

Sistemas de ayuda inteligente para entornos informáticos complejos

Baltasar Fernández Manjón

Dpto de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Ciudad
Universitaria s/n, 28040, Madrid, España, email: balta@sip.ucm.es

Abstract

Los sistemas de ayuda inteligente (SAI) o asistentes inteligentes son una propuesta para mejorar el aprendizaje y el rendimiento en la utilización de entornos informáticos complejos. Habitualmente la ayuda tiene como único fin la mejora del rendimiento del usuario, pero nosotros creemos que dicha asistencia debe tener también un propósito educativo. Si el usuario mejora su conocimiento sobre la aplicación su necesidad de ayuda decrecerá con el tiempo. Un punto clave de los SAI es el énfasis que se hace en la adaptación de la ayuda proporcionada a las necesidades y conocimientos específicos de cada usuario para lo que se utiliza un modelo explícito de usuario. En este trabajo se describe el SAI Aran donde se ha ejemplificado este enfoque.

Palabras clave: Sistemas de ayuda inteligente, Modelado de usuario, Informática educativa, Lenguajes de marcado

Introducción

Las nuevas circunstancias planteadas por la sociedad de la información, como, por ejemplo, la rápida obsolescencia de los conocimientos adquiridos, o la imposibilidad de adquirir previamente todos los conocimientos necesarios para realizar nuestro trabajo, han creado nuevos problemas y desafíos para las aplicaciones educativas. Con la generalización de los entornos cada vez más tecnificados, ya casi nadie duda de la necesidad de una formación continua, adaptada las necesidades específicas de cada uno, y más integrada con el trabajo real [Fischer, en prensa]. Los sistemas educativos inteligentes (e.g. tutores inteligentes, entrenadores inteligentes, asistentes inteligentes), que se han usado durante más de treinta años en dominios muy diversos [Brown 82, Wenger 87, Halff 98], tienen que abordar estos desafíos y proporcionar soluciones que integren nuevas estrategias educativas [Kearsley 00]. Ante

esta situación nosotros proponemos la utilización de los sistemas de ayuda inteligente (SAI o asistentes inteligentes) para resolver parte de los problemas que se plantean en el aprendizaje y uso de los entornos informáticos complejos (p.e. sistemas operativos, paquetes integrados) [Fernández-Manjón 96].

A pesar de los desarrollos de nuevas interfaces más amigables y sencillas, es necesario proporcionar ayuda complementaria a los usuarios de las aplicaciones informáticas. Esto es debido a que estas aplicaciones cada vez incluyen más funcionalidades, están en continua evolución y se han generalizado entre usuarios no expertos. En muchos casos, y a pesar de gran cantidad de documentación de uso asociada, los usuarios esperan poder usar las aplicaciones con un formación específica mínima (o incluso sin ella) y sin una experiencia previa en el uso de ordenadores. Los SAI pueden simplificar la asimilación y contextualización de la información (e.g. filtrado y acceso a datos relevantes) y para ello

consideramos necesario la inclusión de un modelado explícito de usuario (MU) [Kok 91, Wasson 92, Fernández-Manjón 98].

Nuestro enfoque de los SAI como método para mejorar la usabilidad de un sistema complejo es mediante una ayuda con propósito doble. Primero, esta ayuda tiene un propósito a corto plazo, que permita que el usuario termine su tarea actual mejorando su rendimiento. Segundo, la ayuda proporcionada por el SAI debe mejorar y expandir el conocimiento que tiene el usuario sobre la aplicación, de este modo la necesidad de ayuda decrecerá a largo plazo. Este segundo propósito tiene una finalidad educativa que creemos que es tan importante como la anterior. De esta forma los SAI pueden integrar de forma coherente el trabajo y el aprendizaje obteniéndose diversos beneficios, como, por ejemplo, una mayor motivación de los usuarios y un aprendizaje más real.

Esta investigación sobre los SAI se ha contrastado mediante la implementación de un asistente inteligente llamado Aran, aplicado al ya clásico dominio del sistema operativo Unix. En este sistema mediante la integración de tecnologías y la reutilización de información se ha demostrado que es posible producir SAI para sistemas complejos con un coste de desarrollo razonable. Además ahora con la generalización de la tecnología de los lenguajes de marcado producida por Internet y la disposición de muchas herramientas asociadas para la gestión eficiente de información estructurada se simplifica la construcción de estos sistemas de ayuda adaptativa.

En este artículo primero presentamos el contexto de los sistemas de ayuda para aplicaciones informáticas y nuestro enfoque de los SAI. A continuación describimos el SAI Aran haciendo énfasis en los aspectos de contextualización y adaptación de información y de su modelo de usuario que lo hace posible. Finalmente se presentan las conclusiones y trabajo futuro.

