

# Sistema multiagente integrador de bibliotecas digitales

Christian Sánchez-Sánchez, Héctor Jiménez-Salazar,  
Carlos Rodríguez-Lucatero, Esaú Villatoro-Tello, Gabriela Ramírez-de-la-Rosa

Departamento de Tecnologías de la Información,  
División de Ciencias de la Comunicación y Diseño,  
Universidad Autónoma Metropolitana Unidad Cuajimalpa, México D.F.

{csanchez, hjimenez, crodriguez, evillatoro, gramirez}@correo.cua.uam.mx

**Resumen.** Debido a la existencia de múltiples Bibliotecas Digitales en Internet, cuando un usuario pretende obtener una buena parte de la información disponible sobre un tema, éste tiene que invertir mucho tiempo para buscar las fuentes dentro de ellas y después integrar los resultados que considera relevantes. En este artículo se presenta la aplicación de un sistema multiagente que realiza búsquedas de documentos en diferentes bibliotecas digitales dada una determinada consulta y criterio, para que posteriormente integre los resultados y los presente al usuario. El sistema integra las ventajas de los sistemas Multiagente, Arquitecturas Orientadas a Servicios y algunos de los principios de la Integración de Información. En el artículo, se presenta el detalle de la arquitectura propuesta (roles de agentes y protocolos de interacción) y de las fases de Integración de Información, lo que conforma el núcleo del sistema.

**Palabras clave:** sistema multi-agente, arquitecturas orientadas a servicios, integración de información, bibliotecas digitales.

## 1. Introducción

Debido a que existen múltiples Bibliotecas Digitales (BD's) en Internet, se ha constatado la complicación de realizar múltiples búsquedas de documentos (libros, revistas, artículos) sobre un determinado tema tratando de aprovechar la mayor parte de la información disponible. El principal problema radica en que el usuario tiene que invertir mucho tiempo en buscar las BD's, dentro de ellas y después integrar los resultados que considera relevantes.

Tratando de resolver este problema la Federación de Bibliotecas Digitales de los Estados Unidos de Norteamérica está desarrollando un proyecto [1] que pretende unir los sistemas web de más de 33 BD's universitarias de ese país junto con los Archivos de la Nación.

Con la finalidad de ofrecer una solución al problema surge una serie de preguntas cuya respuesta puede facilitar la propuesta de un sistema, por ejemplo: 1) *¿Cómo hacer un sistema dinámico que permita incluir y modificar BD's sin tener que reprogramar el sistema?*, 2) *¿Cómo integrar la información de varias*

fuentes?, 3) ¿Cómo re-ordenar (ranking) información ya ordenada de varias fuentes de acuerdo a una consulta y criterio?

Referente a la pregunta 1, hay que considerar que la mayoría de los sistemas de las BD's ya están hechos por lo que la integración de Sistemas es fundamental. Buscando facilitar la integración de los sistemas se pensó en hacer uso de las Arquitecturas Orientadas a Servicios (SOA por sus siglas en inglés) y los Sistemas Multi-agentes (SMA) para lidiar con el dinamismo del sistema.

SOA propone una forma donde el software se presenta como servicios de aplicación. Estos servicios [2] son: débilmente acoplados, altamente interoperables, se pueden re-usar y por ende permiten desarrollar aplicaciones con mayor rapidez.

Por otro lado tenemos al SMA [3] una red débilmente acoplada de agentes de software que interactúan para resolver problemas que van más allá de su conocimiento o capacidades individuales. Es importante señalar que SMA y SOA se pueden complementar. Según Sycara et al. [4] un problema existente se puede resolver mediante un conjunto de servicios (Web o de Agente) los cuales son descubiertos por un agente que integre los resultados de diferentes servicios, lo cual se consideró para proponer una solución a la pregunta 2 y 3.

