Ingeniería y Competitividad
ISSN: 0123-3033
inycompe@gmail.com
Universidad del Valle
Colombia

Florian, Beatriz E; Valencia, María E.; Rodríguez, Paola J.; Millán, Marta; Gaona, Carlos M.; Carrillo, Javier E.; Ciprián, Mauricio

Diseño de una plataforma experimental para la búsqueda y recuperación de documentos en una biblioteca digital

Ingeniería y Competitividad, vol. 9, núm. 2, 2007, pp. 105-117
Universidad del Valle
Cali, Colombia

Ingeniería y Competitividad, Volumen 9, No. 2, p. 105 - 117 (2007)

INGENIERÍA DE SISTEMAS

Diseño de una plataforma experimental para la búsqueda y recuperación de documentos en una biblioteca digital

Beatriz E. Florián*, María E. Valencia*, Paola J. Rodríguez*, Marta Millán*, Carlos M. Gaona* , Javier E. Carrillo*, Mauricio Ciprián*

** Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Cali, Colombia*
§ e-mail: bflorian@eisc.univalle.edu.co

(Recibido: Mayo 5 de 2007 - Aceptado: Noviembre 13 de 2007)

Resumen

Recuperar información implica representar, organizar y almacenar la información para facilitarle al usuario acceder a ella. Sin embargo, caracterizar la información que el usuario necesita es un problema complejo. Comúnmente, no se puede usar, directamente, una descripción completa de las necesidades de información del usuario sino palabras clave. Por tanto, los mecanismos tradicionales de búsqueda y recuperación, a partir del conjunto de palabras clave dado, devuelven grandes cantidades de información que generan al usuario la necesidad de extraer la que le sea relevante. Los sistemas de personalización, particularmente los de recomendación, son una alternativa de solución. En este artículo se describe la arquitectura de PREDICA, una plataforma experimental que combina sistemas avanzados de consulta y de recomendación, interfaces adaptativas, bodegas de datos y descubrimiento de conocimiento, aplicados a una biblioteca digital. PREDICA integra estrategias de pre-procesamiento, para mejorar la búsqueda de información y estrategias de personalización para recomendar documentos. Una de las más importantes características de PREDICA, con relación a otros sistemas de software para bibliotecas digitales de código abierto como *D-Space*, *Eprints* y *Greenstone*, es que integra un sistema de recomendación para proveer información útil al usuario.

Palabras clave: Biblioteca digital, Sistemas de recomendación, Catalogación de documentos digitales, Tecnologías web 2.0, Recuperación de información, Interfaces adaptativas.

SYSTEMS ENGINEERING

Design of an experimental platform for the search and retrieval of documents in a digital library

Abstract

Information retrieval (IR) deals with representation, organization, and storage of information. IR should provide the user with an easy access to the information in which he or she is interested. However, characterization of the user information needs is a complex problem. Usually, a full description of the user information needs cannot be made directly and a set of keywords should rather be used. Thus, given a set of keywords, an IR system retrieves information, which may be relevant to the user. However, traditional IR systems overload recovery information. Personalization systems, in particular, the recommender ones, have been proposed in order to mitigate the impact of an enormous amount of retrieved information on the user. This paper describes the architecture of PREDICA, an experimental platform for combining advanced query and recommender systems, adaptive interface, data warehouse and knowledge discovery on database in a digital library. PREDICA uses a pre-processing strategy for improving the search of information and personalization strategies for documents' recommendation. Among other open source digital library research projects, like *D-Space*, *Eprints*, and *Greenstone*, PREDICA has an integrated recommender system to provide helpful information to the users.

Keywords: Digital library, Recommender systems, Digital documents cataloguing, Web 2.0 technologies, Information retrieval, Adaptive interfaces.

1. Introducción

La red mundial de información (WWW, de *World Wide Web*) ha generado un vertiginoso crecimiento en la cantidad de información y en el número de usuarios y de computadores conectados a ella. El gran volumen de información generada y almacenada en dispositivos digitales está ahora más que nunca al alcance de todos. Sin embargo, su crecimiento ha hecho también que los usuarios tengan que enfrentarse a la dificultad de recuperar información acorde con sus necesidades. Debido fundamentalmente a la falta de una semántica en los buscadores, normalmente el usuario se ve enfrentado a que buena parte de los resultados de una consulta sobre un tema en particular no satisface sus necesidades. Esto genera la necesidad de acceder a cada documento recuperado para determinar su utilidad.

