

Revisión: tecnología de agentes de *software*

Gabriel Hernan Tolosa
Fernando Raul Alfredo
Bordignon

INTRODUCCION

El concepto de agente

La palabra "Agente" se refiere a todo ente que posee la habilidad, capacidad y autorización para actuar en nombre de otro. A diario, los agentes humanos asisten a las personas en tareas que requieren recursos especializados o conocimiento específico en un dominio. Por ejemplo, una secretaria atiende y resuelve situaciones en nombre de su jefe: administra la agenda, coordina las reuniones, recibe a los visitantes.

Los agentes cumplen con los requerimientos para los cuales fueron entrenados. El usuario "delega" en el agente una o varias tareas que debe llevar a cabo quedando a la espera de los resultados. Dichas tareas son a menudo fáciles de especificar, pero - en algunos casos - complejas de realizar.

Los investigadores en el campo de los agentes computacionales han dado varias definiciones al término, cada uno desde su óptica particular, fundamentada básicamente en la línea de investigación en la cual trabajan (Inteligencia Artificial, Ingeniería de Software, Sistemas Autónomos)

Wooldridge y Jennings (Wooldridge, 1998) definen agente como "*un programa autocontenido capaz de controlar su proceso de toma de decisiones y de actuar, basado en la percepción de su ambiente, en persecución de uno o varios objetivos*".

Según Nwana (Nwana, 1996) "*El término agente se refiere a un componente de software y/o hardware que es capaz de actuar para poder ejecutar tareas en nombre de un usuario*".

Pattie Maes (Maes, 1994), del AI Lab

del MIT (Massachusetts Institute of Technology) definió a los agentes autónomos como sistemas computacionales que habitan en algún ambiente dinámico y complejo, sensando su estado y actuando autónomamente, llevando a cabo una serie de objetivos o tareas para los cuales fueron diseñados".

Según Frankling y Graesser (Frankling, 1996): "*Un agente autónomo es un sistema situado dentro (o es parte) de un medioambiente, que sensa tal medio ambiente y actúa sobre éste, por un tiempo, en persecución de su propia agenda y afectando así lo que sensará en el futuro*".

Russell y Norvig (Russell, 1995): "*Un agente puede ser visto como algo que percibe su ambiente a través de sensores y actúa contra este ambiente a través de efectores*"

Nicholas Negroponte, director del Media Lab del MIT, hizo la observación que el futuro de la computación será de "*delegar a*" y no de "*manipular*" computadoras. Dentro de este marco el término agente se redefine para cada aplicación que persiga dicho objetivo.

Propiedades de los agentes

En base a todas las definiciones anteriores es posible extraer algunas características que deben tener los agentes: deben ser parte de un ambiente, deben poder sensar su entorno y actuar sobre él, y deben responder según los objetivos para los cuales fueron diseñados. Deben - entonces - poseer una serie de atributos o propiedades que lo definen como agente (Wooldridge, 1996):

Autonomía: Capacidad de actuar sin la intervención directa de una persona o de otro agente. Un agente debe poder

Resumen

Los agentes de software en la era de las redes globales son una herramienta vital para superar el fenómeno llamado "sobrecarga de información". El grado de madurez alcanzado en esta tecnología permite que hoy se puedan ver aplicaciones concretas funcionando en organizaciones, como así también en el escritorio del usuario hogareño. El objetivo de este trabajo es presentar una revisión bibliográfica sobre la tecnología de agentes de software, con orientación a los modelos que permiten gerenciar la sobrecarga de información.

Palabras-clave

Agentes de software; Recuperación de información; Revisión.

controlar sus propias acciones y estado interno. Una vez que el usuario activa el agente indicando algún objetivo de alto nivel, éste actúa independientemente, seleccionando estrategias y monitoreando el progreso en busca de la meta. Si falla con una estrategia, usará otra, pero sin intervención humana o con la mínima indispensable.

Habilidad Social: Un agente debe ser comunicativo. Debe tener habilidad para interactuar con otros agentes o incluso con alguna persona, para solicitar información o bien para exponer los resultados obtenidos de la ejecución de las tareas agendadas. La naturaleza de la comunicación dependerá del tipo de agente con quien se comunique (humanos o no), en ambos casos se deberá establecer un protocolo común de intercambio de información entre ambas partes.