Contexto de los sistemas de ayuda

Se han realizado diversos estudios experimentales sobre los problemas asociados con el uso de los entornos informáticos complejos de uso general que tienen un gran número de funcionalidades, tales como los sistemas operativos y, a menor escala, los paquetes integrados (e.g. hojas de cálculo, procesadores de texto). Estos estudios empíricos (que habitualmente han estudiado, entre otros dominios, el sistema operativo Unix) [Duffy 92, Fischer 96] demuestran que los usuarios tienen un conocimiento incompleto del sistema, utilizando de

forma habitual sólo una pequeña parte de las utilidades de las que dispone.

El hecho de que muchos usuarios utilicen la computadora únicamente como una herramienta que facilita la realización de sus tareas hace que no suelen desear aprender nada que no les resulte imprescindible para conseguir sus objetivos. Los usuarios conocen tan sólo un conjunto de operaciones básicas, con las que son capaces de realizar sus tareas habituales de forma satisfactoria, pero de manera que hay partes del sistema que les resultan completamente desconocidas. Esto impide que aprendan otras utilidades más sofisticadas u otras formas de trabajo que les permitirían conseguir un mejor uso de la aplicación.

A partir de estos experimentos, Fischer propone un modelo de la utilización por parte de los usuarios de una aplicación compleja [Fischer 96], modelo que consideramos muy adecuado para explicar cuál es el papel que debe jugar un sistema de ayuda inteligente. Este modelo, que se muestra en la Figura 1, es generalizable a gran parte de los sistemas informáticos.

En el esquema, D_4 representa el sistema real, con el conjunto de funcionalidades (conceptos y órdenes)

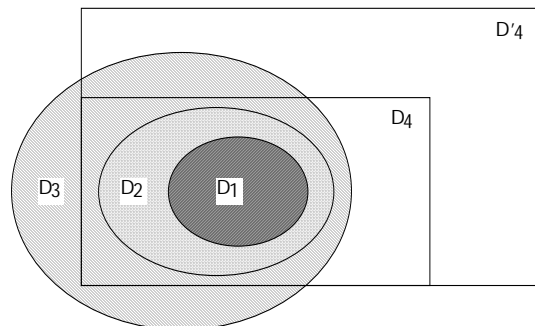


Figura 1: Modelo de niveles de utilización de un sistema informático complejo [Fischer 96]

que están disponibles para el usuario. El subconjunto de conceptos y órdenes que el usuario utiliza de forma regular y sin problemas, está contenido en D_1 . Las funcionalidades incluidas en D_2 son el subconjunto que se usa sólo de forma ocasional. El usuario normalmente no conoce todos los detalles o los efectos de su uso (como tampoco otro tipo de efectos secundarios). D_3 representa el modelo mental que posee el usuario sobre el sistema. Este modelo mental es la representación interna que tiene el usuario sobre la aplicación, es decir, lo que el usuario cree o piensa que es la aplicación y las funcionalidades que ofrece. Aquí no

sólo se incluyen suposiciones correctas sino que también existen otras erróneas. Como muestra la Figura 1, la elipse D_3 incluye partes que no están comprendidas dentro del rectángulo D_4 . La diferencia entre D_4 y D_3 son las operaciones que el sistema permite pero que son completamente ajenas al usuario.

El aumento en las capacidades y funciones proporcionadas por los entornos informáticos está provocando la evolución hacia una situación representada por D'_4 , incrementándose la porción del sistema que es desconocida para el usuario. Facilitar el aprendizaje, acceso y aplicación de las capacidades ofrecidas por el entorno permitirá que suba el porcentaje de las funcionalidades del sistema que se utilicen. También redundará en una mejora de la eficiencia y de la efectividad en la realización de las tareas.

No obstante, el proceso de adquisición de un modelo correcto y completo de la aplicación no es sencillo. Hay partes que como ya están comprendidas dentro del modelo mental del usuario (la intersección entre D_3 y D_4 , o entre D_3 y D'_4) serían más sencillas de adquirir aunque el usuario no las haya utilizado previamente. El problema surge con los conceptos de los que no tiene idea ni siquiera de su existencia y que es improbable que los pueda descubrir sin algún tipo de asistencia. Los conceptos ya presentes en el modelo mental pueden adquirirse mediante exploración libre, pero para los otros se tiene que proporcionar ayuda complementaria.

Nuestra propuesta va dirigida a que este apoyo se

proporcione mediante un SAI, que se adapte a los usuarios y que presente la información dentro de un contexto que facilite su asimilación y su comprensión. Además, esta asistencia tiene que mostrar otra información relacionada que permita un aprendizaje incremental bajo demanda, de forma que sea el usuario el que según sus necesidades decida aumentar su conocimiento del sistema.