Por otra parte, el estado del arte de las nuevas tecnologías de información exige de los usuarios tener conocimientos y habilidades para utilizar diferentes equipos periféricos, comprender menús habitualmente poco amistosos y diversos tipos de interfaces y sistemas operacionales. A pesar del avance significativo en el dominio de las interfaces de usuario y de la preocupación de los creadores de sistemas de la información en mejorar su facilidad o capacidad de ser utilizadas (*usability*) (Nielsen & Loranger, 2006; Lazar, 2005), existen discrepancias entre la representación cognitiva que los usuarios tienen de la tarea y las características y funcionalidades de la imagen de la interfaz. Todavía estamos lejos de sistemas que sean inmunes a dificultades de aprendizaje de parte de quien los utiliza (Cunha, 1999). Ésta no es una característica ajena a las bibliotecas digitales. Particularmente, cuando se trata de bibliotecas y debido a que el costo del hardware es menor, las tasas de utilización de los sistemas suelen ser bajas y pocos o ningún mecanismo de prevención de errores y de ayuda está disponible. Por tanto, la recuperación de la información se vuelve más difícil para un usuario, debido generalmente a la falta de conocimiento sobre la forma en la que debe preguntar para obtener los resultados esperados disminuyendo la necesidad de interpretar, clasificar, priorizar o filtrar grandes volúmenes de información retornados por el sistema (Kafure , 2004).

Se han propuesto muchas estrategias basadas en semántica (Middleton et al., 2003; Middleton et al., 2004; Kurki et al., 1998; Gómez et al., 2003; Corcho, 2006), sin que aun sea posible resolver el problema. Otras propuestas de solución que se destacan tienen relación con los mecanismos y sistemas de filtrado de información (*information filtering*) (Sugiyama et al., 2004; Abbattista et al., 2002; Eirinaki et al., 2004), con técnicas de recuperación de información (*information retrieval*) (Baeza & Ribeiro, 1999; Bollacker et al., 1998; Pazzani & Billsus, 1997) y con sistemas de recomendación (*recommender systems*) (Balabanovic & Shoham, 1997; Chen & Chen, 2001; Smeaton & Callan, 2005; Goecks & Shavlik, 2000; Herlocker et al, 2000; Linden et al., 2003; Miller et al., 2003; Mobasher et al., 2002; O'Donovan & Smyth, 2005; Rashid et al., 2002), entre otros.

Algunas estrategias se apoyan en procesos de personalización del sitio web a las necesidades específicas del usuario, teniendo en cuenta el conocimiento adquirido del análisis de su conducta al navegar, de otro tipo de datos de estructura y contenido del sitio y del perfil de usuario (Mobasher et al., 2002). Generalmente, estas estrategias hacen uso de técnicas de minería de datos en las que se incluyen tareas de clasificación, agrupación por cúmulos (*clustering*) y asociación.

En este artículo se presenta PREDICA, una herramienta de software desarrollada como alternativa para trabajar en la solución de problemas relacionados con la búsqueda y la recuperación de información. PREDICA integra una biblioteca digital con estrategias de minería de datos y modelos de recomendación, convirtiéndose en un poderoso instrumento de experimentación permanente de tecnologías de almacenamiento y recuperación de información orientada al usuario. Se trata de un desarrollo tecnológico que facilita la búsqueda y recuperación de documentos, teniendo en cuenta el perfil de navegación de un usuario. Su desarrollo mediante el uso de software libre y utilizando la red (WWW) como interfaz de comunicación, garantiza, en alguna medida, contar con una herramienta de bajo costo, útil, novedosa y accesible de manera remota.

Este artículo está organizado en secciones. En la Sección 2, se describe la arquitectura de PREDICA haciendo énfasis particular en las vistas tecnológica y funcional. Los módulos funcionales que integran PREDICA se detallan en esta sección. En la Sección 3, se presenta el modelo de recomendación propuesto e implementado que diferencia a PREDICA de otras bibliotecas digitales. Finalmente, en la Sección 4, se presentan algunas de las experiencias recogidas durante su desarrollo.

2. Arquitectura de PREDICA

La arquitectura de un software es la organización básica del sistema representada en sus componentes, en las relaciones entre ellos, en el ambiente y en los principios que orientan su diseño y evolución. Generalmente se presenta a través de diferentes vistas. Una de ellas se puede representar mediante un modelo de capas, donde cada capa incluye componentes agrupados por paquetes, representando la forma en la cual la aplicación se estructura para gestionar la interacción del usuario, las tareas de procesamiento internas, la navegación y la presentación del contenido.

Los componentes de la arquitectura de la aplicación PREDICA están organizados en cuatro capas: presentación (GUI), distribución (servidor web), lógica de la aplicación y almacenamiento persistente. La capa de presentación, permite la comunicación con los usuarios mediante interfaces web que se despliegan usando un examinador (*browser*) y aprovecha los recursos de la máquina del cliente para hacer algunos procesamientos locales del entorno de interfaz. La capa de distribución, se encarga de recibir todas las peticiones del usuario, verificar que exista el recurso (archivo) y ceder el control a la lógica de la aplicación; también es responsable por retornar al usuario la información solicitada. La capa de la lógica procesa las peticiones de todos los usuarios del sistema PREDICA y administra el contenido de la información, autorizaciones, historiales (*web logs*), etc. La capa de almacenamiento contiene los metadatos y los archivos digitales de la información asociada con los documentos, y la información de los usuarios.

En las siguientes secciones se presentan una vista tecnológica y una vista funcional de la arquitectura de PREDICA.