Los agentes deben poseer algún tipo de interface para comunicarse con sus usuarios. Dicha interface puede establecerse simplemente mediante el envío de mensajes por correo electrónico o puede ser todo lo sofisticada que uno quisiera (el deseo es proveer una interface más antropomórfica [human-like] para los agentes)

Reactividad: Se refiere al hecho de que un agente debe poder sentir el estado del ambiente dentro del cual se encuentra inmerso y -en función de esto- actuar, respondiendo de manera adecuada a cambios producidos en el mismo. Los efectos producidos pueden modificar el estado de su entorno. Por ejemplo, un agente de filtrado de correo electrónico está sensando su ambiente, alerta a la llegada de nuevos mensajes. La llegada de un nuevo mensaje (cambio en el medio) produce que el agente responda de acuerdo a las tareas asignadas.

Orientación por objetivos: Un agente no solo debe actuar por cambios detectados en el medioambiente, sino que – además – debe “trabajar” en

función de los objetivos para los cual fue diseñado y las tareas que le fueron delegadas en cada momento.

Un agente busca permanentemente satisfacer su agenda interna. En las aplicaciones convencionales, cuando un usuario ejecuta un comando, se activa un proceso que se ejecuta por determinado tiempo, algunos solicitan datos al usuario, retornan resultados y así sucesivamente. Esta es una visión pasiva de computación, ya que la motivación viene de parte del usuario. En un modelo basado en agentes, el proceso es auto-motivado, es decir, el agente busca satisfacer cierto estado interno con mínima intervención humana.

Por ejemplo, un agente recuperador de archivos tiene especificada una tarea. El agente deberá intentar permanentemente satisfacer la tarea que le fue delegada en base a las estrategias de búsqueda y recuperación con las cuales fue construido, hasta cumplir con el objetivo. Hasta tanto, el agente debe mostrar un comportamiento activo, que lo lleve a tomar la iniciativa de la acción a tomar cuando lo considere apropiado.

Existen – además – otros atributos de los agentes (Franklin, 1996):

Continuidad temporal: Un agente es un proceso temporalmente continuo. A diferencia de un programa convencional del cual se conoce su inicio y fin, un agente debe ejecutarse hasta que se haya alcanzado con el conjunto de objetivos solicitados, o bien, mientras su ciclo perdure y su usuario no desee detenerlo. La continuidad temporal es la propiedad que da “vida” al agente, posibilitando que se mantenga alerta a una solicitud o a algún cambio en el medio. El ciclo de vida de un agente depende de sus características, de las tareas que realice y de los deseos de su usuario en cuanto al tiempo durante el cual el agente debe ejecutarse.

Movilidad: Es la capacidad de un agente de “viajar” por las redes de computadoras, de nodo a nodo, en busca de los recursos que le permitan cumplir con su agenda. El término “viajar” se refiere a que en un determinado instante de tiempo el

agente detiene su ejecución, almacena su estado interno y se dirige a otro sitio dentro de una red de computadoras (tanto el código del agente como su estado) para luego continuar con su ejecución en la nueva ubicación.

Los recursos a los que puede acceder pueden ser de software o hardware. Por ejemplo, un agente puede ir en busca de una base de datos que se encuentra en cierta máquina o bien ejecutarse remotamente haciendo uso del procesador y memoria del equipo destino.

La movilidad no es una propiedad indispensable para un agente, sino que modifica la forma por la cual el agente cumple con sus objetivos, en este caso recurriendo a los recursos que puede ofrecer una red de computadoras. Aporta una nueva forma de computación distribuida.

Clasificación de los agentes

Los agentes pueden clasificarse de varias maneras, teniendo en cuenta algunas de las propiedades que poseen o bien haciendo hincapié en alguna en particular. De esta manera puede armarse un árbol taxonómico que abarque todas las combinaciones de propiedades y tareas que se quieran. En vez de esto se propone una clasificación de acuerdo a líneas de investigación y desarrollo de agentes, donde lo que se prioriza la función u objetivo principal del agente. Puede – entonces – hacerse la siguiente clasificación de los agentes (Berney, 1996):

- Agentes de interface
- Agentes colaborativos
- Agentes móviles
- Agentes de recuperación de información

Agentes de interface

Un agente de interface es un software cuasi-inteligente que asiste a un usuario cuando interactúa con una o más aplicaciones. La motivación es que se les pueda delegar tareas aburridas y laboriosas. Son asistentes

personales que reducen el trabajo por la sobrecarga de información, como por ejemplo el filtrado de los mensajes de correo electrónico o la recuperación de archivos de Internet (Nwana, 1996).