Con la finalidad de que un agente pueda integrar y re-ordenar información existen algunas metodologías para la integración de información (fusión de datos), un ejemplo de estas son los métodos que según Vogt y Cottrell [5] tienen los siguientes efectos: a) *skimming* que toma los elementos mejor posicionados de cada uno de los enfoques de recuperación, b) *coro* que toma los elementos que en varios enfoques de recuperación se consideran relevantes y c) *caballo negro* se toman algunos elementos cuyas estimaciones de relevancia fueron muy altos o muy bajos, en comparación con otros enfoques de recuperación. Estos efectos son utilizados en la Recuperación de Información (RI), aplicables al tipo de resultados que arrojan las BD's.

En este artículo se presenta un SMA que realiza búsquedas de documentos en diferentes BD's dado una determinada consulta y criterio, para que posteriormente integre los resultados y los presente a un usuario. El sistema integra las ventajas de los SMA, SOA y algunos de los principios de la Integración de Información. En el artículo, se presenta el detalle de la arquitectura propuesta (roles de agentes y protocolos de interacción) y de las fases de Integración de Información, lo que conforma el núcleo del sistema.

El resto del documento se encuentra organizado de la siguiente manera: en la Sección 2 se describe brevemente el trabajo relacionado más relevante a la temática en cuestión. En la Sección 3 se presenta la arquitectura: los participantes (los tipos de agentes) y componentes (servicios web y el registro de servicios). La Sección 4 detalla las fases de la composición de la información integrando los resultados de diferentes BD's. Una breve descripción de la aplicación prototipo se muestra en la Sección 5. Algunas pruebas realizadas al sistema se presentan en la Sección 6. Finalmente, la Sección 7 muestra las conclusiones obtenidas y define las líneas de trabajo futuro.

## 2. Trabajo relacionado

Existen conjuntos de programas de software diseñados para crear y distribuir colecciones digitales en forma de BD's, tal es el caso de Greenstone [6] que facilita dicha tarea. Sin embargo, el que haya cada vez más BD dificulta la integración de la información que ofrecen.

Según Ibrahim et al. [7] para la integración de BD se plantean varios retos debido a las diferencias prevaletentes en: los tipos de datos que maneja cada fuente de información, la diversidad de los lenguajes de consulta y manipulación de datos, la gran variedad de tipos de información y formatos así como su semántica.

Los autores identifican dos enfoques utilizados para construir sistemas de integración de datos:

**Enfoque virtual.** El sistema de integración de datos recibe la consulta y la traduce de acuerdo a la fuente de información. Este enfoque es conveniente cuando el número de fuentes de información es muy grande, la información cambia frecuentemente y las necesidades de los clientes es impredecible. Puede ser ineficiente cuando las consultas se generan múltiples veces, las fuentes de información son lentas, costosas o no están disponibles.

**Enfoque materializado.** La información que se puede extraer de las consultas se obtiene con antelación y es guardada en un *Almacén de Datos* para ser consultada de forma centralizada. Se recomienda este enfoque cuando las porciones de información a consultar sean predecibles y los clientes necesiten un buen desempeño de la consulta.

Seaman et al. [8] establece que para facilitar la integración y contar con contenidos más ágiles, es necesario contar con metadatos que se puedan combinar con otros elementos de las BD. Los autores de este trabajo proponen algunos estándares para usarse como metadatos como Open Archives Initiative (OIA), Metadata Encoding and Transmission Standard (METS), CrossRef, DOI entre otros, que ayuden a ligar la información de la BD. Poniendo énfasis que representa un gran reto pero a la vez una gran oportunidad con muchas ventajas cuando exista una implementación adecuada.

Con lo que respecta a las propuestas del desarrollo de prototipos que integran las BD, Lima et al. [9] proponen un sistema cuya arquitectura recae en un SMA. Los autores apostaron por la interoperabilidad semántica, por medio del uso de relaciones inter-ontológicas y una metabase (base de datos de metadatos) que ayudan a describir información en las BD's. Afirman que cuentan con un novedoso servicio de razonamiento ontológico que representa un modelo conceptual sofisticado de términos y relaciones, que captura la semántica de una manera prometedora para la integración de BD. Los autores presentan una aplicación (Sistema Web) que opera con información geográfica y medioambiental.