2.1 Vista tecnológica de PREDICA

La Figura 1 muestra, en un esquema de cuatro capas, los componentes tecnológicos utilizados en PREDICA. En particular, *Ajax* (Eichorn, 2006) y el *framework Qooxdoo* (Qooxdoo, 2007), HTML, XML y CSS se utilizan en el desarrollo de la interfaz web dinámica de usuario, que se ejecuta del lado del cliente.

Figura 1. Una vista tecnológica de la arquitectura de PREDICA.

La capa de distribución es la puerta de enlace (*gateway*) entre la interfaz con el usuario y la aplicación, y está constituida por el servidor web apache que reside en una máquina anfitriona (*host*).

Para la creación de la lógica de la aplicación, que reside en una máquina anfitriona, se utilizó *+XML, PHP, C, Java* y la plataforma (*framework*) de desarrollo *Symfony* (The Symfony Team, 2007).

En la capa de almacenamiento aparece la base de datos operacional de la biblioteca (BD) y la bodega de datos (DWH), implementados con *PostgreSQL*, y un repositorio de archivos digitales. Estos componentes del almacenamiento residen en otra máquina anfitriona.

La BD almacena los metadatos de los documentos de la biblioteca [especificados de acuerdo con el estándar *Dublin Core* (CDP Meta Data Working Group, 2006), y la información relacionada con usuarios y áreas, entre otros. El DWH almacena la historia de navegación y las preferencias de los usuarios. El repositorio almacena los archivos digitales (e.g. texto, audio, video, gráfico y ejecutables) asociados a los metadatos antes mencionados.

2.2. Vista funcional de la arquitectura de PREDICA

La plataforma experimental PREDICA, utiliza una biblioteca de documentos digitales en la que se pueden llevar a cabo tareas de catalogación, búsqueda, consulta y recuperación de documentos, mediante una interfaz adaptable. La plataforma integra también estrategias de recomendación de documentos, basadas en procesos de minería de datos aplicados sobre datos e información de conducta navegacional de los usuarios.

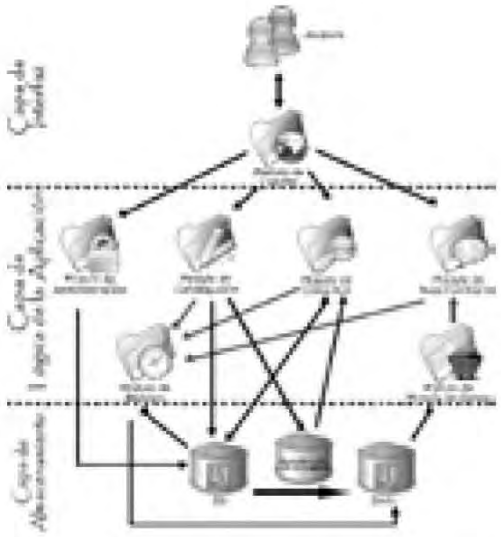

Figura 2. Vista funcional de la arquitectura de PREDICA.

Funcionalmente, PREDICA integra los módulos de interfaz, catalogación, administración, consulta, bitácora, bodega de datos, minería y recomendación y las bases de datos BD, DWH tal y como se muestra en la Figura 2.

2.2.1 Módulos de catalogación y administración

La catalogación es la descripción estandarizada de un recurso bibliográfico que, para el caso particular de la biblioteca digital, consta de metadatos para los documentos digitales, permitiendo simplificar su manejo y facilitando su búsqueda y recuperación.

El *módulo de catalogación* permite catalogar y gestionar documentos digitales, y es uno de los componentes básicos de la biblioteca digital con base en el estándar *Dublin Core* extendido (CDP Meta Data Working Group, 2006), compuesto por metadatos y cualificadores.

El *módulo de administración* gestiona los usuarios de la BD con base en la definición de tipos y perfiles. La información de los usuarios incluye datos personales, áreas de interés y datos demográficos. Cada tipo de perfil determina privilegios. Algunos de los perfiles definidos corresponden a administrador, docente, usuario registrado y usuario no registrado.

2.2.2 Módulo de consultas

El módulo de consultas gestiona los mecanismos mediante los cuales los usuarios recuperan documentos de la biblioteca digital. PREDICA ofrece dos tipos de consulta: básica y avanzada, que se diferencian fundamentalmente por el tipo de interfaz. La consulta avanzada le ofrece al usuario más información (filtros) para restringir y optimizar el proceso de búsqueda. La consulta general, cuya interfaz se muestra en la Figura 3, utiliza un campo de búsqueda que reconoce operadores lógicos como AND para indicar que los documentos a recuperar deben incluir todas las palabras escritas, OR para indicar que los documentos a recuperar pueden contener alguna de las palabras escritas y, " " para indicar la búsqueda de frases completas en los documentos a recuperar.

Los documentos recuperados por el sistema se despliegan al usuario permitiéndole acceder adicionalmente a metadatos sobre información general, resumen, derechos de autor y referencias.