Esta categoría de agentes apoyan y proveen asistencia a su usuario. El agente observa y monitorea las acciones que toma el usuario en la interface, aprende nuevos atajos, y sugiere mejores formas de hacer las tareas. La idea es que el agente pueda adaptarse a las preferencias y hábitos de sus usuarios.

Enfatizan la autonomía y el aprendizaje para llevar a cabo tareas para sus dueños y trabajan en el mismo ambiente que éstos (Nwana, 1996).

A su vez, de los agentes de interface pueden encontrarse subdivisiones debido a diferentes tareas para las cuales son construidos. Las más comunes son:

- **Asistentes:** Trabajan realizando tareas típicas como el manejo de la agenda. Estos agentes ayudan al usuario a planificar las reuniones. Sus acciones incluyen negociar, aceptar o rechazar reuniones.

- **Filtros:** Su tarea principal es la de analizar información según un conjunto de reglas dadas por el usuario. La aplicación típica es el filtrado de mensajes de correo electrónico.

- **Guías:** Asisten a los usuarios en el uso de una aplicación. Estos agentes monitorean las acciones de los usuarios e intentan sugerir qué pasos realizar para alcanzar el objetivo. Algunos ayudan a navegar por la World Wide Web.

Agentes colaborativos

Los agentes colaborativos constituyen un sistema multiagentes, es decir existe más de un agente dedicado a satisfacer los requerimientos de sus usuarios. Para ello es necesario contar con esquemas de comunicación entre agentes que posibiliten la cooperación y el intercambio de conocimiento.

Además, deben poseer un alto grado de autonomía para actuar interactuando con sus pares.

La motivación detrás de la construcción de agentes colaborativos es que los sistemas construidos con unidades relativamente simples proveen mayor funcionalidad que un ente mayor, pudiendo extender la funcionalidad del sistema más allá de las capacidades de uno de sus miembros.

Además, estas arquitecturas posibilitan contar con mayor confiabilidad (debido a la redundancia) y mayor velocidad (debido al paralelismo) en el sistema conjunto (Berney, 1996).

Las áreas de aplicación de este tipo de agentes incluyen:

- Resolución de problemas demasiado grandes.
- Interconexión de múltiples sistemas.
- Manejo de información proveniente de fuentes distribuidas.

Agentes Móviles

Los agentes móviles son procesos capaces de "viajar" por una red de computadoras, interactuando con hosts externos, recolectando información en nombre de su dueño y retornando a "casa" luego de completar las tareas establecidas (Nwana, 1996).

Los agentes forman un nivel de abstracción más para el usuario, detrás del cual se encuentran soluciones a cuestiones técnicas en algunos casos complicadas. Una de estas cuestiones es la distribución, es decir, como manejar recursos computacionales distribuidos. Con la idea de agentes móviles los recursos distribuidos no son completamente ocultos al usuario pero tampoco completamente expuestos (William, 1996).

La noción de movilidad viene del objetivo de reducir el tráfico innecesario dentro de una red, con lo que se pueden reducir los costos de comunicación. Además, al aportar una nueva forma de computación distribuida posibilita el mejor aprovechamiento de los recursos

de la red y permite que los usuarios tengan acceso a una cantidad mayor de recursos.

Por ejemplo, debido a que las sesiones en busca de un recurso determinado ciertas veces son largas, la idea de agentes móviles provee una solución. Un usuario delega la tarea de búsqueda de información a un agente, establece una comunicación con la red y "envía" al agente a cumplir con su misión. La próxima vez que el usuario se conecte, el agente "retorna" con los resultados obtenidos.

Para soportar la movilidad, debe existir una infraestructura de transporte que mueva el código del agente de una ubicación a otra. Además, se debe contar con un entorno de ejecución de agentes, donde los agentes "viven", compuesto por todas las computadoras que los proveen.

Finalmente, para construir sistemas con agentes móviles es necesario resolver algunas cuestiones fundamentales tales como (Nwana, 1996):

- **Transporte:** Cómo se mueven de lugar en lugar?

- **Ejecución:** Cómo ejecutar el agente de forma remota?

- **Autenticación:** Cómo saber si el agente es quien dice ser y a quién representa?

- **Privacidad:** Cómo asegurar que el agente mantenga resguardado su estado interno?

- **Seguridad:** Cómo protegerlo de virus? Cómo prevenir que el agente entre en bucles infinitos o falle?

