



## CAPÍTULO 2

### AGENTES MÓVILES Y RECUPERACIÓN DE INFORMACIÓN

En el capítulo 1 se describió la importancia que tienen los servicios de recuperación y la tecnología de agentes en el contexto de bibliotecas digitales. En el presente capítulo se presentarán algunas definiciones y puntos relevantes sobre agentes móviles, una breve explicación sobre algunos de los algoritmos más utilizados para la recuperación de información, así como una reseña sobre trabajos relacionados en estas dos áreas.

#### 2.1 Agentes Móviles.

La movilidad es una característica que sólo algunos agentes poseen. La diferencia entre los denominados agentes estáticos y los móviles radica en que los primeros se llevan a cabo únicamente en el sistema en donde inició su ejecución, si éstos necesitan información o requieren interactuar con otros agentes ubicados en otro sistema, entonces hacen uso de mecanismos típicos como lo es el Llamado de Procedimientos Remotos (RPC por sus siglas en inglés); por su parte los agentes móviles no se limitan al sistema en el que su ejecución inició, pues poseen la habilidad de transportarse a un nodo remoto y obtener beneficios de él en el momento de realizar su tarea [Lange y Oshima 1998]. Los agentes móviles han sido calificados por diversos autores como una tecnología emergente que facilita el diseño, implementación y mantenimiento de sistemas distribuidos.

##### 2.1.1 Concepto de Agente Móvil.

Los agentes móviles también son denominados como agentes de red [Sánchez 1997], transportables [Gray et al. 1996], itinerantes [Chess et al. 1995], ágiles e incluso objetos móviles [Lange 1997].

Un agente móvil, según Lange y Oshima [1998], es un objeto de software que tiene la capacidad de migrar entre los nodos o dominios de una red en un sistema distribuido. Creado en un ambiente de ejecución, el agente puede transportar su estado (valores de sus atributos que le ayudan a determinar qué hacer cuando éste continúa la ejecución en el destino) y código (instrucciones que el agente debe ejecutar) a otro contexto en donde reanudará la ejecución. Estos agentes aprovechan los recursos del nodo destino en beneficio del nodo que los envió.

En [Castellanos y Sandoval 1997] se dice que por agente móvil puede entenderse a aquel objeto constituido por código, estados de ejecución, datos e itinerario. Dichos elementos deben ser portables para que se puedan migrar y ejecutar en los nodos destino.

De manera general [Pérez 1998] menciona que las características de un agente móvil son:

Ser un proceso autónomo o semi-autónomo. El agente es capaz de decidir cuándo, cómo y en qué condiciones migrar a otro nodo de la red.

Ser persistente pues se envía como objeto conservando su estado de ejecución, código y datos.

Ser capaz de suspender su ejecución y migrar a otro nodo para después reanudar el proceso o iniciarlo nuevamente.

Ser comunicativo con su medio como respuesta a los cambios de su ambiente; con otros agentes con el fin de intercambiar información; o bien con su dueño para informarle sobre su estado o resultados.

Enfocarse a realizar tareas ya sea delegadas por su dueño o en respuesta a los cambios que se produzcan en su ambiente.

Ejecutarse sin conexión, pues el agente es capaz de migrar, desactivarse o esperar mientras la conexión a la red se reanuda.

Procesarse en forma asíncrona gracias a que tiene su propio hilo de ejecución. Esta característica le permite trabajar independientemente de los otros procesos que se estén ejecutando en el nodo.

Duplicarse con el simple hecho de generar un clon.

##### 2.1.2 Ventajas.

Los agentes móviles brindan importantes beneficios en la creación de sistemas distribuidos. Entre las principales ventajas que ofrecen, [Lange y Oshima 1998] mencionan las siguientes:

Permitir el cómputo asíncrono y autónomo. Debido a que los agentes móviles tienen su propio hilo de ejecución, pueden trabajar de manera asíncrona respecto a otros procesos llevados a cabo en el mismo nodo. El hecho de poder decidir qué hacer ante los cambios de su ambiente, así como saber a dónde, cómo y cuándo migrar, son muestra del grado de autonomía de estos procesos.