Nnadi et al. [10] proponen una Infraestructura de Integración de BD's (DLII) ofreciendo un enfoque ligero y sistemático para integrar las colecciones y servicios de una BD. En este trabajo se se detalla como integrar las BD's por medio de metainformación. La principal contribución de esta investigación es proporcionar una infraestructura presumiblemente sencilla y sustentable para la integración de las colecciones y los servicios de BD.

Sin embargo, aunque existen propuestas de sistemas como el de Lima [9], para que puedan operar estos sistemas se requiere de ontologías por cada temática contenida en los documentos de las BD, lo que dificulta su uso con la información actual. Otros sistemas requieren de metadatos que relacionen la información como las propuestas de Seaman [8] y Nnadi [10], que similar al trabajo anterior requieren información adicional que se debe generar para poder utilizar sus propuestas.

Otros sistemas como OLIS [11] y NSDL, desarrollado a partir de la propuesta descrita en [10], también requieren recabar la información de otras BD's, integrarlas a sus RI's para posteriormente poder desplegar resultados de varias fuentes, lo que limita la flexibilidad para agregar nuevas fuentes.

### 3. Arquitectura del sistema

La Figura 1 muestra la arquitectura del sistema propuesto. Se presentan los participantes (tipos de agentes) y los componentes: BD's, servicios web y el registro de servicios. Dicha arquitectura ayudó a ofrecer una solución que responda la pregunta *¿Cómo hacer un sistema dinámico que permita incluir y modificar BD's sin tener que reprogramar el sistema?*, planteada en la Introducción.

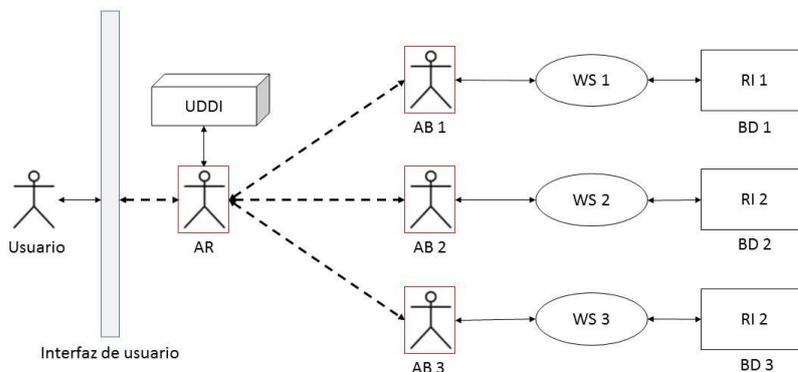


Fig. 1. Arquitectura general del sistema.

En el sistema contempla dos tipos de agentes o roles, los sistemas de BD y otros dos componentes.

- El Agente Representante (**AR**). El cual se encarga de la comunicación con el cliente (recibir la consulta y mostrarle los resultados), la búsqueda de servi-

cios (en el UDDI), así como de comunicarse con los  $AB_j$ . Mayor información se dará en la siguiente sección donde se observan las fases de la Integración.

- El Agente Bibliotecario (**AB**). Es el agente que se encarga de hacer las búsquedas en las BD's, a partir de la consulta hecha por el AR, enviar los resultados y ordenarlos de acuerdo a un determinado criterio.
- Los Servicios Web (**WS**). Son la interfaz a las BD's. Éstos permiten que los agentes o usuarios puedan consultar los sistemas de BD's (bajo diferentes criterios de búsqueda), ofrecen una interfaz estandarizada, reusable e interoperable (características propias de este tipo de sistemas).
- Otro de los componentes es el registro de servicios Web (**UDDI**). En éste se almacena dinámicamente información de servicios web como: descripción de lo que realizan, categoría, ubicación, interfaz de uso, así como los protocolos estándar que permiten comunicarse con dicho servicio.

