). [Weiser94].

permitir el acceso sin cables a Internet, telefonía móvil, etc. El avance de este tipo de computación está también ligado a la disponibilidad de tecnologías básicas, como baterías de alta duración y bajo peso.

8. Conclusiones

La revisión de los métodos y técnicas de interacción persona-ordenador presentada en este artículo evidencia la multidisciplinariedad del área y la necesidad de aplicar conceptos, método y técnicas procedentes de otros campos. Uno de ellos -quizá el que más influencia esté teniendo en el avance hacia interfaces más sofisticadas- es la Inteligencia Artificial. Si tuviéramos que resumir en una frase la tendencia en lo relativo a las interfaces inteligente, diríamos que el objetivo general es conseguir *interfaces aparentemente inteligentes y fácilmente inteligibles*.

Referencias

- [Abascal01] J. Abascal and C. Nicolle. 'Why Inclusive Design Guidelines'. In C. Nicolle and J. Abascal (Eds.) Inclusive Design Guidelines for HCI. Taylor & Francis, (2001).
- [Abrams99] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. Williams, J. E. Shuster. 'UIML: An Appliance-Independent XML User Interface Language'. Proc. XML'99, (1999) (también en http://www.harmonia.com/resources/papers/www_8_0599/)
- [Barfield00] W. Barfield y T. Caudell (Eds.). 'Fundamentals of Wearable Computers and Augmented Reality'. Lawrence Erlbaum, (2000).
- [Browne90] Browne, P. Totterdell and M. Norman (eds.). 'Adaptive user interfaces'. Academic Press, (1990).
- [Brusilovsky02] P. Brusilovsky and D. W. Cooper. 'Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System'. Proc. Int. Conf. on Intelligent User Interfaces, ACM Press, (2002).
- [Burdea94] G. Burdea, and P. Coiffet, 'Virtual Reality Technology'. John Wiley and Sons, (1994).
- [CEC01] Comunicación de la Comisión al Consejo, al Parlamento Europeo, al Comité Económico y Social y al Comité de las Regiones "eEurope 2002: Accesibilidad de los sitios Web públicos y de su contenido". (2001). [http://www.discapnet.es/documentos/tecnica/pdf/com2001_0529es01.pdf]
- [Chignell-90] M. H. Chignell and P. A. Hancock 'Intelligent Interface Design'. In Helander M. (Editor). 'Handbook of Human-Computer Interaction'. North-Holland, (1990).
- [Chung99] G. Chung, P. Dewan and S. Rajaram. 'Generic and Composable Latecomer Accommodation Service for Centralized Shared Systems'. In 'Engineering for Human-Computer Interaction'. Kluwer, (1999).
- [Churchill01] E. F. Churchill, D. N. Snowdon, A. J. Munro (Eds.). 'Collaborative Virtual Environments'. Springer Verlag, (2001).
- [CommACM00] Communications of the ACM, vol. 43, no. 3. (2000). (Número especial dedicado a la programación mediante demostración).
- [Cook95] A. Cook and S. Hussey. 'Assistive Technologies: Principles and practice'. Mosby, (1995).
- [Coutaz95] J. Coutaz, L. Nigay and D. Salber, 'Agent-based Architecture Modelling for Interactive Systems'. In 'Critical Issues in User Interface Engineering', P. Palanque and D. Benyon (eds.) Springer-Verlag, (1995).
- [Coutaz99] J. Coutaz, F. Berard, E. Carraux and J. L. Crowley. 'CoMedi: Using Computer Vision to Support Awareness and Privacy in Mediaspaces'. Proc. CHI 99. ACM Press, (1999).
- [Cypher 93] A. Cypher (ed.). 'Watch What I do: Programming by Demonstration'. MIT Press, (1993).
- [Diaper89] D. Diaper. 'Task Analysis for Human-Computer Interaction'. Ellis Horwood, Books in Information Technology, (1989).
- [Dutta-Roy99] A. Dutta-Roy. 'Networks for Homes'. Spectrum IEEE, vol 36, no. 12, pp. 26-33, (1999).
- [Eisenstein02] J. Eisenstein and C. Rich. 'Agents and GUIs from Task Models'. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Fails02] J. A. Fails and D. Olsen Jr. 'Light Widgets: Interacting in Every-day Spaces'. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Falgueras01] J. Falgueras y A. Guevara. 'Evaluación realista de la usabilidad en la ingeniería de la interfaz persona-ordenador'. En Abascal et al. (eds.). 'Interacción'2001'. Ediciones de la Universidad de Salamanca, (2001).