La consulta avanzada ofrece dos mecanismos para restringir la búsqueda: utilizar un árbol jerárquico que representa la taxonomía de áreas de conocimiento de la biblioteca como se muestra en la Figura 4, o múltiples opciones de filtrado como se muestra en la Figura 5.

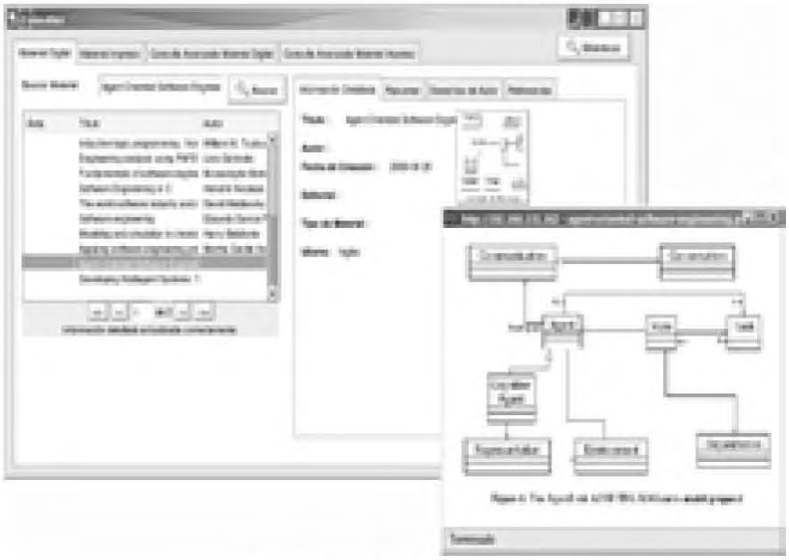

Figura 3. Interfaz de consulta general.

Figura 4. Interfaz de consulta avanzada que utiliza taxonomía de áreas de conocimiento.

Figura 5. Interfaz de consulta avanzada que utiliza múltiples opciones de filtrado.

Entre los filtros que se ofrecen al usuario están: tipo de material (libro, revista, tesis, informe, etc.), idioma, autor, formato (*pdf*, audio, texto, video etc.) y fecha de publicación.

La Tabla 1 muestra resultados de la aplicación de pruebas de esfuerzo al módulo de consultas con el propósito de medir la capacidad de respuesta y el nivel de desempeño del sistema con la arquitectura de hardware actual (una máquina anfitriona *Pentium Dual Core* de 3 GHz y 1 GB de RAM usada para alojar el servidor web y la aplicación del sistema y otra máquina anfitriona *Pentium* de 2.6 GHz con 2 procesadores *Dual Core* y 4 GB de RAM). Para ello se utilizó la herramienta SIEGE (SIEGE, 2007). De acuerdo con los resultados obtenidos, el sistema es capaz de atender adecuadamente hasta 400 transacciones simultáneas por segundo.

Tabla 1. Resultados de pruebas de esfuerzo y rendimiento del sistema de consulta de PREDICA.

	Número de transacciones (peticiones)	Tiempo transcurrido (s)	Datos tranferidos (B)	Tiempo de respuesta (s)	Tasa de transacción (trans / s)	Rendimiento (B / s)	Concurrencia (peticiones)	Transacciones exitosas	Transacciones fallidas
10 usuarios 5 operaciones	50	9.721	0	1.126	5.162	0	5.815	50	0
10 usuarios 10 operaciones	100	18.248	0	1.174	5.482	0	6.441	100	0
10 usuarios 20 operaciones	200	35.629	1	1.166	5.617	0.03	6.546	200	0
20 usuarios 20 operaciones	400	68.851	3	2.648	5.811	0.04	15.388	400	0
30 usuarios 20 operaciones	598.5	102.587	4	4.019	5.837	0.04	23.474	598.5	1.5
40 usuarios 20 operaciones	795.5	135.433	6	5.54	5.875	0.04	32.541	795.5	4.5
60 usuarios 20 operaciones	1185.2	204.433	9	8.808	5.806	0.044	51.149	1185.2	14.8
80 usuarios 20 operaciones	33	482.153	0.2	10.36	0.064	0	0.66	33	1102.7

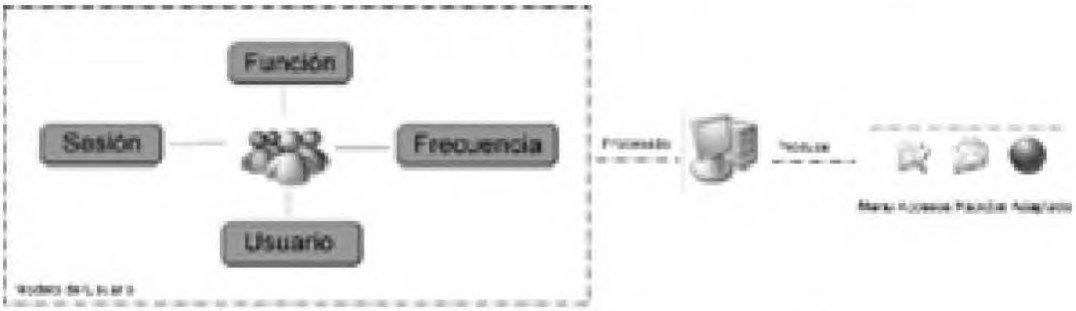

Figura 6. Modelo de usuario para una adaptación por navegación.