Sistemas de ayuda inteligente (SAI)

Como el campo de los sistemas de ayuda es muy amplio, existiendo muchos enfoques, en este apartado se concreta y encuadra nuestra aproximación. Los sistemas de ayuda para aplicaciones informáticas son programas que asisten al usuario en su utilización. Por tanto su propósito principal es el de proporcionar respuestas a los problemas específicos que se presenta en el uso de dicha aplicación [Kearsley 88]. Nosotros consideramos que además de esta mejora de rendimiento deben promover también una mayor comprensión del funcionamiento del sistema (objetivo educativo) que, con el tiempo, hará disminuir la necesidad de la ayuda.

En general, los sistemas de ayuda inteligente (SAI) incorporan técnicas de inteligencia artificial y modelos explícitos, tanto del dominio como del usuario, para mejorar la ayuda proporcionada [Wasson 92]. Otros autores consideran que además un SAI debe poder tomar la iniciativa para proporcionar al usuario ayuda que no ha solicitado. Nosotros consideramos que la característica distintiva es la capacidad de contextualización de la ayuda proporcionada a las circunstancias concretas

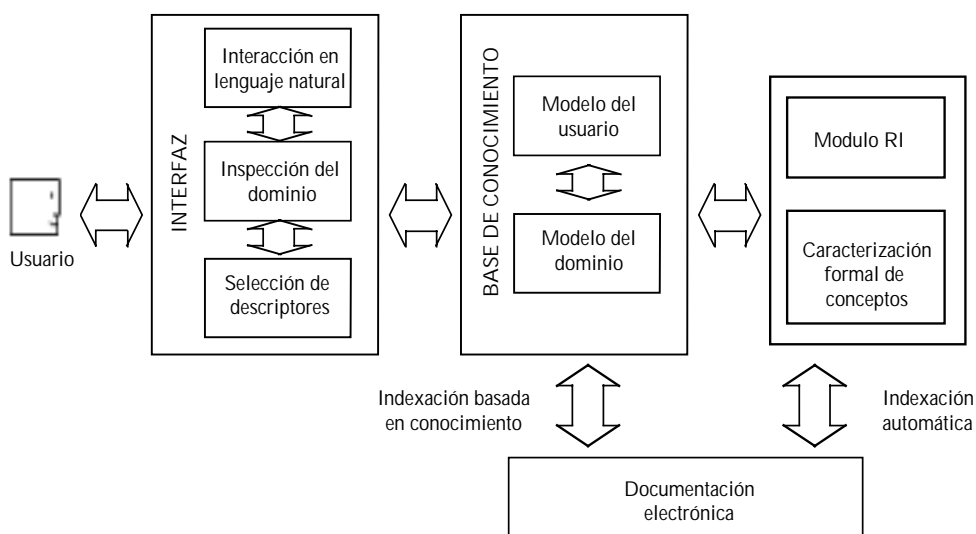


Figura 2: Arquitectura de los módulos del sistema Aran

en que se solicita (e.g. tarea, usuario). Creemos que la incorporación de un modelo explícito de usuario es imprescindible para obtener un comportamiento adaptativo, que se adecúe al usuario y a las tareas que se desean realizar.

El diseño de un SAI es un proceso complejo en el que existen muchas alternativas de diseño que influyen en sus funcionalidades, coste, escalabilidad y factibilidad en dominios reales [Winkels 92]. Las más destacables son: a) quien tiene la iniciativa en la interacción, el usuario debe solicitar la ayuda (sistema pasivo) o el SAI puede ofrecer ayuda cuando detecte errores potenciales del usuario (sistema activo); b) la integración entre el SAI y la aplicación a la que da soporte, integrado o separado, según si el SAI tiene acceso o no al estado interno de la aplicación; y c) tipo de información proporcionada y tipos de usuarios, según el propósito principal de la información que se proporciona (e.g. instructiva, procedimental, explicativa, especificativa) y si se consideran distintos tipos de usuarios según su experiencia o interés.

El objetivo fundamental del modelo que proponemos es la obtención de sistemas de ayuda eficaces, aplicables a dominios reales y que tengan un coste razonable de desarrollo y mantenimiento. Para conseguir este objetivo, nuestra propuesta es un modelo para la construcción de sistemas de ayuda pasivos, que proporcionen fundamentalmente información de referencia sobre cada aplicación

concreta y que hagan uso de un modelo de usuario que permita adecuar la información presentada a cada usuario particular. Las dos ideas en las que se basa la factibilidad del modelo son la integración en un mismo sistema de diferentes tecnologías y la reutilización de información de distinto tipo. Es decir, con respecto al último punto, se propone la creación de asistentes basados en documentación ya existente, para la que se proporcionan nuevos métodos de acceso y una mejor estructuración, mediante la utilización de distintas técnicas. Esta propuesta se ha ejemplificado mediante el SAI Aran para el sistema operativo Unix.