- [Franklin02] D. Franklin, J. Budzik and K. Hammond. ‘Plan-based Interfaces: Keeping Track of User Tasks and Acting to Cooperate’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [García99] F. García, J. Contreras, P. Rodríguez and R. Moriyón. ‘Help generation for task based applications with HATS’. In ‘Engineering for Human-Computer Interaction’. P. Dewan (ed.). Kluwer, (1999).
- [Goldberg84] A. Goldberg, ‘Smalltalk-80: The Interactive Programming Environment’. Addison-Wesley Publ., (1984).
- [Gottlieb02] H. Gottlieb. ‘The Interactive Conversation Interface (ICI): A proposed successor to GUI for an interactive broadband world’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Helander90] M. Helander (ed.). ‘Handbook of Human-Computer Interaction’. North Holland, (1990).
- [Helander97] M. Helander, T. K. Landauer y P. V. Prabhu (eds.). Handbook of Human-Computer Interaction. Second edition. Nort-Holland, (1997).
- [Hewett96] T. Hewett et al. ‘ACM SIGCHI Curricula for Human-Computer Interaction. Report of the ACM SIGCHI Curriculum Development Group. ACM, 1992-96’. ACM Press, (1996). (También disponible en: <http://www.acm.org/sigchi/cdg>).
- [Hinrichs96] T. Hinrichs, R. Bareiss, L. Birnbaum and G. Collins. ‘An Interface Design Tool based on Explicit Task Models’. CHI’96, ACM Press. (1996).
- [Horvitz02] E. Horvitz. ‘Uncertainty, Intelligence, and Interaction’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Kaplan97] R. M. Kaplan. ‘Intelligent Multimedia Systems. A Handbook for Creating Applications’. Wiley, (1997).
- [Küpper02] D. Küpper and A. Kobsa. ‘Generating and Presenting User-Tailored Plans’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Lorés01] J. Lorés (ed.), J. Abascal, I. Aedo, J. J. Cañas, M. Gea, A.B. Gil, A. B. Martínez, M. Ortega, P. Valero, M. Vélez. Curso Introducción a la Interacción Persona-Ordenador. Accesible a través de <http://griho.udl.es/ipo/libroe.html>. Asociación para la Interacción Persona-Ordenador, 2001.
- [Malinowski93] J. U. Malinowski, M. Schneider-Hufschmidt and T. Kühme (eds.). ‘Adaptive user interfaces: principles and practice’. North Holland, (1993).
- [Masthoff02] J. Masthoff and A. Gupta. ‘Design and evaluation of Just-in-time Help in a multi-modal user interface’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (2002).
- [Mattern01] F. Mattern, M. Ortega and J. Lorés. ‘Ubiquitous Computing: The Trend Towards the Computerization and Networking of All Things’. Upgrade. Vol. 2, No. 5, October (2001). <http://www.upgrade-cepis.org/issues/2001/5/up2-5Present.pdf>
- [Maulsby97] D. Maulsby. ‘Inductive Task Modeling for User Interface Customization’. Proc. Int. Conf. on Intelligent User Interfaces. ACM Press, (1997).
- [Maybury93] M. T. Maybury (ed.). ‘Intelligent Multimedia Interfaces’. AAAI/MIT Press, (1993)
- [Maybury98] M. T. Maybury y W. Wahlster. (eds.). ‘Readings in Intelligent user interfaces’. Morgan Kauffman, (1998).
- [Mora02] M. A. Mora, R. Moriyón and F. Saiz. ‘Developing applications with a Framework for the Analysis of the Learning Process and Collaborative Tutoring’. International Journal of Continuing Engineering Education and Lifelong Learning (IJCEELL). En prensa, (2002).
- [Moriyón94] R. Moriyón, P. Szekely and R. Neches. ‘Automatic Generation of Help from Interface Design Models’. Proc. CHI’94, ACM Press, (1994).
- [Myers90] B. A. Myers, D. A. Giuse, R. B. Dannenberg, B. Vander Zanden, D. S. Kosbie, E. Pervin, A. Mickish, and P. Marchal. ‘Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces’. IEEE Computer, vol. 23, no. 11. (1990).
- [Myers92] B. A. Myers y M. B. Rosson. ‘Survey on User Interface Programming’. Proc. CHI’92, ACM Press, (1992).
- [Negroponte95] N. Negroponte. ‘Being digital’. Coronet Books, UK, (1995).
- [Nielsen93] J. Nielsen. ‘Usability Engineering’. Morgan Kaufmann, (1993).
- [Nielsen94] J. Nielsen y R. L. Mack (Eds.). ‘Usability Inspection Methods’. Wiley, (1994)
- [Palanque93] A. Palanque, R. Bastide y L. Dourte. ‘Contextual Help for Free with Formal Dialog Design’. Proc. 5th International Conference on