A partir de este número, de acuerdo con las pruebas, su rendimiento se empieza a degradar.

2.2.3 Módulo de interfaz

Este módulo gestiona un modelo de adaptabilidad, que se fundamenta en la adaptación de navegación (Brusilovsky & Maybury, 2002). Mediante un modelo de usuario (Figura 6), almacenado en forma persistente en la base de datos de la biblioteca, se personaliza el menú de accesos rápidos que se adapta con base en la frecuencia con la cual el usuario utiliza las funciones. El modelo de usuario permite especificar y almacenar la información relacionada con los intereses del usuario representados en la opción del menú de navegación que seleccione. El modelo de usuario tiene en cuenta tres características: la función o representación de cada una de las opciones de navegación o vínculos (*links*), la frecuencia o grado de uso de una determinada función y la sesión, que corresponde al número de visitas a la biblioteca realizadas por el usuario. Las interfaces desarrolladas utilizan tecnología Web 2.0, (e.g. interfaces RIA con AJAX), ofreciendo ventajas con relación a mecanismos de refrescamiento (*refreshing*) y de facilidad o capacidad de uso. De esta manera, se necesita solamente refrescar las zonas de la página que contienen nueva información (a diferencia del refrescamiento total de la página que se debe realizar en las interfaces tradicionales) y se enriquece la interfaz integrándole componentes tales como menús de texto, barras de íconos, árboles desplegables, pestañas, barras deslizantes, tablas interactivas y ventanas flotantes. Esto hace que las interfaces de consulta tengan el aspecto de una aplicación de escritorio y faciliten y agilicen el trabajo del usuario.

Figura 7. Entorno integrado y enriquecido de herramientas en PREDICA.

Adicionalmente, la interfaz provee al usuario un entorno enriquecido e integrado de herramientas tales como: ventanas a diccionarios de términos, de sinónimos, de antónimos, traductores, buscadores web, bibliotecas privadas con las que existe convenio institucional, editor de texto, calculadora y hoja de cálculo como se muestra en la Figura 7.

La evaluación de los aspectos de facilidad o capacidad de uso de PREDICA, se hizo antes de su publicación en la web, con la finalidad de conocer bien a los usuarios y estar en condiciones de definir mejor todos los aspectos del sistema con la superposición de los métodos y técnicas a seguir. Las técnicas usadas fueron:

a) grupo objetivo.

b) evaluación heurística. En esta técnica, como una variante de la inspección de facilidad o capacidad de uso, los usuarios evaluaron los formularios de consulta y las páginas de resultado asociadas utilizando cuatro listas de chequeo. Estas listas fueron construidas por un grupo de expertos, con base en una lista de principios definidos y específicamente aplicables a los motores de búsqueda.

c) análisis de la tarea y de la actividad. La descripción de la tarea fue obtenida por medio de las técnicas de recolección de datos, como en la evaluación heurística, la entrevista semiestructurada y la observación, buscando evidenciar el proceso de su realización. El análisis de la actividad fue ejecutado por medio de observaciones en el local de trabajo, recogiendo información sobre las operaciones efectuadas, su encadenamiento, sus dificultades y frecuencias de uso. Este proceso de evaluación es extendido en el trabajo de Kafure et al. (2007).

2.2.4 Módulos de bitácora, bodega de datos, minería de datos y recomendación

La bitácora es un repositorio temporal de información en donde se registran la huella de navegación y las acciones ejecutadas por un usuario cuando accede a la biblioteca. La bodega de datos (*data warehouse*, DWH) almacena datos e información relacionada con la historia de navegación, consulta y descarga de documentos de los usuarios. La DWH es el repositorio natural para llevar a cabo tareas de minería de datos. Adicionalmente, el administrador de la biblioteca cuenta con opciones de carga, consulta y generación de reportes estadísticos. La DWH soporta el proceso de recomendación utilizando un modelo híbrido (Li & Kim, 2003; Pennock et al., 2000; Rashid et al., 2006; Trujillo et al., 2007) basado en memoria (Herlocker et al., 2000; Konstan et al., 1997; Sarwar et al., 2001) y en modelo (Ahmad, 1999; Kamahara et al., 2005; Kim et al., 2005; Ungar & Foster, 1998) para calcular recomendaciones para un usuario dado y técnicas de agrupación por cúmulos (*clustering*) para agrupar usuarios con base en datos demográficos, áreas de interés y navegación. Las recomendaciones se hacen al usuario una vez se cuenta con suficiente información de su conducta de navegación.