Mayor detalle de la implementación de la arquitectura es mostrada en la subsección 5.2. En la siguiente sección se describen las fases de la composición de información, lo cual muestra el funcionamiento del SMA y como los Agentes cooperativamente resuelven el problema.

#### 4. Fases de integración de las bibliotecas digitales

La construcción de la BD virtual se realiza por medio de 8 fases. Cada fase se describe principalmente por dos eventos: el inicio y la acción que se realiza. La figura 2 muestra la secuencia de las fases así como el agente encargado de la detonación.

Es importante resaltar que estas fases fueron propuestas para dar respuesta a la pregunta *¿Cómo integrar la información de varias fuentes?* Específicamente las fases 5-7 proponen un método para dar solución a la pregunta *¿Cómo reordenar (ranking) información ya ordenada de varias fuentes de acuerdo a una consulta y criterio?*, preguntas que fueron previamente planteadas en la sección 1.

Para que el sistema funcione se asume que se tiene un  $AR$  y un número  $n$  de  $AB$ 's, donde  $n \geq 1$ , y se representa a un  $AB$  cualquiera como  $AB_j$ , donde  $j$  puede ser un número entre 1 y  $n$ . Mayor detalle de la implementación de estas fases se puede encontrar en la subsección 5.3

##### Fase 1. Búsqueda de las bibliotecas digitales

Esta etapa se puede desplegar en dos momentos:

- 1) El primer momento es cuando se arranca el sistema, el  $AR$  consulta en el UDDI que BD's ofrecen sus servicios a través de WS y guarda una lista de ellos.
- 2) El segundo momento se presenta cuando el  $AR$  determina que es necesario consultar de nuevo el UDDI, debido a que no están disponibles algunas de las BD's o se requiere buscar más.

##### Fase 2. Recepción de la Consulta

Esta etapa se presenta cuando el usuario del sistema desea realizar una búsqueda. El  $AR$  obtiene una solicitud del usuario con la consulta y opcionalmente el criterio información que posteriormente le será enviada a cada  $AB_j$ .

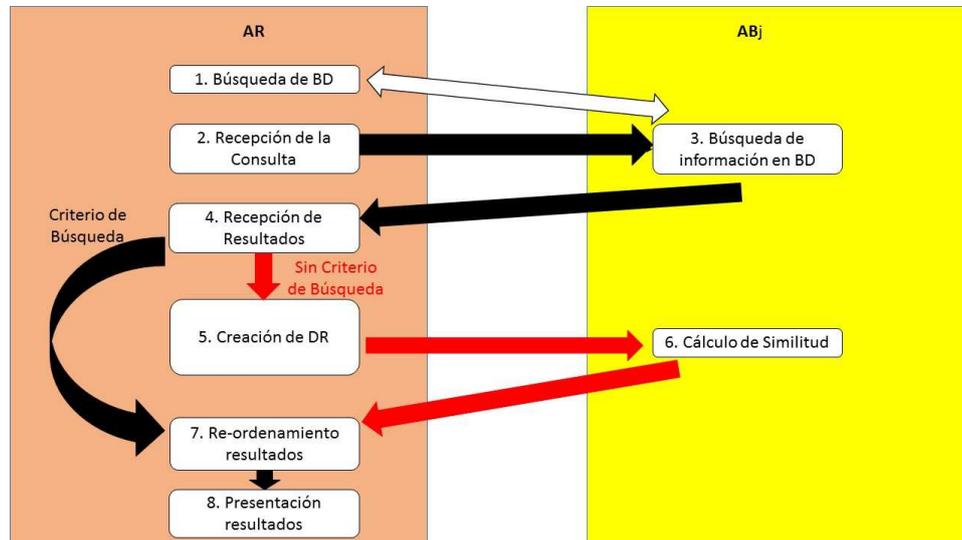


Fig. 2. Fases de Integración de BD's

### Fase 3. Búsqueda de información en las bibliotecas digitales

Esta etapa comienza cuando el *AR* tiene una consulta del usuario. El agente *AR* invita a los *AB<sub>j</sub>* a participar para resolver el problema. Cada *AB<sub>j</sub>* selecciona una *BD* y manda su propuesta donde incluye la *BD* en la cual quiere buscar. Posteriormente, el *AB<sub>j</sub>* busca documentos en la *BD* y envía los resultados a *AR*.