- Human-Computer Interaction, Orlando, Florida. ACM Press, (1993).
- [Paternó97] F. Paternó and L. Paganelli. ‘Remote Automatic Evaluation of Web Sites Based on Task Models and Browser Monitoring’. Proceedings ACM CHI’01, ACM Press, Seattle, (2001).
- [Picard97] Rosalind Picard. ‘Affective Computing’. MIT Press, (1997).
- [Preece01] J. Preece, Y. Rogers, and H. Sharp. ‘Interaction Design’. John Wiley & Sons. 2001.
- [Preece93] J. Preece (Ed.). ‘A guide to usability: human factors in computing’. Addison-Wesley, 1993.
- [Puerta96] A. R. Puerta. ‘The MECANO Project: Enabling User-Task Automation during Interface Development’. Proc. AAAI’96 Spring Symposium on Acquisition, Learning & Demonstration: Automating Tasks for Users, AAAI Press. Standford, (1996).
- [Rubin94] J. Rubin. ‘Handbook of Usability Testing’. Wiley, (1994).
- [Sidner02] C. L. Sidner and M. Dzikovska. ‘Hosting Activities: Experience with and Future Directions for a Robot Agent Host’. Proc. Int. Conf. on Intelligent User Interfaces, ACM Press, (2002).
- [Stephanidis01] C Stephanidis (Ed.). ‘User Interfaces for All - Concepts, Methods, and Tools’. Lawrence Erlbaum Associates,(2001).
- [Stocky02] T. Stocky and J. Cassell. ‘Shared Reality: Spatial Intelligence in Intuitive User Interfaces’. Proc. Int. Conf. on Intelligent User Interfaces, ACM Press, (2002).
- [Sukaviriya90] P. Sukaviriya and J.D. Foley. ‘Coupling a UI Framework with Automatic Generation of Context-Sensitive Animated Help’. Proc. UIST’90, pp. 152-166. ACM Press, (1990).
- [Sullivan91] J. W. Sullivan y S. W. Tyler (eds.) ‘Intelligent user Interfaces’. ACM Press, 1991
- [Szekely92] P. Szekely, P. Luo and R. Neches. ‘Facilitating the Exploration of Interface Design Alternatives: The HUMANOID Model of Interface Design’. Proceedings SIGCHI’92. ACM Press, (1992).
- [VanderZanden96] B.T. Vander Zanden. An ‘Incremental Algorithm for Satisfying Hierarchies of Multi-way, Dataflow Constraints’. ACM Transactions on Programming Languages and Systems, vol. 18, no. 1, (1996).
- [Weiser94] M. Weiser, The world is not a desktop. Interactions, pp. 7-8, January (1994).