Agentes de recuperación de información

El objetivo principal de los agentes dedicados específicamente a la recuperación de información es obtener información por el usuario. La motivación para su construcción es que con el crecimiento vertiginoso de Internet, la cantidad de información

accesible supera la cantidad de tiempo disponible para analizarla. John Naisbitt, de Megatrends, dijo: “*Nos estamos sumergiendo en información pero hambrientos de conocimiento*”.

Las tecnologías de la información han expandido los horizontes de los usuarios en cuanto a las formas de generar y acceder a la misma. Pero esta amplia variedad de información distribuida plantea desafíos en cuanto a las formas de manejar su complejidad y heterogeneidad.

Hoy en día, la información se produce en múltiples contextos, se difunde por medios muy variados y se utiliza en todas partes. El rápido crecimiento de la cantidad de documentos en Internet (especialmente en la World Wide Web) presenta la dificultad de poder acceder a la información relevante.

Se produce así el fenómeno conocido como “sobrecarga de información”, entonces se trata de mejorar, pero no eliminar, el problema específico de la sobrecarga y administración de la información. Además, existe el problema que debido al gran volumen de información disponible, se mezclan el ruido o desperdicio con la información útil o necesaria, lo que determina la gran dificultad de hallar lo realmente resulta de interés para las personas.

Las soluciones actuales a este problema se basan en la construcción de motores de búsqueda, con mecanismos de indexación de documentos, combinados con interfaces de consulta apropiadas a esta tarea, o bien, índices manuales multi-nivel (o directorios), los cuales presentan clasificaciones de los documentos según el criterio de sus autores [Torrissen, 1998]. Ambas técnicas poseen debilidades visibles. Los primeros, son muy propensos a “recuperar” demasiada cantidad de documentos “no deseables” ya que trabajan (básicamente) mediante técnica de búsqueda de la ocurrencia de los términos buscados en los documentos. Por otra parte, los índices manuales, solamente tienen “parte” de los posibles documentos, debido al alto costo que posee la recuperación, manipulación y clasificación manual de

los mismos.

La motivación es poder diseñar una técnica que permita describir los cientos de millones de documentos disponible de manera precisa, creando un índice de alta calidad, con una forma eficaz y eficiente de acceder a éste (ya sea de manera manual o automática). Una de las soluciones posible se basa en los agentes de recuperación de información. Estos agentes pueden asistir a un usuario novato en la formulación de consultas avanzadas, en base a sus necesidades de información. Además, permiten acceder e integrar fuentes heterogéneas y manejar diferentes tipos formatos de información.

Los agentes de recuperación de información poseen métodos para permitir el rápido acceso y recuperación de información relevante. Tienen la tarea de administrar, manipular y juntar información de fuentes distribuidas. Pueden tener mecanismos de búsqueda y navegación flexibles y algoritmos de clasificación poderosos.

El objetivo es construir agentes capaces de “armar” un diario personalizado, sabiendo dónde buscar, cómo encontrar lo buscado y cómo armarlo luego (Nwana, 1996). Los agentes se presentan como una herramienta muy útil en la tarea de resolver el problema de la sobrecarga de información, debido a que éstos pueden realizar sus tareas mucho más rápido que las personas y, además, se encuentran disponibles las veinticuatro horas.

Una aplicación posible: Los agentes de biblioteca

Un interesante ámbito de aplicación de los agentes de recuperación de información son las bibliotecas. Las colecciones de información publicados almacenados en formato electrónico ha aumentado considerablemente, por lo cual las bibliotecas deben incorporar este formato como una alternativa a las publicaciones tradicionales.

Dentro de este ámbito, los agentes de recuperación de información pueden

resultar de gran utilidad como asistentes de los bibliotecarios, manejando los grandes volúmenes de material electrónico almacenado. Dado un contexto de búsqueda (según la petición de material por parte de un usuario), el bibliotecario puede recurrir a un agente para que lo asista en las tareas de buscar, clasificar y filtrar la información solicitada. En base a una solicitud, el agente puede recurrir a estrategias para seleccionar las bases de datos donde buscar y cómo categorizar los documentos encontrados.

Además, pueden colaborar con el bibliotecario en [Zick, 1999]:

1. recuperar información de fuentes distribuidas (internet);
2. realizar consultas con distintas estrategias;
3. generando reportes;
4. creando perfiles de usuarios, con base en sus áreas temáticas o preferencias de material;
5. manteniendo estadísticas de solicitudes/búsqueda/utilidad.