3. El modelo de recomendación de PREDICA

Una de las características más importantes de la biblioteca digital de PREDICA y que la diferencia de otras propuestas, es su modelo de recomendación. Las recomendaciones se hacen con base en un modelo híbrido (Figura 8), construido en dos etapas: una fuera de línea (*off-line*) y otra en línea (*on-line*) (Li & Kim, 2003). En la etapa fuera de línea, se usan algoritmos basados en memoria para calcular cúmulos (*clusters*) de usuarios y de documentos. En la etapa en línea, se aplican algoritmos basados en modelos para hacer la recomendación.

En la etapa fuera de línea, se integra la información de usuarios y documentos y se provee de capacidad de escalamiento al sistema de recomendación PREDICA, al reducir el espacio de búsqueda en la etapa en línea. Un algoritmo jerárquico de agrupación por cúmulos se aplica para identificar grupos de usuarios con base en variables demográficas y psicográficas y en el comportamiento del usuario a la hora de descargar documentos. Mediante el uso de variables demográficas se espera mejorar la calidad de las recomendaciones (Kim et al., 2004).

Para calcular los cúmulos, se propone el uso de ponderaciones en la función de similitud, permitiendo ofrecer recomendaciones a usuarios que no tienen registros de navegación. (Trujillo et al., 2007). Los cúmulos de documentos se calculan a partir de vectores característicos del documento, que contienen información sobre áreas de conocimiento, tipo de documento e idioma. Son estáticos y por lo tanto, solo se actualizan cuando se registran en la biblioteca nuevos documentos. El propósito del proceso de integración de cúmulos de documentos y usuarios es producir agrupaciones de usuarios / documentos. Esta información se usa para el cálculo de probabilidades en la etapa en línea.

En la etapa en línea, se calculan las recomendaciones en tiempo real. Este cálculo está basado en el modelo producido en la etapa fuera de línea. Las recomendaciones se calculan usando el modelo de probabilidad que ha sido derivado de la propuesta de Kim et al. (2004). Este modelo calcula la probabilidad de que el usuario *i* acceda a un documento con un vector de característica *j*, con base exclusivamente en la información del metacúmulo (cúmulo de cúmulos) al que pertenece el usuario.

4. Discusión de resultados y experiencias de desarrollo

Desarrollar PREDICA implicó resolver un problema de gran complejidad, caracterizado por la diversidad de nuevas tecnologías a integrar, por la naturaleza y número de funciones que debía proveer y por la heterogeneidad de los integrantes de los grupos de desarrollo.

Como fruto de la experiencia de este proyecto, recomendamos para desarrollos similares, utilizar una plataforma (*framework*) que soporte el patrón modelo vista controlador. Este modelo garantiza un manejo autónomo y estandarizado del desarrollo de las capas de GUI, de la lógica y de la persistencia, para cada módulo funcional. Adicionalmente, con el propósito de integrar los módulos funcionales, se debería utilizar una herramienta de control de versiones y de integración global como CVS (*control version system*).

El primer modelo de bitácora que se planteó estaba fuertemente ligado al modelo de comunicación entre la interfaz de usuario y la lógica de la aplicación.

Una vez adoptadas las tecnologías de la web 2.0, la versión final implementada se basa en un modelo genérico centrado en los procesos que realiza el usuario y no en las páginas a las que éste accede.

A pesar de haber implementado un agente de software para la adaptabilidad de la navegación, se encontró que al fusionar este agente con las tecnologías web 2.0, el tiempo de respuesta del proceso de adaptabilidad se aumentó significativamente. Para aprovechar el potencial de los agentes, se recomienda plantear la biblioteca como un sistema multiagente.

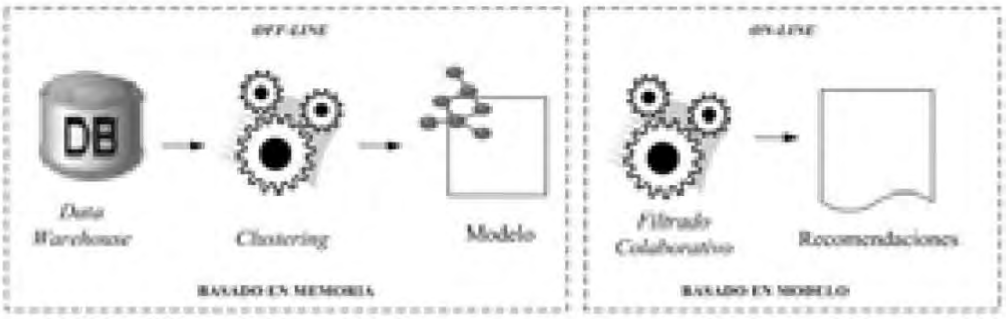

Figura 8. Modelos del sistema de recomendación PREDICA.

La utilización de RIA y AJAX para el desarrollo de las interfaces, mejora el desempeño de la aplicación con relación a las interfaces tradicionales, porque con aquellas sólo se requiere refrescar las páginas con nueva información. Por otra parte, facilitan el desarrollo de interfaces de usuario, similares a las de las aplicaciones de escritorio tan frecuentemente usadas y que se caracterizan por ofrecer al usuario un conjunto integrado y en línea de herramientas de apoyo.