### Fase 4. Recepción de resultados de las bibliotecas digitales

El *AR*, resultado de la comunicación con el *AB<sub>j</sub>*, recibe los resultados de la búsqueda. Si se dio en la solicitud algún criterio (fecha o relevancia de la publicación) se pasa directamente a la fase 7, ya que la fecha o el número de referencias (relevancia) servirán para ordenar los documentos. En el caso de no haberse definido ningún criterio se continúa con la etapa 4, la cual se describe a continuación.

### Fase 5. Creación del documento de referencia

Si en la solicitud no se especificó algún criterio de búsqueda entonces el *AR* hace una expansión de la consulta con la finalidad de proponer un Documento de Referencia (**DR**). Cada *AB<sub>j</sub>* utilizará el **DR** para ordenar sus resultados, de acuerdo a la similitud. Para formar el documento de referencia el *AR* puede usar su conocimiento (por ejemplo, información que haya obtenido de las preferencias del usuario o de los temas), y el conocimiento de cada agente *AB<sub>j</sub>*.

### Fase 6. Cálculo de la similitud de los resultados con respecto al documento referencia

Una vez que el *AB* tiene el **DR**, éste lo compara contra todos los resúmenes. Los agentes se ponen de acuerdo para determinar un criterio de comparación.

### Fase 7. Reordenamiento de resultados

Una vez que el *AR* recibe los resultados ordenados de algún  $AB_j$ , entonces el *AR* tomando en cuenta el criterio de comparación ordena los resultados.

#### Fase 8. Presentación de resultados

Los resultados son presentados al usuario de manera ordenada de acuerdo al criterio (o similitud).

## 5. El sistema integrador de bibliotecas digitales

En esta sección se presentan la interfaz del sistema y mayor detalle de la implementación de la arquitectura y las fases descritas en las secciones 3 y 4 respectivamente.

### 5.1. Interfaz del sistema

El SMA Integrador de BD's es accedido a través de una Sistema Web. Se usa de manera similar a los buscadores de información en Internet y BD's, escribiendo las palabras clave y presionando el botón de búsqueda.

En el caso de que se desee utilizar el criterio de búsqueda “año de la publicación” basta con poner antes o después de las palabras clave *criteria:year*. De manera similar se puede seleccionar el criterio de “relevancia de los artículos” *criteria:relevance*. Si hay resultados de la búsqueda estos son presentados de manera ordena (Ver Figura 3).



Fig. 3. Resultados de una búsqueda en el SMA integrador de BD's

### 5.2. Componentes utilizados en la arquitectura

Con la finalidad de crear y probar el prototipo del Sistema Multiagente Integrador de Bibliotecas Digitales se consideraron tres diferentes BD's: Springer

Link, SciELO e IEEE Xplorer. La selección de estas tres bibliotecas digitales se realizó debido a que son muy consultadas por la comunidad científica, además de que técnicamente permiten hacer consultas por medio del protocolo GET, es decir permiten enviar las consultas en el URL facilitando la integración.

En relación a los componentes, se programaron tres Servicios Web (WS), uno para cada BD, que funcionan como interfaz para que puedan utilizarlos los agentes que integran el sistema. Gran parte del sistema fue desarrollado en lenguaje de programación Java, ya que ofrece herramientas que facilitan el desarrollo de los servicios web y sus clientes, además de que el UDDI utilizado fue el jUDDI, el cual cuenta con un API en Java para que pueda ser utilizado mediante su código.

Con respecto a los participantes, se programaron un AR y tres  $AB_j$  (uno por cada BD).

### 5.3. Detalles de la implementación de las fases

Con la finalidad de facilitar la comunicación entre los diferentes actores se programaron diferentes protocolos de interacción basados en el estándar de FIPA [12]. A continuación se da mayor información sobre la implementación de cada fase.