Ser naturalmente heterogéneos. Los agentes móviles son generalmente independientes del tipo de transporte y cómputo, únicamente dependen de su ambiente de ejecución. Proveen condiciones óptimas para la integración de sistemas distribuidos.

Proporcionar ambientes robustos y a prueba de fallas. La habilidad que poseen los agentes para reaccionar dinámicamente ante las situaciones desfavorables de su ambiente hace más fácil el desarrollo de sistemas distribuidos robustos y tolerantes a fallas. El hecho de que los agentes móviles no requieran de conexiones permanentes y que sus estados estén centralizados en sí mismos, también favorece a que las fallas que se presenten puedan solucionarse de manera más sencilla que con el paradigma cliente - servidor por ejemplo.

Reducir el tráfico de red. Los agentes móviles permiten empaquetar un mensaje y despacharlo a un nodo destino, donde los procesos se llevan a cabo localmente. En el mensaje se envían los datos y procedimientos para la ejecución del proceso, el control del mismo puede llevarse a cabo por medio de supervisiones locales reduciendo así el tráfico de red, el cual es crucial si se considera que existen usuarios que se conectan con un ancho de banda muy bajo. Este aspecto es opuesto al paradigma de Llamados a Procedimientos Remotos (RPC), en donde se establecen dos enlaces de comunicación, uno para enviar al servidor la petición y argumentos necesarios y otro para recibir los resultados de la petición. La figura 2.1 ilustra esta ventaja.

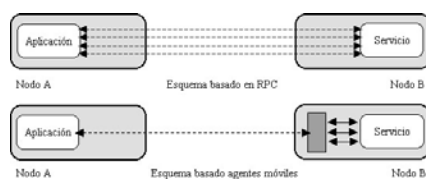


Figura 2.1 Agentes Móviles y reducción de tráfico en red.

(adaptado de [Lange y Oshima 1998])

Favorecer el procesamiento en paralelo. Un trabajo que requiera de alto poder de cómputo podría descomponerse en varios sub-trabajos que fueran asignados a agentes. Estos agentes ejecutarían dichas unidades de trabajo de forma distribuida, es decir, viajarían a otras máquinas en una red y ejecutarían la tarea en otros nodos; al final los resultados serían presentados al agente encargado de administrar el proceso en paralelo.

Mantener comunicación punto a punto. Una característica del paradigma cliente-servidor es la inhabilidad de los servidores para comunicarse. Los agentes móviles se consideran como entidades punto y, como tales, pueden adoptar la postura que más les convenga de acuerdo a sus necesidades. Por ejemplo, cuando un agente solicita un recurso éste toma el papel de cliente; sin embargo, cuando otro agente móvil le hace una consulta, éste se comporta como servidor. Esta característica resulta ventajosa en el contexto de ambientes distribuidos.

Poder operar sin conexión. Los agentes móviles tienen la capacidad de seguir trabajando aún cuando no exista una conexión abierta a red. Esto es posible gracias a que después de que los agentes son despachados, éstos llegan a ser independientes del proceso de creación y pueden operar en forma asincrónica y autónoma. La máquina que envió a dicho agente puede conectarse nuevamente para obtener resultados del mismo. Este aspecto se ilustra en la figura 2.2.

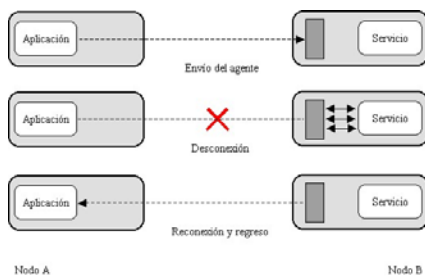


Figura 2.2 Agentes Móviles y operación de desconexión

(adaptado de [Lange y Oshima 1998])

Las desventajas más evidentes en el desarrollo de agentes móviles se refieren a cuestiones de seguridad y a algunas limitantes que presentan los lenguajes de programación para su diseño. En cuanto a seguridad se deben considerar la autenticación de agentes y servidores, así como la protección de los agentes durante su transmisión. Para mayores referencias consultar [Pérez 1998].