Aran

Aran es un SAI para el sistema operativo Unix. Cuando un usuario tiene un problema en el uso del sistema operativo puede activar Aran y mediante un interfaz multimodal expresar su necesidad. Aran simplifica el acceso, la selección y la comprensión por parte del usuario de la información necesaria para resolver dicho problema. Los dos objetivos principales del desarrollo de Aran son:

- a) demostrar la factibilidad de un SAI para un entorno informático complejo
- b) disponer de un marco de trabajo para la investigación en distintos aspectos de la ayuda interactiva, de la estructuración de información, del diseño de interfaces y del modelado del usuario.

Para disminuir el coste de desarrollo además de la

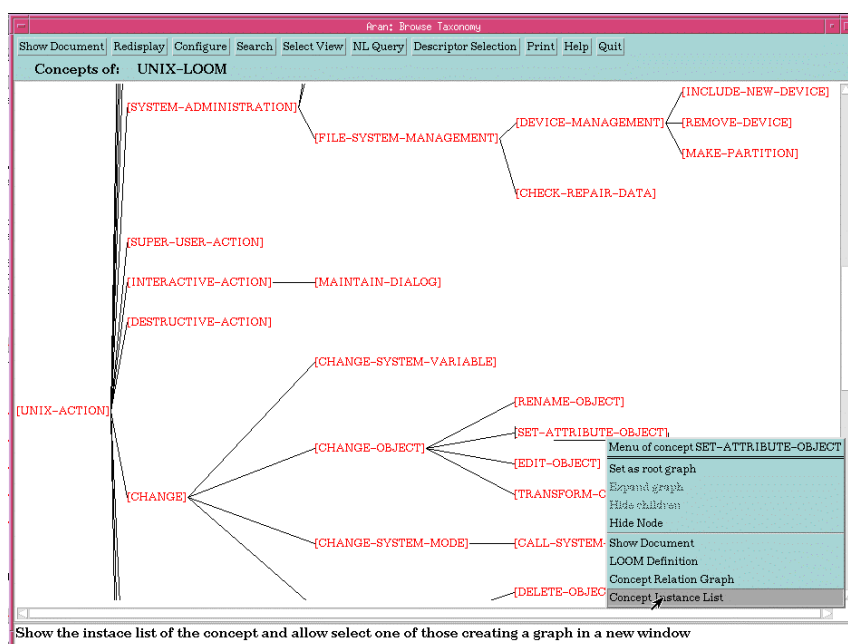


Figura 3: Interfaz de interacción con el modelo de dominio del Unix.

información específica creada para el sistema se reutilizan las páginas del manual electrónico que incorpora Unix. A continuación presentamos una descripción general del SAI haciendo énfasis en la adaptación al usuario y en modelado de usuario. Se describen los conceptos de diseño y las funcionalidades sin entrar en descripciones de detalles técnicos. Una descripción más extensa de Aran, de las tecnologías implicadas y de este enfoque para la creación de sistemas de ayuda se puede encontrar en [Fernández 96, Fernández 98].

Descripción general del sistema Aran

Aran integra diferentes tecnologías estándar para proporcionar las funcionalidades de ayuda. Estas tecnologías son: hipertexto, recuperación de información estadística (RI), análisis formal de conceptos (AFC), modelado de usuario y modelado de dominio (Figura 2). En Aran se utilizan técnicas de hipertexto como herramienta de navegación a través de la documentación de Unix y para interactuar con el modelo de dominio del sistema operativo. Las técnicas de RI y de AFC simplifican la indexación y el acceso a las páginas del manual. Finalmente la adaptación de la ayuda a cada usuario específico se realiza mediante la información contenida en el modelo del usuario.

El núcleo de sistema está formado por la base de conocimiento (compuesto principalmente por el modelo del dominio y por el modelo del usuario) donde se representa, mantiene y organiza distinto tipo de conocimiento e información. El modelo del dominio es un modelo conceptual del Unix que trata de reflejar su información de diseño. Este modelo permite la organización e indexación de diferente información alrededor de los conceptos claves del dominio. Se ha partido de un modelo anterior similar al del Sinix Consultant [Hecking 88], en el que los conceptos del Unix se distinguen entidades, que corresponden con objetos Unix (e.g. fichero, proceso) y acciones o operaciones que involucran a dichos objetos (e.g. cambio, comunicación con un usuario). Estos conceptos se organizan en una taxonomía (jerarquía con herencia) representando que hay objetos o acciones más generales que otros. Las órdenes y los objetos concretos de Unix se representan como instancias o ejemplares de las acciones y entidades respectivamente. En la representación de las órdenes se tienen en cuenta distintos aspectos: sintácticos (e.g. nombre, sintaxis de uso), semánticos (e.g. texto descriptivo de su funcionalidad) y pragmáticos/tutoriales (e.g. órdenes relacionadas, conceptos prerequisites o relacionados).