Al utilizar estrategias de pre-procesamiento de la información (metadatos de los documentos, diccionarios controlados, organización jerárquica de la información) se enriquecen las interfaces de consulta (auto-completado de datos, búsqueda por árboles, entre otros) y se consigue optimizar el tiempo de respuesta de los algoritmos de búsqueda y recuperación de los documentos, porque se disminuye el espacio de búsqueda.

PREDICA, como prototipo de software tiene las siguientes características:

a) modularidad y facilidad de extensión y acceso. En la medida en que se puede dotar de nuevas capacidades funcionales y facilidad de acceso desde sitios remotos (debido al uso de la red como mecanismo básico de comunicación con el usuario final), se facilitan su extensión y la reutilización de código.

b) facilidad o capacidad de uso. Mediante el desarrollo de interfaces adaptativas, se facilitan su aprendizaje y utilización por parte de los usuarios.

c) inteligencia computacional. Debido a que cuenta con mecanismos y técnicas de minería de datos para identificar perfiles de usuario y facilitar la recuperación de información.

5. Agradecimientos

Agradecemos a la Universidad del Valle y al Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología Francisco José de Caldas - COLCIENCIAS la financiación del proyecto PREDICA.

También expresamos nuestro reconocimiento a los siguientes estudiantes de ingeniería de sistemas de la Universidad del Valle: Edward Gallego, Julián Daza, Francisco Javier Restrepo, Jhoan Alejandro García, Albeyro Echeverry, Oswaldo Solarte, Cindy Juliette Bernal, Santiago Gómez Rico, Yaneth Betancourt, Jorge Alexander Arango, Yorley Reyes, Diana María Ramírez y Luis David Males, quienes a través de sus trabajos de grado, participaron en el desarrollo de la primera versión del software de PREDICA.

6. Referencias bibliográficas

Abbattista, F., Degemmis, M., Fanizzi, N., Licchelli, O., Lops, P., Semeraro, G., & Zambetta, F. (2002). *Learning user profiles for content-based filtering in e-commerce.* In Proceedings of the AIIA Workshop *Su Apprendimiento Automatico: Metodi e Applicazioni*, Sienna, Italy. http://citeseer.ist.psu.edu/abbattista02learning.html

Ahmad, M. (1999). *Collecting user access patterns for building user profiles and collaborative filtering.* In Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI'99), Redondo Beach, California, p.57-64.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval.* Boston: Addison Wesley.

Balabanovic, M., & Shoham, Y. (1997). Combining content-based and collaborative recommendation. *Communications of the Association of Computing Machinery (ACM)* 40(3), 66-72.

Bollacker, K.D., Lawrence, S., & Giles, C.L. (1998). *An autonomous web agent for automatic retrieval and identification of interesting publications.* In Proceedings of the Second International ACM Conference on Autonomous Agents, p. 116-123. http://maya.cs.depaul.edu/~classes/csc575/papers/citeseer.pdf

Brusilovsky, P., & Maybury, M.T. (2002). From adaptive hypermedia to the adaptive Web. *Communications of the Association of Computing Machinery (ACM)* 45(5), 30-33.

CDP Meta Data Working Group (2006). *Dublin core metadata best practices v2.1.1.* http://www.cdpheritage.org/cdp/documents/cdpdcmbp.pdf

Chen, H-C., & Chen, A. L. P. (2001). *A music recommendation system based on music data grouping and user interests.* In Proceedings of the 10th International Conference on Information and Knowledge Managament (CIKM'01), Atlanta, Georgia, p. 231-238.

Corcho, O., (2006). Ontology based document annotation: trends and open research problems. *International Journal of Metadata, Semantics and Ontologies* 1 (1), 47-57.

Cunha, M. B. (1999). Desafios na construção de uma biblioteca digital. *Ciência da Informação (Brasilia)* 28 (3), 257-268. http://www.scielo.br/pdf/ci/v28n3/v28n3a3.pdf

Eichorn, J. (2006). *Understanding AJAX: using Javascript to create rich Internet applications.* New Jersey: Prentice Hall.

Eirinaki, M., Lampos, C., Paulakis, S., & Vazirgiannis, M. (2004). *Web personalization integrating content semantics and navigational patterns.* In Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Managament (WIDM' 04), p. 72-79.

Goecks, J., & Shavlik, J. (2000). *Learning users' interests by unobtrusively observing their normal behavior.* In Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI'00), New Orleans, p. 129-132. http://www.cc.gatech.edu/~jeremy/pubs/goecks-iui2000.pdf

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2003). *Ontological engineering.* London: Springer Verlag.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). *Explaining collaborative filtering recommendation.* In Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work (CSCW'00), p. 241-250.