### 2.1.3 Lenguajes de programación para agentes móviles.

Uno de los lenguajes de programación más prometedores para el desarrollo de aplicaciones Web es Java, e inspirados en este lenguaje han surgido interesantes sistemas que soportan la implementación de agentes móviles. Entre los más conocidos se encuentran Mole, Odyssey, Aglets, Voyager y Concordia, cuyas principales características se mencionan a continuación.

**Mole.** Es un proyecto desarrollado por el grupo de Sistemas Distribuidos de la Universidad de Stuttgart, Alemania. Su tercera versión provee de un esquema de nombrado global para agentes, una unidad de manejo de hilos, un director de servicios y esquema de comunicación entre agentes. También considera los conceptos de control, coordinación y terminación de agentes y grupos de agentes, así como la seguridad de estas entidades con transacciones, y su ejecución en nodos dudosos. Mole está formado por varios subproyectos, entre los actuales se encuentran: AIDA, dedicado a la investigación de seguridad de agentes, y HAWK enfocado a la recuperación de información; los concluidos como ASAP y ATOMAS se relacionan respectivamente con la generación de documentos activos y con el desarrollo de agentes transaccionales [Baumann 1999].

**Odyssey.** General Magic Inc. fue el creador del primer sistema de agentes móviles denominado Telescript, cuyo periodo de vida fue muy corto pese a que estaba pensado para trabajar en una arquitectura de red. En respuesta a la popularidad de Internet y al gran auge del lenguaje de programación Java, General Magic decidió reimplementar todos los conceptos considerados en el desarrollo de Telescript pero ahora en este lenguaje. El resultado obtenido fue Odyssey, el cual es una librería de clases Java que permiten al usuario crear sus propias aplicaciones de agentes móviles [General Magic 1997].

**Aglets.** Es una librería de clases Java empleada para el desarrollo de agentes móviles. Fue creada por el Laboratorio de Investigación IBM de Tokyo, en especial por los investigadores Lange y Oshima. Un *aglet* es un objeto Java transportable y persistente que corre asincrónicamente en un nodo que posee un contexto de ejecución. El contexto de ejecución provee de un ambiente seguro, protegiendo tanto al nodo como al *aglet* de *aglets* maliciosos. Un *aglet* refleja el modelo de *applet* en Java pero brindándole la propiedad de movilidad. *Aglet* es una palabra que resulta de la combinación de los términos *agente* y *applet* y representa el siguiente paso en la evolución de contenido ejecutable en Internet [Lange y Oshima 1998].

**Voyager.** Es una plataforma escrita en Java para el desarrollo de cómputo distribuido basado en agentes y fue creado por ObjectSpace. Este provee un extenso conjunto de objetos con capacidades de mensajería, así como objetos que se mueven como agentes en una red. De esta forma se puede decir que Voyager permite crear aplicaciones de red usando las técnicas de programación convencionales o bien fundadas en agentes [ObjectSpace 1999].

**Concordia.** Es un marco de trabajo para el desarrollo y manipulación de aplicaciones basadas en agentes móviles desarrollado por Mitsubishi. Este consiste de múltiples componentes, todos basados en Java, que se combinan para proveer un ambiente completo para la generación de aplicaciones distribuidas. Considera a fondo el concepto de movilidad, seguridad, persistencia y colaboración [Mitsubishi 2000].

En la Tabla 2.1 se presentan de manera comparativa otras características tanto técnicas como prácticas de los sistemas antes mencionados. Esta tabla es una extensión del trabajo realizado por [Pérez 1998] y para su mayor comprensión a continuación se describen los siguientes términos:

**Tipo de migración.** Se dice que es fuerte cuando el agente migra con su código y estado de ejecución, y débil cuando se limita a código que llega de un agente y se liga dinámicamente.

Ambiente de trabajo. Se refiere al servidor o contexto que controla la ejecución de los agentes, o bien como al lugar en el que los agentes migran.

Mecanismo de migración. Es la forma bajo la cual los agentes consiguen viajar a través de los nodos. Esta puede ser por medio de serialización, RMI o CORBA.