Concretamente la documentación textual se indexa de tres formas diferentes:

- a) *Indexación basada en conocimiento*: los documentos se indexan manualmente utilizando los conceptos del modelo de dominio creado por un experto.
- b) *Indexación estadística*: los documentos se indexan mediante términos extraídos automáticamente de los textos.
- c) *FCA o indexación por atributos*: los documentos se indexan mediante un conjunto de descriptores o atributos.

El diseño del interfaz gráfico de usuario de Aran trata de simplificar el acceso y la comprensión de la información sobre el dominio proporcionando tres métodos o modos de interacción distintos pero relacionados. En la *inspección de dominio* se proporciona una representación del modelo conceptual del sistema proporcionado por un experto con la que se puede interactuar directamente (e.g. menús, ratón) para acceder a la información y documentación del dominio (Figura 3). Esta interacción del usuario con el modelo del dominio le ayuda a obtener una visión más completa y correcta del sistema operativo. En el modo de *interacción en lenguaje natural* o de pregunta, el usuario puede expresar su solicitud de información mediante un texto libre obteniendo como respuesta una lista ordenada por relevancia de los documentos relacionados. En el modo de *selección de descriptores*, el usuario elige incrementalmente de una lista (proporcionada por Aran) los descriptores que mejor se ajustan a su necesidad de ayuda obteniendo los documentos donde dichos descriptores están presentes. Si el usuario selecciona un documento utilizando el modo de selección de descriptores o de pregunta y cambia al modo de inspección del dominio, la visualización de dominio estará centrada alrededor de los conceptos que indexan dicho documento.

Adaptación al usuario

Como se hemos descrito Aran contiene y organiza mucha información de distintos tipos sobre el dominio del Unix y sus órdenes. Para obtener el doble propósito de la asistencia es necesario que la ayuda presentada se adapte a las características concretas de cada usuario. Esta adaptación se realiza mediante el modelo de usuario que almacena la información y suposiciones sobre cada individuo que pueden ser relevantes para ayudarlo a asimilar esta información compleja. A diferencia de otros modelados (e.g. de estudiante) su propósito es únicamente mejorar la interacción y no pretende capturar el estado de conocimiento de cada persona.

Concretamente esta adaptación a las necesidades del usuario se hace mediante:

- a) *Mejora de la interacción en lenguaje natural o texto libre.* Se tienen en cuenta no sólo la pregunta actual sino también las anteriores así como la información sobre la relevancia o interés para el usuario de los documentos recuperados previamente. De esta forma Aran puede filtrar documentos irrelevantes.
- b) *Ordenación por relevancia de la información recuperada.* De acuerdo con esta ordenación Aran presenta al usuario primero la información nueva (no presentada antes) y que se supone por el modelo de usuario que es desconocida. A continuación los documentos que se supone que el usuario puede conocer y finalmente aquellos que ya han sido presentados en esa sesión. De esta forma Aran trata de no repetir la presentación de información.
- c) *Modificación de la visualización del modelo de dominio.* Cuando el usuario accede a una instancia (e.g. orden de Unix) Aran trata de adaptar su presentación según el conocimiento esperado del usuario. La visualización se modifica en función de los elementos conocidos o no por el usuario. Por ejemplo, si se supone que el usuario está familiarizado con la orden no se le presentan los ejemplos de uso. Tampoco se presentan directamente aquellos conceptos prerequisite o relacionados que se supone que ya conoce.

No obstante como el filtrado puede producir efectos indeseados de ocultación de información, el usuario siempre tiene una posibilidad de obtener la

información filtrada. El usuario también puede desactivar la adaptación automática.

Modelado del usuario

Las peculiaridades de la asistencia a los usuarios de aplicaciones informáticas nos ha llevado a centrarnos en modelos de rápida adquisición inicial, sin preguntas directas al usuario, que posteriormente se refina y actualiza durante la interacción con el sistema [Kobsa 89]. También es necesario que dicho modelo sea capaz de predecir el conocimiento del usuario a partir de información parcial. Esto nos ha hecho decantarnos por un modelo de usuario individual y dinámico basado en *estereotipos* [Rich 89]. Esto permite clasificar inicialmente al usuario para posteriormente, y según se interactúa con Aran, completar y actualizar dicha información.