En los próximos años, es impensable una biblioteca que no incorpore asistentes digitales. La diversidad de soportes, formatos, idiomas, motores de consulta ,etc están a disposición de los usuarios con la finalidad de brindar una mayor calidad de servicio informativo, pero él usuario debe abstraerse de cuestiones tecnológicas que lo dispersen de sus objetivos principales. Esta última cuestión es la que tienden a realizar los asistentes o agentes de bibliotecas, operando como interfaces inteligentes entre el usuario y las tecnologías de información.

Ejemplos de modelos de agentes:

Websailor

Es un agente del área de recuperación de información, posibilita a sus usuarios atacar el problema conocido como sobrecarga de información y su consecuente baja eficiencia a la hora de realizar consultas por medio de los modelos tradicionales. Websailor

(Jaccard, 1999) actúa como intermediario entre el usuario y el espacio World Wide Web. Analiza el comportamiento de los usuarios al navegar, determinando la relevancia de cada una de las páginas accedidas, a los efectos de mantener un perfil. En cada nueva consulta toma ventaja de estos datos acerca de las preferencias del usuario, a los efectos de poder filtrar y reorganizar los links devueltos por buscadores tradicionales. Como interface de consulta genera páginas de respuesta adaptadas a los hábitos del usuario.

Se implementa del lado del usuario en la forma de un applet que un servidor proxy (adaptado especialmente) incluye en cada una de las páginas HTML que los navegadores soliciten. De esta forma se pueden capturar datos tales como: tiempo de lectura, utilización de vínculos, impresión o grabación de las páginas, etc. Este proyecto es parte de una línea de investigación que tiende a detectar y aprovechar al máximo los hábitos de los usuarios con la finalidad de incorporar este conocimiento a asistentes personales.

Softbot

Softbot (Etzioni, 1994) es un ejemplo clásico de agentes de interface, de la subcategoría de los asistentes personales, desarrollado en la Universidad de Washington. Utiliza un shell UNIX y permite interactuar con diversos servicios de Internet. Sus efectores usan algunos servicios como ftp, telnet, mail y sus sensores otros como netfind, archie, gopher.

Softbot provee una interface integrada para utilizar los recursos de Internet, elige dinámicamente cuáles servicios invocar y en qué secuencia. Además, cambia su comportamiento en base a la información obtenida del medio (por ejemplo si se encuentra desabilitado un servicio intentará con otros).

Como actúa como un asistente personal inteligente, el softbot permite que el usuario especifique una solicitud de alto nivel. Luego utilizará búsquedas, inferencias y conocimiento para determinar cómo satisfacer el requerimiento. La interface permite

completar una especie de formulario que acepta incompletitud de datos (en algunos casos) que luego deberá tratar. Dicha interface se diseñó en base a los siguientes principios:

- *Orientada por objetivos*: La solicitud expresa "qué" desea el usuario. El "cómo" obtenerlo y "cuando" satisfacer el pedido es responsabilidad del softbot.
- *Flexible*: Cuando un objetivo no está completo o correcto, se intenta descifrarlo para luego satisfacerlo.
- *Balanceada*: El softbot balancea el costo de encontrar la información contra el tener que "molestar" al usuario con mas preguntas.
- *Integrada*: Provee una interface uniforme a una amplia gama de servicios de Internet.

Ejemplo de utilización: Supóngase la necesidad de enviar un grupo de archivos sobre un tema a una determinada persona de una universidad. Supóngase además que no se sabe la dirección de correo de dicha persona. La interface del softbot permitirá completar los datos que se conocen (Apellido y nombre de la persona, universidad, nombre de los archivos a enviar, ubicación). Ya en acción el softbot puede emplear algún servicio tipo netfind para tratar de localizar a la persona, o bien revisar el site de la institución en busca de la misma. Una vez localizada puede enviarle el grupo de archivos vía mail (si lo es que obtuvo su dirección) o vía ftp (si lo es que obtuvo es su directorio dentro del site).

Virtual Mattie

Otro ejemplo de asistente personal es Virtual Mattie (VMattie) (Franklin, 1996). Es un agente autónomo e inteligente, que "vive" en un ambiente UNIX. Está programado en Java y Perl y realiza las tareas de un bedel, liberando a la persona y aumentando la eficiencia. Trabaja sin intervención humana y se comunica mediante e-mail, utilizando lenguaje natural.