Seguridad. En este contexto la seguridad considera distintos aspectos: la seguridad que existe entre los ambientes de ejecución, la que posee el ambiente y la que incluyen los agentes. La seguridad entre ambientes puede emplear esquemas de autenticación, la seguridad en ambientes aplica técnicas de control de acceso, mientras que los agentes consideran algunas técnicas de cifrado o identificación única.

Lenguaje	Autor	Tipo Migración	Seguridad	Contexto trabajo	Mecanismo Migración	Contacto con el usuario	Comunicación entre agentes	Requerimientos	Plataformas soportadas	Descargas disponibles	Docum. disponible
Mole	Univ. de Stuttgart	débil	contexto	sitio y máquina	RMI, CORBA	applets	mensajes sincronicos y asincronicos	JDK 1.1.3-1.1.6	-----	v. 3.0	mínima
Odyssey	General Magic	débil	contexto	----	RMI, CORBA, DCOM	agentes en hojas HTML	-----	JDK 1.1	Solaris, W95, WNT	-----	mínima
Aglets	IBM	fuerte	contexto, autenticación, derechos de acceso	aglet, context	Serialización Java, RMI	Tahiti, Fiji, Applets, Servlets	mensajes sincronicos y asincronicos	JDK 1.1.*	Solaris 2.5, SPARC, W95, WNT, Macintosh, Netscape4	v1.0.3, v1.1 beta, fiji no disponible	suficiente
Voyager	ObjectSpace, Inc.	fuerte	contexto	servidor	RMI, CORBA, DCOM	-----	mensajes sincronicos y asincronicos	JDK 1.2, 1.1	-----	v2.0.2	suficiente bajo costo
Concordia	Mitsubishi Electric ITA	fuerte	encriptación	sitio y máquina	-----	applets	-----	JDK 1.1.*	W95, WNT, Solaris	v1.1.2, v1.1.4	suficiente

Tabla 2.1 Tabla comparativa de lenguajes de programación para agentes móviles

Aunados a los lenguajes de programación para agentes móviles presentados anteriormente, también se pueden mencionar otros cuyas características favorecen su desarrollo. Sistemas como Agent Tcl, Ara, TACOMA, Telescript, Scheme y Python se describen detalladamente en [Castellanos y Sandoval 1997], [Pérez 1998] y [Lange y Oshima 1998].

#### 2.1.4 Aplicaciones.

Algunos de los campos idóneos en donde se pueden aplicar los agentes móviles son, desde el punto de vista de [Lange y Oshima 1998], los siguientes: procesamiento en paralelo, comercio electrónico, asistencia personal, recuperación de información distribuida, servicios de redes y telecomunicaciones, aplicaciones de software para grupos, monitoreo y notificación, y disseminación de información. Los agentes móviles también pueden aplicarse para generar software personalizable, software distribuido e incluso representar a usuarios desconectados [Milojicic et al. 1998].

Dado que el presente proyecto se basa en crear agentes móviles que sean capaces de ofrecer el servicio de recuperación en una biblioteca digital, en la sección siguiente se describirán brevemente algunos de los métodos más conocidos en cuanto a recuperación de información se refiere.

#### 2.2 Recuperación de información.

La recuperación de información [Salton y Michael 1983] busca organizar y administrar la información con el objetivo de extraer, de un vasto cúmulo de documentos, aquellos que sean relevantes para las necesidades o preferencias del usuario. Al mismo tiempo se pretende minimizar la cantidad de información irrelevante recuperada. A grosso modo, la recuperación consiste en apuntar sólo a los documentos que reflejan interés o utilidad.

En base a lo anterior se puede decir que uno de los principales servicios que brindan las bibliotecas digitales consiste en la recuperación de información. Ante este hecho es necesario desarrollar aplicaciones en las que se consideren factores claves como la relevancia que tienen los documentos recuperados ante el usuario, el constante y rápido cambio de la información, los lugares en donde buscarla, la incompatibilidad de algunos de sus formatos, su disponibilidad y tiempo de acceso, así como su ubicación geográfica [Cabrera 1997]. Aunado a estos factores podemos mencionar otro cuyos beneficios favorecen en gran medida al usuario final, este es el hecho de personalizar el servicio de recuperación de información.