Estereotipos y modelado individual del usuario

El modelado mediante estereotipos se basa en la identificación de grupos de usuarios que tengan ciertas características homogéneas respecto a la aplicación. Para cada estereotipo se identifican un pequeño número de características clave que permiten al sistema identificar si para el usuario se puede clasificar o no dentro de un determinado grupo. De esta forma a un usuario concreto se le pueden aplicar uno o varios estereotipos. Los estereotipos son muy importantes al principio de la sesión ya que permiten proporcionar un

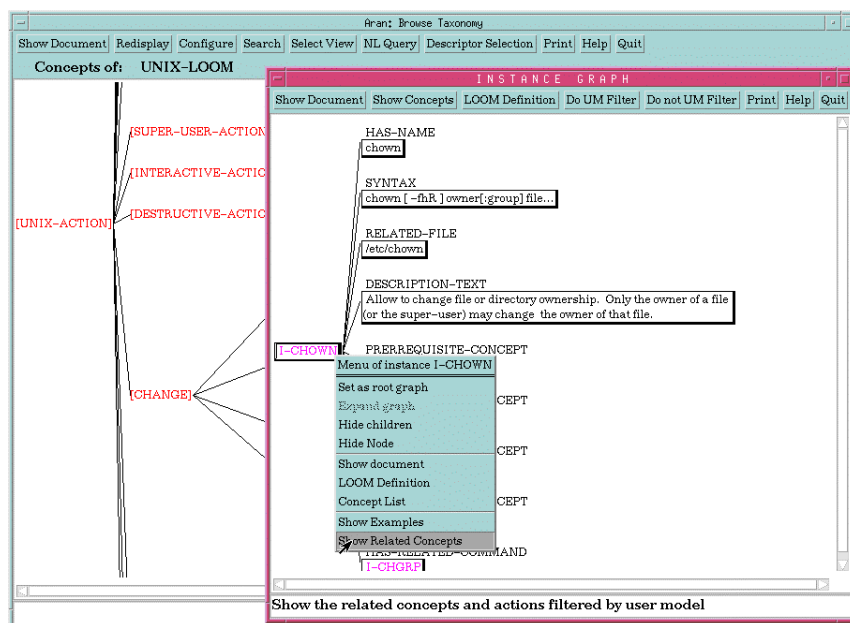


Figura 3: Ejemplo de adaptación de información en la visualización del modelo de dominio. Se han filtrado los ejemplos y los conceptos relacionados de la orden *chown*.

conocimiento por omisión sobre el usuario que posteriormente se actualizará con los datos obtenidos directamente en la interacción.

Los estereotipos se clasifican en una jerarquía con herencia formando un grafo acíclico. Estos estereotipos tratan de representar los intereses y grado de experiencia de los usuarios con respecto a diferentes contextos y subdominios del Unix.

La raíz de la jerarquía es el estereotipo *user-prototype*, en el que se define la plantilla del conocimiento que se va a tener en cuenta para modelar a un usuario. El resto de estereotipos incluyen las características típicas de usuarios especializando los contenidos heredados de prototipos más generales. En cada estereotipo se representan explícitamente las suposiciones aplicables a cualquier usuario de ese subgrupo, así como las relaciones entre estereotipos. Por ejemplo, a un usuario no se le puede clasificar a la vez como novato y experto en el uso de la red, por tanto son estereotipos excluyentes entre sí. En la figura 4 se presenta la jerarquía básica de estereotipos de Aran.

En el modelo se consideran tres tipos distintos de información:

- a) *Información inicial*, datos obtenidos directamente del sistema Unix.
- b) *Hechos objetivos*, datos adquiridos dinámicamente de la interacción con el usuario
- c) *Hechos subjetivos*, suposiciones obtenidas a partir de la interacción.
- d) *Características clave*, datos que permiten determinar la asignación al estereotipo.

La información inicial almacena datos como el nombre, identificación y grupo del usuario, junto con el histórico de interacción con el sistema. Los hechos objetivos contienen las palabras utilizadas en

consultas anteriores y los conceptos, documentos o componentes a los que se ha accedido. Los hechos subjetivos representan las suposiciones respecto a los conceptos y ordenes conocidas o desconocidas por el usuario. Un modelo de usuario concreto heredará inicialmente las suposiciones y las características clave de los estereotipos que le sean aplicables.

Creación, adquisición y mantenimiento del modelo

La adquisición inicial del modelo es automática e implícita a partir de la información obtenida directamente del sistema Unix. A partir de la identificación y del histórico de interacción Aran crea una instancia de la plantilla modelo (el modelo individual para ese usuario) y determina los estereotipos se le pueden aplicar inicialmente.