VMattie cada semana envía un e-mail con el anuncio de los seminarios agendados en un departamento académico. Para ello, se comunica con

los organizadores de los seminarios, arma los anuncios y los envía a todos los integrantes de una lista de correo que él mantiene.

Si bien la tarea no es complicada para el bedel real, requiere un alto nivel de percepción para reconocer e interpretar el contenido de los mensajes de los organizadores y los interesados, los cuales poseen un formato libre. Además, debe manejar fechas y componer distintos tipos de mensajes a distintas personas: anuncios, erratas, recordatorios, confirmaciones.

La arquitectura de VMattie consta de tres módulos centrales: Objetivos, Comportamientos y Registros de Atención.

VMattie posee varios objetivos operando en paralelo:

- 1) obtener a tiempo la lista de seminarios en la semana;
- 2) mantener completa la información acerca de cada seminario;
- 3) mantener actualizada la lista de correo;
- 4) confirmar cada mensaje entrante.

Estos objetivos tienen niveles de urgencia variados conforme lleguen mensajes, y se aproxime la fecha de envío del anuncio del seminario.

Los comportamientos son acciones de nivel medio, por ejemplo:

- 1) agregar-dirección-a-lista;
- 2) asociar-organizador-con-seminario;
- 3) componer-recordatorio.

Los registros de atención mantienen información extraída de los mensajes entrantes, por ejemplo:

- 1) nombre-del-organizador;
- 2) dirección-de-correo;
- 3) fecha;

4) orador;

5) nombre-del-seminario.

Existe un registro especial el cual registra el tipo de mensaje que ingresó. El dominio de los tipos de mensajes en VMattie es limitado. Por ejemplo, solicitudes de baja de una lista de e-mail, mensajes de los organizadores enviando el título de un seminario y/o el nombre del disertante. La información en los registros de atención provee un entorno de activación para los comportamientos, que operan con su contenido.

Un módulo especial (módulo de procesamiento de conocimiento) se encarga de interpretar los mensajes, "entendiendo" varios formatos de algunos vocablos que pueden escribirse completos o abreviados, con mayúsculas o minúsculas. La información almacenada en este módulo se obtuvo del análisis del cuerpo de los mensajes recibidos por el bedel real en dos años. Se identificaron distintos tipos de mensajes, abreviaturas utilizadas, costumbres y tratamientos, títulos, etc.

Luego de interpretar un mensaje entrante, VMattie envía un mensaje de confirmación con el contenido de los que entendió del mensaje, e invita a enviar las correcciones necesarias.

Finalmente, una base de conocimiento contiene las plantillas para el armado de los mensajes a enviar (al igual que con los mensajes entrantes existen unos pocos tipos de mensajes posibles de enviar), como así también la información necesaria para completar dichos mensajes y la lista de las direcciones de correo de los destinatarios. Esta base es actualizada por los comportamientos.

Visitorbot

La tarea de agendar las visitas a un laboratorio suele ser bastante sencilla y rutinaria, pero consume demasiado tiempo. La secuencia normal de

actividades que involucra incluye la recepción de mensajes con las solicitudes de visita, la lista de las personas a quienes se desea ver y los horarios preferidos. Luego, es necesario avisar a dichas personas solicitando la respuesta y sus horarios. Finalmente, se debe armar – según las restricciones impuestas – la visita, agendarla y enviarla a todos los participantes indicando lugar, teléfono y demás datos.

Un agente llamado VisitorBot (Kautz, 1994) se especializa en manejar la tarea antes descrita. Inicialmente, el usuario se comunicaba directamente con el agente vía e-mail, lo que permitió observar lo siguiente:

- La comunicación vía e-mail resultaba difícil ya que se debían escribir los mensajes de una forma predefinida y el menor error de tipeo ocasionaba el mal funcionamiento, lo que determinó que la siguiente versión utilizaría una interfaz gráfica.

- Era necesario utilizar rutinas redundantes para el manejo de errores, por mensajes incompletos o incorrectos.

- La tarea de crear una buena agenda no era una tarea que requiera planificación avanzada sino que podía traducirse a un problema de programación entera.

Estas observaciones condujeron al desarrollo de "userbots", es decir, agentes (de interface) que se comuniquen con el usuario para permitir especificar los requerimientos y que luego los envíen al Visitor Bot de una forma normalizada. Esto se tradujo en la construcción de un sistema de agentes colaborativos destinados al objetivo final.