##### 2.2.1 Recuperación de Información Personalizada.

Ubicados en el contexto de cómputo distribuido resulta obligatoria la siguiente pregunta: ¿Cuántas veces nos hemos pasado buscando información en la red durante horas y los resultados obtenidos han sido insatisfactorios? La respuesta suele ser fácil de contestar y es "la mayoría de las veces". Esta situación nos lleva a pensar que para la recuperación de información es necesario contemplar las necesidades de cada usuario. Lo anterior quiere decir que si se conocen los requerimientos, el perfil de los usuarios y su espacio personal, la información que se obtenga a través de la búsqueda debe reflejar sus necesidades, tal vez evitando así la pérdida de tiempo y minimizando la cantidad de información irrelevante recuperada. Con todo lo anterior se pretende que el usuario al delegar la tarea de recuperación a un agente, éste le ofrezca los resultados del servicio en su ambiente personal de una manera práctica y eficiente, y cuyo origen de los documentos sea digno de explorar.

##### 2.2.2 Modelos de Recuperación de Información.

El proceso de recuperación de información implica buscar, filtrar y obtener la información a partir de un vasto depósito de datos. Para lograr tal objetivo existen algunos modelos propuestos por [Salton 1988] conocidos como *recuperación booleana* y *espacios vectoriales*. Estos métodos realizan una búsqueda textual con el fin de obtener aquellos documentos que posean cierto grado de similitud con la solicitud del usuario.

###### 2.2.2.1 Modelo de recuperación booleana.

La aproximación booleana es la más común y sencilla de implementar en cuanto a recuperación de información se refiere, sin embargo presenta tres grandes inconvenientes [Iglesias et al. 1999] :

El operador lógico AND restringe demasiado la consulta, ya que cuando dos o más términos están conectados por este operador, todos ellos deben aparecer en el documento; por otro lado el operador OR libera demasiado la consulta, pues se recuperarán todos aquellos documentos en

los estén presentes al menos uno de los términos especificados.

Este modelo no permite ordenar los documentos recuperados según el grado de similitud con la consulta.

No se considera asignar factores de importancia o pesos a los términos conectados ya sea a los documentos o a las consultas, por lo que asume que todos los términos incluidos en los documentos y consultas tienen la misma importancia.

El modelo de espacios vectoriales elimina algunas de las desventajas de las búsquedas booleanas.

### 2.2.2.2 Modelo de espacios vectoriales.

Es un método de recuperación de información que se caracteriza por la sencillez de su conceptualización e implementación. Este algoritmo consiste en representar a los documentos y a los términos de las consultas como vectores, y en asignarle a cada una de sus dimensiones un peso que represente la importancia que ese término tiene con respecto a los demás. Se entiende por término a las palabras claves que conforman la consulta o bien, aquellas palabras que sean de considerable importancia y que describan el contenido del documento. La representación de dichos vectores puede ilustrarse en la Figura 2.3.

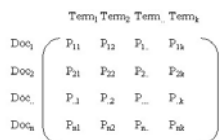


Figura 2.3 Matriz términos-documentos (adaptada de [Salton 1988])

La similitud entre dos documentos, o bien, un documento y una consulta, está determinada por el ángulo que forman sus vectores correspondientes en el espacio n-dimensional, donde n es el número de términos significativos [Iglesias 1998]. La parte medular de este modelo consiste en realizar el cálculo del coseno de dicho ángulo para hallar así el grado de similitud. Este valor representa la distancia que tiene el término con respecto al origen en un plano cartesiano. Entre más pequeño sea el ángulo, mayor similitud tendrán los respectivos documentos y consultas. La figura 2.4 muestra la fórmula para calcular el grado de coincidencia.

$$\text{Coseno}(\text{Doc}, \text{Cons}) = \frac{\sum_{k=1}^n (\text{DocTerm}_{k,a} * \text{ConsTerm}_{k,b})}{\sqrt{\sum_{k=1}^n (\text{DocTerm}_{k,a})^2 + \sum_{k=1}^n (\text{ConsTerm}_{k,b})^2}}$$

Figura 2.4 Ángulo de similitud en el método de espacios vectoriales.