En la aplicación de los estereotipos iniciales hay dos posibilidades. La primera es que por alguno de los datos iniciales se le pueda asignar directamente alguno de los estereotipos. Por ejemplo, si el usuario tiene número de identificación cero se le clasifica como superusuario del sistema y también se podría inferir que es un usuario experto y que es un usuario programador (Figura 4). La segunda posibilidad es su asignación mediante heurísticas en función del histórico y de los contenidos de las características clave y de los hechos subjetivos de cada estereotipo. Por ejemplo, si un usuario ha utilizado alguna de las herramientas de programación en lenguaje C se le podría clasificar como programador en dicho lenguaje.

Una vez determinados los estereotipos iniciales, Aran completa dinámicamente el modelo individual para esa sesión. A partir de la interacción se van adquiriendo los hechos objetivos (e.g. documentos

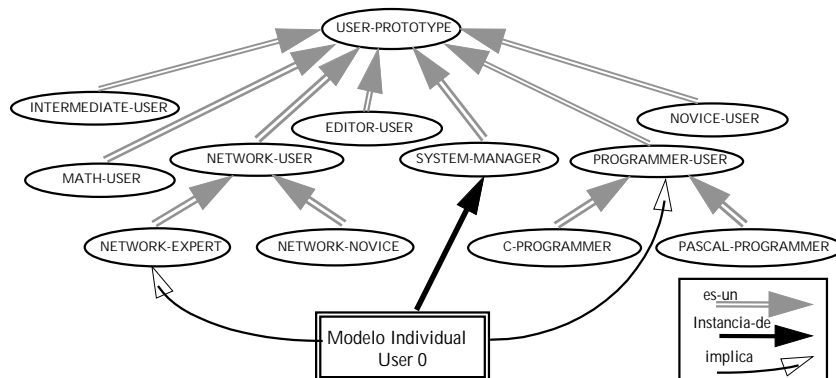


Figura 4: Jerarquía de estereotipos en el dominio Unix

mostrados, palabras utilizadas en las consultas). Utilizando estas mismas interacciones y aplicando heurísticas propias del dominio se derivan suposiciones sobre que elementos (e.g. órdenes, conceptos) son conocidos o desconocidos. Por ejemplo, si un usuario accede a los ejemplos de uso de una orden se supone que previamente no la conocía.

Todos estos nuevos hechos objetivos y subjetivos van siendo incorporados en el modelo pudiendo producirse incoherencias. Si la incoherencia es por tener información contradictoria se corrige con una estrategia conservadora. Por ejemplo, si un concepto aparece como conocido y desconocido al mismo tiempo, se elimina de los conceptos conocidos. Si la incoherencia proviene (por herencia) de la asignación de un estereotipo a un usuario se retracta dicha clasificación.

La adquisición de nuevos datos tiene que producir también una reevaluación de los estereotipos aplicados al usuario mediante heurísticas. De esta forma Aran tiene en cuenta los cambios de interés o de conocimiento del usuario y los refleja asignándole nuevos estereotipos.

Conclusiones y trabajo futuro

En la sociedad de la información la adquisición de nuevas habilidades y el aprendizaje de nuevas formas de trabajo es un proceso continuo e integrado en el mundo laboral. Los sistemas educativos inteligentes deben simplificar estos procesos.

Con este propósito, y dentro del campo de las aplicaciones informáticas, nosotros proponemos los SAI como una forma de proporcionar asistencia y enseñanza bajo demanda de los usuarios para mejorar su rendimiento. Las técnicas de modelado de usuario son claves para lograr la contextualización y adaptación de la ayuda a los intereses y conocimientos del usuario mejorando por tanto la efectividad de la asistencia. El sistema Aran es una ejemplificación de como, integrando distintas tecnologías y reutilizando información, se pueden aplicar estos principios a un dominio real con un esfuerzo de desarrollo admisible.

En el desarrollo de Aran se detectaron algunos problemas, como la granularidad de la información documental presentada (sólo se podían presentar documentos completos) o la dificultad de la reutilización y mantenimiento de la información textual existente. Proponer soluciones a estos aspectos nos condujo a los lenguajes de marcado (SGML, XML) [Golfard 90], que actualmente han

tenido tanto impacto con el desarrollo de Internet. Ahora nuestros esfuerzos se centran más en como simplificar el diseño, desarrollo y mantenimiento de aplicaciones educativas (e.g. módulos multimedia o hipermedia) que en los aspectos de adaptación o modelado mediante bases de conocimiento [Fernández 97, Fernández-Valmayor 00]. Como los sistemas educativos son aplicaciones muy complejas, en las que intervienen distintos expertos (e.g. profesores, desarrolladores, diseñadores) y en las que durante su desarrollo se realizan continuos cambios, creemos que es necesario hacer un esfuerzo en la construcción de herramientas que simplifiquen el diseño, la gestión y actualización de los contenidos [Navarro 00, Sierra 00].