La interacción normal entre el usuario y el Visitor Bot se realiza de la siguiente forma. El Visitor Bot envía un mail a cada userbot. Éste determina el tipo de comunicación posible con el usuario (por ejemplo, una ventana si se trata de una interfaz gráfica) y le solicita

si quiere agendar la solicitud hecha por un visitante. Si la respuesta es afirmativa, el userbot pasa la información al Visitor Bot, el cual solicita una lista de horarios preferidos por el usuario. Finalmente, el userbot genera un mensaje conteniendo las preferencias del usuario y lo envía de regreso al Visitor Bot.

Este diseño posee algunas ventajas. Primero, el Visitor Bot no necesita saber acerca del tipo de interfaz con la cual comunicarse con el usuario, reduciendo la complejidad de su construcción. Segundo, los usuarios pueden tener acceso a diferentes recursos, por lo cual tendrían distintos userbots.

Finalmente, el agente que realiza una tarea específica solicita qué información se necesita para llevar a cabo la tarea pero no cómo obtenerla, ya que esto queda a elección de userbot en función de sus medios de comunicación (voz, mail, fax) y las preferencias del usuario.

Web Watcher

Para una búsqueda en el espacio Web, los usuarios recurren generalmente a un motor de búsqueda que trabaja con el principio básico de la aparición (ocurrencia) de los términos buscados, para retornar una lista de hipervínculos que permitan acceder a la información. No siempre un usuario conoce con exactitud los términos que lo llevarán a lo buscado y en función de esto – luego de algunas búsquedas iniciales – va refinando los criterios para tratar de alcanzar su objetivo.

Con la idea de mejorar las búsquedas existe WebWatcher (Joachims, 1995), un agente de interface (de la subcategoría de guías) que acompaña al usuario en su navegación por la World Wide Web, e interactivamente le va sugiriendo dónde continuar. Además, el usuario puede comunicarse con el sistema para retroalimentarlo. WebWatcher funciona como un aprendiz ya que observa y aprende de las acciones del usuario. Con el tiempo, registra partes de la WWW por donde pasó y los tipos de temas en los cuales usuarios anteriores tuvieron interés.

A partir de la página principal de la

School of Computer Science, Carnegie Mellon University se puede solicitar un "tour" acompañado por WebWatcher. En primer lugar se debe ingresar una frase que corresponda al tema de interés, luego WebWatcher retorna a la página inicial y desde allí comienza el "tour". El sistema funciona como un servidor proxy, el cual entrega las páginas solicitadas por el usuario con algunos "agregados" y "modificaciones". En la parte superior de la página es agregado un conjunto de comandos para comunicarse con el agente. Cada hipervínculo es reemplazado por la URL del servidor de WebWatcher y finalmente, si alguno de los vínculos es altamente recomendado lo remarca como una sugerencia para el usuario.

Los consejos de WebWatcher se basan en conocimiento obtenido en tours anteriores. Cada vez que el usuario selecciona un vínculo (recomendado o no), WebWatcher lo acompaña a la siguiente página, registrando la selección como un ejemplo de entrenamiento para mejorar las recomendaciones futuras. Al finalizar el "tour" el usuario tiene la posibilidad de indicar si alcanzó el objetivo o no, brindando una última forma de retroalimentación para el agente.

Para sugerir un vínculo apropiado dado el interés del usuario el sistema requiere del conocimiento de la

siguiente función:

Calidad-del-Vínculo : Página x Interés x Vínculo ® [0, 1]

La cual expresa la probabilidad de que un usuario seleccione un Vínculo dada la Página actual y el Interés. Para aplicar dicha función se utilizan tres aproximaciones para aprender de la experiencia:

- aprendizaje de tours anteriores;
- aprendizaje de la estructura del hipertexto;
- métodos que combinan las dos estrategias anteriores.

CONCLUSIONES

La tecnología de agentes de software ha mostrado una nueva forma de trabajar con computadores, permitiendo "delegar" en un programa (el agente) ciertas tareas que, de otra forma, solamente podría hacer el usuario. En algunos casos, puede tratarse simplemente de tareas repetitivas, pero en otros, puede ser una solución a problemas tales como la sobrecarga de información.

Los agentes han surgido como una solución posible y adecuada a la creciente dificultad de las personas para encontrar información útil, sobre todo con la creciente popularidad de internet. En este sentido, pueden verse como una buena herramienta que asiste en el proceso de abastecimiento y demanda de información, actuando como intermediarios entre quienes poseen la información y quienes la necesitan.