Con respecto a la **Fase 1**, para encontrar todas las fuentes de documentos disponibles, el AR consulta el UDDI, donde solo hay registrados WS de BD's. Todos los servicios que AR encuentra los agrega a su lista para posteriormente enviarlos a los  $AB_j$ 's. Esta consulta la realiza cuando se arranca el sistema o cuando un agente  $AB_j$  falla al tratar de consumir al WS de la BD que le corresponde. En ese caso  $AB_j$  comienza el protocolo de interacción FIPA-Request-When, la condición es que el WS que le corresponde no es localizado, no contesta, o no envía los resultados en un formato que pueda procesar el agente. Como resultado del protocolo de interacción el agente  $AB_j$  esperaría una nueva asignación de WS el cual consultar.

En la **Fase 2**, el usuario, a través de la interfaz del sistema, activa el protocolo de comunicación FIPA-Request, enviando una solicitud de búsqueda de documentos en bibliotecas digitales al AR. En el mensaje de solicitud (request) se le envía al agente, la consulta y opcionalmente el criterio de búsqueda (fecha ó relevancia de la publicación). La comunicación entre el AR y el usuario terminará si el agente se niega, falla o reporta los resultados.

La **Fase 3** comienza cuando el AR arranca el protocolo de interacción FIPA-Iterated-Contract-Net. Envía una llamada a participar a los  $AB_j$ 's, en la llamada incluye las condiciones de ejecución, la consulta, el criterio y la lista de posibles BD's en las cuales puede buscar. El AR puede estar revisando las propuestas con la finalidad de hacer búsquedas en la mayoría de las BD's. Cada  $AB_j$  tiene un cliente de WS para consultar al WS de la BD, por el momento todas las interfaces de los WS son iguales.

El AR almacena, en la **Fase 4**, temporalmente los resultados de las búsquedas. En el caso de que se haya escogido algún criterio de búsqueda pasa a la fase 7, de lo contrario se continua con la siguiente fase.

En la **Fase 5**, debido a que en el sistema no se considera el conocimiento de *AR* para crear el DR, solo se toma en cuenta el primer resultado que regresa cada  $AB_j$  asumiendo que es el que mejor satisface la consulta. Luego entonces, el DR se crea por medio de la unión de los términos contenidos en los resúmenes de los artículos (retroalimentación ciega) que fueron obtenidos como el primer resultado de cada BD.

Posteriormente dicho documento se pre-procesa (DRP): eliminando palabras cerradas y signos de puntuación, transformado todas las letras a minúsculas, y utilizado un lematizador. Una vez que se cuente con el DRP éste se envía a cada  $AB_j$ .

En la **Fase 6**,  $AB_j$  compara los resúmenes de los documentos encontrados contra el DRP de la siguiente manera:

Se preprocesan todos los resúmenes de la colección  $r_j$ , donde el índice  $j$  representa al  $AB$  que lo encontró. Se obtiene un nuevo documento  $rp_j$ : eliminando palabras cerradas, signos de puntuación y se transforman todas las letras a minúsculas para que finalmente se utilice un lematizador.

Una vez teniendo todos los  $rp_j$  se calcula la similitud (coeficiente Jaccard)  $sim(DRP, rp_j) = \frac{T_{DRP \cap T_{rp_j}}}{T_{DRP \cup T_{rp_j}}}$ . Es decir cantidad de elementos que contiene la intersección de los conjuntos de términos de ambos documentos entre la cantidad de términos contenidos en la unión. Calculadas todas las similitudes estas se mandan junto con los índices de los documentos re-ordenados al agente *AR*.

Para acomodar los resultados en la **Fase 7**, el *AR* utiliza el ordenamiento por inserción, tomando en cuenta las similitudes calculadas por los  $AB_j$ , con respecto al DR.

En la última fase, **Fase 8**, se presentan todos los documentos encontrados por los  $AB$ 's y por cada documento encontrado se muestra: el nombre del artículo, un fragmento del resumen y la liga al sitio de la BD donde se puede consultar el artículo completo.