(adaptado de [Salton 1988])

## 2.3 Trabajo previo.

En la actualidad se han desarrollado diversos proyectos que contemplan la integración de las áreas de bibliotecas digitales, recuperación de información y agentes. En esta sección se presentan algunos trabajos relacionados en estos campos con el fin de tener una visión más amplia de las ventajas, arquitecturas y resultados obtenidos por éstos. El proyecto desarrollado por [Dale 1998] propone una arquitectura de agentes móviles para el manejo de recursos de información distribuida. Trabajos como FRA [Cabrera 1997] y SAM [Pérez 1998] fueron realizados previamente por el grupo de Bibliotecas Digitales del Laboratorio de Tecnologías Interactivas y Cooperativas (ICT) de la Universidad de las Américas, en ellos se integran agentes al contexto de bibliotecas digitales.

### 2.3.1 Una arquitectura de agentes móviles para soportar el manejo de recursos de información distribuida.

Este proyecto considera la integración de dos tecnologías; el manejo de información distribuida para proveer herramientas que le permitan al usuario descubrir, navegar y controlar información, y la tecnología de agentes móviles que le permita al usuario acceder y manejar información que se encuentra distribuida en diversas áreas. [Dale 1998] presenta un marco de trabajo en el cual los agentes móviles viajan a través de ambientes distribuidos, interactúan con recursos locales y otros agentes móviles, y comunican los resultados de sus actividades al usuario a través de interfaces basadas en agentes.

El marco de trabajo descrito está basado en un modelo orientado a agentes, esto quiere decir que cada una de las partes que conforman el sistema es considerado como un agente independiente (Figura 2.5). El sistema está formado tanto por agentes móviles como estáticos, éstos últimos proveen los recursos y facilidades para los móviles y éstos a su vez son los que migran a través de los dominios de la red para tomar ventaja de estos recursos y cumplir con sus objetivos.

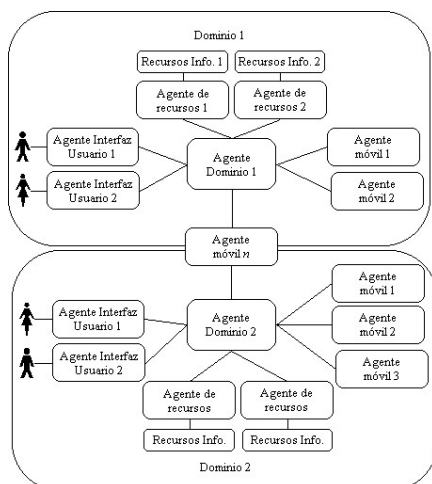


Figura 2.5 Arquitectura de agentes móviles para soportar el manejo de información distribuida. (adaptado de [Dale 1998])

Los principales componentes de este marco de trabajo son:

**Agente de dominio.** Es un agente estático que supervisa las actividades que ocurren en el dominio o nodo. Este agente tiene la responsabilidad de controlar los recursos de información, y los agentes y usuarios en el dominio dado. Brinda el servicio de migración a los agentes móviles que deseen abandonar el nodo, provee la autenticación y validación de los agentes que deseen entrar al dominio, hospeda a los agentes móviles en un ambiente de ejecución, provee de un ruteador central de mensajes y sirve de mediador para el acceso global de los recursos de información.

**Agente de recursos.** Es un agente estático que existe en el dominio y que sirve de intermediario entre los agentes móviles y los recursos de información. Este agente conoce la forma de cómo interrogar a los recursos y de cómo acceder las estructuras asociadas a dichos recursos. Es capaz de brindar un conjunto de servicios genéricos y estándar que todos los agentes móviles pueden utilizar, advierte la presencia de los recursos de información al registrar los servicios disponibles en el agente del dominio, también provee la conversión entre la representación de información de los agentes móviles y la representación de información de los recursos que el agente de recursos está manejando.