Además la difusión de estas tecnologías de marcado que ha impuesto Internet hace que se generalicen las herramientas para el diseño y gestión eficiente de la información (e.g. editores, parsers, kits de desarrollo) a la que posteriormente se puede dar distinto tratamiento (e.g. multimedia, publicación web). La combinación de las técnicas de lenguajes de marcado junto con las de modelado de usuario creemos que dará lugar a nuevas aplicaciones educativas más cooperativas y centradas en las necesidades de los usuarios.

Agradecimientos

A los miembros del grupo de Ingeniería del Software e Inteligencia Artificial (ISIA) del Departamento de Sistemas Informáticos y Programación de la Universidad Complutense de Madrid con cuya colaboración se ha desarrollado este trabajo.

Este trabajo ha sido parcialmente financiado por el proyecto europeo Galatea (TM-LD-1995-1-FR89) y por los proyectos CYCIT (TIC97 2009-CE y TIC98-0733)

Referencias

Duffy, T.M., Palmer, J., Mehlenbacher B., 1992. *On Line Help Design and Evaluation*. Ablex Publishing Corporation. Norwood, New Jersey.

Fernandez-Manjon, B., 1996. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*. Tesis Doctoral, Universidad Complutense de Madrid.

Fernandez-Manjon, B., Fernández-Valmayor, A. 1998. Building Educational Tools based on Formal Concept Analysis. *Education and Information Technology*, 3(3), pp. 187-201.

Fernandez-Manjon, B., Fernandez-Valmayor, A., Fernández Chamizo, C. 1998. Pragmatic User Model Implementation in an Intelligent Help System. *British Journal of Educational Technology*, 29(2), pp. 113-123.

Fernandez-Valmayor A., Lopez-Alonso C., Sere A., Fernandez-Manjon B., 2000. A hypermedia design for learning foreign language text comprehension. S. D. Franklin, E. Strenski (Eds) *Building University Electronic Educational Environments*, Kluwer Academic Publishers, pp 51-66.

Fischer, G., (en prensa). Lifelong Learning - More Than Training. Mizoguchi, R., Kommers, A. M. (Eds) Special Issue on "Intelligent Systems/Tools in Training and Life-Long Learning". *International Journal of Continuing Engineering Education and Life-Long Learning*.

Fischer, G., 1999. User Modeling: The Long and Winding Road. *Proceedings of the Seventh International Conference on User Modeling (UM99)*, Banff, Canada, 20-24 June 1999

Half, H. 1998. (ed) *Intelligent Tutoring Systems'98*, Lecture Notes in Computer Science 1452, Springer-Verlag, Berlin

Hecking M *et al.* (1988) *The SINIX Consultant - A Progress Report* Memo Nr.28, KI-Labor, Informatik IV, Universitaet des Saarlandes, Saarbruecken, Germany.

Kearsley, G., 1988. *Online Help Systems: Design and Implementation*. Ablex Publishing Corporation, Norwood, New Jersey, USA.

Kearsley, G., 2000. *Explorations in Learning & Instruction: The Theory Into Practice Database*. School of Education and Human Development , George Washington University, Washington. (URL, <http://www.gwu.edu/~tip>)

Kobsa A and Whalster W (1989) (eds) *User Modelling in Dialog Systems* Springer, Berlin.

Kok A J (1991) A Review and Synthesis of User Modelling in Intelligent Systems *The Knowledge Engineering Review* 6 (1) 21-47.

Navarro A., Fernandez-Manjon B., Fernandez-Valmayor A., Sierra J.L., 2000. A Practical Methodology for the Development of Educational Hypermedias. *Proceedings of ICEUT 2000, 16th IFIP World Computer Congress 2000, Information Processing Beyond Year 2000*.

Rich E, 1989. Stereotypes and User Modelling in Kobsa A and Wahlster W (eds) *User Models in Dialog Systems* Springer, Berlin.

Sierra J.L, Fernandez-Manjon B., Fernandez-Valmayor A., Navarro A., 2000. Integration of Markup Languages, Document Transformations and

Software Components in the Development of Applications: the DTC Approach. *Proceedings of ICS 2000, 16th IFIP World Computer Congress 2000, Information Processing Beyond Year 2000*.

Sleeman, D & Brown, J.S. (Eds), 1982. *Intelligent Tutoring Systems*, Academic Press Inc.

Wasson B and Akselsen S, 1992. An Overview of On-line Assistance: from On-line Documentation to Intelligent Help and Training *The Knowledge Engineering Review* 7 (4) 289-322.

Wenger, E., 1987. *Artificial Intelligence and Tutoring Systems*, Morgan Kaufman Publishers Inc., Los Altos, CA.

Winkels R, 1992. *Explorations in Intelligent Tutoring and Help* IOS Press, Amsterdam.