## 6. Pruebas

Las pruebas que se realizaron con respecto a la arquitectura fueron las siguientes:

- Se probó el sistema con 1 a 5  $AB_j$ , debido a que hasta al momento solo hay disponibles (a través de WS) tres Bibliotecas Digitales, el sistema con 3 a 5 Agentes no mostró diferencia con respecto los resultados considerados (tomando en cuenta las 3 BD's) y el tiempo de respuesta del sistema fue de 2 a 3 minutos para un aproximado de 75 resultados.
- Se agregaron y eliminaron los WS de las BD's del UDDI para probar el dinamismo del sistema, se examinó con una, dos y tres WS de BD's y no se detectó ningún problema con respecto a la operación del sistema. El *AR* operó sobre las BD's que se encontraban registradas (una, dos o tres) mostrando los resultados del mismo número de BD's.

- Con respecto a la Integración de Información y Ordenamiento de Resultados de Diferentes fuentes se diseñó un pequeño experimento para probar dos formas de integración. Se realizó la búsqueda en las tres BD's con la consulta "Web Services and MultiAgent System Integration", recuperándose 70 resultados.

Posteriormente, con la finalidad de evaluar la fusión de documentos se hizo un etiquetado manual de los resultados marcando solo aquellos que hablaban sobre la integración de estas tecnologías, de este etiquetado solo 37 resultados fueron relevantes.

Los algoritmos probados para fusionar documentos fueron: como *Gold Standard* tomamos la integración Round-Robin (efecto *Skimming*) la cual consta en ingresar los resultados de las 3 BD en 3 pilas, los elementos se ingresan del resultados con peor *ranking* al mejor, quedando en el tope de la pila el mejor resultado. Se toma un elemento de cada pila intercalándolos hasta quedar vacíos.

El segundo algoritmo probado fue el descrito en la subsección 5.3 para las fases 5-7. Se tomó el mejor resultado de cada BD para la consulta previamente mencionada. Teniendo los tres resultados y mediante la retroalimentación ciega se logró extender la consulta con conjuntos de términos relevantes como: "service oriented computing", "loose coupled", "infraestructure", "distributed applications", "autonomic computing", "service discovery", "service description", "service invocation", "dynamic integration", "seamless integration", "service interoperability", "FIPA", "service oriented architecture".

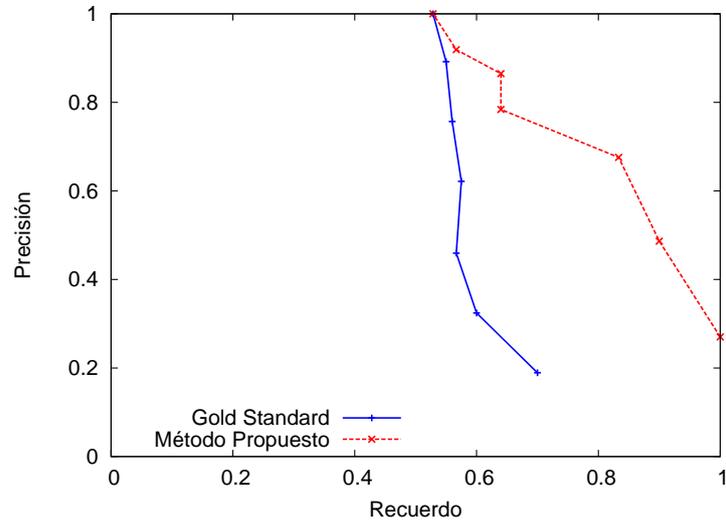
Se obtuvo el DR y se pre-procesó para obtener el DRP. Se pre-procesaron todos los resúmenes de los resultados obtenidos de las BD, y cada  $AB_j$  cálculo la similitud (coeficiente Jaccard) de sus documentos encontrados con respecto a DRP. Cada  $AB_j$  envió las similitudes al  $AR$  quien se encargó de ordenarlos (por inserción) de acuerdo a su similitud.