**Agentes móviles.** Como su nombre lo indica, estos agentes son los que migran a través de los nodos de la red para conseguir la información que el usuario requiere. Los agentes móviles son equipados con los objetivos y límites definidos por el usuario, pero se puede dar el caso en el que también su ejecución se vea limitada por cuestiones de seguridad y autenticación de los nodos a los que visiten.

**Agente de interfaz de usuario.** Es un agente ubicado en el dominio que sirve de intermediario entre el usuario y el marco de trabajo de los agentes móviles. Este agente es capaz de presentar una ventana en la que se puedan consultar tanto resultados como estados de los agentes; también puede interpretar las necesidades del usuario y generar agentes móviles para cubrir sus requerimientos

**Agente de puerto.** Es una entidad estática en el dominio y provee acceso a otras redes o a un dominio restringido. Para lograr su objetivo manipula y estandariza protocolos. Se encarga de controlar los problemas relacionados con la heterogeneidad de las redes y las barreras de seguridad (firewalls).

Las herramientas empleadas para la implementación de esta arquitectura fueron las siguientes: el lenguaje de programación Java con el cual se desarrollaron los agentes móviles; y el lenguaje ACL (Lenguaje de Comunicación de Agentes) para controlar lo referente a la comunicación entre agentes. ACL está formado por tres componentes: un vocabulario, un lenguaje de comunicación estándar entre los agentes denominado KQML (Knowledge Query Manipulation Language), y un lenguaje de contenido llamado KIF (Knowledge Interchange Format) el cual sirve en este caso para manipular la base de conocimiento de los agentes.

### 2.3.2 FRA

Flora Retrieval Agent es un agente desarrollado por [Cabrera 1997] que se incorporó a la iniciativa denominada Biblioteca Digital Florística o "FDL" con el objetivo de mejorar la búsqueda y recuperación de información relevante a los intereses y preferencias del usuario. FRA filtra y elimina aquella información botánica que no considera de interés para el usuario. Este agente tiene la capacidad de trabajar de manera semi-autónoma porque el usuario en cualquier momento puede desactivarlo o suspenderlo, el usuario tiene la capacidad de calificar los documentos recuperados mejorando con esto próximas recuperaciones. El agente cuenta con el servicio de notificación vía correo electrónico para cuando nuevos resultados hayan sido obtenidos. La Figura 2.6 muestra la pantalla principal de este agente.



Figura 2.6 Pantalla principal de FRA; el agente en estado activo.

(adaptado de [Cabrera 1997])

El esquema general del agente FRA está compuesto por 6 módulos que se interrelacionan entre sí enviándose mensajes. Para implementar el agente se tomó en consideración el ambiente distribuido en el que va a estar inmerso y el acceso a una base de datos vía remota. El agente se desarrolló en Java accediendo al manejador de base de datos Illustra por medio del controlador mijdbc [Kalvelagen 1997].

FRA puede recuperar información botánica a partir de una biblioteca digital florística mediante el modelo de espacios vectoriales. Este agente tiene la capacidad de presentarle al usuario los resultados de una forma clara y ordenada, además de que le permite ver el documento completo al que se refiere un resultado en especial; este aspecto se ve ilustrado en la Figura 2.7. FRA permite al usuario calificar a su modo de ver los resultados que el agente le presenta con el objetivo de ir refinando las búsquedas y acercarse lo más posible a los gustos del usuario. Tiene la capacidad de usar un archivo de preferencias en el que el usuario puede definir con toda libertad las palabras claves que le interesan para darle orientación a la recuperación de la información.

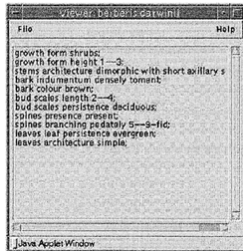


Figura 2.7 Pantalla que muestra el contenido completo de un documento.