Internet es el ambiente ideal para desarrollar agentes, ya que presenta un medio cambiante y complejo pero por sobre todo muy extenso, donde los agentes de software pueden asistir a

los usuarios a encontrar "la información correcta en el tiempo justo".

La evolución de esta tecnología puede, además, plantear el uso de agentes móviles para resolver el problema de la sobrecarga de información [Dale, 1997], exponiendo las fuentes de información a grupos de agentes administradores, los cuales "entiendan" la arquitectura, complejidad y dominio del recurso que manejan. Estos agentes pueden ser consultados por agentes itinerantes que, ante una solicitud por parte de un usuario, "naveguen" la red en busca de satisfacer el requerimiento.

Los agentes, ya sea en forma individual, o bien trabajando en forma conjunta, poseen el potencial de llegar a ser una herramienta valiosa en esta nueva sociedad de la información.

REFERENCIAS BIBLIOGRAFICAS

1. JENNINGS, N.; WOOLDRIDGE, M. *Agent Technology - Foundations, Applications, and Markets*. Springer-UNICOM. 1998.
2. NWANA, H. Software Agents: An Overview. *Knowledge Engineering Review*. Cambridge University Press. v.3, p.1-40. 1996.
3. MAES, P. Modeling Adaptive Autonomous Agents. *Artificial Life Journal*, v.1, n. 1&2, p.135-162. MIT Press. 1994.
4. FRANKLIN, S; GRAESSER, A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*. Springer-Verlag. 1996.
5. RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall. 1995
6. WOOLDRIDGE, M.; JENNINGS, N. Software Agents. *IEE Review*, p.17-20. 1996.
7. BERNEY, B. *Software Agents – A Review*. Manchester Metropolitan University. 1996.
8. WILLIAM, J. *BOTS and other Internet Beasties*. Sams.Net Publishing. First Edition. 1996
9. ETZIONI, O; WELD, D. A Softbot-Based Interface to the Internet. *Communication of ACM*. n.37. 1994.
10. FRANKLIN, S.; OLDE, B.; SONG, H.; NEGATU, A. Virtual Mattie - an Intelligent Clerical Agent. *AAAI Symposium on Embodied Cognition and Action*, Cambridge MA. 1996.
11. KAUTZ, H.; SELMAN, B.; COHEN, M.; KETCHPEL, S. An Experiment in the Design of Software Agents. *Proceedings of the 1994 AAAI Spring Symposium on Software Agents*, Palo Alto. 1994.
12. JOACHIMS, T.; FREITAG, D.; MITCHELL, T. WebWatcher: A Tour Guide for the World Wide Web. *Proceedings of IJCAI97*. 1997.
13. ZICK, LAURA. *The Work of Information Mediators: A Comparison of Librarians and Intelligent Software Agents*. for CSCI 590: Intelligent Systems, IUPUI. 1999.
14. TORRISSEN, BJORN CHRISTIAN. *Dewey Goes Surfing: Agent-Based Information Retrieval and Clasification Support*. Norwegian University of Science and Technology. Faculty for Physics, Informatics and Mathematics. 1998.
15. DALE, J.; DE ROURE, D. Towards a Framework for Developing Mobile Agents for Managing Distributed Information Resources. Department of Electronics and Computer Science. University of Southampton. Technical Report Nro.97-1. 1997
16. JACCARD, J.; GAUTERO, M.; TOMASSINI, M. Websailor: Smart Agent for Information overLOAD Reduction. *Proceedings of the Fourth International Conference on the Practical Application of Intelligent Agents and Multi-Agent technology*, p 463-464, 1999.

Review: Technology of agents of software

Abstract

Software agents in the era of global networks are a vital tool to overcome the so-called "information overload" phenomenon. The degree of maturity reached by this technology reveals concrete applications already functioning in organizations, as well as in the home user's desktops. The objective of this work is to display a bibliographical overhaul on the technology of software agents, in particular in connection with models enabling the management of information overload.

Key words

Software agents, Information retrieval, review.

Gabriel Hernan Tolosa

Licenciado en Sistemas de Información. Auxiliar docente, Departamento de Ciencias Básicas, Universidad Nacional de Luján, Argentina.

tolosoft@mail.unlu.edu.ar

Fernando Raul Alfredo Bordignon

Licenciado en Sistemas de Información. Profesor Adjunto, Departamento de Ciencias Básicas, Universidad Nacional de Luján, Argentina.

bordi@mail.unlu.edu.ar
<http://www.unlu.edu.ar>