Kafure I., (2004) *Usabilidade da imagem na recuperação da informação no catálogo público de acesso em linha.* Tesis de Doutorado, Universidade de Brasilia, Departamento de Ciência da Informação e Documentação, Brasil. http://repositorio.ibict.br/ibict/bitstream/123456789/84/1/HPTese_2004+-+Completa.pdf

Kafure, I., Valencia, M. E., Rodríguez, P. J., Florián, B. E., Carrillo, J. E., Solarte, O., & Ciprian, M. (2007). *Evaluación de la usabilidad de la biblioteca digital PREDICA.* Memorias del Seminario Internacional de Bibliotecas Digitales, Brasil.

Kamahara, J., Asawaka, T., Shimojo, S., & Miyahara, H. (2005). *A community-based recommendation system to reveal unexpected interests.* In Proceedings of the 11th International Multimedia Modelling Conference (MMM'05), p. 433-438.

Kim, H., Kim, J., & Herlocker, J. (2004). *Feature-based prediction of unknown preferences for nearest-neighbor collaborative filtering.* In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM'04), p. 435-438.

Kim, Y. S., Yum, B-Y., Song, J., & Kim, S. M. (2005). Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Systems with Applications* 28 (2), 381-393.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J., Gordon, L., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the Association of Computing Machinery (ACM)* 40 (3), 77-87.

Kurki, T., Jokela, S., Sulonen, R., & Turpeinen, M. (1998). *Agents in delivering personalized content based on semantic metadata.* In Proceedings of the American Association for Artificial Intelligence Spring Symposium Workshop on Intelligent Agents in Cyberspace, Stanford, California.
http://www.soberit.hut.fi/publications/SmartPush/sp_papers/agents_md_aaai99s.pdf

Lazar, J. (2005). *Web usability: a user-centered design approach.* Boston: Addison Wesley.

Li, Q., & Kim, B.M. (2003). *Clustering approach for hybrid recommender system.* In Proceedings of the 2003 IEEE / WIC International Conference on Web Intelligence, p. 33.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7 (1), 76-80.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2003). *Capturing interest through inference and visualization: ontological user profiling in recommender systems.* In Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP'03), Florida, p. 62-69.

Middleton, S. E., Shadbolt, N. R., De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems* 22 (1), 54-88.
http://eprints.ecs.soton.ac.uk/8926/01/tois2004.pdf

Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). *MovieLens unplugged: experiences with an occasionally connected recommended system.* In Proceedings of the International Conference on Intelligent User Interfaces (IUI'03), Miami, p. 263-266.
http://www.grouplens.org/papers/pdf/miller-iui03.pdf

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery* 6 (1), 61-82.
http://robotics.stanford.edu/~ronnyk/WEBKDD-DMKD/Mobasher.pdf

Nielsen, J., & Loranger, H. (2006). *Prioritizing web usability.* Berkeley: New Riders Press.

O'Donovan, J., & Smyth, B. (2005). *Trust in recommender systems.* In Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05), California, p. 167-174.

Pazzani, M., & Billsus D. (1997) Learning and revising user profiles: the identification of interesting Web sites. *Machine Learning* 27, 313-331.
http://www.ics.uci.edu/~pazzani/Publications/SW-MLJ.pdf

Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). *Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach.* In Proceedings of the 16th Conference on Uncertainty and Artificial Intelligence (UAI-2000), Stanford, California, p. 473-480.
http://dpennock.com/papers/pd-uai-00.pdf

Qooxdoo (The New Era of Web Development). (2007). *Qooxdoo documentation.* http://qooxdoo.org/documentation

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002). *Getting to know you: learning new user preferences in recommender systems.* In Proceedings of the International Conference on Intelligent User Interfaces (IUI'02), San Francisco, p. 127-134.

Rashid, A. M., Lam, S. K., Karypis, G., & Riedl, J. (2006). *ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm.* In Proceedings of the Knowledge Discovery and Data Mining on the Web Workshop (WEBKDD'06), Philadelphia.
http://www.grouplens.org/papers/pdf/clustKNN.pdf

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms.* In Proceedings of the 10th International Conference on World Wide Web (WWW2001), Hong Kong, p. 285-295.
http://www.grouplens.org/papers/pdf/www10_sarwar.pdf

SIEGE. (2007). *SIEGE Home Page.*
http://www.joedog.org/JoeDog/Siege

Smeaton, A. F., & Callan, J. (2005). Personalisation and recommender systems in digital libraries. *International Journal on Digital Libraries* 5 (4), 299-308.

Sugiyama, K., Hatano, K., & Yoshikawa, M. (2004). *Adaptive Web search based on user profile constructed without any effort from users.* In Proceedings of the 13th International Conference on World Wide Web (WWW2004), New York, p. 675-684.

The Symfony Team. (2007). *Symfony documentation.*
http://www.symfony-project.org/doc/1_0/

Trujillo, M., Millán, M., & Ortiz, E. (2007). A recommender system based on multi-features. *Lectures Notes in Computer Science* 4706, 370-382.

Ungar, L., & Foster, D. (1998). *A formal approach to collaborative filtering.* In Proceedings of the Conference on Automated Learning and Discovery (CONALD'98).