(Adaptado de [Cabrera 1997])

### 2.3.3 SAM

Este proyecto desarrollado por [Pérez 1998] introduce el uso de agentes móviles en el proceso de búsqueda y recuperación de información en una Biblioteca Digital Florística altamente distribuida. Los agentes viajan a través de diferentes repositorios de datos y recuperan la información empleando el modelo de Recuperación Booleana Extendida. La información obtenida es relevante para el usuario puesto que se conocen previamente sus necesidades y perfil. Después de presentar los resultados al usuario, éste puede retroalimentar al agente calificando cada una de los documentos desde excelentes para afinar el proceso de recuperación, hasta pésimos para mejorarla.

A este conjunto de componentes de FDL relativos a agentes móviles que se desarrollaron se les denominó Sistema de Agentes Móviles (SAM). Los elementos principales de SAM son 5:

**Módulo de Interfaz.** Genera la interfaz gráfica del agente por medio de la cual el usuario se comunica con los otros módulos del sistema. A partir de este módulo el usuario puede definir su archivo de preferencias, configurar al agente y mandar al agente a recuperar información. Una vez que el usuario posee los resultados, este módulo le da la posibilidad de recuperar el documento completo y revisarlo en pantalla, así como poder calificar los resultados obtenidos.

**Módulo de Configuración.** En él se definen los parámetros requeridos por SAM tanto como para poder migrar como para poder localizar información relevante.

**Manejador del Perfil de Usuario.** Permite que el usuario recupere y modifique su archivo de preferencias que incluye estructuras morfológicas y su respectivo peso, conformando así el perfil del usuario.

**Módulo de recuperación.** Ejecuta la migración hacia cada uno de los nodos que conforman el itinerario y la búsqueda, filtrado y recuperación de la información en base al Modelo de Recuperación Booleano Extendido. Una vez obtenidos los resultados regresa al nodo origen. La información recuperada la muestra en la interfaz de una manera clara y le permite al usuario consultar el documento o bien calificarlo.

**Módulo de Retroalimentación.** Se actualiza cuando el usuario evalúa los resultados presentados. Se encarga de actualizar los pesos de las palabras clave del archivo de preferencias. Las pantallas alusivas a la presentación general de resultados y a la retroalimentación, así como la presentación de documentos completos se muestran en las Figuras 2.8 y 2.9

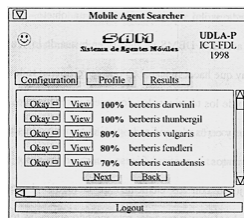


Figura 2.8 Pantalla de Resultados. (adaptado de [Pérez 1998])

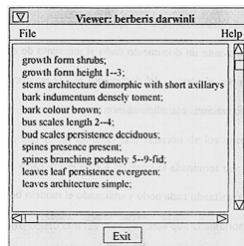


Figura 2.9 Documento Completo (adaptado de [Pérez 1998])

Las herramientas utilizadas para la implementación de SAM fueron: el lenguaje de programación de agentes móviles Aglets WorkBench [Lange y Oshima 1998], Java, CGI, HTML y SQL para Illustra.

### 2.4 Trabajos previos y Viajerus

Los proyectos FRA y SAM descritos anteriormente, nos dan una clara idea de las ventajas que ofrece el hecho de incorporar agentes estáticos o móviles al contexto de bibliotecas digitales para ofrecer un servicio de recuperación de información. Considerando dichos beneficios, Viajerus también integra algunos elementos de la arquitectura definida por [Dale 1998] para el proyecto U-DL-A, de tal forma que su diseño sea capaz de ofrecer el servicio de recuperación en forma transparente para el usuario, que permita presentarle resultados precisos y que facilite la

explotación de los acervos de otras colecciones de tesis digitales federadas.

En el siguiente capítulo se presenta el diseño de Viajerus, una arquitectura que soporta la creación de agentes móviles recuperadores de información en el contexto de la biblioteca digital U-DL-A.

sinopsis índice 1 2 3 4 5 6 A B C D E bibliografía



Chevalier Dueñas, C. A. 2000. Agentes móviles para la recuperación personalizada de información. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Mayo. Derechos Reservados © 2000, Universidad de las Américas Puebla.