Se compararon los resultados de las fusiones obtenidas de los dos algoritmos, siendo algoritmo propuesto el que arrojó mejor Precisión y Recuerdo (Ver Figura 4), ordenando la mayoría de documentos relevantes a la consulta en los primeros documentos mostrados al usuario. El mejor desempeño del algoritmo propuesto se alcanzó con los primeros 40 resultados ( $P@40$ ) con  $P=.725$   $R=.78$  contra  $P=.575$   $R=.6216$  del primer método, razón por la cual se seleccionó para implementarlo en el sistema.

## 7. Conclusiones y trabajo futuro

Consideramos que la combinación de las tecnologías de SMA y SOA son una buena opción para generar sistemas de integración de información, ya que los agentes se pueden organizar para conseguir y ordenar la información. Con lo que respecta a la integración de información, proveniente de RI's, los algoritmos de integración y ordenamiento utilizados mostraron, para la mayoría de las pruebas, resultados aceptables.

Debido a que este trabajo enmarca el principio de la investigación y desarrollo del sistema se omitieron otras pruebas, sin embargo se ha planteado incluir otros



**Fig. 4.** Resultados (P y R) de los integradores Round Robin y método propuesto

protocolos de interacción entre agentes, evaluar comportamiento con un mayor número de agentes, consultas y BD's.

Así mismo, se ha planteado probar otros métodos de integración y ordenamiento de información. Otra de las tareas futuras es integrar a la tecnología de la Web Semántica, para desarrollar descripciones de servicios semánticos y mecanismos para que los agentes puedan realizar búsquedas más eficientes.

Así mismo, se propone dotar de herramientas al agente para extender consultas en base a su conocimiento del perfil del usuario.

**Agradecimientos.** Agradecemos a la Universidad Autónoma Metropolitana (UAM) y al proyecto, auspiciado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), número CB2010/153315 por el apoyo para la asistencia a este evento. También queremos hacer un reconocimiento especial a la alumna de la UAM-Cuajimalpa Ruth Esquivel Pérez por haber realizado la programación de algunos módulos del sistema aquí descrito, como parte de su Proyecto Terminal

## Referencias

1. New Jersey Institute of Technology, Project: Service Integration of Digital Libraries. <http://nsdl.org/about> (Última visita en Diciembre de 2014)
2. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison Wesley (2005)

3. Durfee, V.R., Lesser, E.H., Corkill, D.D.: Trends in Cooperative Distributed Problem Solving In: IEEE Transactions on Knowledge and Data Engineering, Vol. March 1989, pp.63–83, IEEE (1989)
4. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services Journal of Web Semantics, Vol. 1, No. 1, pp. 27–46, Elsevier (2003)
5. Vogt C. C. and Cottrell G. W.: Predicting the performance of linearly combined ir systems. In: 21st International Conference on Research and Development in Information Retrieval, Melbourne, Australia (1998)
6. <http://www.greenstone.org> (Última visita en Diciembre de 2014)
7. Ibrahim,I. K., Schwinger, W.: Data Integration in Digital Libraries: Approaches and Challenges. In: Software Competence Center Hagenberg, Austria (2001)
8. Seaman, D.: Aggregation, Integration and Openness: Current Trends in Digital Libraries. In Proc. Digital Libraries and Knowledge Communities in Networked Information Society, Tsukuba, Ibaraki, Japan, pp.105–110 (2004)
9. Lima, T., Sheth, A., Ashish,N., Guntamadugu, M., Lakshminarayan, S., Palse-na,N., Singh,D.: Digital Library Services Supporting Information Integration over the Web. In: Workshop on Information Integration on the Web, pp. 19–26 (2001)
10. Nnadi, N., Bieber, M.: Towards Lightweight Digital Libraries Integration. (2004)
11. <http://www.bodleian.ox.ac.uk/bdlss/olis-ils> (Última visita en Diciembre de 2014)
12. <http://www.fipa.org/> (Última visita en Diciembre de 2014)